

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

PROPOSTA DE UMA METODOLOGIA PARTICIPATIVA PARA O
DESENVOLVIMENTO DE *SOFTWARE* EDUCACIONAL

Gabriela Trindade Perry

PORTO ALEGRE, NOVEMBRO DE 2005

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

**PROPOSTA DE UMA METODOLOGIA PARTICIPATIVA PARA O
DESENVOLVIMENTO DE *SOFTWARE* EDUCACIONAL**

Gabriela Trindade Perry

Orientador: Fernando Gonçalves Amaral, Dr.
Co-orientador: Agostinho Serrano de Andrade Neto, Dr.

Banca Examinadora:

Professora Liane Werner, Dra.
Professor Luís Claudio Del Pino, Dr.
Professor Milton Zaro, Dr.

Dissertação submetida ao Programa de Pós-Graduação em Engenharia de
Produção como requisito parcial à obtenção do título de
MESTRE EM ENGENHARIA DE PRODUÇÃO

Modalidade: Acadêmica
Área de concentração: Sistemas de Produção

Porto Alegre, novembro de 2005

...dissertação foi julgada adequada para a obtenção do título de Mestre em Engenharia de Produção na modalidade
...têmica e aprovada em sua forma final pelo Orientador e pela Banca Examinadora designada pelo Programa de
Pós-Graduação em Engenharia de Produção.

Prof. Fernando Gonçalves Amaral, Dr.

Orientador PPGEP/UFRGS

Prof. Luis Antonio Lindau, Dr.

Coordenador PPGEP/UFRGS

Banca Examinadora:

Professor Liane Werner, Dra. (PPGEP/UFRGS)

Professor Luis Claudio Del Pino, Dr. (IQ/UFRGS)

Professor Milton Zaro, Dr. (PPGIE/UFRGS)

DEDICATÓRIA

Agradeço à minha mãe, Vera Lúcia Maidana Trindade, e ao meu marido; Fabrício Celso, pelo apoio incondicional.

Agradeço à professora Adriana Lopes Barros, da ETCOM/UFRGS, por oferecer espaço para em sala de aula e ao professor Del Pino, pelo auxílio durante o mestrado. Em especial – logicamente – ao meu orientador, professor Fernando Amaral, por ter me iniciado no método científico. Estes ensinamentos serão fundamentais daqui para frente.

E como não poderia deixar de ser, agradeço ao professor Agostinho, pelo seu entusiasmo e confiança. Sem ele, nada disso teria sido possível.

A todos a certeza: mesmo que o acaso jamais tivesse cruzado meu caminho, eu não poderia estar melhor do que estou agora

SUMÁRIO

INTRODUÇÃO.....	8
1.1 Apresentação do tema	8
1.2 Objetivos	10
1.3 Justificativa.....	11
1.4 Questões de pesquisa.....	12
1.5 Metodologia.....	12
1.6 Delimitação do estudo	12
1.7 Estrutura do trabalho	13
2 REVISÃO BIBLIOGRÁFICA	15
2.1 A evolução da informática na educação.....	15
2.2 Estudos de caso - ambientes educacionais para o ensino de Química	18
2.3 Procedimentos, processos ou ciclo de desenvolvimento de <i>software</i> , segundo modelos de engenharia.....	22
2.3.1 Modelo <i>waterfall</i>	22
2.3.2 Prototipação	23
2.3.3 Modelo espiral	23
2.3.4 <i>Agile development processes - Extreme programming (XP)</i>	24
2.3.5 CESD - <i>Cooperative experimental system development</i>	25
2.4 Ciclo de vida de <i>software</i> educacional	26
2.5 Trazendo o usuário para dentro do processo	37
2.6 Atividades de projeto – levantamento ou especificação de requisitos	40
2.7 Atividades de Implementação – <i>Guidelines e Design Participativo</i>	42
2.8. Atividades de avaliação da interface	47
2.8.1 Modelagem da performance do usuário: KLM - <i>keystroke level model</i> e GOMS – <i>goal, operators, methods, selection rules</i>	47
2.8.2 <i>Cognitive walkthroughs</i>	48
2.8.3 <i>Checklists</i>	48
2.8.4 Análise heurística.....	49
2.8.5 Avaliação com usuários	51
2.8.6 Sistemas de monitoramento	52
2.9 Avaliação preditiva de <i>software</i> educacional.....	53
2.9.1 Questionário MicroSIFT – <i>Micro computer software information for teachers</i>	53
2.9.2 Método JIGSAW	54
2.9.3 MAQSE – Manual para avaliação de qualidade de <i>software</i> educacional	55
2.9.4 TICESE – Técnica de inspeção e conformidade ergonômica de <i>software</i> educacional	56
2.10 Considerações a respeito da Revisão Bibliográfica	58
3. PROPOSTA DE METODOLOGIA PARTICIPATIVA PARA O DESENVOLVIMENTO DE <i>SOFTWARE</i> EDUCACIONAL	61
3.1 Detalhamento da etapa Projeto Educacional.....	65
3.2 Detalhamento da etapa Implementação.....	66
3.3 Detalhamento da etapa Avaliação	67

4 RESULTADOS	69
4.1 Elaboração do primeiro conjunto de requisitos: o projeto educacional	69
4.2 A implementação.....	72
4.3 Avaliação do <i>software</i> pelos usuários	79
4.3.1 Entrevista com professores de química.....	79
4.3.2 O teste piloto	82
4.3.3 Avaliação do <i>software</i> pelos usuários - o experimento em sala de aula..	83
4.4 Considerações sobre o processo	97
5. DISCUSSÃO	99
CONCLUSÃO.....	112
APÊNDICE A - Termo de consentimento livre informado	119
APÊNDICE B - Guia de simulação.....	120
APÊNDICE C - Pós-teste	133
APÊNDICE D - Manual para interpretação das respostas dos estudantes do pós-teste.....	139
APÊNDICE E - Tabelas do χ^2 significativo a 95%	142

RESUMO

Este trabalho visa agilizar e sistematizar o desenvolvimento de *softwares* educacionais, maximizando suas chances de favorecer a aprendizagem, através da descrição de cada uma de suas etapas, desde a primeira coleta de requisitos até a avaliação da aprendizagem, compondo, desta forma, uma metodologia. Para atingir essa meta, é importante favorecer a estruturação da integração da equipe de desenvolvimento com os professores e estudantes, durante o planejamento educacional, a implementação e a avaliação. Como produto desta metodologia, construiu-se um *software* educativo (cujo contexto é equilíbrio químico focado em ensino no nível médio), e sua eficiência em termos de aprendizagem foi avaliada experimentalmente, através de uma atividade educacional, em ambiente de ensino presencial. Ao final deste processo investigativo, constatou-se o fato de que a etapa de *design* da interface tem tanta importância quanto a etapa do planejamento educacional. Além disso, discute-se a necessidade de haver um especialista em educação no time, pois há indícios de que sua presença seja mais determinante para o sucesso do produto do que a dos demais profissionais (por exemplo: programadores, *designers*, roteiristas, etc). Assim, conclui-se que a proposta metodológica feita neste trabalho otimiza a participação dos integrantes da equipe, direcionando suas habilidades para as atividades necessárias, e que essa estratégia se reflete em bons produtos de *software*.

Palavras-chave: desenvolvimento de *software* educacional – avaliação de *software* – ergonomia participativa

ABSTRACT

The aim of this work is to systematize the development of an educational software, maximizing its capacity to support learning, through the description of each one of its phases, from the first requirement analysis to the evaluation of learning, putting a methodology together. To achieve this goal, it is important to supply a framework for the integration of the development team with teachers and students, during the educational project, the implementation and evaluation. As a product of this methodology, an educational software (whose context is chemical equilibrium, for high school) was built, and its efficiency (in terms of learning) was experimentally evaluated through an educational activity, on a presence teaching environment. In the end of this investigative process, it is verified that the interface's design phase is as important as the educational planning phase. In the end of this work is discussed the need of an expert in education on the team, because there is an indication that his presence is more significant for the success of the product than the presence of other professionals (for instance: programmers, designers, writers, etc). Therefore, it follows that the methodological proposition made in this work optimizes the participation of the members of the team, routing their abilities to the required activities, and that this strategy reflects in good software products.

Keywords: educational software development – software evaluation – participative ergonomics

INTRODUÇÃO

1.1 APRESENTAÇÃO DO TEMA

No contexto educacional atual, a inserção do computador está se tornando cada vez mais freqüente. Ela abrange o uso de aplicativos de escritório (por exemplo, MS Word®, MS Excel®, Open Office®), a Internet (como ferramenta de busca, comunicação, divulgação e de aprendizagem à distância) e *softwares* produzidos com o objetivo específico de contribuir para a aprendizagem. Estas classes de ferramentas são utilizadas em todos os níveis educacionais dentro das escolas e universidades, e também estão auxiliando o treinamento em indústrias. Restringindo a análise ao contexto acadêmico, em relação às ferramentas baseadas na *web*, por exemplo, abre-se uma gama muito grande de serviços a explorar, principalmente em benefício da educação à distância. De fato, no Brasil, tais produções têm sido bastante incentivadas pelo governo federal, como por exemplo, o projeto Educom, e vêm despertando o interesse tanto de pesquisadores quanto de professores. Podem ser citados dois casos de sucesso, um de âmbito nacional, o TelEduc, ambiente desenvolvido pelo NIED/UNICAMP e distribuído gratuitamente, e outro regional, o ROODA, plataforma da UFRGS para cursos à distância. Uma das vantagens destas tecnologias é a transmissão de informação e conhecimento a membros de comunidades distantes dos centros de produção cultural. Defendendo esta nova mídia como ferramenta legítima de comunicação do século XXI, capaz de transformar nossas atividades e nossa percepção, Levy (1999) oferece o resguardo teórico para todos que nela acreditam.

Em relação aos *softwares*, por sua vez, duas linhas de argumentação sustentam seu uso em ambientes educacionais. Uma, de ordem prática, legitima a utilização destas ferramentas como forma de mostrar ao estudante processos aos quais ele não pode ou não deve ficar exposto. Tais casos são típicos nas áreas das ciências exatas e biológicas. A outra linha de argumentação favorável ao uso de *softwares* como ferramentas educacionais apóia-se no pensamento de Papert (1982), para quem o computador é uma ferramenta cultural que permite à criança (estudante) relacionar-se com formas de raciocínio lógico não acessível fora dele. Papert (1982) acredita, por exemplo, que, ao usar um *software* de modelagem (também conhecido por micro-mundo) como o Logo, a criança aprenderia algumas estratégias típicas da computação, como o *debugging* e a execução de procedimentos através de *loops* aninhados. Além disso, pode-se referir o interesse que *softwares* educacionais e ferramentas

baseadas na *web* despertam no estudante do novo milênio como um terceiro aspecto positivo do uso. Cabe, neste ponto, lembrar que a própria separação entre *software* e Internet tende a ficar cada vez mais tênue, na medida em que a popularização de *standarts* de comunicação entre linguagens de programação permite que sejam estabelecidas conexões entre *softwares* e *web sites*.

Constatado este fato, é relevante, tanto do ponto de vista acadêmico como industrial, a preocupação em compreender e estruturar o processo de desenvolvimento de *softwares* educacionais, através do emprego de uma metodologia de produção, assumindo que esta envolva as etapas de: projeto educacional, implementação e avaliação.

Devido à especificidade do uso destes produtos, as questões colocadas à equipe de desenvolvimento são muito peculiares. Além das conhecidas dificuldades da produção de *softwares* (por exemplo: incorporação de novos requisitos aos protótipos, manutenção de prazos e orçamento, comprometimento real com as etapas de teste com usuários e *design* da interface), soma-se uma nova dimensão: a educacional. Na verdade, ela é o maior desafio destes times, por dois motivos principais. Primeiro porque o fato de uma estratégia ter sido adequada para um determinado produto, isto não é garantia de que este sucesso será repetido. O segundo motivo é que não é possível simplesmente transpor uma dada teoria que oriente a prática pedagógica para o *software* a ser criado, haja visto que algumas destas sequer são teorias de aprendizagem. A teoria de Jean Piaget, por exemplo, que tem informado a prática docente desde a década de 80, é uma teoria de caráter psicológico sobre a gênese do conhecimento, não sobre o ensino ou a aprendizagem. A proposta comportamentalista de Skinner, por outro lado, muito em voga durante os anos 70, sequer configura uma teoria, de acordo o próprio cientista (MOREIRA, 1985). Some-se a isto a crescente participação e reconhecimento de teorias cognitivas sobre o processamento da informação, que são de grande valor na defesa do uso de sistemas de simulação e modelagem, por exemplo. Por outro lado, as teorias focadas no ensino e na aprendizagem, como as de Ausubel, Novak e Hannesian, foram concebidas antes do uso educacional do microcomputador. A teoria construcionista de Seymour Papert, por sua vez, oferece validação teórica apenas para projetos de *software* de simulação ou de modelagem, deixando um tanto desamparadas outras estratégias de interação. Desta forma, a expressão popular matar um leão por dia ganha muito significado no contexto de desenvolvimento de *software* educacional.

Para superar estas dificuldades, podem ser tomadas contribuições de diversas áreas do conhecimento: educação e psicologia cognitiva, para compreender e orientar o processo de

aprendizagem; ergonomia e interação homem-computador, para promover a integração entre os usuários e desenvolvedores e entre os usuários e a ferramenta; e engenharia de *software*, para sistematizar o desenvolvimento. No entanto, reunir estas habilidades no time de produção não é suficiente. Conduzir trabalhos com uma equipe multidisciplinar e ainda validar as decisões com o auxílio de usuários reais não é tarefa simples. Algumas sugestões são encontradas na literatura, como por exemplo, designar um líder que tenha condições de dialogar com os membros da equipe de desenvolvimento e com os usuários. Há autores de engenharia que sugerem a adoção de formalismos pertinentes a cada uma das áreas citadas supra que, posteriormente, possam ser integradas no protótipo do *software*. De todo modo, é necessário adotar uma postura de estruturação e controle desta integração. Este é o requisito principal que se busca atender. Além disto, pretende-se também especificar medidas e momentos para realizar avaliações de qualidade/conformidade, de forma a orientar o avanço dos trabalhos.

Por tudo isso, se considera necessário propor uma metodologia para estruturar e sistematizar o processo de desenvolvimento de *software* educacional, e que esta seja de caráter participativo, para integrar os conhecimentos e as experiências da equipe, dos estudantes e dos professores, durante o projeto, a implementação e a avaliação.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Este trabalho visa, de maneira geral, sistematizar e agilizar o desenvolvimento de *softwares* educacionais, maximizando suas chances de favorecer a aprendizagem, através da descrição de cada uma de suas etapas de concepção, desde o primeiro levantamento de requisitos até a avaliação da aprendizagem.

1.2.2 Objetivos Específicos

Como objetivos específicos deste trabalho citam-se:

- a) possibilitar a estruturação da integração da equipe de desenvolvimento com os professores e estudantes, durante o projeto educacional, a implementação e a avaliação, com base em metodologia apropriada;

- b) produzir um *software* educacional de acordo com esta proposta metodológica, cujo assunto é equilíbrio químico¹ para ensino no nível médio;
- c) realizar uma avaliação experimental da aprendizagem, através de uma atividade educacional que envolva o uso deste *software*, em ambiente de ensino presencial.

1.3 JUSTIFICATIVA

Em relação ao desenvolvimento de *softwares* educacionais, Gomes e Wanderley (2003) afirmam que mesmo soluções geradas em centros de pesquisa tendem a abordar essa atividade de forma não sistemática. Isso significa que tais equipes trabalham de acordo com o primeiro paradigma de desenvolvimento de *software*, um *loop* envolvendo as etapas *code-test-fix*, adotado antes da década de 70. Esperam-se desta abordagem as seguintes conseqüências: (1) dificuldade crescente em incorporar novos requisitos; (2) perda de estrutura do código, com conseqüente perda da capacidade de reutilização e manutenção e (3) diminuição sobre o controle do projeto. Outro aspecto negativo é o elevado índice de retrabalhos, gerados principalmente por lacunas na comunicação entre os desenvolvedores, professores e estudantes e por falta ou deficiência nas especificações das funcionalidades. Por causa do retrabalho pode-se esperar: (1) maior tempo de desenvolvimento, acarretando aumento de gastos e (2) desmotivação da equipe e do cliente pela demora em ter resultados. Desta maneira, é necessário sistematizar ao máximo os processos de projeto, implementação e avaliação, com o objetivo de diminuir o tempo de produção e maximizar as chances de entregar um produto eficiente. Acredita-se que a sistematização contribui para conduzir à qualidade, aumentando as chances de ser bem sucedido. Um benefício adicional presumido da sistematização dos processos é a diminuição do tempo de desenvolvimento de cada produto de *software*. Esta sistematização dos processos de desenvolvimento deve considerar a integração entre os desenvolvedores, professores e estudantes, sendo justificada nos seguintes termos: (1) os professores detêm as estratégias de ensino, a experiência em relação à receptividade dos estudantes e o conhecimento profundo do tópico; (2) os estudantes, ao participarem da avaliação, contribuem para dar a medida da eficiência do *software* e (3) os

¹ Em Brown, LeMay e Burnsten, encontra-se, à página 353, a seguinte definição para equilíbrio químico: “no equilíbrio, a velocidade de formação dos produtos a partir dos reagente é igual à velocidade em que os reagentes são regenerados pelos produtos”.

desenvolvedores, quando em contato com professores, têm condições para estruturar o código de modo mais eficiente.

Justifica-se desta maneira a necessidade de propor uma metodologia participativa para o desenvolvimento de *software* educacional. Ressalta-se que o termo “participativa” refere-se à postura adotada pelo autor desta dissertação em relação à participação do usuário durante o desenvolvimento do software, ou seja, que eles devem ser incluídos; participar do processo. Este termo também deve ser interpretado como uma referência à escola escandinava (*design*), representado por um grupo de *designers* e ergonomistas que propunham tal atitude². Assim, deseja-se esclarecer que não há relação com a metodologia de pesquisa qualitativa denominada participativa, conforme descrita em Moreira (1990).

1.4 QUESTÕES DE PESQUISA

As questões norteadoras desta pesquisa podem ser expressas da seguinte forma: de que maneiras devem estar organizadas as atividades de projeto, implementação e avaliação? Que ferramentas e métodos podem ser empregados, para que ocorra a integração dos conhecimentos e experiências de desenvolvedores, professores e estudantes?

1.5 METODOLOGIA

A metodologia utilizada foi a pesquisa-ação. Sua utilização se deu pela proposição da construção do *software* e posterior avaliação.

1.6 DELIMITAÇÃO DO ESTUDO

Em primeiro lugar, é necessário esclarecer que o fato de a metodologia se mostrar adequada ou não para um determinado produto não permite fazer generalizações sobre sua eficiência para outros casos. Espera-se, contudo, que, por terem sido atacados pontos críticos do desenvolvimento de *software* educacionais apontados pela literatura e utilizados procedimentos de eficácia comprovada maximizem suas chances de ser positiva para outros casos.

² Mais detalhes na Revisão Bibliográfica.

Não será abordado, nesta pesquisa, o projeto de *software* / interface para usuários com dificuldades de acesso ou portadores de deficiência, nem assuntos restritos à engenharia de *software* e ciências da computação.

Além disto, a metodologia proposta neste trabalho não prevê a inserção do *software* em longo prazo em ambientes educacionais, nem aspectos logísticos e de marketing associados ao produto. Estas etapas não foram implementadas por razões de tempo. Por este motivo não foram projetados materiais de apoio ao uso do software, quer eletrônicos (*web site*, tutoriais, ajuda) quer sob a forma de polígrafos ou apostilas.

Ressalta-se ainda que o sucesso da aplicação de uma metodologia relaciona-se à experiência e ao conhecimento dos sujeitos que a utilizam e que, por causa do caráter participativo da metodologia proposta, depende-se em grande medida da colaboração e da motivação dos integrantes da equipe de desenvolvimento e dos usuários que participam das avaliações.

1.7 ESTRUTURA DO TRABALHO

Este trabalho está dividido em seis capítulos, sendo o primeiro destes a presente introdução. O segundo capítulo, a Revisão Bibliográfica, trata de temas pertinentes à criação de *softwares* educacionais. Assim, inicia-se fazendo uma introdução à informática na educação, para em seguida passar a estudos de caso de *software* para ensino de química, tema do programa produzido durante esta pesquisa. Seguem orientações, princípios ergonômicos e técnicas de *design*, cujo objetivo é trazer o usuário para dentro do processo. Após esta exposição, trata-se das questões de avaliação, considerada em três etapas: avaliação da interface, avaliação preditiva (relaciona-se com aspectos educacionais) e avaliação da interface.

O capítulo três apresenta a metodologia proposta, com o detalhamento de cada uma das etapas. Após a especificação das ações de cada uma delas, passa-se ao relato do desenvolvimento, desde o projeto pedagógico até a realização do teste piloto e o teste em sala de aula.

O quarto capítulo traz os resultados obtidos em cada uma das etapas, além da análise dos resultados e a discussão do experimento conduzido em sala de aula.

O capítulo cinco é destinado à discussão de questões que não puderam ser respondidas durante esta pesquisa, mas que concernem à temática desenvolvimento de *software*

educacional. São hipóteses em relação ao emprego da metodologia, levantadas durante a produção do *software* e que não puderam ser verificadas. Ao final do capítulo, é feita uma comparação entre o processo de construção dos *softwares* Le Chat e Equil (produto desta dissertação). Finalmente, as conclusões desta pesquisa encontram-se no capítulo seis.

2 REVISÃO BIBLIOGRÁFICA

2.1 A EVOLUÇÃO DA INFORMÁTICA NA EDUCAÇÃO

O primeiro computador, o ENIAC, criado em 1946 na Universidade de Pensilvânia, havia sido projetado para fins militares. Possuía 18.000 válvulas, 1.500 relés e emitia o equivalente a 200 quilowatts de calor. Essa enorme máquina foi alojada em uma sala de 9m por 30m. Este quadro é muito diferente da realidade contemporânea, na qual os computadores estão presentes em nossa vida de tal modo que sua presença sequer é notada. Eles estão embutidos em eletrodomésticos, celulares, automóveis. Permitiram a comunicação de pessoas, empresas e mercados através de rede mundial de computadores, determinando o início da Era da Informação. É utilizado ostensivamente para fins militares, de pesquisa e entretenimento. Pode-se dizer, inclusive, que algumas áreas modernas de conhecimento não poderiam ter se desenvolvido sem ele, vide o caso da Engenharia Genética e do projeto Genoma. Assim, como não poderia deixar de acontecer, o computador entrou também nas salas de aula.

Este processo começou na década de 60, através de *softwares* classificados como CAI (*Computer Aided Instruction*), cujo enfoque pedagógico remete à proposta behaviourista. Segundo Cosmann (1996), um dos primeiros exemplos desta série foi Plato (*Programmed Logic for Automatic Teaching Operation*), criado em 1960, que oferecia 7.000 horas de instrução em 150 assuntos, para uma rede de escolas de alcance nacional, nos Estados Unidos da América. Porém, devido a seu alto custo e necessidade de suporte técnico para operação, não chegou a ser utilizado amplamente nas salas de aula americanas.

Há ainda os micro-mundos, onde são elaboradas circunstâncias difíceis, ou até mesmo impossíveis de serem criadas fora de um ambiente computacional (CARRAHER, 1992). O expoente desta categoria de *softwares* é o Logo, um ambiente de programação para crianças, criado em 1967. Ele representou uma ruptura na maneira de inserir atividades com computadores em ambientes educacionais, pois ao utilizá-lo as crianças têm controle sobre o computador, e não o contrário (PAPERT, 1982). Seu impacto foi tão grande que até hoje existem grupos de pesquisa trabalhando com Logo e variantes, como por exemplo, o NIED - UNICAMP. Existem ainda outros tipos de *software* educacionais, por exemplo, jogos e simuladores. A produção e a inserção destes *softwares* na sala de aula têm sido incentivadas pelos governos de diversos países, desde a década de 70. Em relação aos Estados Unidos da América, Valente e Almeida (1997) afirmam que a inserção do computador começa nesta

década, com a utilização do programa Plato, descrito anteriormente. O uso de computadores na educação era descentralizado e independente das decisões governamentais. No início dos anos 80 os PC se disseminaram nas escolas, o que incentivou uma enorme produção e diversificação de CAI, como tutoriais, programas de demonstração, exercício-e-prática, avaliação do aprendizado, jogos educacionais e simulação.

Na França, segundo La Taille (1980), a implantação da informática na educação foi planejada levando em consideração público alvo, materiais, *software*, meios de distribuição, instalação e manutenção do equipamento nas escolas. Houve, como nos Estados Unidos da América, a utilização de CAI nos anos 70, e do Logo, nos anos 80.

Sobre a experiência latino-americana, Muñoz (2001) ressalta que, apesar de todos os países estarem fazendo esforços para aumentar o acesso às novas tecnologias (não apenas o computador), existe uma grande preocupação com alguns problemas básicos, tais como: alfabetização, aumentar o alcance das tecnologias, aumentar o acesso ao nível superior, melhorar a qualidade da educação. No mesmo trabalho, a autora apresenta ainda as iniciativas de diversos países nesta direção.

No Brasil, de acordo com Andrade (1994), a inserção da informática na educação teve efetivamente seu início com o projeto Educom. Buscava-se, segundo o autor, a compreensão dessa intersecção e a maximização dos benefícios mediante a implantação de centros-piloto, antes da adoção em massa pelo sistema de ensino, sem ter o conhecimento correto dos critérios de utilização. Estes centros foram implantados em 1984, em cinco universidades brasileiras: UFPE, UFMG, UFRJ, UFRGS e UNICAMP.

A receptividade da comunidade docente a tais ferramentas, porém, não foi homogênea. Rattner (1992, p.31), professor da faculdade de Economia e Administração da USP declara, em relação ao contato e a familiaridade precoces com o computador em estabelecimentos fabris, que:

Ocorre, com o uso do computador, uma perda da função de memória, assumida crescentemente pelos chips magnéticos. Da mesma forma, perde-se a habilidade manual, em virtude não só dos robôs, mas, sobretudo da redução das operações ao simples ato de apertar teclas e botões. [...]. Esse perigo de atrofiamento das características essenciais do ser humano, de sua natureza de Homo Faber, não se manifesta apenas nas fábricas e escritórios. Interesses comerciais, tão poderosos quanto inescrupulosos, forçam a entrada e inundam com sua publicidade escolas e lares, instigando educadores e pais a adquirirem equipamentos indispensáveis para a formação e o futuro das crianças.

Sobre a percepção do uso de computadores na escola, Grünberg (1994), que realizou entrevistas sistemáticas com 91 professores do nível secundário uruguaio, relatou que muitos educadores sequer consideraram haver necessidade do uso do computador, e que a maioria dos benefícios poderia ser alcançada através de outros meios, de forma mais ágil e com menor custo. Além disso, foram apontados motivos concretos para a não utilização do computador por parte dos professores pesquisados, listados a seguir:

- a) utilizar computadores demanda mais tempo de preparação, devido à imprevisibilidade do desenvolvimento da aula (devem estar preparados para tratar com mais questionamentos);
- b) professores experientes normalmente investem pouco tempo para preparar uma aula convencional;
- c) o tempo investido na preparação não pode ser recuperado em todas as escolas em que os professores trabalham, já que nem todas dispõem do mesmo equipamento;
- d) a necessidade de reservar o laboratório de informática para dias pré-fixados introduz rigidez na preparação do docente, o que reduz sua auto-suficiência por depender da equipe técnica;
- e) a privacidade é considerada ameaçada pela presença de um membro da equipe do Centro de Processamento de Dados na sala de informática.

Há também diversos argumentos apresentados por educadores, que defendem a inclusão digital. La Taille (1980) relaciona entre os efeitos pedagógicos benéficos da utilização do computador: aumento da capacidade de leitura; valorização dos materiais impressos como fonte de informação e recreação sadia; desenvolvimento do hábito de estudo independente; facilitação das revisões periódicas e descoberta e correção de erros e dificuldades. Papert, em seu livro *Mindstorms*, (1982), enumera as vantagens pedagógicas da utilização de computadores. Segundo ele, por ser uma ferramenta lógico-matemática muito presente em nossa cultura, teria condições de estimular este tipo de raciocínio em crianças. Defende ainda que a atividade criativa de programar pode inclusive estimular o raciocínio formal. Seguindo esta linha, apresenta certos procedimentos mentais corriqueiros empregados por programadores, tais como utilizar *loops* aninhados, realizar *debug* etc. que favorecem a sistematização, e, portanto, o raciocínio formal. No prefácio da edição de 1982, Papert faz uma ressalva sobre um ponto que julgou mal compreendido pelos leitores das edições anteriores: ele acredita que os professores são um elemento-chave no processo de

ensino/aprendizagem e que a instrução formal, apoiada por computadores ou não, precisa ser mediada por eles.

Desta forma, conclui-se que o aumento das atividades educacionais mediadas por computadores pode causar um impacto positivo nas atividades educacionais e que elas devem aumentar. Além disso, o governo brasileiro continua mantendo programas de incentivo à inclusão digital, enquanto a comunidade acadêmica mantém o interesse neste ponto, tema do *Workshop* de Informática na Educação de 2005, realizado durante o XXV encontro da Sociedade Brasileira de Computação. Espera-se que, conforme o computador comece a fazer parte da rotina educacional, as resistências diminuam. Também não se devem desprezar os anseios de uma nova geração de estudantes nascida na era da informação e que é familiarizada com tais ferramentas. Assim, o *software* educacional desponta como o produto mais rico desta nova realidade.

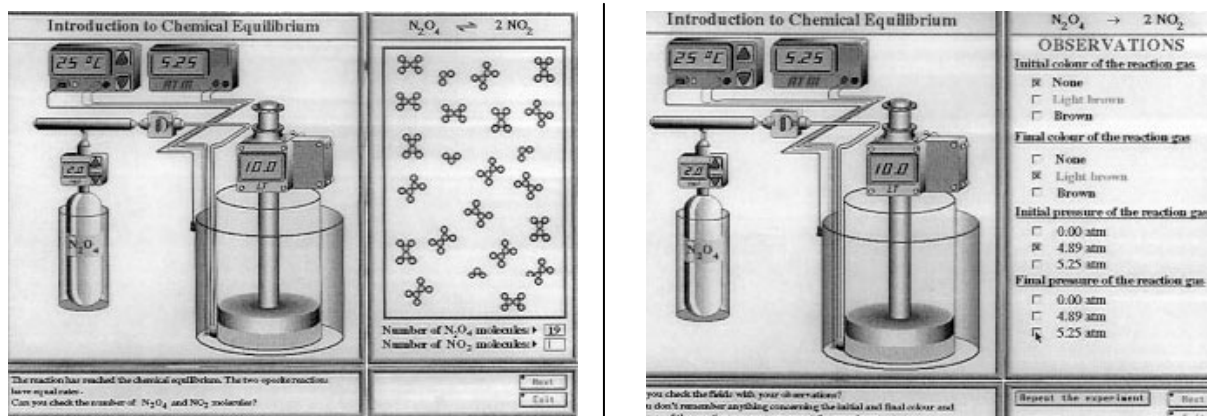
A partir deste ponto, a revisão bibliográfica deste trabalho passará a focar assuntos que fazem parte do escopo proposto, ou seja, o desenvolvimento de *softwares* educacionais. O primeiro tema apresentado são estudos que apontam para as boas chances de sucesso da utilização de *softwares* educacionais no ensino de química. O tema do *software* que será construído é equilíbrio químico, daí a importância desta exposição.

2.2 ESTUDOS DE CASO - AMBIENTES EDUCACIONAIS PARA O ENSINO DE QUÍMICA

Diversos trabalhos relatam o sucesso do uso de *softwares* como auxiliares no ensino de química, como por exemplo, Solomonidou e Stavridou (2001), que desenvolveram um *software* para ensino de equilíbrio com base nas concepções alternativas dos estudantes. A Figura 1 apresenta uma das telas do *software* que, segundo definição das autoras é um ambiente de aprendizado interativo, inspirado pela visão construtivista de aprendizagem, e que objetiva auxiliar os estudantes a melhorar suas concepções sobre equilíbrio químico.

Abordando o mesmo assunto, em Portugal, Gil e Paiva (1994) desenvolveram o programa *Le Chat*, para alunos dos 10º ao 12º anos de Portugal (equivalente ao ensino médio brasileiro). Nele, pode-se escolher uma dentre seis reações químicas simuláveis, sendo permitido manipular condições, tais como: volume do sistema; concentração; a temperatura e o uso de catalisadores. À medida que a reação ocorre podem ser visualizados gráficos de concentração em função do tempo, gráficos do quociente da reação *versus* a constante de

equilíbrio ou as simulações das próprias moléculas em um recipiente. O programa dispõe ainda de uma ajuda / glossário com uma interpretação das conseqüências para o sistema das alterações a que o sistema químico é submetido.



Tela com a simulação representada no nível microscópico e macroscópico.

Tela com perguntas sobre a observação.

Figura 1: Telas do *software* desenvolvido pela equipe coordenada por Solomonidou e Stavridou (2001, p.10).

Em sua tese de doutorado, Paiva (2000) apresentou a versão 2.0 do programa *Le Chat*³. Entre os melhoramentos implementados estão: acréscimo de um modo de jogo; subprograma de acerto de equações químicas; múltipla simulação de qualquer reação no estado gasoso; e atenção particular às concepções alternativas dos alunos. Em relação a estas, o autor afirma que o programa não tem uma relação de causa-efeito com as concepções alternativas, ou seja, não é um conjunto explícito de ferramentas para ‘desmontá-las’ sistemática e diretamente. Segundo o autor, as reações dos professores ao *software* foram positivas, com críticas relacionadas à sobrecarga visual da interface, e não aos aspectos pedagógicos. Em relação à avaliação com os estudantes, Paiva aplicou um questionário com 5 alternativas comuns à todas perguntas, e obteve um índice de 21% de ocorrência da resposta **tinha esta idéia / dificuldade, mas corrigi / resolvi com o uso do programa** [grifo pessoal]. A Figura 2 mostra a tela de entrada do *software*.

³ O download pode ser feito em http://nautilus.fis.uc.pt/wwwqui/equilibrio/port/eqq_lechat2.html

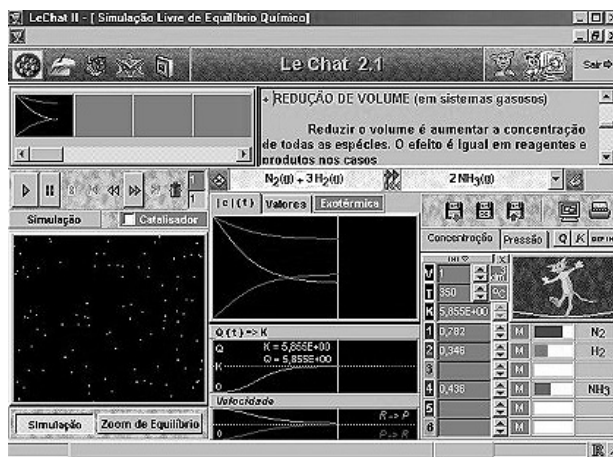


Figura 2: Tela do *software Lechat 2.0*.

Barnea e Dori (1996), por sua vez, investigaram o uso de *softwares* para CMM (*Computerized Molecular Modeling*) integrados à instrução regular. As autoras afirmam que a vantagem do uso desta classe de *software* está relacionada principalmente à dificuldade de manipular tais moléculas no mundo físico. No experimento conduzido, o uso foi integrado a uma estratégia de ensino que valorizava a exploração e a elaboração de raciocínio lógico, através da elaboração de hipóteses e posterior derivação de conclusões, a partir de dados obtidos do *software*. Os professores haviam sido instruídos sobre a função dos modelos moleculares no ensino de química; sobre o aplicativo em si e sobre a integração deste ao ensino, e utilizaram-no com o apoio de material didático produzido pela equipe que conduziu a pesquisa. As vantagens do uso de CMM para transmitir conhecimentos sobre a teoria da repulsão dos pares eletrônicos da camada de valência (também referenciada como VSEPR, de *vallence shell electronic pair repulsion*) listadas pelos professores foram:

- trabalhar com computadores motiva a maior parte dos estudantes;
- estudantes que utilizam CMM (experimental) são mais ativos e participantes que os que utilizam outros métodos (controle);
- o fato de o CMM ser tridimensional permite que os estudantes rotacionem a molécula, melhorando sua habilidade espacial;
- construir modelos moleculares com *software* é mais fácil e rápido do que com modelos físicos.

As desvantagens listadas foram:

- o custo de pacotes CMM pode ser proibitivo para muitas escolas;

- b) é necessário certo tempo até que todos estejam familiarizados com o *software*;
- c) alguns aspectos da modelagem da simulação podem não estar contemplados. No caso do pacote utilizado, não foi considerada a polaridade das moléculas e ligações intermoleculares, como Van der Waals e de hidrogênio.

O desempenho dos estudantes em questões que requeriam aplicação direta de princípios da teoria da repulsão dos pares eletrônicos da camada de valência (alvo do pacote CMM) está ilustrado na Figura 3.

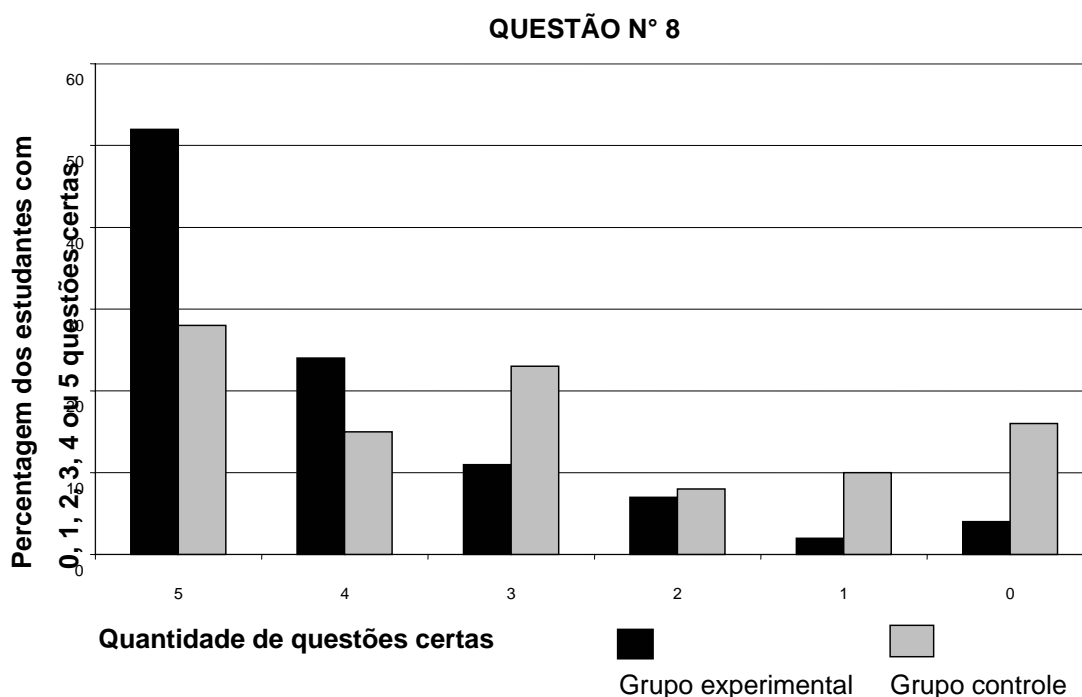


Figura 3: Desempenho dos estudantes do grupo controle (não utilizaram o *software*) e experimental (utilizaram o *software*) em questões envolvendo conhecimento de VSEPR. Mostra que o grupo experimental teve desempenho superior, acertando maior quantidade de questões relacionadas com o tema. Retirado de Barnea e Dori (1996, p. 6).

Barnea e Dori (1993) também desenvolveram um *software* para instrução em polímeros. Os motivos alegados foram: falta de modelos adequados para lidar com os aspectos dinâmicos do processo e com a estrutura tridimensional dos polímeros; o risco e o custo envolvidos em experimentos e a necessidade de estudar o assunto em diferentes níveis de complexidade. O pacote trata de três assuntos, com as informações organizadas em mapas conceituais e bases

de dados. É feito uso extensivo de recursos gráficos e de animação e, ao final de cada tópico, são propostos problemas, para os quais é dada uma resposta imediata. A conclusão das autoras é que o pacote supriu as necessidades que motivaram seu desenvolvimento.

A exposição destes resultados – retirados de trabalhos idôneos publicados em periódicos internacionais reconhecidos – permite concluir que há boas perspectivas para o uso de ambientes computadorizados específicos para ensino de química. Porém, nenhum dos estudos relaciona o sucesso do ensino apenas à introdução do computador. Em todos eles, seu uso estava integrado a uma estratégia instrucional que valoriza a exploração e a elaboração de raciocínio lógico, através da elaboração de hipóteses e posterior derivação de conclusões, a partir de dados obtidos do *software*.

2.3 PROCEDIMENTOS, PROCESSOS OU CICLO DE DESENVOLVIMENTO DE SOFTWARE, SEGUNDO MODELOS DE ENGENHARIA.

Em Pressman (1995), procedimento é definido como o elo que mantém juntos os métodos e as ferramentas, possibilitando o desenvolvimento racional e oportuno do *software* de computador. Os procedimentos definem a seqüência em que os métodos serão aplicados; os produtos (*deliverables*) que se exige que sejam entregues (documentos, relatórios, formulários etc.), os controles que ajudam a assegurar a qualidade e a coordenar as mudanças e os marcos de referência (*milestones*), que possibilitam aos gerentes avaliar o progresso.

A diferença entre procedimento e métodos, segundo Boehm (1988), é que o foco destes últimos é como transitar entre fases do desenvolvimento e como representar seus produtos. Ao utilizar processos de desenvolvimento, o desenvolvedor aumenta a probabilidade de obter produtos de alta qualidade, através da organização e sistematização das tarefas. Boehm (1988) afirma que, caso o procedimento adotado fosse codificar e depois corrigir, depois de algumas correções o código perderia a estrutura. Este é o argumento para uma fase de projeto (*design*) anterior à de codificação. A fase de análise de requisitos é anterior à de *design* para que o *software* esteja adequado ao usuário.

2.3.1 Modelo *waterfall*

O modelo Cascata ou Waterfall, apresentado por Royce (1970) para o desenvolvimento de grandes sistemas de *software*, representou uma mudança de paradigma em relação à abordagem análise-programação. Neste modelo estão definidas fases de

desenvolvimento, *milestones* claras e documentos gerados, e está previsto um *loop* entre as fases, nos períodos de transição. Em Pressman (1995) são encontradas as seguintes críticas a este modelo:

- a) projetos reais raramente seguem um fluxo seqüencial, pois sempre há iterações;
- b) muitas vezes é difícil para o cliente declarar todas as exigências explicitamente;
- c) uma versão funcional do programa estará disponível apenas no final do cronograma.

2.3.2 Prototipação

Segundo Pressman (1995), uma abordagem orientada à prototipação é adequada em casos que o cliente não definiu requisitos de entrada, processamento e saída detalhados ou em que o desenvolvedor não tem certeza da eficiência de um algoritmo ou da forma que a interação homem-máquina deve assumir. Pernin (1996) define protótipo como um modelo operacional, modificável e extensível de um sistema, porém não necessariamente representativo do sistema completo, e que fornece aos usuários da aplicação final uma representação física dos principais aspectos do sistema depois de sua implementação.

As desvantagens desta abordagem, de acordo com Pressman (1995) são:

1. o cliente pode não entender que um protótipo não é o produto final, e que precisa ser reconstruído;
2. o desenvolvedor pode terminar construindo o produto final a partir do protótipo.

Pernin (1996) questiona em que medida a abordagem da prototipação se diferencia da abordagem Cascata. Para o autor, a única diferença seria que naquela o ciclo inteiro deve ser repetido muitas vezes. Como será visto adiante, diversas metodologias prevêm a construção de protótipos e iterações entre fases.

2.3.3 Modelo espiral

O processo espiral foi proposto por Boehm (1988) como um avanço em relação aos processos de *software* existentes. A diferença fundamental entre eles é que o modelo espiral é orientado pelos riscos (*risk-driven*), ao invés de orientado por documentação e código. Também foi planejado para lidar com o desenvolvimento de grandes sistemas

Segundo Boehm (1988), o modelo espiral tem a vantagem de se adaptar aos diferentes tipos de riscos envolvidos no projeto (relacionados a orçamento, prazos, interface com usuário e patamares de *performance*); tem alta capacidade de eliminar erros e identificar alternativas equivocadas logo no início e estabelece critérios claros para passar para outra fase.

2.3.4 Agile development processes - Extreme programming (XP)

As metodologias pertencentes à família dos *agile development process*, da qual XP é um dos representantes, têm em comum: o foco em indivíduos e interações, ao invés de processos e ferramentas; *software* funcionando ao invés de documentação ostensiva; colaboração do cliente ao invés de negociações nos contratos; responder às mudanças ao invés de seguir um plano. Isto não significa que os itens da direita não sejam valorizados, mas que os da esquerda são mais (BECK et al., 2001). Os processos ágeis estão relacionados à filosofia de produção de *software* livre. Como exemplo, pode-se verificar que, nas duas últimas edições do Fórum Internacional de *Software* Livre (5ª e 6ª edições, de 2004 e 2005), foram apresentados estudos de caso de programas desenvolvidos de acordo com este método.

O modelo XP é adequado para equipes pequenas e médias que desenvolvem *software* com requisitos vagos e em constante mudança, que têm equipes mistas com desenvolvedores júnior e sênior e que precisam mostrar o progresso do trabalho através de protótipos. Está baseado nas práticas listadas a seguir.

- a) A maior prioridade é satisfazer o cliente entregando *software* funcionando desde o início.
- b) A mudança nos requisitos é bem-vinda.
- c) Mostre *software* funcionando com a maior frequência possível.
- d) Desenvolvedores e clientes devem trabalhar juntos.
- e) Desenvolver projetos com pessoas motivadas.
- f) A comunicação face a face é o meio mais eficiente de distribuir informação.
- g) *Software* funcionando é a medida do progresso.
- h) Ágil não significa instantâneo. O ritmo da equipe deve ser mantido durante todo o processo.
- i) Atenção contínua à excelência técnica e à qualidade do *design* garantem a agilidade.
- j) Simplicidade.
- k) Auto-organização das equipes.
- l) Regularmente, a equipe reflete sobre seu desempenho e ajusta seu comportamento.

2.3.5 CESD - *Cooperative experimental system development*

Finalmente, apresenta-se o modelo CESD (*Cooperative Experimental System Development*), proposto por Grønbaek et al. (1997). As motivações desta proposta são: integrar ativamente o usuário durante todo o processo; testar protótipos em situações reais ou muito próximas das reais e fazer projetos sob medida. O modelo CESD foi criado por um grupo de pesquisadores ligados à abordagem escandinava, ou seja, que empregam métodos de *design* participativo⁴. O aspecto cooperativo é dado pelo extenso uso de tais métodos, para que todos os atores contribuam ostensivamente durante o projeto. O aspecto experimental é dado pela abordagem iterativa, onde alternativas são exploradas e implantadas através de experimentação em situações de uso reais. São definidas 4 dimensões: interesses; atividades; domínios envolvidos e atributos de projeto. A categoria de interesses engloba as atividades que definem conceitualmente um projeto, e a categoria atividades contém as ações concretas. Os domínios relacionam-se com as atividades, e os atributos de projeto dizem respeito ao tipo de projeto desenvolvido: contratual; de pesquisa; desenvolvimento *in-house* ou de pacotes de *software*. As atividades envolvem as práticas de desenvolvedores e dos usuários, a tecnologia e visões sobre a tecnologia em uso. Os interesses, representados na camada do meio, são empreendidos através de atividades. Uma atividade pode contribuir para um ou mais interesse, mas tipicamente há um interesse principal em foco. Desta forma, durante o desenvolvimento do projeto, as atividades devem acontecer em paralelo, pois estão sempre contribuindo com os diversos interesses. A Figura 4 mostra o relacionamento entre atividades e interesses em um projeto desenvolvido com o modelo CESD.

Este modelo não permite a definição de uma seqüência de fases, pois não há como terminar a análise e começar o *design*, pois uma mesma atividade influenciar ambos. Esta é, nas palavras dos autores, uma das principais fontes de problemas em ciclos que propõe a iteração de fases, como por exemplo, o modelo espiral de Boehm (1988).

Espera-se que, após a finalização desta exposição, tenha-se conseguido alcançar o objetivo de mostrar diferentes abordagens em relação ao ciclo de desenvolvimento de *software*. Apesar de nenhuma das abordagens ter sido esgotada, julga-se que esta apresentação é suficiente para enquadrar as propostas metodológicas de desenvolvimento de *software* educacional em um contexto de produção.

⁴ *Design* participativo será abordado na seção 2.7 desta revisão.

No entanto, para completar a análise da produção desta classe de programas, convém focar as propostas específicas para este tipo de ferramenta. Serão apresentadas sugestões de ciclos de desenvolvimento, pois eles são os eixos de trabalho no processo de articulação dos aspectos educacionais, de comunicação e computacionais.

Atividade	Etapa	Gerenciamento	Análise	Design	Realização	Atividades Apoiadas pelo Computador
1. Início do projeto		●	●	●		
2. Pesquisas iniciais e reuniões		●	●	●		
3. Experimentação com protótipo feito no HyperCard		●	●	●	●	●
4. Experimentações com protótipos baseados em Dexter, em Macintoshes		●	●	●	●	●
5. Desenvolvimento de um <i>framework</i> baseado em Dexter e em base de dados orientada a objeto		●		●	●	
6. Experimentação com hipermídia aberta em Windows/NT		●	●	●	●	●
7. Uso da hipermídia aberta			●	●	●	●

Figura 4: Relacionamento entre atividades e conceitos para o projeto EuroCoOp/EuroCODE⁵, extraída de Grønbaek et al. (1997). O diâmetro dos círculos indica o grau de comprometimento entre as tarefas e cada etapa.

2.4 CICLO DE VIDA DE SOFTWARE EDUCACIONAL

Inicia-se esta parte da revisão com o ciclo para produção de materiais didáticos, proposto por Dorrego (1994). As etapas (de um ciclo linear) são apresentadas na Figura 5.

Zambrano et al. (1995), por sua vez, sugerem o emprego método PROSDOS Ampliado, para sistematizar e orientar o processo de produção de programas educacionais. São definidas as quatro etapas, listadas com suas respectivas sub-atividades na Figura 6.

⁵ O projeto EuroCODE visa construir um ambiente de desenvolvimento aberto para CSCW –*Collaborative Support / Collaborative Work*. EuroCoOp procura integrar componentes de diversos sistemas e desenvolver novas ferramentas colaborativas. Informações retiradas de <http://dbs.cordis.lu/>

1. Planificação.
2. Plano didático.
 - Fase 1. Formação de objetivos terminais.
 - Fase 2. Análise estrutural, condutas de entrada, processos cognitivos implicados e requisitos prévios.
 - Fase 3. Formação de objetivos específicos.
 - Fase 4. Determinação das estratégias instrucionais, seleção e organização do conteúdo, especificação das variáveis técnicas do meio.
 - Fase 5. Avaliação formativa e somativa.
3. Plano de produção.
 - Fase 6. Entrega de documentação técnica.
4. Realização.
5. Avaliação formativa dos materiais instrucionais.

Figura 5: Desenvolvimento de materiais didáticos segundo Dorrego (1994).

1. Etapa de concepção pedagógica.
 - 1.1 Revisar e decompor o tópico.
 - 1.2 Ordenar e agrupar os elementos do tópico.
 - 1.3 Definir os objetivos de cada lição.
 - 1.4 Identificar os conceitos relevantes de cada lição.
 - 1.5 Especificar cada lição.
 2. Etapa de transição pedagógica-informática.
 - 2.1 Especificar os objetos e classes de objetos presentes em cada lição.
 - 2.2 Especificar os recursos de *software*.
 - 2.3 Selecionar os recursos de hardware
 3. Etapa de desenvolvimento.
 - 3.1 Construir o diagrama de interconexão de instâncias de classe para cada lição.
 - 3.2. Construir a interface gráfica.
 - 3.3 Elaborar os recursos audiovisuais.
 4. Etapa de estudo do produto.
 - 4.1 Avaliar o suporte didático, através de questionários, que devem ser aplicados respectivamente a um grupo de estudantes, à equipe pedagógica que participou do desenvolvimento, à equipe de desenvolvimento e à equipe de desenhistas.
- OBS.: Não foram especificadas iterações entre fases.

Figura 6: Desenvolvimento de *software* educacional segundo Zambrano et al. (1995).

Campos e Rocha (1996) propõem as seguintes etapas para o desenvolvimento de um programa educacional: definição do ambiente de aprendizagem, análise de viabilidade, seleção do tipo de documento, seleção do método para autoria, planejamento da interface, planejamento do documento, seleção do sistema de autoria e das ferramentas, implementação, avaliação e validação. As autoras afirmam que o ciclo de vida mais adequado para produção de *software* educacional é o de prototipagem evolutiva. Comenta-se ainda que não estão explícitos os atores que participam de cada fase; os critérios para passar de uma fase a outra e como elas se relacionam ou os momentos de iteração ou revisão entre as fases.

Castro e Aguiar (200) também propõem um processo de desenvolvimento de programas educacionais com um perfil de prototipação evolutiva. A Figura 7 esquematiza o ciclo proposto, com as fases da iteração indicadas no círculo, e na Figura 8 é mostrado o diagrama *use case* para as fases de iteração. Na descrição do caso **Preparação**, Castro e Aguiar (1999) sugerem a elaboração de um plano instrucional conforme eventos relacionados à aprendizagem descritos, referenciando os trabalhos de Gagné⁶. Durante a fase **Projeto**, na qual toma corpo a interface do programa, os autores sugerem a escolha de metáforas para a interface, oferecem uma lista de *guidelines* e, até mesmo, um *template* para uma janela de navegação. Acredita-se que a disponibilidade de informação a respeito do projeto de interfaces é ampla e acessível o bastante para que não se restrinja a este tipo de recomendação.

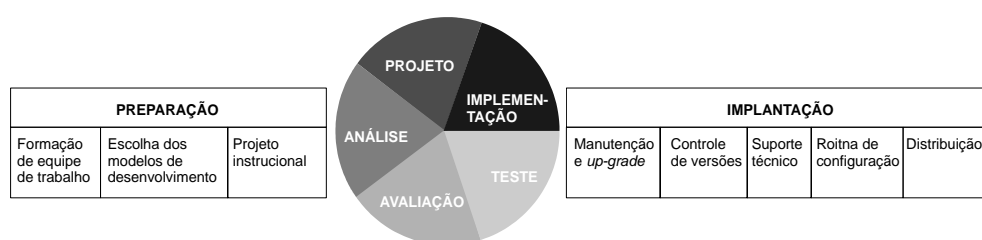


Figura 7: Ciclo de vida segundo Castro e Aguiar (2000, p. 2).

⁶ Em relação a isto faz-se a crítica que este posicionamento lida com a proposta de Gagné como *behaviorista*, classificação na qual ela não se encaixa, segundo Moreira (1985).

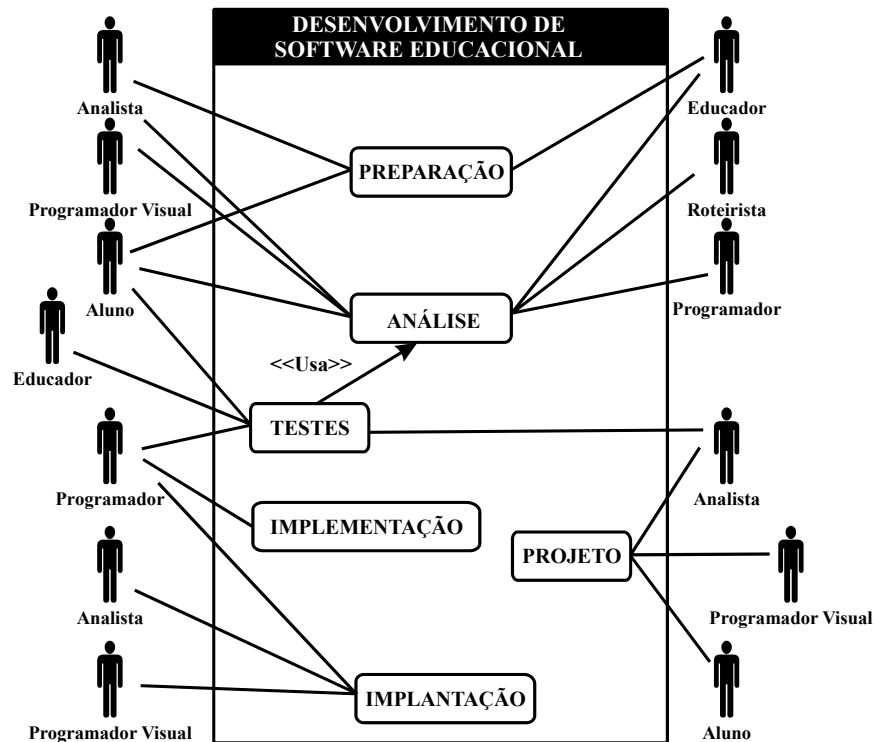


Figura 8: Diagrama *use case* para as fases de iteração do ciclo de Castro e Aguiar (2000, p. 3).

Batista (1997) é outro autor que recomenda a utilização de ciclos baseados em prototipagem evolutiva durante o desenvolvimento. Seu argumento é que esta escolha, aliada à utilização de métodos orientados ao objeto, contribui para a utilização de protótipos sujeitos a testes desde cedo e facilita a gestão do projeto, pois ao invés de gerir todo o sistema, gerencia-se pequenos protótipos. Em seu modelo, incentiva a participação dos professores e estudantes (os últimos participam através de testes) e do conceitor – que pode ou não coincidir com o professor. O conceitor é quem decide os requisitos de cada protótipo. As reuniões são chamadas de **interações formais**, cujas principais atividades são: decisão de iniciar o projeto; elaboração da tabela de responsabilidades estendida; apresentação do protótipo ao professor e ao aluno e realização de testes; negociação do contrato e decisão e encerrar o desenvolvimento. A Figura 9 esquematiza um exemplo de reunião de interação formal, e a Figura 10 mostra um exemplo da tabela de responsabilidades estendida.

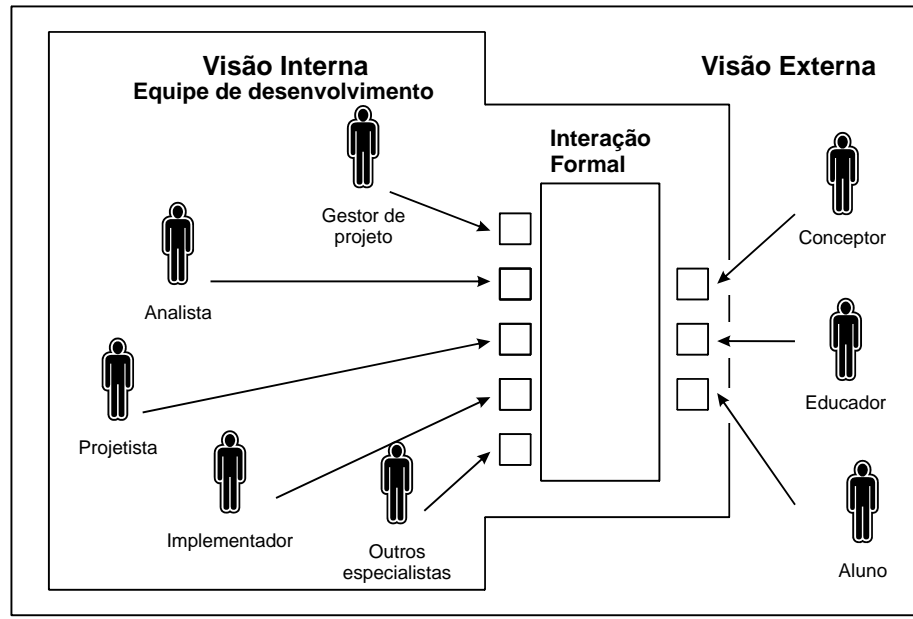


Figura 9: Representação do envolvimento de entidades internas e externas em uma seção de interação formal. Extraída de Batista (1997, p. 5).

Classe Cenário	Aluno [Cardinalidade 1-2]	Programa Educativo [Cardinalidade 1]	Educador [Cardinalidade 1]
Relógio	A: Tempo por volta	A: Tempo por volta	A: Tempo por volta
	P: Pedir tempo por volta	M: Recebe pedido de alteração do tempo por volta	P: Pedir alteração do tempo por volta
	D: Alterar tempo por volta	D: Alterar tempo por volta	D: Alterar tempo por volta
	A: Posição do ponteiro	A: Posição do ponteiro M: Avançar (1/60) tempo por volta	

Figura 10: Fragmento da tabela de responsabilidade estendida. Extraída de Batista e Figueiredo (1997, p. 7).

O desenvolvimento se dá entre as interações formais, conforme a Figura 11. Nela estão representadas as entidades A – equipe de desenvolvimento – e B – todos os membros envolvidos. Nota-se que cada reunião é sucedida por um intervalo no qual apenas a equipe de desenvolvedores trabalha (chamado **patamar de evolução atual**), seguido da apresentação de um protótipo e de outra sessão de interação formal.

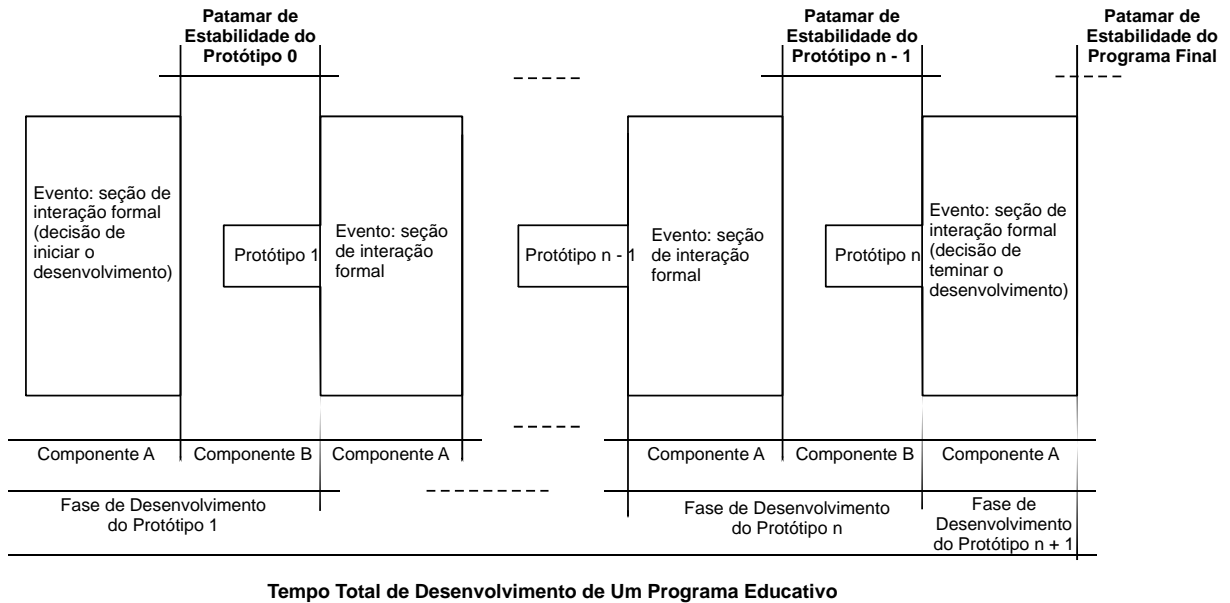


Figura 11: Processo de desenvolvimento de *software* educacional através de prototipagem evolutiva, segundo proposta de Batista (1997, p. 7).

Burd (1999) propõe um ciclo de vida de *software* educacional apoiado em preceitos da Teoria da Atividade. O processo, orientado à prototipagem evolutiva, está esquematizado na Figura 12.

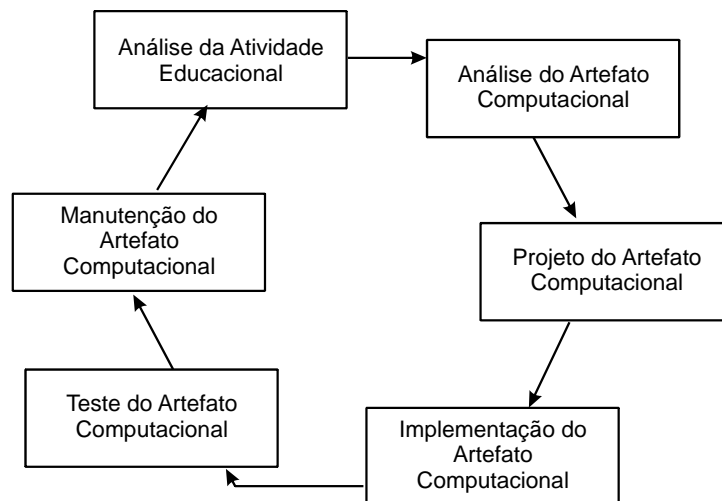


Figura 12: Ciclo de vida conforme Burd (1999, p. 93).

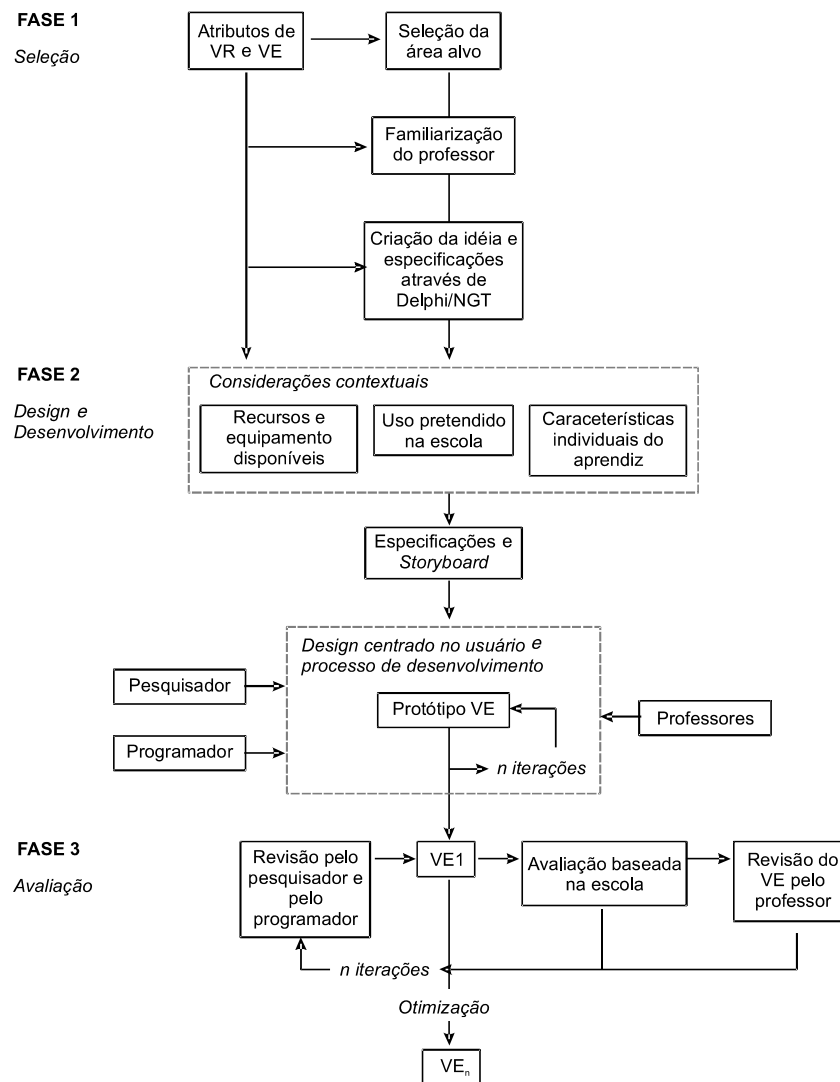
As etapas análise da atividade educacional e análise do artefato computacional são as que recebem maior ênfase. Segundo Burd, a função da primeira é descrever a atividade

educacional da forma mais completa, realçando os aspectos sócio-histórico-culturais que deverão ser alcançados e obedecidos. Ele sugere uma lista de questionamentos que podem orientar a elaboração de um documento que marca a realização desta fase. A função da segunda é delinear o que será feito pelo artefato computacional. Esta análise é feita frente à atividade, daí o caráter complementar destas duas etapas iniciais. Para orientar o desenrolar desta fase, o autor propõe questionamentos que possibilitam compreender esta interação, no contexto da Teoria da Atividade.

Crosier et al. (2002), por sua vez, propõem um ciclo para desenvolvimento de simulações educativas fortemente embasadas pela demanda dos professores. Neste sentido, são especificadas fases iterativas, cujo produto é um protótipo, representadas na Figura 13.

Fazendo do desenvolvimento de *software* educacional assunto de sua tese de doutoramento em informática, Van der Mast (1995) embasa sua proposta na abordagem de Seligman (apud VAN DER MAST, 1995) para investigação de sistemas informacionais. É estabelecido um jeito de pensar, de modelar, de trabalhar e de controlar, complementado por um jeito de auxiliar. O de pensar é descrito em três níveis: micro, meso e macro. O nível micro contém a interação do aprendiz com o computador. O meso representa a influência do professor, e o macro o ambiente: a escola e suas determinações curriculares.

No jeito de modelar são empregadas técnicas (modelos, formalismos) para descrever conhecimento sobre conceitos e ações sobre a interface. Cada um dos três níveis descritos para o jeito de pensar requer diferentes estratégias de modelagem. Por exemplo, a modelagem do nível macro pode ser mais flexível, não sendo necessário utilizar formalismos. Seu resultado é uma descrição do assunto a ser aprendido e ensinado, os principais objetivos de aprendizagem e os critérios para avaliar o desempenho dos estudantes. Já a modelagem pedagógica e didática feita no nível meso, define componentes do *software*, objetivos, regras, estratégias instrucionais, tarefas dos estudantes e definição do meio. A interação com o estudante é especificada no nível micro. Van der Mast (1995) ressalta que tais atividades devem ser conduzidas por pessoas com experiências em cada uma das áreas, orientadas por um gerente que, idealmente, tem conhecimento suficiente para dialogar com todos os membros da equipe. As especificações produzidas para cada um destes níveis são os pontos iniciais para o *design* técnico. Tais técnicas (de modelagem) podem ser escolhidas de acordo com a preferência da equipe. A divisão da modelagem em três níveis (micro, meso e macro) resulta na divisão em atividades independentes para o jeito de trabalhar, conforme a Figura 14.



OBS.: As siglas VR e VE significam *virtual reality* e *virtual environment* (como sinônimo de ambiente de simulação). DELPHI e NGT (*Nominal Group Technique*) são técnicas para levantamento de requisitos.

Figura 13: Representação das etapas e dos atores do método proposto por Crosier (2002, p. 11).

Ressalta-se que o autor expressa sua indicação para o uso de protótipos nos três níveis, inserindo sua metodologia em um contexto de prototipagem evolutiva. Para sustentar as atividades constituintes do jeito de controlar, devem-se especificar os documentos a ser entregues no final de cada uma das fases do jeito de trabalhar, além de prever um planejamento de avaliação. O autor sugere que tais responsabilidades sejam delegadas a uma pessoa.

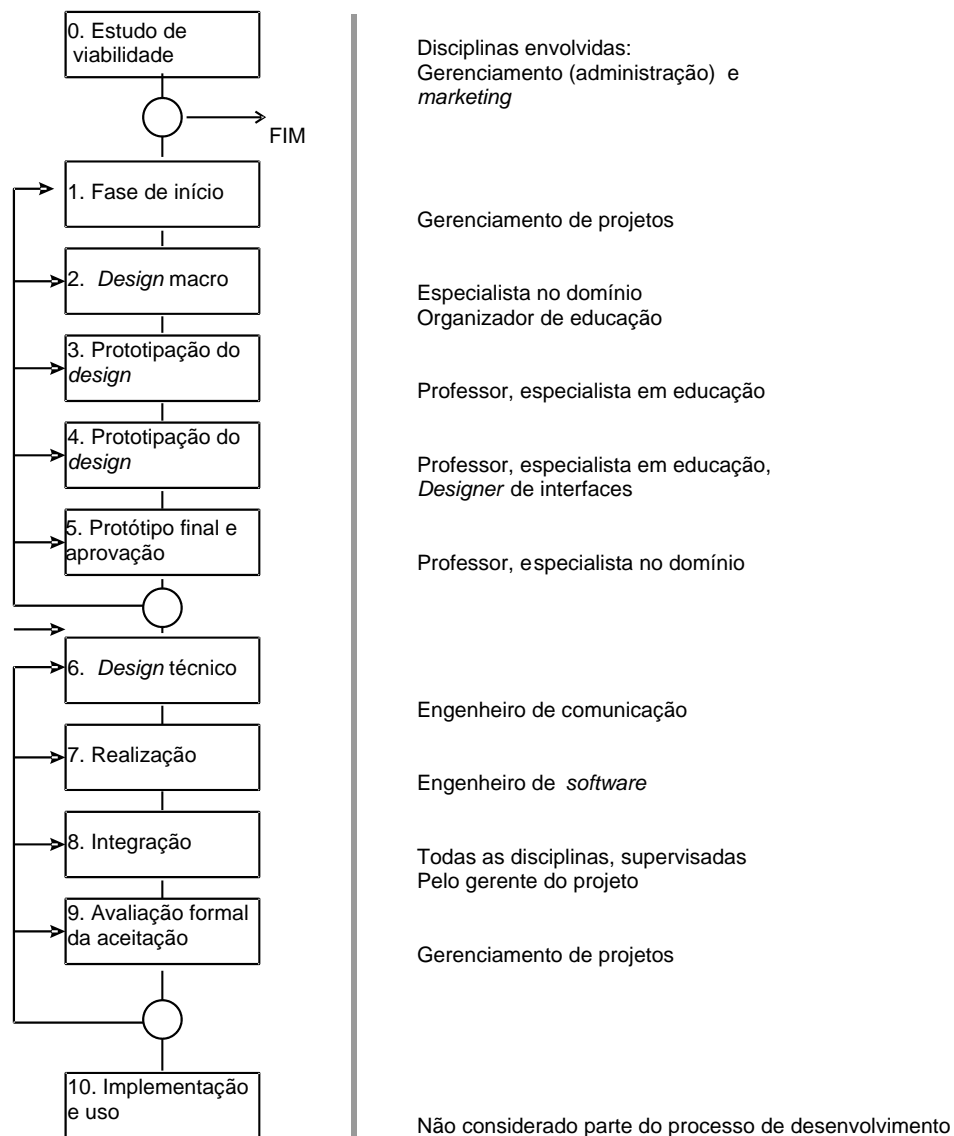


Figura 14: O jeito de trabalhar, segundo Van der Mast (1995, p. 95).

Van der Mast (1995) chega às seguintes conclusões:

- após o problema (educacional) ser identificado, alguém deve ser declarado como especialista no problema (o autor refere-se a *problem owner*). Esta pessoa, em conjunto com o gerente de desenvolvimento deve focar explicitamente nos aspectos educacionais e tecnológicos;
- a modelagem dos níveis micro, meso e macro deve ser feita separadamente;
- o projeto da interface deve ser considerado prioritário;

- d) o comprometimento das pessoas envolvidas deve ser baseado em definições claras de suas responsabilidades;
- e) o *design* e a realização devem ser realizados separadamente;
- f) o líder do projeto deve ter conhecimento em todas as áreas envolvidas;
- g) medidas e avaliações de qualidade são necessárias;

Pernin (1996) alega motivos semelhantes para propor uma metodologia de desenvolvimento de *software* educacional. Assim como Van der Mast (1995), afirma que a comunicação entre os diferentes membros da equipe merece atenção durante o processo, uma vez que ela se mostra difícil. Pernin (1996), porém, foca sua atenção para o processo de desenvolvimento de simulações computacionais pedagógicas (não excluindo *software* para treinamento). O autor considera apropriado propor um ambiente de produção integrado que automatize as tarefas de concepção e realização. O ciclo que orienta a construção das simulações está ilustrado na Figura 15. Nele, as tarefas de especificação do cenário pedagógico, da representação da interface e da modelagem da simulação ocorrem em paralelo, com o auxílio de formalismos pertinentes a cada uma das fases e com ferramentas automatizadas para construção e validação de protótipos. As motivações declaradas por ele são:

- a) o desenvolvimento de simulações pedagógicas necessita da definição de um processo específico que considere o fato que diversas tarefas de especificação são realizadas ao mesmo tempo;
- b) tais tarefas devem ser realizadas de maneira independente por cada conceptor;
- c) a cooperação indispensável entre os membros demanda uma formalização precisa para as especificações de cada conceptor, sendo que tais formalismos devem ser compatíveis entre si.

O modelo proposto chama-se MARS (*Modèle, Scénario, Adaptation et Représentation*), e traz à discussão a noção de espaços de trabalho, onde intervém cada um dos atores, munidos de competências bem específicas e de tarefas bem delimitadas. Um espaço de trabalho é definido como um ambiente autônomo, onde cada um dos atores dispõe de formalismos, de dados e de ferramentas que lhe permitem produzir especificações e produzir um resultado avaliável. A Figura 16 esquematiza o modelo.

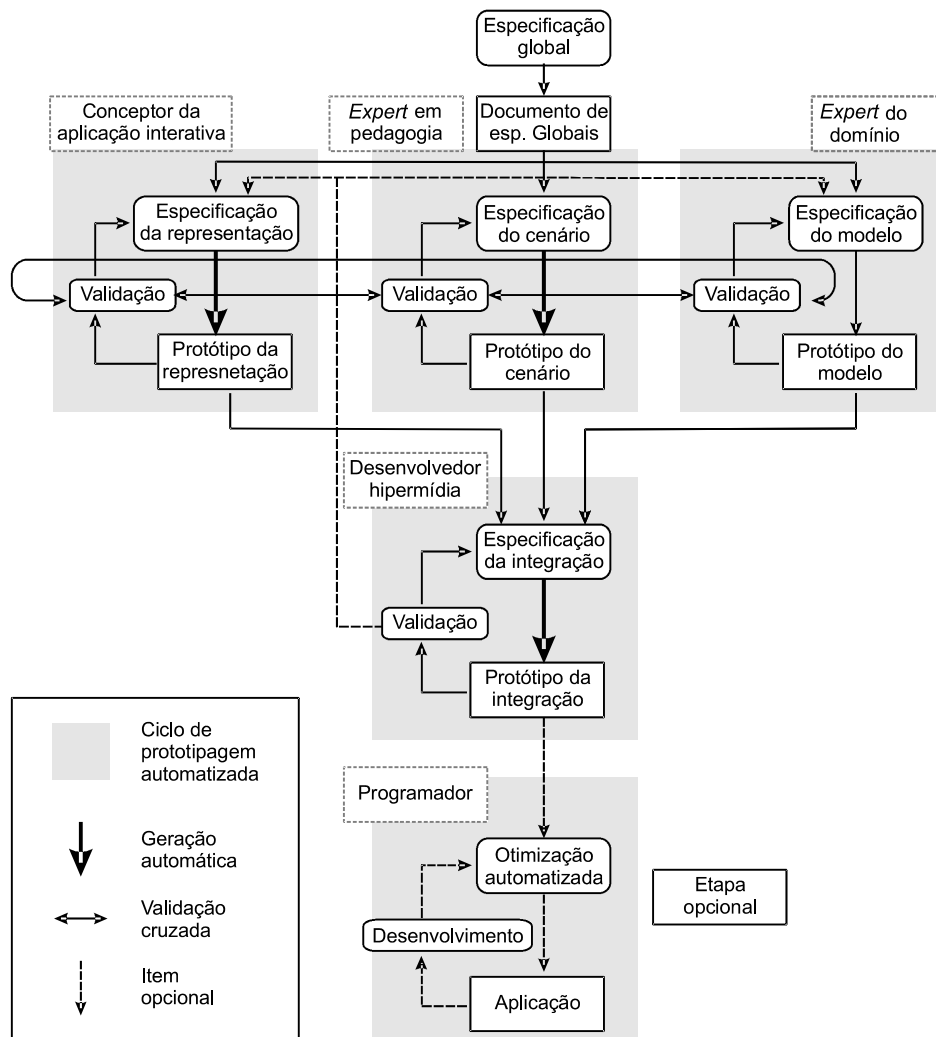


Figura 15: Ciclo de desenvolvimento proposto por Pernin (1996, p. 102).

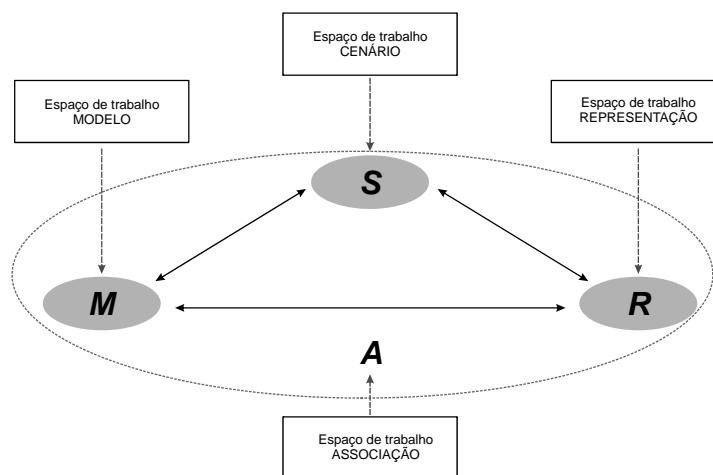


Figura 16: Espaços de trabalho do modelo MARS (PERNIN, 1996, p. 114).

Pernin (1996) define cada espaço de trabalho da seguinte forma: o espaço M é utilizado pelos especialistas do domínio, na tarefa de modelar a simulação. O espaço A é onde se realiza a integração da aplicação, através da associação dos resultados produzidos pelos espaços de trabalho M, R e S. O espaço R será utilizado pelos responsáveis pela realização da aplicação: *designers*, programadores e outros profissionais necessários. O espaço S, por fim, será utilizado por professores, para a definição de cenários pedagógicos.

Com esta revisão espera-se ter abordado alguns dos problemas mais freqüentes e algumas das soluções propostas. As metodologias pesquisadas apontam para algumas direções comuns.

- a) O projeto pedagógico (didático, instrucional, etc.) é uma etapa distinta do processo, e este deve ser concluído com o auxílio de profissionais de educação e especialistas do domínio. Esta etapa não consiste simplesmente em tentar adequar / transportar alguma teoria educacional para um conjunto de regras que define a abordagem didática do *software*.
- b) As etapas de avaliação são necessárias.
- c) A equipe de desenvolvimento deve ser multidisciplinar, e no ciclo deve estar especificado cada profissional que participa de determinada fase.
- d) Uma abordagem iterativa que prevê a construção de protótipos parece ser favorecida.

Prossegue-se com a revisão, abordando a partir deste ponto técnicas para envolver o usuário no processo de desenvolvimento. No final deste capítulo de revisão bibliográfica, as conclusões acima serão retomadas e somadas com as demais, para avaliar o caminho a seguir pela metodologia proposta neste trabalho.

2.5 TRAZENDO O USUÁRIO PARA DENTRO DO PROCESSO

De acordo com a revisão de metodologias de desenvolvimento de *software* feita nas páginas anteriores, conclui-se que integrar o usuário no processo é um aspecto determinante do sucesso do produto. No entanto, a dificuldade dos usuários em lidar com o computador chega a ser folclórica. O reflexo desta situação é visível nos custos e riscos associados a um projeto. Myers (1994), sobre uma pesquisa envolvendo dezenas de projetos, verificou que em média 48% do código de um sistema interativo é dedicado à interface. Quanto ao tempo total de desenvolvimento, consome-se com a interface o equivalente a 48% do tempo de projeto de todo o sistema, 50% de toda a implementação e 37% da manutenção. Segundo o mesmo

autor, economias da ordem de milhões de dólares podem ser realizadas em interfaces onde o usuário comete poucos erros, pois o tempo de descrição das tarefas a serem realizadas seria minimizado, a distração do usuário reduzida e os treinamentos eliminados. Ou seja, a interface interfere economicamente na utilização de um sistema⁷.

Para atingir estes objetivos são inseridos profissionais de *design* e ergonomia nas equipes, um movimento feito por grandes *software houses* internacionais. Em relação à percepção da importância destas competências no meio industrial, a Figura 17 mostra informações de empresas norte-americanas e dinamarquesas.

	KMD	Danfoss	B&O	IBM	AIR	Microsoft
Composição da equipe						
Fatores humanos						
Psicologia						
Design visual						
Engenharia						
Antropologia						
Ciências da computação						
Atividades						
Estudos de campo						
Avaliações com heurísticas						
Prototipagem						
Testes de design						
Testes de usabilidade						
Testes pós-lançamento						

Continua

⁷ Os aplicativos aos quais Myers (1994) referiu-se eram sistemas comerciais de grande porte. No entanto, optou-se por dar a informação, pois, respeitadas as dimensões, esta situação também se verifica no presente caso.

continuação

		Canais de comunicação					
Relatórios							
Pôsteres							
Vídeo							
Protótipos							
Reuniões							
		Participação do usuário					
Testadores							
Designers							
Sendo observados							
		Usabilidade no processo de desenvolvimento					
Cedo (<i>design</i>)							
Tarde (avaliação)							
Durante							

Legenda




	Alto, predominante		Algum, médio		Pouco, nenhum
---	--------------------	---	--------------	---	---------------

Figura 17: Extraída de Borgholm e Madsen (1999, p. 4).

A fim de retratar com maior detalhe a relevância das atividades relacionadas à ergonomia e ao *design* com participação do usuário, cita-se que estão refletidas em um conjunto de normas internacionais que buscam orientar as etapas de produção e avaliação de interfaces homem-computador. Este conjunto é composto pelas normas ISO 13407 e ISO 9241. A Norma ISO 13407 – Processo De Concepção Centrado No Operador Humano Para Sistemas Interativos, trata das recomendações para as atividades de concepção durante o ciclo de vida, fornecendo um guia de fontes de informação e de normas que relacionadas à abordagem centrada no usuário (BASTIEN, SCAPIN 1993). A Norma ISO 9241 – Requisitos Ergonômicos Para Trabalho Em Escritório Com Terminais Visuais, é um conjunto de normas que confere à usabilidade o *status* de qualidade ergonômica. Para a avaliação de interfaces são propostos conjuntos de *guidelines*. A Figura 18 mostra os itens desta norma. Observando-a percebe-se que os itens 10 a 17 referem-se à usabilidade da interface.

Lista das partes da Norma ISO 9241 – Requisitos Ergonômicos Para Trabalho Em Escritório Com Terminais Visuais	
Parte	Título
1	Introdução geral
2	Guia para requisitos de tarefas
3	Requisitos para displays visuais
4	Requisitos para teclados
5	<i>Layout</i> de estações de trabalho e requisitos posturais
6	Requisitos ambientais
7	Requisitos para displays
8	Requisitos para displays coloridos
9	Requisitos para dispositivos de entrada, menos o teclado
10	Guias para diálogo
11	Especificação de usabilidade
12	Apresentação da informação
13	Orientando o usuário
14	Diálogos de menu
15	Diálogos de comando
16	Diálogos de manipulação direta
17	Diálogos de preenchimento de formulário

Figura 18: - Itens da norma ISO/CD 9241.

2.6 ATIVIDADES DE PROJETO – LEVANTAMENTO OU ESPECIFICAÇÃO DE REQUISITOS

No início de qualquer projeto de *software* está a etapa de levantamento de requisitos. O primeiro conjunto de requisitos, muitas vezes pouco preciso, pode definir questões importantes como a estimativa da complexidade e do tamanho da aplicação e a escolha da(s) ferramenta(s) a utilizar. A qualidade dos levantamentos anteriores à codificação, por sua vez, é fundamental para fazer projeções a respeito de itens como: modelagem do *software* e estimativa de horas/programação até o primeiro protótipo. Assim, devido à sua grande importância, são apresentadas, nesta revisão, três ferramentas para realizar a primeira coleta de requisitos. O critério para a escolha destas ferramentas foi o fato de todas proporcionarem a participação dos usuários nesse processo.

A primeira ferramenta configura uma lista de questionamentos para orientar a elaboração do projeto educacional, descrita por Van der Mast (1995). O projeto é dividido em três níveis (macro, micro e meso), de acordo com a Figura 19.

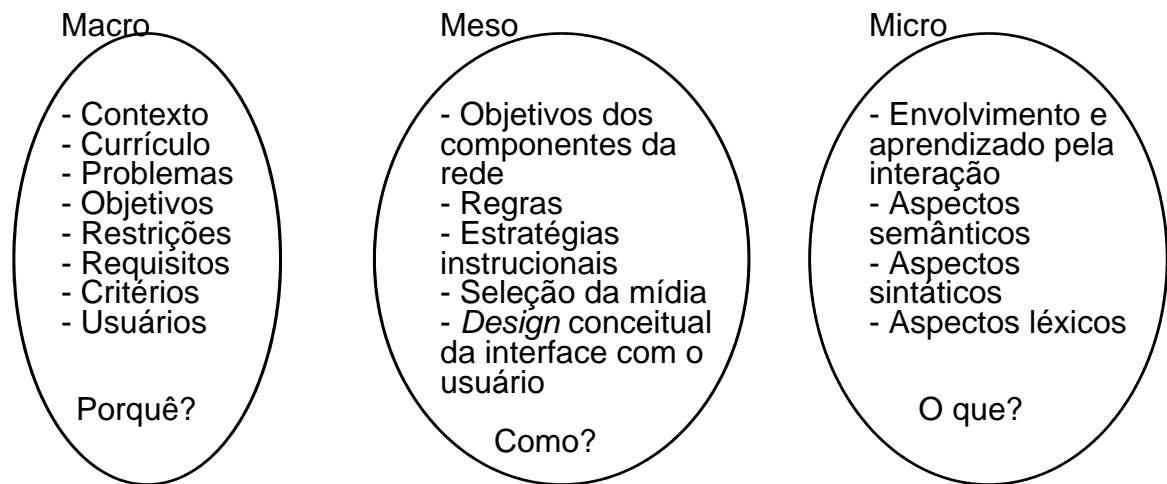


Figura 19: Reapresentação dos níveis que compõe o jeito de pensar, de Van der Mast (1995, p. 92).

Outra ferramenta que pode ser bastante útil, e que é utilizada principalmente por equipes ágeis, são reuniões com o formato JAD (*Joint Application Development*). Neste processo de grupo, os participantes interagem livremente, o que substitui a técnica de empregar entrevistas com usuários para determinar requisitos de um sistema. Segundo Carmel et al. (1993) participam das reuniões: um facilitador; usuários, gerentes e desenvolvedores; um secretário; um observador. Segundo os mesmos autores, uma sessão tem cinco fases: definição do projeto; pesquisa; preparação; reunião; elaboração do documento final. Ainda segundo o mesmo autor, problemas relacionados à JAD são: grande dependência do desempenho do facilitador; geração de idéias não-efetivas; participação e domínio da discussão por apenas alguns membros; não resolução de conflitos.

A terceira ferramenta sugerida para auxiliar no levantamento inicial de requisitos do projeto educacional é nativa da engenharia de produção, chamada *Design Macroergonômico* – DM – proposto por Fogliatto e Guimarães (1999). Esta ferramenta, segundo os autores, foi projetada para incorporar as necessidades dos usuários em produtos e postos de trabalho. A implementação do DM contempla as seguintes etapas:

1. identificação do usuário e coleta organizada de informações acerca de sua demanda ergonômica;
2. priorização dos itens de demanda ergonômica;

3. incorporação da opinião de especialistas;
4. listagem dos itens de *design* a serem considerados;
5. determinação da relação de força entre os itens de demanda ergonômica e os itens de *design*. O objetivo é identificar grupos de itens de *design* a priorizar;
6. tratamento ergonômico dos itens de *design*;
7. implementação do novo *design* e acompanhamento.

2.7 ATIVIDADES DE IMPLEMENTAÇÃO – GUIDELINES E DESIGN PARTICIPATIVO ⁸

Guidelines são recomendações ou princípios de *design*, ou seja, cabe ao *designer* escolher segui-las. Na literatura, podem ser encontradas quer sob a forma de recomendações gerais ou sobre algum item específico. Em Lewis e Rieman (1994), encontram-se os seguintes exemplos: agrupar componentes que executam funções parecidas; dar mais destaque visual aos componentes mais usados; usar telas parecidas para funções parecidas; usar a cor como um código secundário e não pôr informações demais na tela. Schneiderman (1998), por sua vez, dedica um capítulo de seu livro exclusivamente às *guidelines*. Lá estão enumerados princípios para a construção de telas; mensagens; entrada de dados; componentes de interação; ajuda e uso da cor. Apesar de seguir *guidelines* ser uma prática encorajada para evitar problemas de *design*, isso não significa que exista um comprometimento com um projeto de interface centrado nas necessidades do usuário. Para alcançar este objetivo deve ser utilizada uma abordagem baseada em *design* participativo.

A cultura de *design* participativo, segundo Spinuzzi (2002, p. 3), originou-se na Escandinávia, no início dos anos 80, com o projeto UTOPIA, desenvolvido pela Universidade de Aarhus, Dinamarca. O trecho a seguir foi retirado do referido artigo e ilustra a proposta do movimento.

⁸ Também referenciado como abordagem escandinava e *design* cooperativo.

O projeto (UTOPIA) era uma colaboração entre os acadêmicos da universidade e o sindicato dos digitadores, que queriam participar do processo de inserção de computadores na produção de um jornal. Por ter orientação marxista, estes pesquisadores percebiam o ambiente de trabalho como um campo de conflito entre trabalhadores e patrões. Apoiados pelo sindicato, os desenvolvedores buscavam uma forma pró-ativa para que os trabalhadores implementassem uma democracia no espaço de trabalho. Uma parte importante desta estratégia foi o uso de *mock-ups*, que permitiu que trabalhadores e desenvolvedores tivessem uma linguagem comum.

Uma abordagem de *design* parecida surgiu uma década depois nos Estados Unidos, com motivações e orientações políticas diferentes, porém fundamentada nas práticas do projeto UTOPIA. Este conjunto de práticas foi batizado de *design* contextual. Constituíam-se em um conjunto de estratégias para promover um *design* centrado no cliente, e que sugeria o uso de técnicas como observação em campo e análise da tarefa. Em seguida, protótipos baseados nestas modelagens deveriam ser construídos e testados.

Ainda segundo Spinuzzi (2002), é importante ressaltar que naquela época (1980), 90% dos trabalhadores escandinavos eram sindicalizados, e que a orientação política (marxista) da pesquisa era muito forte. Outro dado importante fornecido pelo autor é que o aumento da eficiência estava claramente excluído dos objetivos do projeto, pois era considerado sinônimo de aumento de trabalho e desvalorização das habilidades. Isto culminou na conclusão, feita no projeto UTOPIA, de que era preciso dar poder de decisão aos trabalhadores para que eles pudessem dar alternativas às iniciativas da gerência. Nos Estados Unidos, porém, a situação era diferente. Apenas 15% dos trabalhadores eram sindicalizados no início da década de 90, de forma que foi necessário encontrar outras vias para dar voz aos trabalhadores.

A título de comparação com outras técnicas, apresenta-se a Figura 20, que mostra alguns métodos de *design* participativo organizados conforme o nível de integração (usuário/desenvolvedores). A Figura 21 mostra uma comparação entre JAD (*Joint Application Development*, modelo de reunião dirigida, bastante utilizada por equipes ágeis) e *design* participativo.

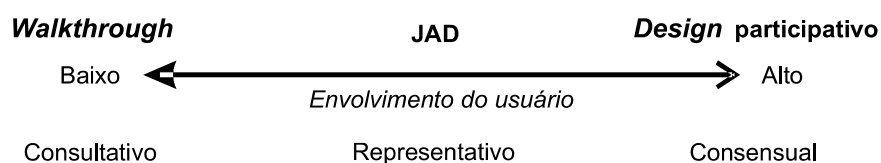


Figura 20: Intensidade do envolvimento do usuário. Extraída de Carmel et al. (1993, p. 42).

Ponto de comparação	JAD	Design participativo
Crítérios para validação	Quantitativos: otimização econômica, índices de <i>performance</i> , diminuição de tempo.	Qualitativos: democracia, aprendizado e educação mútuos, resolução de conflitos.
Background / teoria	Dinâmica de grupos e engenharia de <i>software</i> .	Relações no trabalho e aprendizagem em grupo.
Objetivo	Melhorar sistemas.	Melhorar o ambiente de trabalho.
Origens	Indústria (Estados Unidos da América e Canadá)	Governo, sindicatos e universidades escandinavas.
Temas	Trabalho em equipe, <i>design</i> acelerado.	Democracia no local de trabalho, contexto social, democracia industrial, humanização.
Atividades-alvo	A reunião (tempo limitado e agenda pré-estabelecida).	Os processos de grupo (a satisfação é o limitante, agendas negociáveis).
Ênfase técnica	Na estrutura.	Na criatividade.
Perspectiva em relação aos usuários	<ul style="list-style-type: none"> - Tanto os usuários operacionais quanto os gerentes de projeto são considerados usuários. - A seleção de usuários acontece pelo critério de competência. - Usuários são vistos como uma das fontes de informação. 	<ul style="list-style-type: none"> - Usuários são os usuários operacionais. - Participação dos usuários é regra. - Usuários são vistos como a principal fonte de informação.

Figura 21: Comparação entre *design* participativo e JAD. Extraída de Carmel et al. (1993, p. 44).

Diversas técnicas de construção de protótipos foram criadas para alcançar o máximo envolvimento dos usuários. A mais simples delas é o *mock-up*⁹, que tem a função de servir de elemento de comunicação entre *designers* e usuários, de modo que possam compartilhar habilidades e experiências. O objetivo é simular o trabalho. No projeto UTOPIA, especificamente, *mock-ups* substituíram gráficos e diagramas de fluxo de dados por que os trabalhadores não conseguiam entendê-los (SPINUZZI, 2002). A prototipagem cooperativa, por sua vez, objetiva ser uma atividade articulada de usuários e *designers*, ao invés de destes simplesmente utilizando requisitos fornecidos pelos usuários. O objetivo é experimentar uma situação real de trabalho com a futura ferramenta computacional. Espera-se que isso permita

⁹ Usualmente referido como protótipo de baixa fidelidade.

entender o comportamento do usuário e do sistema em casos de interrupções do trabalho. A diferença do protótipo para o *mock-up*, no contexto de desenvolvimento de sistemas informacionais àquela época, é que os últimos eram construídos com ferramentas de autoria como HyperCard® (SPINUZZI, 2002). Outra técnica de prototipagem classificada como *design* participativo é PICTIVE (Plastic Interface for Collaborative Technology Initiatives Through Video Exploration), apresentada por Muller (1991). PICTIVE combina artefatos com pouca tecnologia e gravações em vídeo. Os primeiros são objetos físicos que representam funcionalidades do sistema, e por serem de simples produção e operação, permite que todos contribuam com idéias. Estes materiais se dividem em duas categorias: de escritório e os preparados pela equipe de desenvolvimento (ícones, diálogos, etc). A idéia é que o *mock-up* produzido seja modificado pelos usuários em tempo real. O vídeo é a documentação do processo.

Muller (1991) ressalta ainda que é adequado utilizar PICTIVE em projetos que têm disponíveis usuários com conhecimento da tarefa. Por esta razão, o autor não aconselha seu uso em projetos de desenvolvimento de novas tecnologias. CARD (*Collaborative Analysis of Requirements and Design*), por sua vez, é uma técnica participativa que objetiva analisar e redesenhar o fluxo da tarefa em sistemas computacionais, através de jogos de cartão, informais e semi-estruturados. Cada ícone ou componente da tela é impresso em um cartão, e enquanto o usuário descreve uma determinada tarefa, o facilitador mostra o cartão relacionado. Em seguida, o usuário é convidado a expressar suas idéias para melhorar algum aspecto do sistema em estudo através dos cartões. Assim, ele representa um fluxo melhorado da tarefa, mostrando um cartão após o outro, sendo o procedimento filmado e analisado posteriormente (MULLER, 2001). Para uma melhor visualização do conjunto destas técnicas, a Figura 22 mostra no eixo vertical em que nível a cooperação acontece, e no horizontal quando a técnica deve ser empregada.

As dificuldades de utilizar uma abordagem de *design* participativo foram enumeradas por Bodker e Iversen (2002). As autoras afirmam que, uma vez que essas atividades são pouco estruturadas, torna-se mais difícil implementá-las, e que por este motivo, simplesmente inserir o usuário no processo não basta para assegurar o sucesso. Assim, ressaltam que é necessário adotar uma postura de estruturação e controle desta integração, o que significa ir além do fascínio com o primeiro contato com o usuário em campo. Bjercknes (1993), por sua vez, afirma que há obstáculos para a adoção de *design* participativo comercialmente, e dá algumas sugestões para uma implementação eficiente e efetiva. Um dos pontos de partida é

dar-se conta de que, mesmo que usuários e desenvolvedores estejam dispostos a cooperarem, estas atividades interferem em seu trabalho, em seus compromissos. Por isso, deve ser especificado no contrato quanto tempo será disponibilizado para a participação. Incluir pessoas com habilidade e poder de decisão para intervir caso seja necessário (como em casos de tensões não resolvidas) também é uma estratégia para controlar o processo.

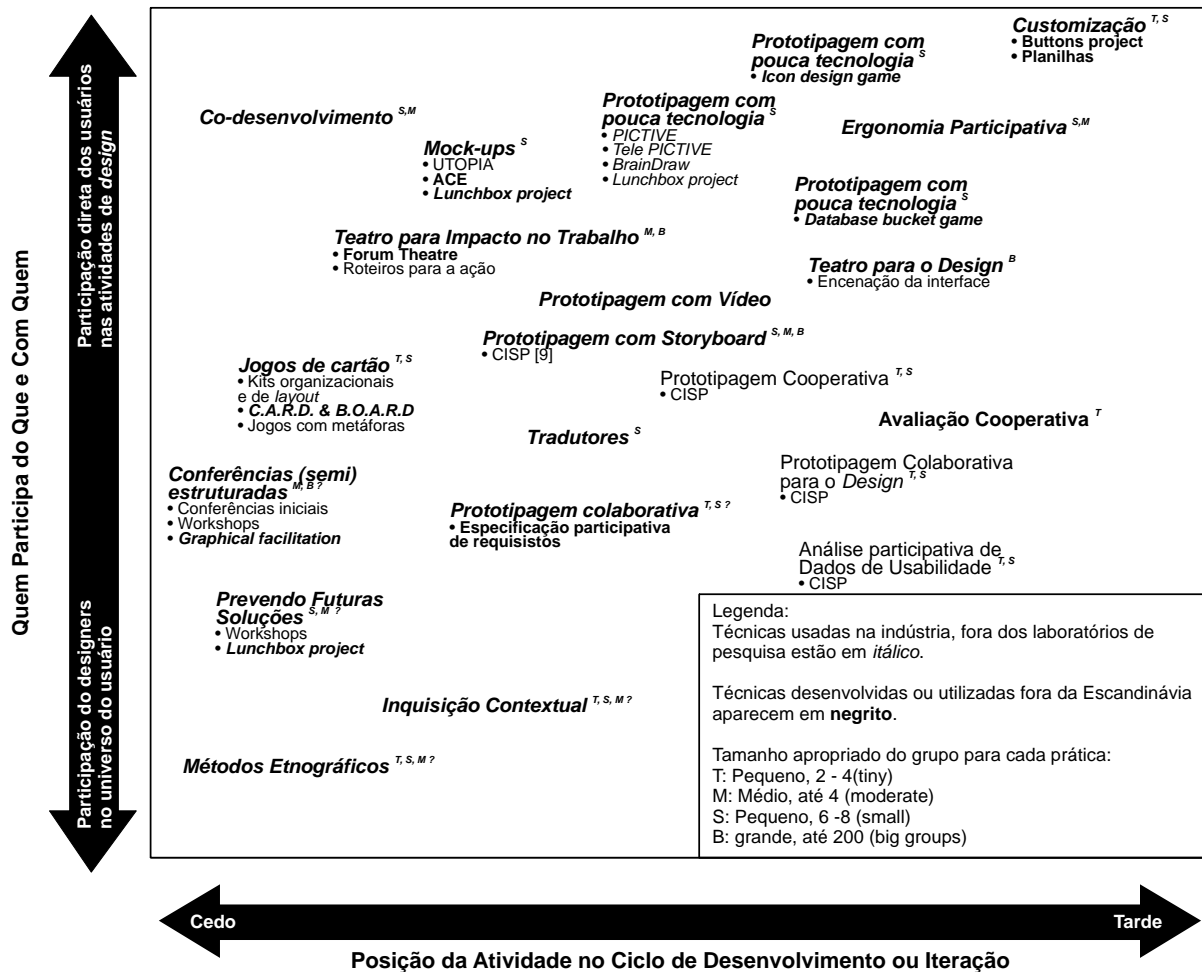


Figura 22: Taxonomia das práticas de PD, segundo Muller et al. (1993, p. 27).

Sem fazer uma crítica direta às técnicas de *design* participativo, mencionando ao invés a abordagem de *design* centrado no usuário, Rubin (1994) destaca que envolvê-los no projeto pode trazer complicações, pois a equipe de desenvolvimento talvez não saiba interpretar as reações dos usuários. Schneiderman (1998), por sua vez, relata que o envolvimento extensivo de usuários no processo pode: encarecer e talvez atrasar o desenvolvimento; gerar antagonismo com quem não está envolvido ou cujas sugestões são rejeitadas; forçar *designers*

a comprometer seus projetos para satisfazer participantes incompetentes e simplesmente gerar oposição à implementação. Em relação à iteratividade, Myers (1994) adverte que a intuição de um *designer* de como consertar um problema observado pode estar errada, de modo que a versão nova pode ficar pior que a anterior. Finalmente, o *design* iterativo pode ser demorado e caro. Bailey (1993) afirma que dados experimentais dão suporte à idéia que mudanças introduzidas para resolver um problema de usabilidade acabam gerando outro problema. O mesmo artigo indica que iterar um *design* ruim melhora-o bastante, mas este método não faz com que o *design* fique tão bom quanto ele seria se tivesse sido originalmente bem projetado. Outra informação interessante extraída deste artigo é que a primeira iteração dos *designers* apresentou menos erros de *design* do que a última dos programadores. Resumindo, a iteração não substitui bons *designers*.

2.8. ATIVIDADES DE AVALIAÇÃO DA INTERFACE

A seguir serão abordadas as atividades de avaliação, divididas em duas partes principais: avaliação da usabilidade e avaliação preditiva (relacionada aos aspectos educacionais). Na primeira parte serão revistas técnicas de interação homem computador, como modelagem do usuário, *cognitive walkthroughs*, *guidelines*, análise heurística e *checklists*. Na segunda encontram-se processos de avaliação de interface voltados a *software* educacional: TICESE, MAQSE, questionário MICROSIFT e modelo JIGSAW. Todas são atividades feitas depois da construção do programa ou de protótipos.

2.8.1 Modelagem da performance do usuário: KLM - *keystroke level model* e GOMS – *goal, operators, methods, selection rules*

KLM é um modelo para prever a *performance* de um usuário especialista operando uma interface dada. É baseado na forma mais primária de interação com o computador, os operadores. É útil para prever erros e dificuldades de manuseio. Foi proposto em 1980, por Card, Moran e Newell, e apesar de ser pouco utilizado atualmente, é apresentado por ser uma ferramenta de modelagem pioneira.

Para dar início à modelagem, deve ser fornecida uma tarefa (tarefas grandes devem ser divididas); os métodos para realizá-la (seqüência de passos - expressa em operadores - memorizada pelo usuário para executar tarefas mínimas da operação do sistema); os comandos utilizados para realizá-la; os parâmetros da habilidade motora do usuário e os

tempos de resposta do sistema. Parte-se do pressuposto que o usuário modelo executa a tarefa sem cometer erros e empregando os métodos de forma otimizada (menor caminho).

Também proposto por Card, Moran e Newell (HOCHSTEIN, 2002), GOMS consiste no uso de métodos empregados para atingir certo objetivo. Se há mais de um método para atingir um objetivo, são aplicadas regras de seleção. Segundo Hochstein (2002), pode ser classificada como uma ferramenta de interação homem computador que suporta a descrição, a predição e a prescrição. Segundo a autora, os principais eixos de crítica ao modelo GOMS se relacionam à não consideração do erro e da aprendizagem e ao não reconhecimento do paralelismo durante o processamento de informação.

2.8.2 Cognitive walkthroughs

Os *Cognitive Walkthroughs* têm como objetivo imaginar as ações dos usuários durante a primeira vez que estão usando uma interface do tipo *walk-up-and-use* (por exemplo, caixa eletrônico). Lewis e Rieman (1994, cap. IV), em seu livro eletrônico, apresentam o seguinte exemplo sobre o uso de uma máquina copiadora.

O usuário quer fazer uma cópia, e sabe que a máquina deve estar ligada. Então ele aperta o botão Ligar e passa para a próxima ação. Porém esta história não é crível. Pode-se esperar que um usuário saiba que uma máquina, para funcionar, deva estar ligada. Mas porque ele iria esperar que ela não estivesse ligada quando ele quer realizar a tarefa? Mesmo que ele imagine que está desligada, ele consegue achar a chave? Esta chave tem algum rótulo? Depois que o usuário aperta este botão, como ele sabe que a máquina ligou?

2.8.3 Checklists

São verificações que podem ser conduzidas por não especialistas em IHC, que diagnosticam problemas gerais da interface. A qualidade da verificação está diretamente relacionada à da lista. Segundo Cybis (2003) a avaliação através de *checklists* apresenta as características seguintes:

- a) não exige especialistas em ergonomia;
- b) sistematiza a avaliação, o que garante a consistência dos resultados mesmo quando a ferramenta é aplicada por diferentes avaliadores;
- c) facilita a identificação dos problemas de usabilidade, devido à especificidade das questões;

- d) reduz a subjetividade;
- e) reduz o custo.

O ErgoList¹⁰, um checklist eletrônico produzido pelo LabiÚtil (Laboratório de Utilizabilidade) da Universidade Federal de Santa Catarina, e disponibilizado na internet, é um exemplo deste tipo de ferramenta.

2.8.4 Análise heurística

Realizar uma avaliação através de heurísticas significa seguir uma estratégia. Essa estratégia está baseada em certos parâmetros que o avaliador deve levar em consideração ao fazer sua inspeção. Assim, para realizar uma avaliação heurística, deve-se escolher este conjunto de parâmetros. Esta é a grande diferença entre *guidelines* e heurísticas, pois as primeiras são regras para resolver um problema, como se fossem uma receita para construir uma boa interface. As segundas, por sua vez, são mais flexíveis, permitindo que o *designer* encontre sua própria solução.

Bastien e Scapin (1993) fornecem este tipo de orientação em um relatório de pesquisa intitulado Critérios Ergonômicos para Avaliação de Interfaces. Nele, são estabelecidas estratégias para verificação de problemas a respeito dos seguintes itens: orientação; carga de trabalho; controle explícito da tarefa; adaptabilidade; gerenciamento de erros; consistência; significado dos códigos e compatibilidade. Alguns destes critérios estão divididos em sub-critérios, sendo cada um deles apresentado com sua descrição, justificativa e comentários.

Outro conjunto de heurísticas é fornecido por Molich e Nielsen (1990). Eles realizaram uma pesquisa com 77 *designers* e programadores, da indústria e acadêmicos, para investigar se conseguiam encontrar problemas de usabilidade em uma interface. A hipótese dos autores é que as *guidelines* não orientam o *design* de maneira adequada, por não serem específicas e porque os manuais são muito grandes. Como solução, propõe nove heurísticas de usabilidade (Figura 23), identificadas pela experiência dos autores, e que, segundo os mesmos, estavam implícitas ou explícitas na maioria das listas.

¹⁰ O ErgoList está disponível na URL <http://www.labiutil.inf.ufsc.br/ergolist>

1. **Diálogos devem ser simples e naturais.** Simples significa não ter informações irrelevantes. Natural significa uma ordem que faça sentido no contexto da tarefa.
2. **Fale a linguagem do usuário.** Use palavras e conceitos compreensíveis, que façam parte do universo do usuário. Não use termos específicos ou de engenharia.
3. **Minimize a carga de memória do usuário.** Não o force a lembrar de coisas de uma ação para outra. Deixe a informação na tela até que ela não seja mais necessária.
4. **Seja consistente.** Os usuários devem ser capazes de aprender uma seqüência de ações em uma parte do sistema e aplicá-la em outra.
5. **Forneça retorno (*feedback*).** Os usuários devem saber o resultado de suas ações.
6. **Forneça saídas claras.** Se o usuário entrar numa parte do sistema que não o interessa, ele deve conseguir sair rapidamente, sem causar danos.
7. **Forneça atalhos.** Atalhos ajudam usuários experientes.
8. **Boas mensagens de erro.** Permitem que o usuário saiba qual o problema e como corrigi-lo
9. **Prevenção de erros.** Quando se escreve uma mensagem de erro deve-se perguntar se ele não pode ser evitado.

Figura 23: As nove heurísticas de Molich e Nielsen (1990, p. 249).

Em Nielsen (1992) é mostrado um procedimento para aplicar estas heurísticas, baseado na observação de que nenhum avaliador consegue detectar todos os erros de usabilidade de uma interface, e que diferentes avaliadores acharão diferentes problemas. Assim, cada avaliador deve analisar a interface separadamente. Em seguida, estes documentos devem ser combinados para formar um só. Molich e Nielsen (1990) submeteram quatro interfaces à avaliação de acordo com esta metodologia. As conclusões são que o sucesso da análise está intimamente relacionado à qualidade do avaliador, e profissionais experientes e com formação em ergonomia têm melhor *performance* que especialistas em computação. Além disso, afirma que algumas interfaces são mais difíceis de avaliar heurísticamente do que outras, e que esta diferença acentua ainda mais a importância de avaliadores experientes. Os autores também afirmam que este tipo de avaliação não deve ser feito por um único avaliador (recomendam de três a cinco profissionais). Os autores ainda citam algumas vantagens do uso deste método: é rápido e barato; é fácil motivar a equipe a conduzir um experimento deste tipo e pode ser usado desde cedo no processo de desenvolvimento. A Figura 24 mostra a performance dos avaliadores para um dos experimentos. Nela, cada coluna corresponde a um avaliador, e cada linha corresponde a um problema de usabilidade. Quadrados negros

significam que o problema foi apontado pelo avaliador. Os avaliadores foram classificados de acordo com a quantidade de problemas que encontraram

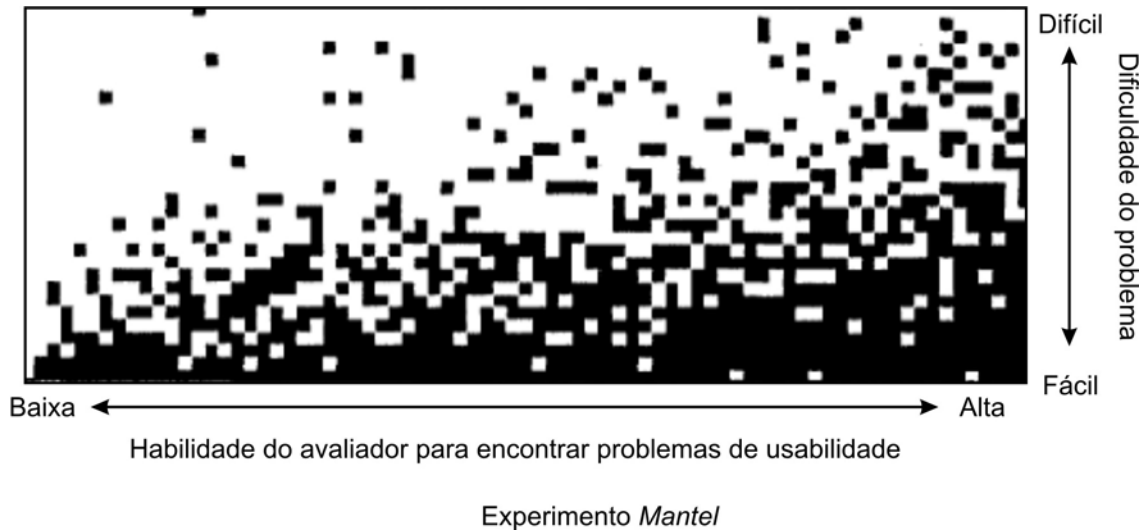


Figura 24: *Performance* dos avaliadores com uso de heurísticas. Extraída de Molich e Nielsen (1990, p. 253).

2.8.5 Avaliação com usuários

O estudo dos testes com o usuário é um assunto de grande importância para a ergonomia. Em Rubin (1994), por exemplo, encontram-se desde tutoriais passo a passo para sua realização, o momento de utilizá-los e sugestões para configuração do ambiente (sala, laboratório) onde serão realizados.

No entanto, podem ser feitas críticas em relação à participação de usuários nestes testes. Myers (1994) é um dos autores que questiona seu desempenho, afirmando que, como os usuários normalmente são voluntários, é possível que estejam mais interessados e motivados que o usuário real. Além disso, quando sabem que estão participando de um teste, eles se comportam de maneira diferente. Em relação à quantidade de usuários necessária para realizar um teste, Nielsen e Landauer (1993) apresentam um modelo baseado na distribuição de Poisson para encontrar problemas de usabilidade, concluindo que 5 usuários encontram cerca de 80% dos problemas. É importante ressaltar que o teste com usuários deve ser executado, pois é o que têm maior capacidade de encontrar problemas.

Rubin (1994) faz algumas considerações em relação à escolha de participantes, conforme a lista a seguir.

- a) Documente o teste e defina os termos com precisão.
- b) Escolha o número de participantes de acordo com o grau de confiança desejado para o teste, a quantidade de recursos disponíveis, a disponibilidade do tipo de participante desejado, o tempo disponível para preparar e rodar o teste.
- c) Inclua pelo menos alguns participantes menos competentes, hábeis e experientes.
- d) Cuidados ao usar participantes internos, membros da equipe ou funcionários.

Foi extraída do livro digital de Lewis e Rieman (1994, cap.V) uma citação a respeito de aspectos éticos a serem considerados ao realizar testes com usuários. Os depoimentos são um tanto dramáticos, mas servem ao propósito de proteger os sujeitos de constrangimentos. Os autores recomendam voluntariado e uso de termos de consentimento assinados pelo usuário.

Já ouvimos falar de usuários que deixaram a sala de teste às lágrimas, e de uma pessoa em um estudo psicológico que teve de ser removida sob sedação e de ambulância, pois não tinha sido capaz de resolver o que era um “simple puzzle lógico”. Outro aspecto a ser seriamente considerado é o constrangimento. Algumas pessoas talvez se sintam mal se um vídeo delas usando o sistema seja mostrado a terceiros, ou mesmo se imaginarem que seu nome pode estar associado a *performances* fracas.

Outro método para testar interfaces com usuários é chamado pensar em voz alta (*think aloud*). O plano é dar uma tarefa para o usuário realizar e pedir que ele fale enquanto isso. Esta sessão deve ser gravada para análise posterior.

2.8.6 Sistemas de monitoramento

Segundo Cybis (2003), sistemas de monitoramento são *softwares* que capturam as ações (toques no teclado, clique e movimentação do mouse, acesso a componentes e diálogos, etc.) do usuário. Têm a vantagem de não provocar constrangimento (como uma câmera de vídeo pode provocar, por exemplo), sendo úteis como instrumentos de apoio. O autor cita os programas *MS CamCorder*® e *Lotus ScreenCam*®. Deve-se lembrar que é possível inserir no protótipo funções para realizar estes procedimentos¹¹. Porém, é importante referir que

¹¹ Como, por exemplo, em um *spyware* ou um *cookie*.

algumas atitudes neste sentido têm sido motivo de disputas judiciais, pois diversos usuários entenderam que foi invadida sua privacidade.

Concluída a revisão a respeito de avaliação de interfaces, prossegue-se, nos próximos parágrafos, com a apresentação de alguns modelos para avaliação preditiva de *softwares* educacionais, cujo objetivo é ser um instrumento para avaliar os benefícios futuros da compra e uso de determinada ferramenta.

2.9 AVALIAÇÃO PREDITIVA DE *SOFTWARE* EDUCACIONAL

A separação entre métodos de avaliação da preditiva e de usabilidade de um *software* educacional foi observada por Squires e Preece (1996) em importante artigo, no qual os autores esboçam a motivação para a criação do método por eles denominado JIGSAW¹². Os autores afirmam que se lida com a usabilidade de uma maneira *ad hoc* durante a avaliação de programas educacionais, não havendo consideração das características de usabilidade da ferramenta para atingir seus objetivos específicos. Squires e McDougall (1994) avaliam, por sua vez, que não é suportada a avaliação para diferentes estratégias de ensino ou tipos de *software*. Julgam também que os critérios técnicos têm uma valorização excessiva, pois estes não fundamentam a educação. Esta preocupação em particular fundamentará os trabalhos de Squires nos anos seguintes à publicação daquele livro.

Nas subseções seguintes serão apresentados três *checklists* para avaliação de *software* educacional.

2.9.1 Questionário MicroSIFT – *Micro computer software information for teachers*

Squires e McDougall (1994) afirmam que este é um dos instrumentos mais conhecidos para avaliação preditiva, por ter sido produzido por uma organização pioneira, fundada em 1979, com fundos federais norte-americanos. O guia data de 1982, sendo formado um conjunto de formulários e um guia do avaliador. Em Gamez (1998, cap. V) encontram-se as seguintes críticas ao guia, na versão eletrônica de sua dissertação.

¹² Será abordado mais adiante.

A decisão de categorizar itens em termos de conteúdo, qualidade instrucional e qualidade técnica denota uma abordagem não integrada. Uma inspeção mais profunda dos itens educacionais mostra uma preocupação muito limitada quanto à temática de educação e usabilidade. Dos 14 itens das secções 1 e 2, apenas um aborda especificamente esta integração entre usabilidade e aprendizagem: Gráficos/ cores/ sons são utilizados por apropriadas razões instrucionais? Em suma, o MicroSIFT *checklist* não atende às especificações de integração entre a temática da aprendizagem com a usabilidade.”

2.9.2 Método JIGSAW

A inspiração para a criação deste método foi, segundo Squires e Preece (1996), a separação entre critérios educacionais e de usabilidade. Por ter inspiração construtivista compreende a aprendizagem como um fenômeno que envolve os estudantes, o ambiente e a ferramenta. Desta forma, a preocupação com a forma que os critérios de usabilidade impactam na aprendizagem se justifica. De acordo com este modelo, a tarefa de aprender envolve dois tipos de atividades:

- a) entender os conceitos relacionados ao domínio em estudo;
- b) entender os conceitos pré-requisitos.

A tarefa operacional (usar o *software*) também envolve duas atividades:

- a) compreender a interface;
- b) compreender o sistema operacional, hardware e periféricos sobre os quais se opera.

É ainda necessário que o estudante entenda a integração destas duas áreas. Todas estão representadas na Figura 25 que esquematiza o modelo JIGSAW.

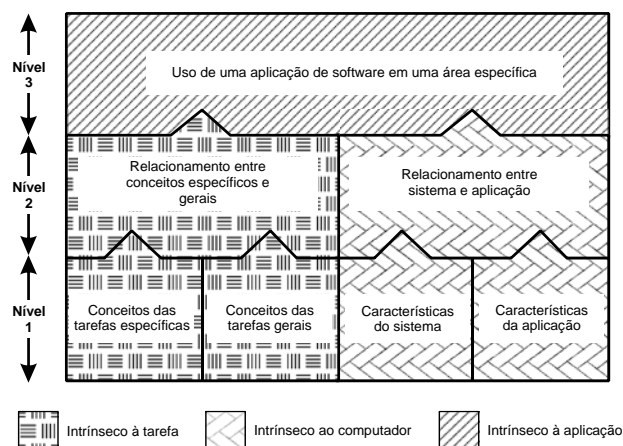


Figura 25: Modelo JIGSAW. Extraído de Squires e Preece (1996, p. 19).

2.9.3 MAQSE – Manual para avaliação de qualidade de *software* educacional

Apresenta-se como um manual dividido em três seções: uma para qualquer modalidade de *softwares*, outra para documentos hipermídia e outro para *softwares* educacionais disponíveis na internet. Em cada seção são definidos objetivos, divididos em fatores. Cada fator pode ser decomposto em diversos subfatores, avaliados separadamente. São também identificados critérios para avaliação dos subfatores. A Figura 26 mostra como exemplo o plano de avaliação para o objetivo **Usabilidade**, fator **Operacionalidade**.

Cada um dos critérios é avaliado segundo a classificação a seguir:

- 0,95 – 1,00 => Elevada qualidade de *software*
- 0,90 – 0,94 => Qualidade boa, devendo-se resolver em paralelo os problemas citados.
- 0,60 – 0,89 => Qualidade mediana; problemas existentes resultaram em um produto final pobre.
- 0,00 – 0,59 => Sem qualidade; problemas existentes não justificam o uso do produto.

Em Gamez (1998, cap. V) encontra-se a seguinte crítica ao MAQSE.

É confusa a ramificação proposta na estrutura do método, ou seja, a divisão entre fatores, subfatores e critérios. A aplicação do método permite apenas verificar os aspectos gerais do *software*, mas não permite identificar os problemas de não conformidade com as recomendações ergonômicas. O método é bastante subjetivo, pois solicita a opinião do avaliador a respeito de uma dada característica, a partir de uma escala variando entre 0, 0.25, 0.50, 0.75, e 1.0. O fato de o resultado final ser dado a partir de uma nota é controverso, pois este tipo de tratamento prima pela classificação em detrimento da identificação pontual dos problemas específicos do produto em cada critério.

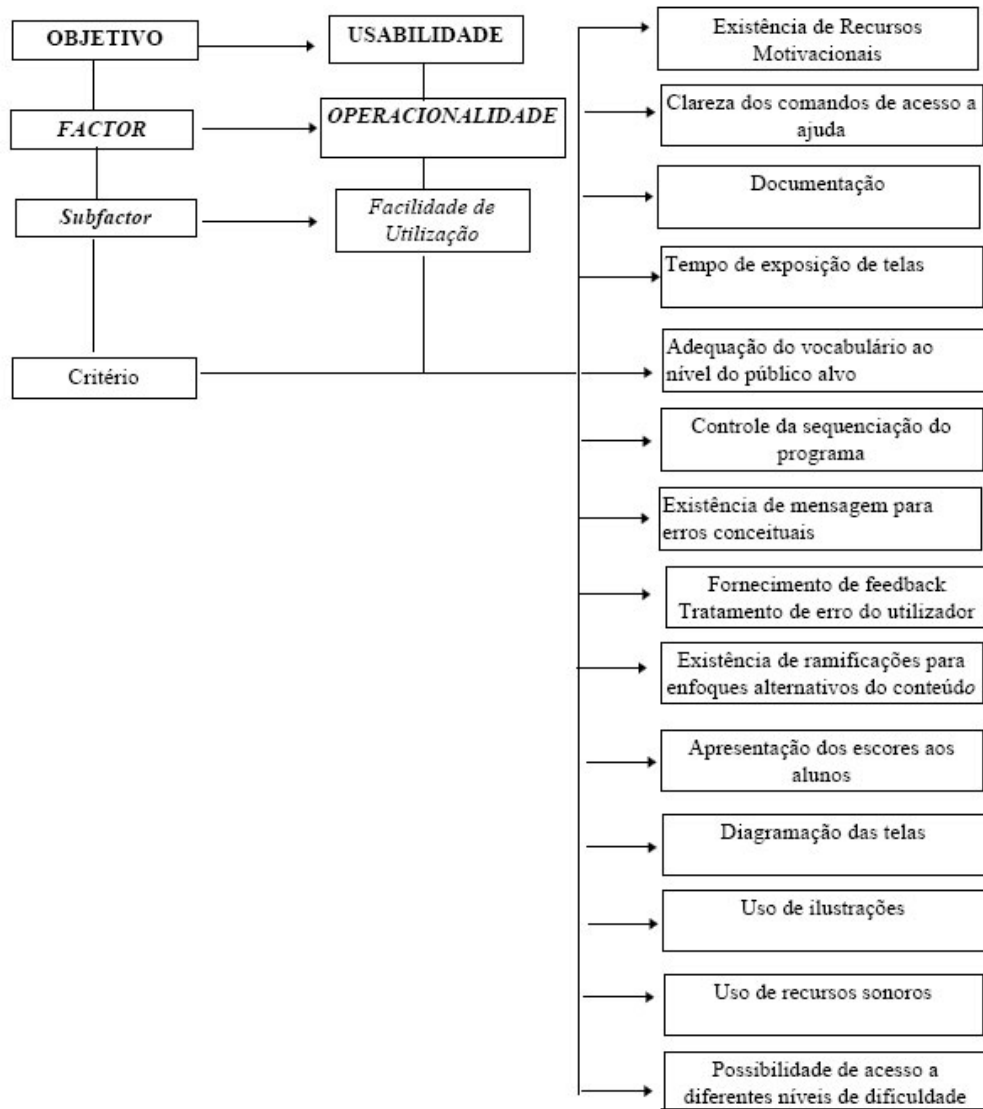


Figura 26: Estrutura de avaliação do MAQSE, de Campos *apud* Gamez (1998).

2.9.4 TICESE – Técnica de inspeção e conformidade ergonômica de *software* educacional

O TICESE (GAMEZ, 1998) é um modelo de avaliação proposto a partir do método JIGSAW (SQUIRES *apud* GAMEZ, 1998), do método MAQSE (CAMPOS *apud* GAMEZ, 1998), dos critérios ergonômicos de Bastien e Scapin (1993) e da revisão de métodos de avaliação preditiva de *software* feita pelo autor. Apresenta-se sob a forma de manual, composto por três seções:

- a) descrição e justificativa dos critérios;
- b) diretrizes para o tratamento quantitativo da informação;
- c) formulário de inspeção ergonômica de *software* educacional.

A Figura 27 mostra a estruturação do módulo de inspeção ergonômica, com seus critérios e sub-critérios.

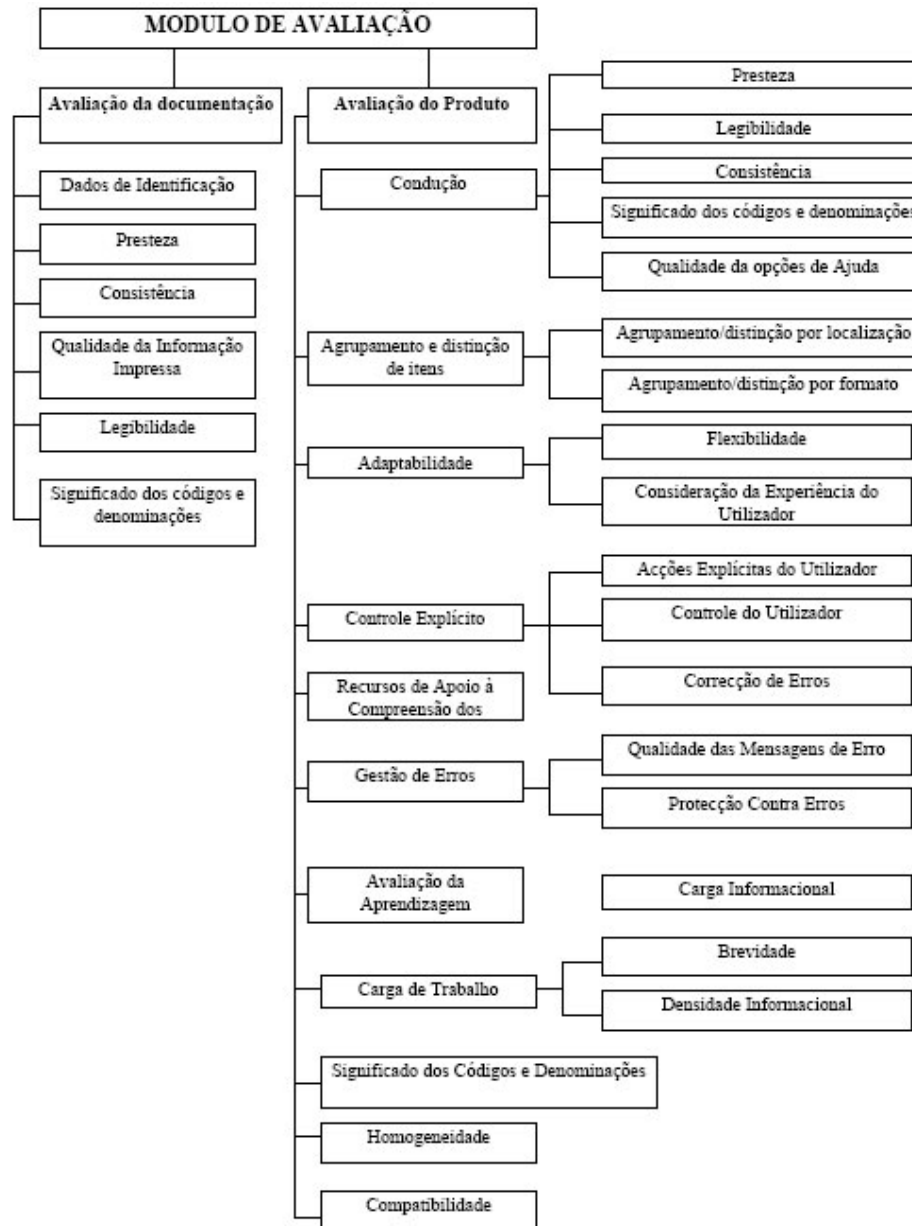


Figura 27: Módulo de avaliação do TICESE. Extraída de Gamez (1998).

O avaliador deve responder às questões do *checklist*, classificando-as em três níveis, segundo sua pertinência, conforme segue.

- a) Se a questão não se aplica, o peso é 0 (zero).
- b) Se a questão se aplica parcialmente o peso é 1 (um).
- c) Se a questão se aplica, o peso é 1,5 (um e meio).

Cada questão é também pontuada de acordo com a resposta dada: à resposta sim se atribui valor 1 (um); à resposta não se atribui valor 0 (zero); à resposta mais ou menos se atribui valor 0,5 (meio).

Pode-se observar que a TICESE recebe, como o próprio autor admite, influência de diversos outros trabalhos. A atribuição de pesos diferentes conforme a pertinência e a resposta dada (sim, não, mais ou menos) está presente no guia MicroSIFT. A integração de questões de usabilidade e educacionais foi, declaradamente, devido aos textos de Squires.

No entanto, por mais completos que sejam tais guias, não se pode utilizá-los para inferir sobre a eficiência dos *softwares* em relação à aprendizagem, pois seu objetivo é fazer uma avaliação da usabilidade, levando em consideração aspectos pedagógicos. Por este motivo, julga-se essencial realizar um experimento em sala de aula, para verificar esta questão. Por ser demorado, custoso e complexo, não faz sentido que seja realizado pelo professor que está decidindo se utilizará ou não a aplicação. Acredita-se que esta tarefa deva ser delegada aos produtores do programa, que podem fornecer estas informações como um capítulo da documentação.

2.10 CONSIDERAÇÕES A RESPEITO DA REVISÃO BIBLIOGRÁFICA

Para finalizar o apanhado de idéias apresentado nesta revisão, cabe citar os trabalhos considerados mais importantes para a elaboração da metodologia. O primeiro deles é o de Grønbaek et al. (1997), que no modelo CESD apóiam o uso de práticas de *design* participativo, inseridas em um ciclo onde as etapas se fundem e se sobrepõem. Desta forma, atividades de *design* e análise de requisitos interferem no gerenciamento do projeto, por exemplo. A pesquisa de Van der Mast (1995), por sua vez, forneceu suporte ao posicionamento adotado nesta pesquisa a respeito do projeto educacional. De acordo com o autor, esta etapa deve ser realizada separadamente e antes de todas as demais, que irão, ao longo do ciclo, reportar-se aos critérios nele definidos. Além disso, o ciclo iterativo proposto é parecido com o de Van der Mast, no sentido que, depois de estabelecido o projeto educacional, não se volta mais a esta etapa. O terceiro *benchmark* é a tese de Pernin (1996), que com o ambiente M.A.R.S. oferece uma linguagem de modelagem adequada para promover a integração dos envolvidos com a produção.

Inspirado nestas propostas pretende-se estabelecer uma metodologia de desenvolvimento de *software* educacional, que leve em consideração as dificuldades

apontadas na literatura. Estas estão relacionadas, principalmente, à integração dos membros da equipe (dela podem fazer parte: profissionais da educação, do *design*, da computação e especialistas no domínio abordado pelo programa, além de professores e estudantes) e às iterações de *design* para a construção dos protótipos. Em relação às primeiras, observa-se os seguintes aspectos:

- a) o gerenciamento de uma equipe com tal caráter multidisciplinar é complexo. Seu gerente deve ter condições de dialogar com todos os membros da equipe;
- b) a comunicação entre os membros desta equipe também apresenta obstáculos, por causa da falta de um vocabulário comum e pela dificuldade em compreender globalmente a importância dos objetivos dos demais especialistas para o projeto.

De sua parte, os modelos de desenvolvimento de *software* educacional consultados entram em acordo nos seguintes pontos:

- a) todos os processos estudados incorporam a avaliação da interface, e a maioria prevê atividades de avaliação preditiva (relacionada aos aspectos educacionais) e algum tipo de avaliação com estudantes. Medidas e avaliações de qualidade são necessárias, pois determinam os critérios para avançar para a fase seguinte;
- b) a construção do *software* ocorre em um processo iterativo, preferencialmente de acordo com a abordagem de prototipação evolutiva.

Contudo, na literatura sobre interação homem computador encontra-se algumas ressalvas em relação a esses pontos. Em primeiro lugar, sobre o envolvimento de usuários no processo de *design* e avaliação de *softwares*, Bodker e Iversen (2003) afirmam que é necessário adotar uma postura de estruturação e controle desta integração, o que significa ir além do fascínio com o primeiro contato com o usuário em campo. Rubin (1994) destaca que este envolvimento pode trazer complicações, pois a equipe de desenvolvimento talvez não saiba interpretar as reações e sugestões dos usuários. De maneira complementar, Schneiderman (1998) relata que o envolvimento extensivo de usuários no processo pode: encarecer e talvez atrasar o desenvolvimento, gerar antagonismo com quem não está envolvido, ou cujas sugestões são rejeitadas, forçar *designers* a comprometer seus projetos para satisfazer participantes incompetentes e simplesmente gerar oposição à implementação.

Em relação à iteratividade, Myers (1994) adverte que a intuição de um *designer* de como consertar um problema observado pode estar equivocada, de modo que a nova versão pode ficar pior que a anterior. Ele também adverte que o *design* iterativo pode ser demorado e caro. Bailey (1993), por sua vez, afirma que dados experimentais dão suporte à idéia que

mudanças introduzidas para resolver um problema de usabilidade acabam gerando outro problema. O mesmo artigo indica que iterar um *design* fraco melhora-o bastante, mas isso não faz com que o *design* fique tão bom quanto ele seria se tivesse sido originalmente bem projetado.

Deste ponto do trabalho em diante, as pesquisas e argumentos expostos neste capítulo que se encerra serão articulados para a criação de uma metodologia para desenvolvimento de *softwares* educacionais, cujo maior objetivo é ser participativa, no sentido de oferecer a todos os membros da equipe chances iguais de colaborar com o projeto. Acredita-se que o caminho para um processo que leve à *softwares* educacionais de boa qualidade educacional passa, necessariamente, por um conjunto de atividades que confere grande importância às competências e saberes de cada especialista participante. Isso justifica o compromisso em apoiar práticas de *design* participativo, com ciclos de iteração onde todos os participantes têm voz ativa, além de testes periódicos com usuários, em atividades com diferentes graus de formalização.

3. PROPOSTA DE METODOLOGIA PARTICIPATIVA PARA O DESENVOLVIMENTO DE SOFTWARE EDUCACIONAL

No final do capítulo de Revisão Bibliográfica foram expostos os obstáculos postos às equipes que decidem produzir *softwares* educacionais, além dos pontos em que as propostas metodológicas pesquisadas entram em acordo. Estas informações formaram a base da configuração da metodologia proposta, determinando a ordem de realização das etapas e os participantes necessários em cada uma delas. Importante também foi estar ciente dos custos e benefícios de dois dos pontos mais caros ao ciclo de desenvolvimento de *software*: a necessidade de um ciclo de iteração e de avaliações com usuários. Uma vez tendo conhecimento dos prós e contras relacionados a cada um destes aspectos, foi possível tomar as decisões que moldaram a orientação técnica¹³ da metodologia proposta.

Avaliando os resultados encontrados na literatura a respeito destas questões, escolheu-se utilizar uma abordagem iterativa do desenvolvimento, através de prototipagem evolutiva e da inserção de testes com usuários. Em relação à primeira opção, acredita-se que um bom projeto educacional pode diminuir a quantidade de iterações necessárias para concluir o projeto, mantendo, dessa forma, os prazos e custos estabelecidos pela equipe sob controle. Além disso, é importante frisar que, atualmente, há uma grande disponibilidade e variedade de ferramentas para a prototipação rápida de interfaces, que contribui para tornar os argumentos anti-iteração obsoletos. Por exemplo, os cada vez mais populares processos ágeis são solidamente embasados na construção de protótipos. Quanto à segunda opção, a de realizar avaliações com usuários, argumenta-se que, se bem planejados e conduzidos, estes podem cumprir duas importantes missões: contribuir com idéias para o projeto e dar real medida da eficiência do projeto educacional e da interface.

A metodologia, portanto, se propõe a arcar com os custos da iteratividade e do envolvimento com usuários. Por isso, devem ser respondidas algumas perguntas.

- a) Como pode ser feita a estruturação da integração dos membros da equipe entre si e da integração da equipe com os usuários?
- b) Quais as ferramentas que podem ser utilizadas para gerenciar/conduzir as tarefas propostas?

¹³ Ao falar em técnica refere-se às de *design* e ergonomia.

c) Como e quando realizar avaliações de qualidade do produto?

No entanto, os seguintes pontos ficarão em aberto, pois não é possível dar uma resposta conclusiva a eles, sendo, portanto, delimitadores da metodologia proposta.

a) Em que medida esta metodologia é dependente de especialistas?

b) Há funções que podem ser exercidas por não-especialistas, ou mesmo por pessoas sem formação acadêmica na área?

c) Esta metodologia é aplicável para construção de *softwares* de médio e grande porte?

d) Um bom projeto educacional realmente diminui a quantidade de iterações, e consequentemente contribui para a manutenção dos prazos?

e) Como se dá a dependência da interface com o projeto educacional?

A metodologia posta em prática para atingir os objetivos propostos, está representada na Figura 29 ilustra as recomendações quanto ao tipo de profissional que deve participar de cada etapa. Nela são descritos os três sujeitos que participam da produção do *software*, ou seja, equipe de desenvolvimento: o especialista em educação, o *designer* e o programador. Também é recomendado que professores do domínio e especialistas do domínio possam, eventualmente, prestar colaboração.

Estes três atores principais devem, antes de iniciar os trabalhos, firmar o compromisso de tentar compreender os problemas colocados pelos colegas. O cumprimento deste acordo está relacionado à motivação e à personalidade de cada um. Os membros da equipe precisam dar-se conta que, caso o respeito pela opinião e pelos problemas alheios e o interesse sobre o trabalho dos colegas prevaleça, todos enriquecerão sua prática, pois desenvolverão novas habilidades. Isto é especialmente valioso caso a equipe planeje trabalhar junta no futuro. A Figura 30 mostra o comportamento ideal de cada sujeito em relação aos outros.

O último item a ser mencionado diz respeito à gerência do projeto. Caso se considere necessário indicar alguém para este posto (o *software* pode não ter a figura do gerente), recomenda-se que seja o especialista em educação, pois espera-se que este profissional tenha mais experiência com as questões relacionadas ao ensino e à aprendizagem. Caso isto não seja possível, deve ser escolhido em primeiro lugar o sujeito com maior experiência interdisciplinar, senão o que tiver mais disponibilidade.

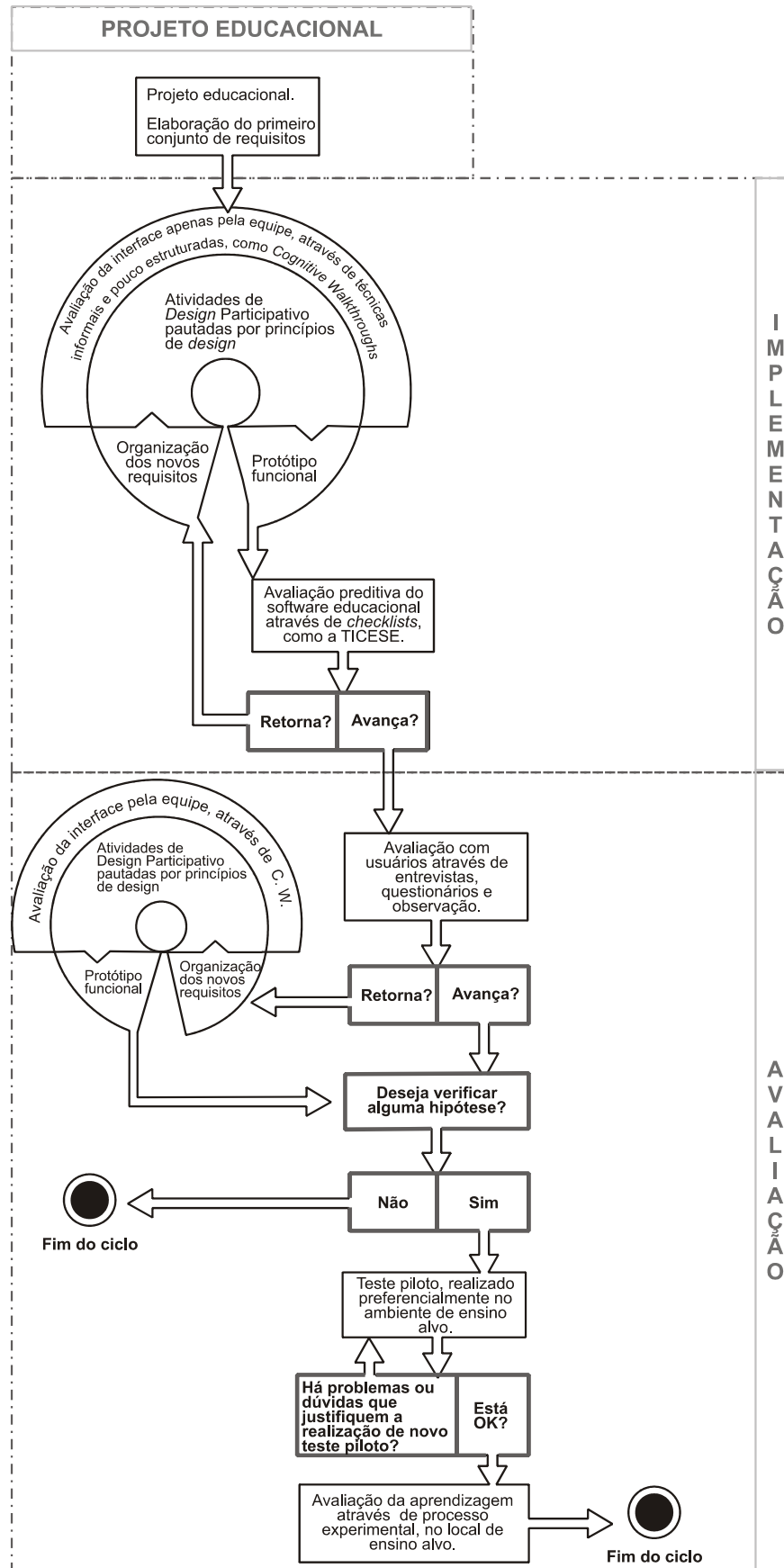
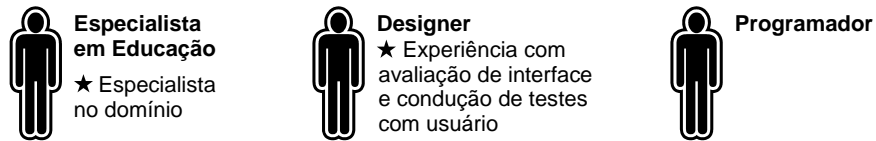


Figura 28: Metodologia proposta para desenvolvimento de *software* educacional.

Desenvolvem a ferramenta:



Além disso, convém ter acesso a:

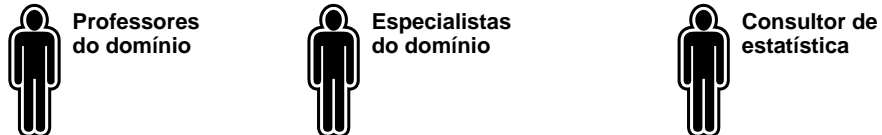


Figura 29: Sujeitos requeridos para o processo.

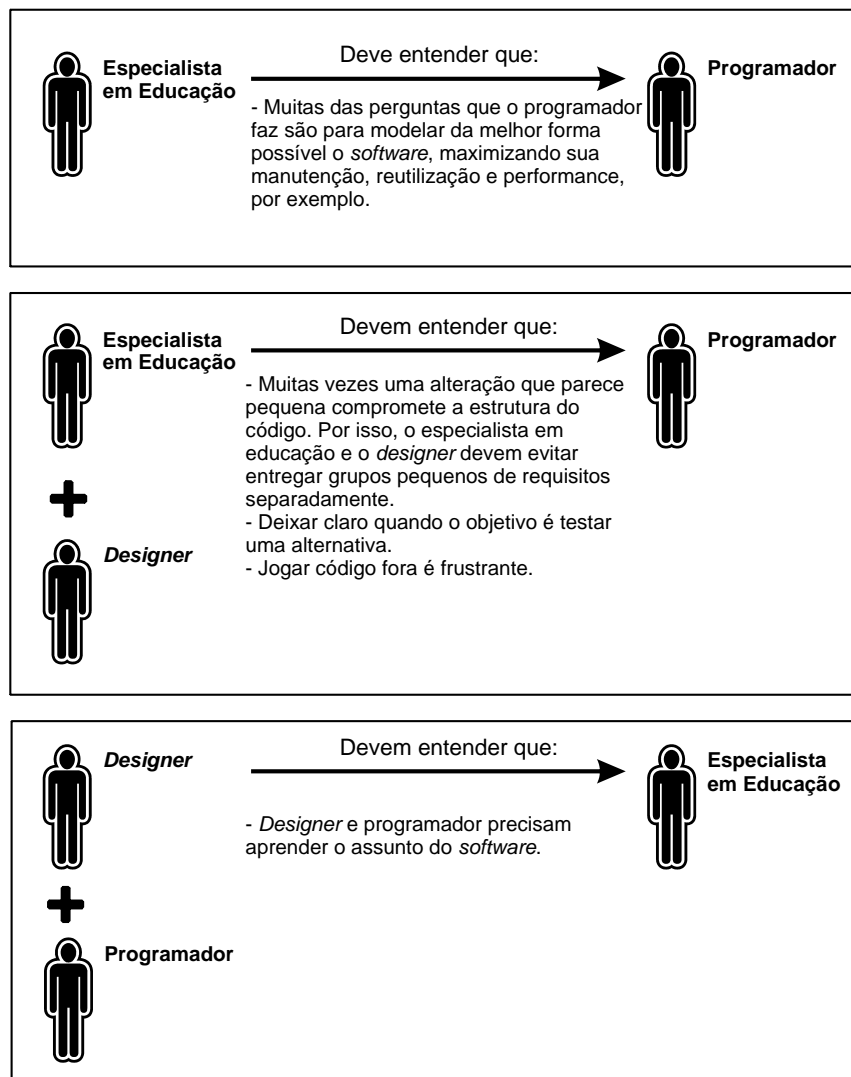


Figura 30: Relacionamento entre os sujeitos da equipe.

3.1 DETALHAMENTO DA ETAPA PROJETO EDUCACIONAL

Como se pode perceber pela Figura 28, o projeto educacional não sofre iterações. A justificativa é o alto custo associado a esta etapa. Há dois motivos para a pouca formalização desta etapa, sendo o primeiro a especificidade de cada projeto. Isto significa que elaborar o projeto educacional de uma forma rígida poderia limitar a liberdade de cada professor expressar convenientemente suas orientações teóricas e metodológicas. O segundo motivo é a extrema dificuldade (se é que há possibilidade) de transferir os princípios das teorias cognitivas de ensino e aprendizagem para o projeto. Sustenta-se que o fato de haver uma estratégia didática não significa que o programa seja construtivista, sócio-interacionista, *behaviorista* etc. Além disso, acredita-se que o significado do *software* se realiza durante seu uso, que é definido pelo professor. Buscou-se, então, evitar rótulos para esta metodologia, que não é mais adequada para projetos de orientação construtivista do que *behaviorista*. Deve-se, ainda, lembrar que ela objetiva ser útil para o maior número possível de desenvolvedores, motivo pelo qual não se restringiu a um tipo de *software*. Desta forma, diferentes tipos de inserção podem ser planejados.

A confiança nesta estratégia se deve à crença em que, se o projeto for bem feito, o *software* terá aumentadas suas chances de ser eficiente, ou seja, de facilitar a aprendizagem. Partindo do pressuposto que projeto é um instrumento formal de previsão, antecipação e exploração, assume-se que um projeto educacional deve servir como meio de traçar objetivos e determinar estratégias para alcançar a aprendizagem de determinado(s) assunto(s). Deve, além disso, levar em considerações desvios, erros e outros problemas que venham a ocorrer durante a execução da tarefa-alvo (aprendizagem). Sendo assim, seu conteúdo é diferente para cada *software*, uma vez que os projetos devem contemplar características muito particulares: a delimitação do tópico abordado; as dificuldades de aprendizagem do tópico; as orientações teóricas da equipe desenvolvedora e as experiências de seus membros. Contemplados estes itens, ele pode ser considerado satisfatório.

Recomenda-se que todos os membros participem desta etapa, pois durante seu desenvolvimento importantes decisões serão tomadas, e elas irão afetar todas as áreas de especialização envolvidas. No entanto, naturalmente, cabe ao especialista em educação dirigir os trabalhos.

Para passar para a etapa de implementação, qualquer que tenha sido a forma escolhida para criar o projeto educacional, é importante que o conjunto de perguntas a seguir tenha sido respondido.

- a) Qual a delimitação do assunto?
- b) Quais os problemas que os estudantes têm para compreender este assunto?
- c) Qual a estratégia para abordar estas dificuldades?

A partir destas respostas, podem ser traçadas estratégias para a forma de interação com o usuário, escolha de mídias, etc.

3.2 DETALHAMENTO DA ETAPA IMPLEMENTAÇÃO

Uma vez concluído o projeto educacional, deve-se avançar em direção da construção do primeiro protótipo. Esta tarefa acontece dentro do ciclo iterativo de implementação, no qual se realizam três tipos de atividades: especificação de requisitos; *design* participativo e construção do protótipo, descritas na lista que segue.

1. O ciclo tem início com as especificações para criação do primeiro protótipo funcional. Esta atividade deve ser coordenada pelo *designer* e deve contar com a participação pelo menos do programador. Nesta etapa, pode-se prescindir do especialista em educação, consultando-o quando houver necessidade de dirimir alguma dúvida. O objetivo é projetar a aparência da interface e a forma de interação que o usuário irá experimentar. Balizar a construção destes protótipos por *guidelines*, como por exemplo, as de Lewis e Rieman (1994) e Schneiderman (1998) e por heurísticas, como por exemplo, as de Molich e Nielsen (1990) e Bastien e Scapin (1993) é especialmente indicado, mesmo que a equipe conte com um especialista em *design*. Caso seja possível, pode-se desenvolver diversos protótipos funcionais para testar módulos separados do programa. De qualquer forma, os protótipos gerados – funcionais ou não – devem ser submetidos à avaliação de todos os membros da equipe, de preferência em uma sessão conjunta de *Cognitive Walkthrough*. Além disso, é conveniente mostrar tais protótipos para professores do domínio, pois eles podem dar sugestões muito ricas. Também é importante, principalmente em *softwares* de modelagem e simulação, que um especialista do domínio conheça a interface e os módulos que estão sendo criados. Provavelmente, esta será a etapa mais longa do processo. No momento em que nenhum dos membros da equipe tiver sugestões para alterações, deve-se passar para a próxima fase do ciclo de iteração: a construção do protótipo funcional.
2. O protótipo deve ser construído pelo programador. A ele cabe a responsabilidade de escolher a ferramenta que julgue mais adequada. Os demais membros da equipe são

- acionados apenas se o programador achar necessário. Depois de concluída esta fase, passa-se para a avaliação preditiva com o uso de *checklists* (recomenda-se a TICESE).
3. Por serem ferramentas simples, os *checklists* podem ser aplicados mesmo por pessoas sem experiência ou conhecimento de *design* e ergonomia. Caso seja necessário responder algum item sobre o conteúdo abordado, pode ser solicitado auxílio do especialista em educação. Esta aplicação trará um resultado, que a equipe deve avaliar. Se forem encontrados problemas, se deve voltar para o ciclo de iteração descrito supra. A lista de problemas identificada será o novo conjunto de requisitos que o próximo protótipo deve atender. Se, por outro lado, o desempenho for considerado satisfatório, e se considere que não há necessidade de alterações no protótipo, passa-se à etapa seguinte: a avaliação com usuários.

3.3 DETALHAMENTO DA ETAPA AVALIAÇÃO

O objetivo é fazer um levantamento de novos requisitos (novas sugestões) e submeter o protótipo à crítica de professores do domínio. Recomenda-se, portanto, que os professores participantes desta etapa não conheçam o *software*. Os membros da equipe que participaram desta etapa são o *designer* e o especialista em educação, este orientando o primeiro na confecção do material de apoio. Sugere-se que o levantamento de sugestões e de novos requisitos seja feito através de entrevistas. Estas podem ser conduzidas por qualquer um dos membros da equipe que se sinta em condições de ouvir professores do domínio. Isto é especialmente importante, pois, nestas reuniões, serão dadas sugestões que precisam ser corretamente compreendidas. Deve-se levar em consideração que um profissional que está sendo entrevistado para colaborar com a construção de um programa não irá se comportar como um professor, e, portanto, não irá tomar o entrevistador por aluno. Em seguida, devem ser tabuladas estas sugestões e críticas, o que irá levar a modificações no protótipo. Se for este o caso, inicia-se mais uma vez o ciclo de iteração da etapa de implementação. Porém, ao finalizar o protótipo funcional, não é necessário realizar a avaliação através de *checklists* novamente.

Ao final desta fase, chega-se à pergunta: deseja verificar alguma hipótese? Se a resposta for **Não**, os trabalhos podem ser encerrados. No entanto, se a resposta for **Sim**, será dado início ao processo de projeto de experimento. As hipóteses a serem verificadas podem ser a respeito do *design* da interface, da interação com o usuário, do projeto educacional, etc. O importante é que elas sejam formuladas como uma pergunta que aceita apenas duas

respostas: sim ou não. É imprescindível que a equipe tenha consciência de que projetar, executar e interpretar dados em um experimento é uma tarefa extremamente cara, trabalhosa e demorada, e que necessita do acompanhamento de sujeitos com habilidades muito especializadas. Será necessário, então, contar com a disponibilidade de um consultor em estatística, função que pode ser acumulada apenas pelo especialista nas variáveis em estudo (por exemplo, o *designer*, caso o experimento seja sobre a forma de interação da interface).

De toda forma, qualquer que seja o arranjo escolhido pela equipe, é fundamental que se realize um teste piloto em condições as mais próximas possíveis da realidade. Afinal, dificilmente os instrumentos de coleta e interpretação de dados não necessitarão de correções. Além do mais, é uma boa oportunidade para entender como transcorrerá o experimento no ambiente escolar.

Por último, realiza-se o experimento final. Esta avaliação acontece apenas uma vez devido ao seu alto custo, composto não apenas por fatores financeiros, mas, principalmente por: (a) tempo despendido no projeto do experimento e na elaboração de materiais para coleta de dados e do manual para análise dos resultados; (b) necessidade de condução de teste piloto, para testar o *software* e o material de coleta; (c) dificuldade de conseguir um professor que tenha uma classe com número suficiente de estudantes e que disponha do seu tempo e que incentive os estudantes a participar do teste e (d) tempo gasto na análise de dados. Deve-se ter o cuidado de documentar amplamente esta atividade, tendo em vista as inspeções pelo Comitê de Ética do órgão ao qual se está filiado ou que financia a pesquisa. Esta precaução, que protege o pesquisador e os demais participantes, pode ser tomada com a formalização das correspondências entre este e a instituição onde será realizado o teste, e pelo preenchimento de um Termo de Consentimento Livre Informado, presente no Apêndice A.

Esta é a última etapa da produção de um *software* educacional. Certamente há outras atividades a realizar, por exemplo, a produção de material de apoio para os professores e a divulgação. Porém os aspectos pós-produção não foram contemplados em razão do tempo, e não por não serem considerados importantes.

Segue, no capítulo quatro, a apresentação dos resultados obtidos com a aplicação da metodologia aqui exposta, para a construção de um *software* sobre equilíbrio químico.

4. RESULTADOS

Antes de proceder a descrição do desenvolvimento do *software*, algumas palavras devem ser ditas, em benefício de uma visão apropriada das condições em que ele ocorreu. Participaram ativamente do projeto duas pessoas, ou seja, uma equipe muito pequena, composta por um especialista em educação (que acumulou a função de especialista em estatística e foi o autor do projeto educacional) e por um *designer* (que acumulou a função de programador). Eventualmente, outras pessoas colaboraram com o desenvolvimento, em especial químicos (especialistas no domínio). O fato de nenhum dos dois membros da equipe ser químico não apresentou muitas dificuldades. Como o desenvolvimento aconteceu dentro de universidades (ULBRA – Universidade Luterana do Brasil e UFRGS – Universidade Federal do Rio Grande do Sul) e, devido aos contatos do professor especialista em educação, jamais faltou a quem pedir auxílio. No entanto, foi necessário que ambos os membros da equipe tivessem pleno domínio do assunto, pelo menos até o ponto em que o *software* o aborda.

O tempo decorrido do início à finalização dos trabalhos foi aproximadamente dois anos. Deve-se considerar que, para um projeto pequeno e simples como este é um período consideravelmente longo. No entanto, alguns fatores podem explicar esta demora, em especial a impossibilidade de dedicar-se integralmente à construção do *software*. Os demais fatores incluem a inexperiência com a metodologia, com temas educacionais e com a programação.

4.1 ELABORAÇÃO DO PRIMEIRO CONJUNTO DE REQUISITOS: O PROJETO EDUCACIONAL

Equilíbrio químico é um conceito abstrato, que exige domínio de uma vasta quantidade de conceitos subordinados, tendo sido considerado um dos mais difíceis de ensinar, envolvendo um alto índice de concepções alternativas¹⁴. Provavelmente, esta afirmação poderia ser utilizada para a maioria dos assuntos de que trata a química, pois os

¹⁴ Em Driver e Easley (1978, p.62), encontra-se a seguinte definição para concepção alternativa: “As concepções ingênuas não parecem refletir uma compreensão fragmentada de conceitos científicos. Ao invés disso, parecem fazer parte de uma estrutura que dá sentido às informações e que permite fazer predições, sendo um dos maiores obstáculos para o sucesso da instrução em ciências”.

fenômenos acontecem em nível atômico, que não está acessível aos sentidos do ponto de vista humano.

Para investigá-los é possível analisar evidências de mudanças macroscópicas, descrevê-los através de equações, fórmulas, gráficos ou símbolos. A compreensão de um fenômeno químico a partir do ponto de vista microscópico, porém, é bastante difícil. Por exemplo, a relação entre o estado macroscópico e microscópico não é simples nem direta, e a tentativa de compreendê-lo através da observação de alterações na aparência da matéria é uma concepção alternativa muito presente nos estudantes. Eles podem, por exemplo, pensar que uma molécula de água derrete quando o gelo se liquefaz. Por outro lado, a resistência dos estudantes à modelagem e linguagem matemática é notória. Camacho e Good (1989) afirmam que lhes falta inclusive conhecimento básico de operações matemáticas, como propriedades dos logaritmos, proporções, expoentes, raízes e isolar parâmetros em uma equação. Wu, Krajcik e Solloway (2001) afirmam que o entendimento de representações do nível microscópico e do simbólico é especialmente difícil, porque tais representações não são visíveis e são abstratas, enquanto o raciocínio dos estudantes se baseia fortemente em informações sensoriais. Infelizmente, para este problema não há saída: a química é inerentemente representacional e simbólica.

Some-se a isto o fato de que estudantes secundaristas não apresentam habilidades de bons solucionadores de problemas. Para investigar a veracidade dessa hipótese, Camacho e Good (1989) conduziram um experimento no qual filmaram 13 novatos (colegiais e não graduados) e 10 especialistas (doutorandos e professores universitários) durante atividade de resolução de problemas típicos de equilíbrio químico. Como esperado, os especialistas saíram-se melhor que os novatos, sendo que os professores universitários tiveram um desempenho verdadeiramente notável. O objetivo do trabalho era identificar diferenças nas abordagens de resolução de problemas destes grupos. A Figura 31 mostra tais diferenças.

Assim, sustenta-se que os estudantes apresentam três tipos principais de dificuldades: (1) em relação ao domínio de conceitos matemáticos e de estratégias de raciocínio lógico; (2) em relação ao assunto equilíbrio químico e (3) em relação a conceitos pré-requisitos para a compreensão e equilíbrio químico. Desta forma, é possível estabelecer a seguinte linha de atuação para a interface:

- a. as dificuldades de raciocínio não serão abordadas de maneira direta, ficando a cargo do professor esta tarefa;

- b. as concepções alternativas sobre equilíbrio químico serão atacadas diretamente pelo *software* (como as investigadas por Hackling e Garnett (1985); Niaz (1995); Bergquist e Heikkinen (1990))
- c. sobre os conceitos pré-requisitos, considerou-se importante dar atenção especial às concepções relacionadas à natureza particulada da matéria¹⁵. Barnea e Dori (2000) fornecem uma explicação para algumas destas concepções ao afirmarem que no ensino de ciências, insuficiente ênfase é posta no fato de que modelos são simulações da realidade, baseados em certa teoria, e que moléculas não são miniaturas dos modelos que as representam.

Heurística	Uso por sujeitos bem sucedidos	Uso por sujeitos mal sucedidos
1. Ir para outra parte do problema quando incapaz de resolver uma parte anterior.	Raramente	Freqüentemente
2. Uso do método mais simples quando dois ou mais são disponíveis.	Freqüentemente	Raramente
3. Procura por similaridades entre os problemas.	Freqüentemente	Raramente
4. Comparação das respostas com o solicitado pelo problema.	Freqüentemente	Raramente
5. Eliminação de variáveis problemáticas e trabalhar com uma variável por vez.	Freqüentemente	Raramente
6. Verificação da consistência das assunções.	Freqüentemente	Raramente
7. Verificação da consistência das respostas e justificativas.	Freqüentemente	Raramente
8. Uso de estratégia.	Freqüentemente	Raramente
9. Uso de tentativa e erro.	Raramente	Freqüentemente
10. Uso de outros símbolos químicos (seus próprios), mais fáceis de lidar do que aqueles impostos pelo problema.	Freqüentemente	Raramente
11. Uso de aproximações que facilitavam o cálculo de resultados.	Freqüentemente	Raramente

Figura 31: Emprego de heurísticas pelos sujeitos do estudo, extraída de Camacho e Good (1989, p.268).

Atendendo aos pontos listados supra, definiu-se como ponto central do *software* construído a tentativa de focar a natureza particulada. Desta forma, assume-se a hipótese de Gabel (1993), que afirma que a química pode ser ensinada utilizando-se três níveis de representação: microscópico, com átomos e moléculas; macroscópico e simbólico. A autora conduziu um experimento que indica que nas turmas onde houve instrução sobre a natureza particulada da matéria os estudantes tornaram-se mais aptos a fazer relacionamentos entre os três níveis de representação. A hipótese da autora é que, quando a natureza particulada é

¹⁵ A dualidade energia/matéria que caracteriza a compreensão atual tanto da Física quanto da Química não foi considerada por não estar no escopo dos fenômenos estudados no ensino médio.

ênfatisada, ela não pode prescindir de um ou mais tipos de representação para gerar significados. Esta hipótese é central para o projeto educacional do presente trabalho.

Outra hipótese que norteia o projeto educacional deste *software* foi apresentada por Kozma e Russell (1997), que em experimento envolvendo especialistas (sujeitos graduados em Química) e novatos (sujeitos não graduados), verificaram que os primeiros demonstravam ser mais capazes de traduzir uma determinada representação em outra. Os autores acreditam que esta habilidade é importante para a compreensão dos fenômenos químicos, pois estes não estão sob alcance da nossa percepção. Concluem que o projeto de ambientes multimídia deve promover a articulação de múltiplas formas de representação, pois isto auxiliaria os novatos adquirirem uma habilidade apresentada por especialistas. Esta é outra idéia muito cara a este projeto.

Antes de passar às etapas seguintes, deseja-se esclarecer a razão de não ter sido projetado nenhum tipo de material de apoio ao *software*. Em primeiro lugar alega-se falta de tempo, o que significa que é reconhecida importância de prover este tipo de auxílio. Em segundo lugar, o experimento (para verificação da aprendizagem do tema aliado ao uso do *software*) proposto nesta pesquisa não demanda este tipo de material. Afinal, inserções prolongadas do software não foram objeto de avaliação.

4.2 A IMPLEMENTAÇÃO

Seguindo com a descrição da aplicação da metodologia, são abordadas as iterações de *design* da interface. Ao iniciar este processo, o projeto pedagógico encontrava-se pronto, o que contribuiu para que não houvesse necessidade de mudanças bruscas no rumo a ser seguido. De toda a forma, isto não impediu o surgimento de situações imprevistas, nem que algumas escolhas tivessem que ser revistas. Por exemplo, a primeira versão do *software* tratava quase que exclusivamente do nível de representação microscópico, pois se acreditava que ele seria suficiente para alavancar a compreensão dos estudantes para níveis mais elevados. No entanto, a pesquisa de Orlandi (2004) mostrou que essa abordagem era insuficiente, pois os estudantes que usaram o *software* Le Chat (que contém uma representação muito simplista desse nível de representação), saíram-se melhor, em testes, do que aqueles usando a primeira versão do Equil (à época chamado Colisões).

Após rever esta escolha, definiu-se como prioritária a seguinte característica: a articulação dos níveis microscópico, macroscópico e simbólico, em todos os momentos da

evolução da reação. Os motivos que levaram à escolha da reação gasosa envolvendo o hidrogênio, o iodo e o ácido iodídrico (Figura 32) foram os seguintes: modelagem computacional simples, ter poucos elementos e todos serem gases; a reação é utilizada para ensino de equilíbrio químico no nível médio e a cor é uma indicação da evolução em direção ao equilíbrio. No entanto, não foram modelados todos os aspectos desta reação. Dois exemplos disto são: a não representação da reação rápida e a modelagem das colisões, que não leva consideração à orientação das moléculas. Estas informações dão suporte à hipótese que o projeto educacional determina em grande parte a aparência e até mesmo o estilo de interação da interface.

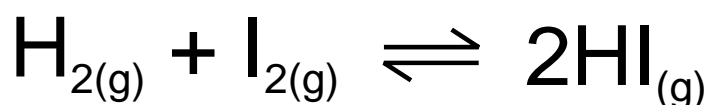


Figura 32: Reação simulada pelo *software*.

A maioria dos elementos da interface atual do *software* passou por diversos testes e modificações. Em alguns momentos houve a necessidade de relacionar itens da interface com a modelagem interna do *software*. Outras vezes chegou-se a extremos em que era necessário sacrificar uma funcionalidade em detrimento de outra, sem falar nos *bugs* pelos quais todos os programas passam. Por exemplo, durante a criação dos protótipos, houve ocasiões em que o funcionamento de certos elementos da interface não puderam ser configurados conforme desejado. Um caso pode ilustrar esta situação. Como afirmado anteriormente, uma das crenças mais caras do projeto experimental é a força que a visualização do nível microscópico teria na compreensão dos estudantes sobre equilíbrio químico. Por isso, desejava-se que ele refletisse a evolução da reação escolhida nos demais níveis (macroscópico e simbólico). Cogitava-se inclusive adicionar um botão que permitisse contar a quantidade de cada uma das espécies existentes no momento, e que houvesse coerência entre os gráficos e essa quantidade, a exemplo do que há no programa Le Chat (Figura 33). Infere-se que no Le Chat seja possível parar a simulação e verificar que a quantidade de espécies coincide com a indicada nos gráficos porque o

posicionamento das moléculas parece ser aleatório¹⁶, e não determinado, como é o caso do Equil.

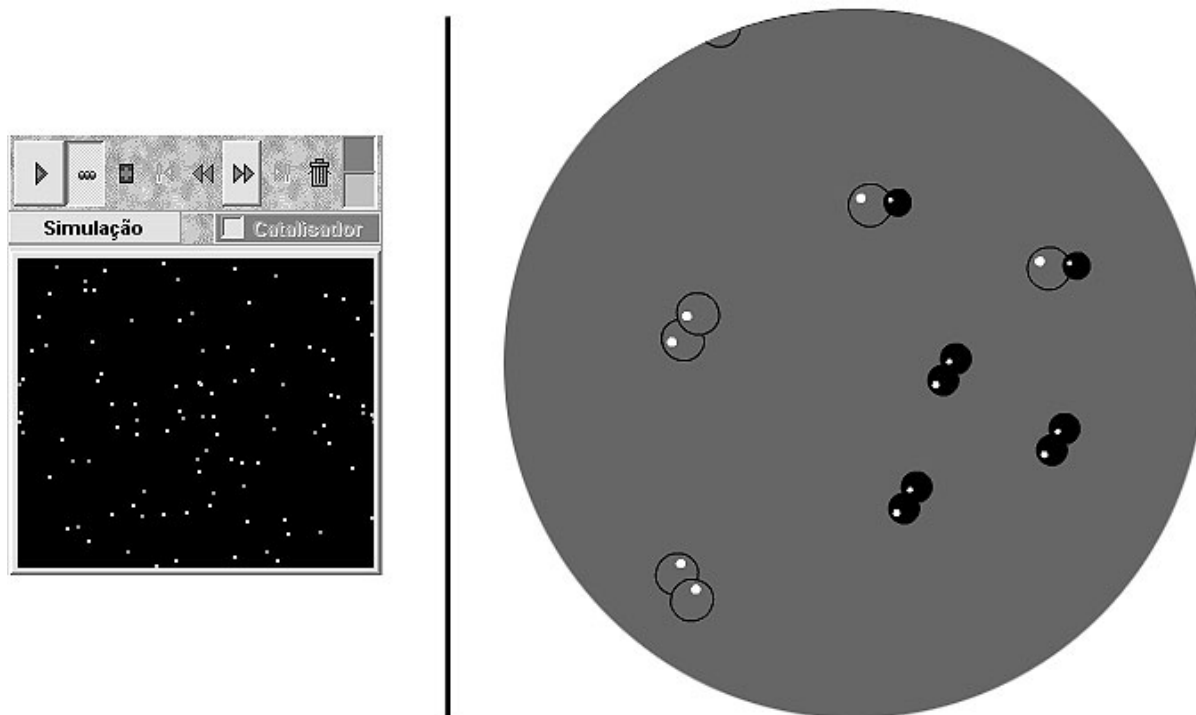


Figura 33: Representação do nível microscópico nos *softwares* Le Chat (à esquerda) e Equil (à direita), ambos em tamanho real em uma tela de 17", com resolução de 1152x864 *pixels*.

Porém, isso não podia ser realizado caso houvesse poucas moléculas colidindo. Dessa forma, criou-se um protótipo onde várias moléculas pudessem ser inseridas (25 para cada espécie). No entanto a representação do nível microscópico ficou muito prejudicada, pois cada objeto era muito pequeno, e esta idéia foi abandonada. Os gráficos, como os demais elementos da interface, foram separados do programa de colisão das moléculas. Por isso é correto dizer que há dois programas, um para a representação do nível microscópico e outro

¹⁶ O fragmento **parece ser aleatório** necessita explicação. Usou-se o verbo parecer porque não se tem acesso ao código do Le Chat. A visualização oferecida pelo software do nível microscópico poderia ser obtida (por exemplo) aleatorizando as coordenadas (x,y) dos pixels da área e colorindo-os de acordo com a quantidade de espécies determinada pela concentração no momento em que a função é executada, o que faria com que sempre houvesse a quantidade certa de espécies de cada tipo. Isto não é possível no Equil.

que controla todo o resto. Esta decisão sepultou a intenção de integrar os três níveis de representação. Isto gerou um problema de interface que nunca será resolvido: a quantidade de moléculas de uma determinada espécie não é a mesma indicada numericamente ou nos gráficos de concentração. A saída encontrada foi representar o nível microscópico dentro de um círculo, de modo que os estudantes pudessem imaginar que um **pedaço** da reação estava sendo mostrado. Isso desobrigaria a articulação das quantidades de espécies com os gráficos e números. A Figura 34 mostra uma troca de *e-mail* sobre este tema. A Figura 35 ilustra esta proposta.

Membro A pergunta - Pensei em uma solução para lidar com o problema dos gráficos. Por exemplo, caso a quantidade de moléculas seja pequena, o gráfico não vai ficar bom. Então, eu pensei em fazer a área das colisões mascarada com o círculo, para que fosse como uma parte do vidro. Para que esta abordagem seja sustentada, na tela de entrada, ao invés de mostrar as quantidades em termos de unidades (como está implícito agora), poderia mostrar em termos de mol.

Por que assim dá para fazer um gráfico de equilíbrio a partir de uma equação qualquer, que levasse em conta os fatores relevantes (quantidade de produtos, velocidades). O problema, é que talvez o estudante não entenda quando o gráfico chegar ao equilíbrio e as moléculas não terem refletido isto.

Membro B responde - Entendo. Realmente, o estudante vai seguramente prestar atenção no número de moléculas que irão estar se transformando, ele esperaria que o número de moléculas (de produtos) aumente até chegar em um valor estável no equilíbrio. E que fique, estatisticamente, estável. Neste caso, você terá que assegurar que sempre haja moléculas de reagentes e produtos no equilíbrio. Naturalmente a proporção pode alterar, mas, na média, tem que ser a que se espera no equilíbrio. Isto seguramente ocorre no COLISOES (EQUIL) anterior, mas o problema é que a média ($1/1 = \text{produtos/reagentes}$) é sobre um número tão pequeno que as flutuações são significativas, fazendo, em alguns momentos, reagentes desaparecerem (ou produtos). Assim, o estudante fica com a impressão que o equilíbrio vai e volta. Ou seja, que quando só tem produtos, atingiu o equilíbrio (reforço de uma concepção errônea), mas que pode sair e voltar. Quando tiver um número suficiente de moléculas, isto não ocorrerá, sempre haverá algum produto e reagente para que o estudante perceba que no equilíbrio, produtos e reagentes coexistem. Um ponto importante é você participar da confecção do guia, e explicitamente mencionar a natureza ESTATISTICA do equilíbrio, em nível microscópico. Isto ficou faltando. Minha sugestão é assegurar ao estudante um número suficientemente grande de produtos e reagentes (tipo 20 de cada no equilíbrio) como o mínimo. Aí você utiliza as equações para o número molar de espécies de produtos e reagentes (deve ser alguma exponencial, fala com tua mãe) para confeccionar o gráfico e consequentemente a cor do nível macroscópico.

Membro A responde - Setar a quantidade mínima de objetos em 10, por exemplo, poderia ser uma saída? O estudante não vai conseguir contar os objetos na tela cada vez que a função é executada. Neste caso, não haveria problemas de o gráfico não fechar com a tela das colisões. O que tu achas?

Figura 34: Debate sobre a quantidade de moléculas e sua representação em cena.

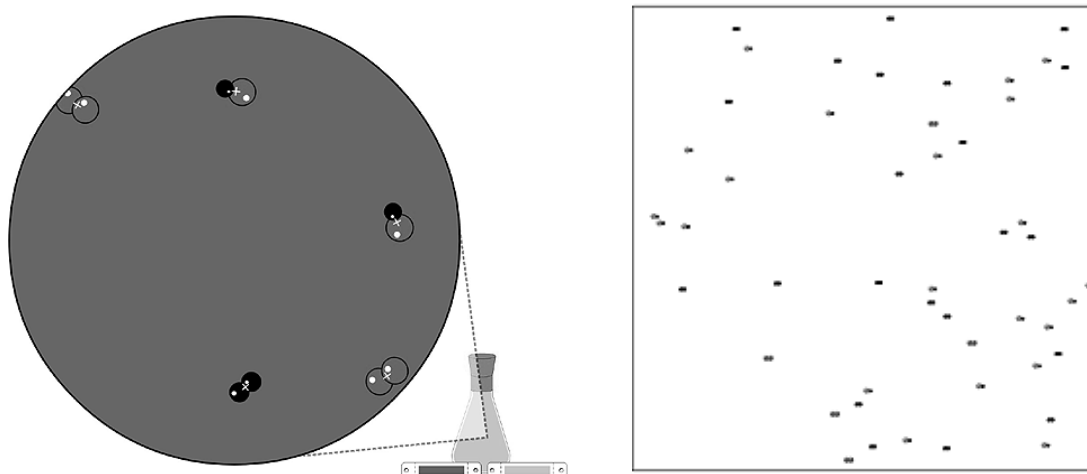


Figura 35: Duas propostas de representação do nível microscópico. A da esquerda foi adotada.

Houve ainda outra questão não prevista no projeto educacional, que se refere à quantidade máxima e mínima de moléculas, que podem estar presentes para gerar a simulação. Como dito anteriormente, foi construído um protótipo que permitia a entrada de até 75 moléculas. Neste ponto, a questão que ocupava a atenção era o desenho do gráfico e a contagem de espécies. Porém, logo ficou claro que era preciso estabelecer duas quantidades: máxima e mínima, que gerassem gráficos cujos desenhos ocupassem a mesma área e que pudessem usar a mesma escala. Por exemplo, para a reação estudada, caso se iniciasse a simulação com 20 mol.L^{-1} de H_2 e 20 mol.L^{-1} de I_2 , ao final haveriam aproximadamente 28 mol.L^{-1} de HI . Assim, o gráfico deveria ter uma marcação, no eixo vertical, que fosse de 0 a 30. Da mesma forma, em uma simulação com 5 mol.L^{-1} de H_2 e 5 mol.L^{-1} de I_2 , o resultado final seria aproximadamente 7 mol.L^{-1} de HI ¹⁷. Estes dois gráficos teriam uma diferença muito grande de altura, o prejudicaria sua compreensão. A Figura 36 mostra, à esquerda, um controle de entrada de dados através de barras deslizantes, e à direita uma das primeiras interfaces de simulação. Ambas constam no primeiro protótipo.

¹⁷ À temperatura de 308K

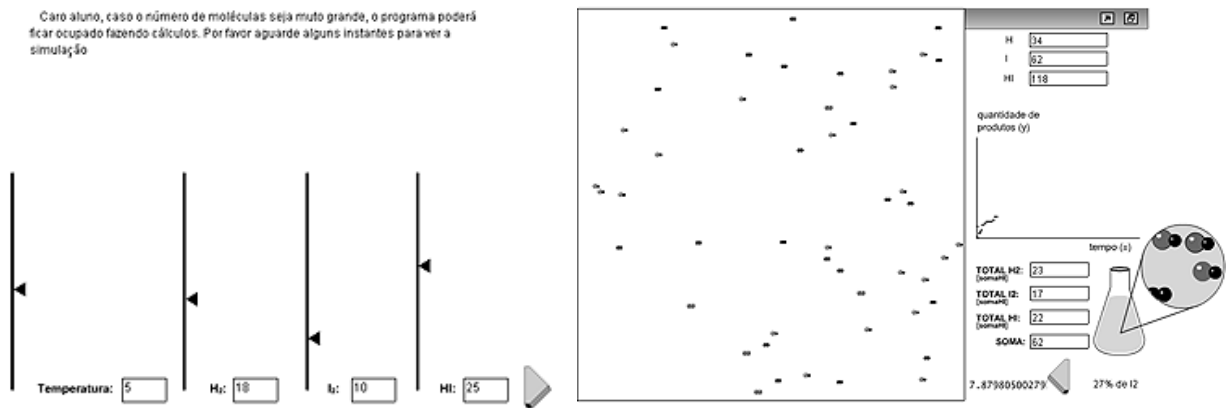


Figura 36: Interfaces dos primeiros protótipos.

Como se pode perceber na Figura 36, o frasco não tinha uma rolha para tampá-lo, um descuido que poderia ser grave se não tivesse sido notado. Na mesma figura, a imagem reduzida da ação das moléculas está sobre um fundo que muda de cor, o que poderia denotar que existe algo mais além das moléculas no frasco. O fundo da área onde acontecem as colisões é branco, quando o preto é o código usado para indicar espaço vazio. A apresentação da quantidade de cada espécie era feita através de caixas dispostas na vertical, outra clara demonstração da prevalência das representações do desenvolvedor sobre as do usuário. A Figura 37 mostra uma ampliação do botão que mostrava/escondia a direção e a velocidade das moléculas. A Figura 38 mostra a primeira versão para o módulo numérico (não estava presente na versão testada por Orlandi, 2004).



Figura 37: Ferramenta direção da molécula. Excluída das versões finais.

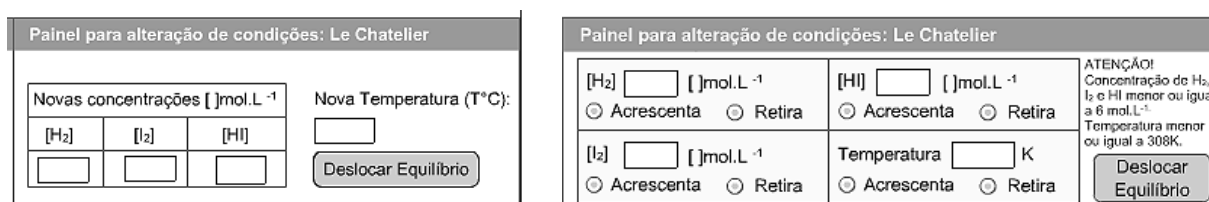


Figura 38: À esquerda, primeira versão do módulo Le Chatelier, cuja interface é um tanto confusa. Era necessário entrar a quantidade desejada seguida de um sinal de + ou –, indicando desejo de inserir ou retirar moléculas. À direita a versão final.

Como atividades de avaliação, conforme recomendado no capítulo três, foram conduzidos *cognitive walkthroughs*, pelos sujeitos que participavam da rotina de construção dos protótipos. Estas atividades não foram estruturadas de forma alguma: não havia data para acontecer, nem documentos para serem entregues, nem itens a analisar. Bastava que estivessem reunidas pessoas aptas a julgar a interface e o funcionamento do *software* para que estas sessões acontecessem. Por causa destas avaliações informais descobriu-se um erro na modelagem da simulação: as moléculas, ao colidir, retornavam formando um ângulo de 90°. Estas seções de avaliações informais certamente foram responsáveis pela aparência do *software*.

Quando se julgou que o protótipo estava pronto, foi feita uma avaliação preditiva com o *checklist* TICESE (GAMEZ, 1998). No caso do programa em questão, foram acrescentadas informações quanto à versão, nome dos produtores, criação de uma tecla para rever a simulação e configuração de valores *default*. Na versão que aborda o princípio de Le Chatelier, que apresenta um nível maior de interatividade, foi dada atenção especial ao tratamento das ações do usuário e das respostas dadas pelo programa. Conclui-se, desta forma, que consultar *checklists* é uma forma eficaz de corrigir erros ou esquecimentos de menor impacto.

As próximas seções dedicam-se aos relatos dos resultados dos procedimentos de avaliação pelos quais o protótipo passou.

4.3 AVALIAÇÃO DO *SOFTWARE* PELOS USUÁRIOS

4.3.1 Entrevista com professores de química

A avaliação do *software* com os professores foi realizada durante o mês de Agosto de 2004, através de entrevistas e de questionários sobre o tema do *software*. Após responder ambos, os professores conheciam o *software* e eram convidados a fazer sugestões. Estas entrevistas não foram gravadas. Os professores tinham 5, 7, 13, 30 e 44 anos de experiência com ensino de química. Apenas o menos experiente leciona para o nível médio. Os demais atuam pelo menos em cursos de graduação em química, e os dois mais antigos dão aula para pós-graduação (em educação).

Na entrevista, uma pergunta foi feita: que ferramentas o senhor (a) consideraria úteis em um *software* para ensino de equilíbrio químico? A Figura 39 mostra o resultado. Ela é uma prova da robustez do projeto educacional deste programa. Como se pode ver, a maioria das sugestões dos professores já havia sido implementada antes da consulta (coluna **Existente**). A exceção é feita por duas sugestões, números 8 e 11, que foram trabalhadas em conjunto e proporcionaram um grande ganho para o *software*. O resultado foi uma ferramenta que pode auxiliar os professores a explicar um dos conceitos mais difíceis de equilíbrio químico, as velocidades das reações direta e inversa (Figura 40).

Sugestão	Quantidade de citações	Implementada	Existente	Não aceita
1- Estabelecer o que seria a situação de equilíbrio.	5		X	
2- Mostrar a simultaneidade das reações inversa e direta.	4		X	
3- Alterações nas condições de equilíbrio.	4		X	
4- Comparação de diferentes sistemas.	3		X	
5- Animação das colisões e da reação ocorrendo.	3		X	
6- Representação através da cor.	2		X	
7- Aplicar o conceito em processos reais.	2			X
8- Representação da direção da equação, através do comprimento da seta.	2	X		
9- Possibilidade de determinar a constante de equilíbrio.	1		X	
10- Gráficos dinâmicos.	1		X	
11- Representação da equação balanceada.	1	X		
12 - Dar uma situação $A+B \rightarrow AB$ e $AB \rightarrow A+B$ e mostrá-las separadamente.	1			X
13- Representação do fulcro (balança).	1			X
14 - Representação de animações através de letras.	1			X

Figura 39: Sugestões dadas pelos professores.

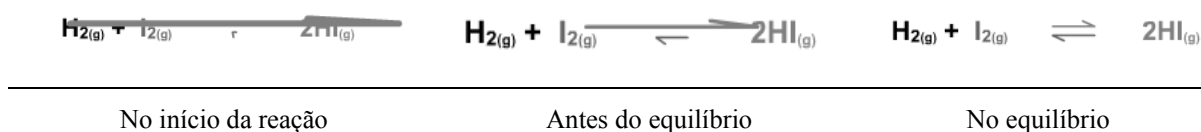


Figura 40: Ferramenta para acompanhamento da evolução das velocidades inversa e direta. Disponível nas versões 1 e 2 do *software*.

Na figura 39, para as sugestões da coluna **Não Aceitas**, há justificativas (sugestões 7, 12, 13 e 14). Assim, não foi implementada a sugestão 12 porque para mostrar a reação inversa basta rodar a simulação iniciando apenas com produtos. A sugestão 7 não foi implementada porque não foi encontrada uma reação simples, computacionalmente, que tivesse os três níveis de representação definidos e que fosse utilizada industrialmente. Um dos professores sugeriu a reação de síntese da amônia, que tem inclusive importância histórica, devido ao seu uso durante a 1^o guerra; porém, ela não é facilmente representável macroscopicamente. A representação em balança (sugestão 13) não foi acrescentada, porque poderia confundir os estudantes que estejam estudando equilíbrio reversível. A sugestão 14 vai contra a representação microscópica utilizada no *software*.

Logo, após responder à pergunta da entrevista, o professor era convidado a conhecer o *software* e fazer sugestões. A Figura 41 mostra os resultados. Apenas um professor já conhecia o programa, tendo inclusive acompanhado sua evolução.

Sugestão	Implementada	Implementada após TICESE	Não aceita	Não resolvida
1. Fontes do módulo numérico devem ser mais claras.		X		X
2. Está muito carregado.		X		X
3. Colocar nomes nos gráficos de concentração e velocidade.		X		
4. Fazer módulo numérico à parte.			X	
5. Chamar mais atenção para o frasco.	X			
6. Aumentar um pouco a velocidade das moléculas.	X			
7. Colocar as unidades na tela de configuração.	X			

Figura 41: Sugestões dadas pelos professores ao observar o *software* pela primeira vez.

Observa-se, nesta figura, que a primeira sugestão, referente à legibilidade das informações do módulo numérico, não foi considerada satisfatória nem mesmo após a segunda modificação. Ou seja, este problema (que já havia sido acusado pela TICESE), foi tratado e apontado novamente pelos professores. A questão refere-se ao fato de os valores das concentrações finais de reagentes e produtos estarem expressas em números fracionários em fonte pequena, o que dificultaria a leitura. Como não é possível, devido ao *layout* da tela, aumentar a fonte, o problema se manterá. Arredondar os valores não é uma solução desejada, pois, para valores pequenos como os da figura a seguir, valores decimais são importantes. A Figura 42 mostra o módulo numérico da versão 2 do *software*.

Gráficos simulação						
t_0			$t_1 \rightarrow t_2$			T(K): 299
C. iniciais []mol.L ⁻¹			C. finais []mol.L ⁻¹			
[H ₂]	[I ₂]	[HI]	[H ₂]	[I ₂]	[HI]	Kc: 16
0	0	6	0.477	0.477	5.045	P _{atm} : 1

Figura 42: Módulo numérico da versão 2, representado com tamanho real, em uma tela de 17" e resolução 1152 x 864 *pixels*.

A sugestão 2 (Figura 41), sobre a quantidade de elementos na tela, foi detectada pela TICESE e pelos professores. É verdade que, tanto na versão 2 como na 3, ela é realmente muito carregada. Para minimizar os efeitos de uma possível sobrecarga, foi acrescentado um botão **Rever**, que permite que cada aspecto da simulação seja acompanhado com detalhe. A sugestão 4, **deixar o módulo numérico à parte** não foi aceita, porque ele é essencial à compreensão da formação dos gráficos. Isto foi argumentado com o professor que deu a sugestão, ao que ele respondeu que sabia disso, mas acreditava que os estudantes não iriam prestar atenção nele, nem utilizá-lo para compor gráficos. Infelizmente, a previsão deste professor se confirmou, pois nenhum estudante fez referências à esta ferramenta, nem no teste piloto, nem no experimento. Contudo, decidiu-se manter o destaque a esta ferramenta, pois pode ser de grande auxílio ao professor que desejar dar destaque ao desenho dos gráficos ou ao cálculo de concentrações no equilíbrio.

Três sugestões dadas pelos professores após observar o *software* (sugestões 5, 6 e 7 da Figura 41) não haviam sido detectadas pela TICESE. Estas são questões específicas do assunto que o programa aborda. Finalizando esta etapa, apresenta-se, na Tabela 1, o resultado

do questionário aplicado a estes professores, com a média da atribuição de importância pelos professores. O assunto era a importância do domínio de cada um dos conceitos apresentados para a compreensão de Equilíbrio químico. A intenção era avaliar se os professores estavam de acordo com o conjunto de conceitos abordados pelo programa, listados a seguir. Depois de implementadas as sugestões dos professores, passou-se à realização do teste piloto.

4.3.2 O teste piloto

Após a entrevista com os professores foi realizado um teste piloto com uma turma de formandos do curso de Técnico em Química do Colégio Cristo Redentor ($n=12$, $X_{\text{BAR}}=21,3$, $s=4,47$). Esta foi a última etapa antes da realização do experimento relatado. O teste piloto apontou melhorias para os materiais de coleta de dados. Após este teste, devido aos comentários dos estudantes, decidiu-se adotar uma versão mais prática e mais rápida de ser preenchida, com questões de múltipla escolha. Esta pode parecer uma contribuição pequena, porém o tempo necessário para completar o pós-teste diminuiu aproximadamente 66% (de uma hora para vinte minutos).

Tabela 1

Atribuição de importância para cada conceito para a compreensão de equilíbrio químico no ensino médio. Notas de 0 a 10.

Conceitos apresentados	Média	Desvio padrão
C1→ Representação macroscópica de um sistema em equilíbrio desde o instante inicial.	10	0,0
C2→ Representação atomística clara.	10	0,0
C3→ Aproximação do equilíbrio no nível representacional microscópico.	9,2	1,10
C4→ Representação microscópica do Equilíbrio químico.	9,6	0,89
C5→ Aproximação do equilíbrio para concentração de reagentes e produtos.	9,2	1,10
C6→ Concentração de reagentes e produtos no Equilíbrio químico.	9,2	1,10
C7→ Aproximação do equilíbrio para as velocidades da reação.	8,2	2,05
C8→ Velocidade da reação no Equilíbrio químico.	8,8	1,30
C9→ Representação da constante de Equilíbrio químico.	8,6	2,19
C10→ Alterações das condições de equilíbrio.	9,4	1,34

4.3.3 Avaliação do *software* pelos usuários - o experimento em sala de aula

Nesta seção são descritas as condições em que foi conduzido o experimento. Este foi realizado no dia 31 de Março de 2005, durante os 3º, 4º e 5º períodos da manhã (cada período tem duração de 50 minutos), no laboratório de informática da ETCOM – UFRGS (Escola Técnica e de Comércio), uma escola federal do município de Porto Alegre. A duração total foi de 120 minutos. Não foi necessário utilizar todo o último período, de modo que os estudantes combinaram com a professora para sair mais cedo, depois de encerradas as atividades. Os estudantes não tinham estudado o assunto no curso que estavam fazendo, porém alguns já tinham estudado no colégio (ensino médio), outros no cursinho, outros em ambos.

Ao ler estes resultados, deve-se ter em mente que as condições em que este experimento foi realizado foram bastante favoráveis. Os estudantes estavam bastante motivados, e demonstraram muita maturidade, respondendo prontamente a todas as solicitações, e mantendo uma postura de ordem e respeito ao longo do processo. A direção da escola estava ciente e apoiou a realização do teste, mostrando interesse em saber os resultados. O corpo técnico do laboratório de informática se encarregou de entregar os equipamentos da sala prontos para o uso. Os materiais utilizados pelos estudantes (guia de simulação e pós-teste) tinham anexado um Termo de Consentimento Livre Informado, que informava do caráter absolutamente sigiloso da pesquisa. Todos eles estão assinados pela pesquisadora, pela professora e por cada um dos estudantes. O modelo utilizado está no Apêndice A deste trabalho. A Figura 43 mostra a configuração da sala.

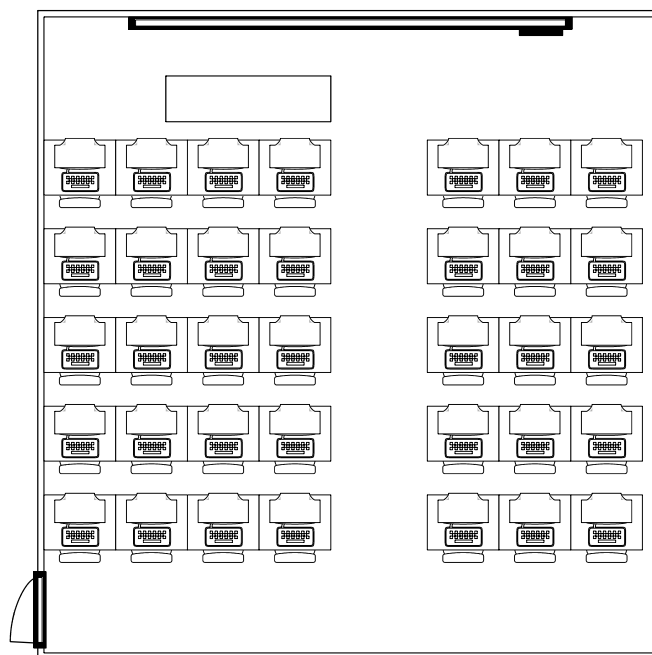


Figura 43: Configuração da sala onde foi conduzido o teste.

4.3.3.1 Suporte metodológico para o experimento

Antes de iniciar esta seção deseja-se frisar que a análise quantitativa de dados não é a única possível em pesquisa educacional. Muito antes, as técnicas qualitativas, de eficiência reconhecida, colocam-se como alternativa valorizada. No caso desta pesquisa não seria diferente, porém a análise qualitativa impõe um grande obstáculo: é imperativo que o profissional que conduz a avaliação tenha domínio de técnicas inquisitivas e um entendimento sólido e global do tema pesquisado. Se este profissional não faz parte da equipe, será necessário conjugar sua disponibilidade para intervenção com a participação dos estudantes. Algumas das vantagens deste tipo de técnica são: sua alta aceitação por parte da comunidade de pesquisa educacional; a necessidade da participação de poucos estudantes; a possibilidade de agendar as entrevistas em horários diferentes, o que facilita e encoraja a participação voluntária e a capacidade de responder com grande detalhe as questões específicas fechadas ao escrutínio quantitativo.

No entanto, acredita-se que uma abordagem quantitativa ofereça duas vantagens que a tornam muito adequada. A primeira relaciona-se ao fato de que é mais fácil conduzir uma avaliação quantitativa. O fator experiência, ainda que certamente melhore a qualidade do experimento, já não influencia tanto. Por isso, é possível que ele seja planejado e revisado por mais de uma pessoa e aplicado com relativa facilidade. A outra vantagem diz respeito à sua natureza objetiva. Ao final da análise, tem-se uma resposta numérica à pergunta que se fez, e

esta pode ser interpretada com bastante segurança. Certamente existem algumas limitações, sendo o tamanho das amostras necessárias a primeira delas. Ainda que seja possível realizar experimentos com poucos sujeitos, uma amostra pequena não confere legitimidade científica à pesquisa (a menos que amostras maiores sejam difíceis de conseguir). Mobilizar classes inteiras de estudantes é tarefa de uma dificuldade notável. A outra limitação, por paradoxal que pareça, está relacionada ao seu caráter objetivo. Quando se planeja um experimento, se deseja responder a uma pergunta, fazer um teste de hipóteses. Por isso não é possível uma investigação ampla, envolvendo a investigação de processos de gênese e transformação de conceitos, por exemplo. São duas abordagens distintas, que respondem a perguntas diferentes. Assim, com o intuito de colaborar para a sistematização da etapa de avaliação da aprendizagem, serão feitas considerações sobre a condução de experimentos.

Há necessidade de minimizar a influência de fatores que comprometam sua validade interna e externa. Campbell e Stanley (1963), por exemplo, listaram fatores que, se não forem controlados, comprometem a validade interna do experimento.

1. Eventos ocorridos entre a primeira e a segunda medição.
2. Processos relativos à passagem do tempo: ficar mais velho, mais cansado, com fome etc.
3. Realizar um teste sobre a pontuação de um segundo teste.
4. Mudanças no instrumento de medida, nos observadores ou nos avaliadores.
5. Regressão: operante quando os grupos foram selecionados com base em pontuações extremas.
6. Tendenciosidade, parcialidade, inclinação (*bias*): operante durante a seleção dos componentes de cada grupo.
7. Mortalidade, desistência: não comparecimento de um dos sujeitos em uma das observações.

Campbell e Stanley (1963) enumeram os fatores que comprometem a validade externa do experimento, ou seja, sua capacidade de estender suas conclusões a outras populações, configurações etc.

1. Reatividade ou efeito de interação do teste: influência do pré-teste na sensibilidade do respondente às variáveis experimentais.

2. Efeitos da interação entre a tendenciosidade da seleção e as variáveis experimentais.
3. Efeitos do ambiente experimental, por si só poderiam impedir a generalização dos resultados para ambientes não-experimentais.
4. Inferências baseadas em múltiplos tratamentos. Pode ocorrer quando múltiplos tratamentos são aplicados aos mesmos respondentes, pois os efeitos de tratamentos anteriores não podem ser excluídos nem negligenciados.

Para minimizar os efeitos destes fatores, os autores propõem o uso preferencial de dois arranjos experimentais. A Figura 44 mostra o primeiro deles. Ele oferece um controle robusto da validade interna do experimento. O segundo *design*, chamado *Design Solomon*, com quatro grupos (Figura 45) possui mais prestígio, devido à sua capacidade de generalizar conclusões.

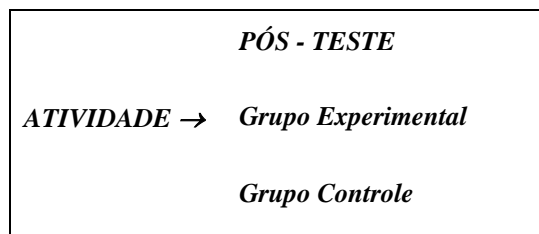


Figura 44: Configuração dois grupos aleatoriamente definidos, uma observação.

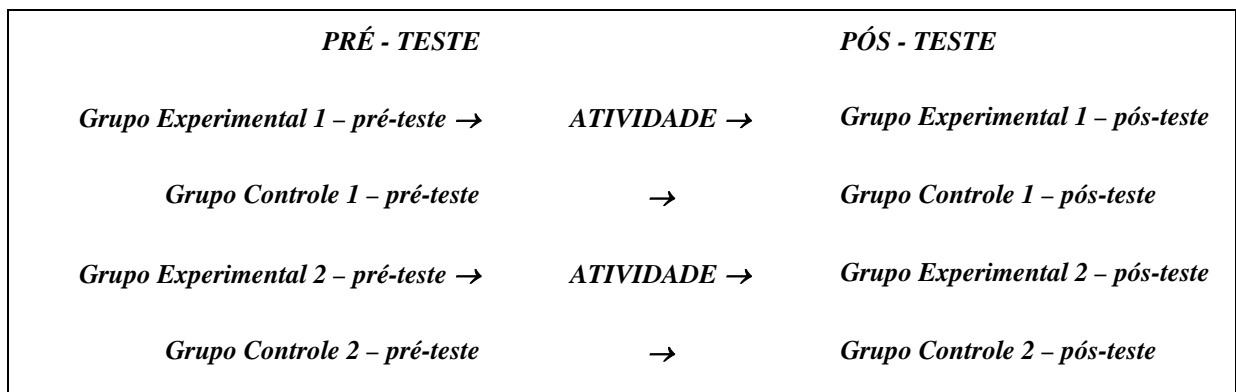


Figura 45: Configuração Solomon com quatro grupos aleatoriamente definidos

O arranjo escolhido para o experimento será o primeiro (Figura 44), pois pode ser feito de uma vez só, e precisa de menor número de participantes.

4.3.3.2 Descrição do ambiente e da amostra do experimento

Participaram 40 estudantes, sendo 19 do curso de Técnico em Química e 21 da Biotecnologia. A média da idade dos alunos é 19,72 anos ($s=1,20$). Todos afirmaram ter acesso a computadores, sendo que 80% (32 estudantes) usam o computador para acessar a internet e realizar trabalhos escolares. 85% (34 estudantes) têm acesso em casa, os demais têm acesso pelo menos na escola. 53% (18 de 34) dos que têm computador em casa também o utilizam na escola. A composição da turma em relação aos cursos e ao gênero está na Tabela 2.

Para determinar quais estudantes ficariam em cada grupo (Controle e Experimental, doravante C e E), foi solicitado que eles retirassem de uma lata um papel que podia conter a letra **E** ou a letra **C**. Havia 20 de cada, de forma que cada estudante tinha a mesma chance de estar em cada um dos grupos. Foi informado que não seria possível trocar ou escolher o grupo. Os estudantes que tiraram a letra E foram para o lado esquerdo da sala e realizaram a atividade com o guia de simulação, em duplas. A divisão da turma nos grupos C e E é mostrada na Tabela 3.

Tabela 2
Composição da amostra

	Mulheres	Homens	Total
Biotecnologia	18	3	21
Técnico em Química	7	12	19

Tabela 3
Composição dos grupos C e E.

	C	E
Biotecnologia	11	10
Técnico em Química	9	10
Homens	9	6
Mulheres	11	14

A primeira atividade para o grupo E foi preencher o guia de simulação (dividido em três partes: previsão, observação guiada e com auxílio do *software* e explicação), seguido de um questionário sobre a atividade com o Equil e sobre a percepção da dificuldade em equilíbrio químico. Não houve referências a problemas em entender o funcionamento do

software, tampouco dúvidas sobre o guia ou sobre o questionário. Alguns estudantes fizeram comentários sobre o *software*, dando sugestões de melhoria. Enquanto isso, os estudantes do grupo C realizaram uma pesquisa orientada pela professora (sem relação com equilíbrio químico). Esta atividades duraram cerca de 80 minutos.

A etapa seguinte, para ambos grupos, foi o preenchimento do pós-teste, realizado por toda a turma ao mesmo tempo, com duração total de 40 minutos (passados 20 minutos, mais da metade da turma já havia terminado). Além disso, os estudantes do grupo E também preencheram um questionário. O teste parece ter sido de fácil compreensão e muito ágil, pois não foram referidas questões de interpretação. Depois que o último teste foi entregue a turma foi liberada pela professora para ir embora. Um estudante do grupo C pediu explicações sobre os objetivos do *software* ao final do teste. O teste não foi filmado ou fotografado.

4.3.3.3 Materiais e métodos

Nesta seção são descritos os materiais utilizados para a coleta de dados, bem como os métodos empregados para interpretação dos dados. O objetivo geral é verificar se o *software*, em conjunto com uma atividade dada, favorece a aprendizagem. É importante ressaltar que não é possível separar os efeitos da atividade dos efeitos do *software*, sendo que ambos devem ser considerados como uma unidade. Além disso, será possível responder as questões abaixo.

1. Os estudantes que utilizaram o *software* e realizaram a atividade educacional apresentam notas melhores que os que não realizaram?
2. Algum grupo de conceitos não foi afetado pelo uso do *software*?
3. A compreensão de algum dos três tipos de representação (microscópico, macroscópico e simbólico) é favorecida pelo uso do *software*?

As respostas de cada estudante foram classificadas em uma escala ordinal de 0 até 4, com cada posição representando um nível de compreensão do estudante em relação ao assunto abordado, de acordo com lista a seguir.

- No nível quatro sejam enquadrados estudantes que não apresentem concepções alternativas para aquele nível.
- No nível três se constatar que a presença de concepções alternativas não impede a chegada do estudante ao nível de classificação mais elevado, por provocar lacunas na articulação do raciocínio;

- No nível dois se constatar a presença de concepções alternativas que provoquem dificuldades de compreensão ou uso de conceitos científicos de forma confusa;
- No nível um se constatar a total falta de articulação de conceitos científicos e de raciocínio lógico ao responder o questionário;
- No nível zero se constatar o não preenchimento da resposta.

O conjunto de variáveis avaliado neste experimento é apresentado na Figura 46.

A próxima questão a ser levantada é sobre os materiais utilizados pelos estudantes. Em primeiro lugar se tem o guia para simulação (disponível no Apêndice B), que deve ser distribuído para a turma E, e será a atividade educacional realizada em conjunto com o uso do *software*. Este material está estruturado em três momentos: previsão, observação e explicação, de maneira a incentivar que o estudante compare suas idéias anteriores com as (possíveis) novas conclusões derivadas do uso do *software* e do questionamento proposto na atividade. O outro material de coleta de dados empregado nesta pesquisa é um pós-teste (disponível no Apêndice C), com questões de múltipla escolha.

Estes materiais são similares aos utilizados por Orlandi (2004). As diferenças entre eles são: a distribuição de um guia de cores; a solicitação para os estudantes identificarem os gráficos de concentração e de velocidade e o *layout* do teste. Estas mudanças foram introduzidas por causa da recepção dos estudantes do teste piloto. A opção por um código de cores se justifica na medida em que os estudantes do teste piloto não demonstraram possuir um vocabulário comum e conciso para responder às questões. A solicitação para identificação dos gráficos foi feita porque todos os estudantes do teste piloto desenharam gráficos sem rótulos. Os estudantes também afirmaram que era cansativo preencher os testes, e que as questões eram confusas, pois pareciam ser sempre as mesmas.

<p>1. RepMac: representação macroscópica de um sistema em equilíbrio desde o instante inicial da reação Exploração das características macroscópicas das reações químicas trabalhadas, mais especificamente a cor da reação.</p>
<p>2. ReAtom: representação atomística Análise de quais concepções e representações utilizadas pelos estudantes no nível representacional microscópico para reações químicas em fase gasosa.</p>
<p>3. AproEqMi: aproximação do equilíbrio no nível representacional microscópico Exploração das concepções dos estudantes através de ilustrações no nível microscópico para a aproximação do equilíbrio.</p>
<p>4. RepEqMI: representação microscópica para o equilíbrio químico Complementar ao item 3. Verificação de quais as concepções no Equilíbrio químico Dinâmico de uma reação gasosa ilustrada no nível microscópico.</p>
<p>5. AproEqCo: aproximação do equilíbrio para as concentrações de reagentes e produtos Exploração das concepções sobre a reação química, nos estágios pré-equilíbrio dentro do nível de representação simbólico, tanto a partir de respostas referentes ao estado atual e a tendência de uma reação nos instantes iniciais, quanto na representação gráfica das concentrações.</p>
<p>6. RepEqCo: concentração dos reagentes e dos produtos no equilíbrio químico Complementar ao item 5. Análise das idéias dos estudantes sobre o que acontece com as concentrações dos reagentes e dos produtos apenas no Equilíbrio químico de uma reação em fase gasosa.</p>
<p>7. AproEqVel: aproximação do equilíbrio para as velocidades da reação Exploração das idéias dos estudantes sobre o que acontece com as velocidades direta e inversa de uma reação nos instantes iniciais. Para isso, baseia-se em respostas sobre o estado atual e a tendência de uma reação, bem como os gráficos de velocidades.</p>
<p>8. RepEqVel: velocidade da reação no equilíbrio químico Complementar ao item 7. Análise através de gráficos e respostas das concepções dos estudantes sobre o que acontece com as velocidades das reações direta e inversa no Equilíbrio químico.</p>

Figura 46: Grupos de variáveis estabelecidas.

A última questão que se coloca é a da classificação das respostas dos estudantes. A intenção era utilizar uma lista elaborada por Orlandi (2004), que contém as prováveis respostas para cada variável, em cada um dos níveis de compreensão, num total de 32 descrições (8 variáveis * 4 níveis), como mostra a Figura 47.

Todavia, já tendo posse dos materiais preenchidos pelos estudantes, conforme se procedia a avaliação das respostas, surgiu a necessidade de criar um instrumento que classificasse de forma mais objetiva as respostas. A lista montada por Orlandi (2004) se revelou de difícil uso e subjetiva, pois não tratava de todas as possíveis combinações de respostas. Assim, optou-se por criar um manual para interpretação dos resultados, disponível no Apêndice D.

Representação Macroscópica de um sistema em equilíbrio desde o instante inicial da reação

As questões analisadas nesta categoria são:

Pós-Teste: 1.5, 2.0 e 2.5.

Compreensão Total (4)

O estudante reconhece que uma reação química em equilíbrio, apesar de parecer estática macroscopicamente, é dinâmica microscopicamente. Assim, ele identifica que a cor de um sistema desde o instante inicial até atingir o equilíbrio vai alterando, com tendência a diminuir, sem ficar incolor. Nesse nível de compreensão, o estudante reconhece que no início da reação a cor é mais intensa, e essa intensidade diminui até atingir o equilíbrio, permanecendo inalterada. Inclusive, alguns estudantes utilizam, para definir as cores, termos como **mais claro** ou **mais escuro**, justificando a resposta.

Compreensão com elementos errôneos (3)

O estudante não mostra uma representação correta da cor do sistema na aproximação do equilíbrio ou no estado de Equilíbrio químico. Alguns estudantes acertam a cor para o instante t_0 e t_1 , mas no instante t_2 deduzem que fica incolor.

Figura 47: Excerto de guia para interpretação das respostas. Extraída da dissertação eletrônica de Orlandi (2004).

4.3.3.4 Aspectos conceituais não avaliados

Alguns aspectos do conteúdo, presentes em todas as oito variáveis não foram considerados durante a avaliação. Assim, para a variável 1 (Representação Macroscópica) não foram penalizados, da mesma forma, estudantes que responderam haver 100% ou 0% de I_2 no frasco. Neste caso, admitiu-se que alguns estudantes tenham se confundido com o fato de haver, em dado momento, 100% de reagentes. A variável 4 (Representação do Equilíbrio em Nível Microscópico), por sua vez, não penalizou estudantes que desenharam a concentração de reagentes maior que a de produtos no equilíbrio. Nenhuma das variáveis que analisa os desenhos (2, 3 e 4, respectivamente Representação Atomística, Aproximação do Equilíbrio para o Nível Microscópico e Representação do Equilíbrio no Nível Microscópico) penalizou estudantes que não mantiveram o mesmo número de moléculas ao longo das questões. Acredita-se que isto não caracterize uma violação do princípio de conservação de massa. Foi assumida a hipótese que os estudantes não se deram conta disto ao desenhar, pois não é uma tarefa a qual eles estejam acostumados. Em relação às variáveis do nível simbólico: concentração e velocidade, não foram penalizados estudantes que não cruzaram os gráficos de concentração de reagentes e produtos nem os que desenharam os gráficos da velocidade inversa e direta com a mesma altura.

4.3.3.5 Resultados do experimento em sala de aula

Antes de iniciar a exposição dos resultados, será feita uma breve discussão sobre as modificações introduzidas em relação ao experimento conduzido por Orlandi (2004), e

referidas anteriormente na descrição do experimento. Alguns estudantes tiveram dúvidas sobre o uso do guia de cores, pois afirmavam que o valor da concentração de Iodo esperado por eles não estava expresso, por exemplo:

- Em t_0 deveria ter 50% de I_2 , mas não há esta opção. O que respondo?

Nestes casos os estudantes foram orientados sobre a função do guia, que era permitir avaliar se os alunos sabem se a tonalidade fica mais forte, mais fraca ou se ela não se altera. Em relação aos gráficos, apesar de ter sido solicitado diversas vezes que ambos fossem desenhados e identificados, três estudantes não o fizeram. Em relação aos desenhos, alguns estudantes afirmaram, durante o teste, que não sabiam como responder a questão. Nestes casos, foi sugerido que eles imaginassem o que veriam se pudessem enxergar os átomos e moléculas no frasco. Os estudantes ficaram livres para responder “não sei” a qualquer das questões. Não foi pedido que deixassem em branco aquelas questões que não soubessem resolver. Os problemas relatados acima foram mais freqüentes no grupo C.

Outra questão que merece atenção é o cuidado em relação ao plágio. Há possibilidade que isto tenha acontecido nas questões sobre as variáveis 1 a 4 (respectivamente: Representação Macroscópica, Representação Atomística, Aproximação do Equilíbrio para o Nível Microscópico e Representação do Equilíbrio no Nível Microscópico). Porém, acredita-se que a hipótese mais provável é que os estudantes da mesma dupla tenham adotado o mesmo estilo de desenhar e as mesmas conclusões sobre a cor do frasco compartilhadas durante o uso do guia. Isto pode dar suporte à hipótese de que o trabalho em dupla favorece a evolução de estudantes com um nível de compreensão menor. No entanto o material utilizado não é adequado para este tipo de conclusão.

Uma vez atribuídas notas para todos os estudantes em cada uma das variáveis em estudo, agrupou-se, para cada variável, as notas iguais a 0, 1 e 2 em um sub-conjunto e as notas 3 e 4 em outro. O motivo desta escolha é que as notas 3 e 4, na escala utilizada, representam os melhores desempenhos, ou seja, os estudantes com esta pontuação têm uma compreensão maior do tema em estudo. Esse artifício foi utilizado também para satisfazer a condição de uso do teste qui-quadrado, que exige no máximo 20% das caselas com freqüência menor do que 5. Como haviam 40 estudantes divididos em dois grupos de 20, classificados em 5 postos, haveria 2x5 caselas na tabela, o que tornaria impossível satisfazer a condição acima referida. O nível de significância utilizado para aceitar a hipótese alternativa da diferença entre as pontuações dos grupos E e C é 95%. A seguir, é mostrada a Tabela 4, com as variáveis que apresentaram diferença significativa entre o desempenho dos grupos E e C

(as tabelas completas estão no Apêndice E). Também foi realizada uma análise de confiabilidade para as respostas dadas no questionário de avaliação da atividade educacional (apenas estudantes do grupo E realizaram, após a atividade com o guia de simulação). O teste de Alfa de Cronbach ($\alpha = 0,1113$) para este questionário mostrou que as respostas não foram consistentes. Por este motivo ele não será discutido.

Tabela 4

Variáveis onde foi detectada diferença significativa entre os desempenhos dos grupos E e C (E >C).

RepMac – Representação Macroscópica	RepEqMi – Representação do Equilíbrio ao Nível Microscópico
ReAtom – Representação Atomística	AproEqCo – Aproximação do Equilíbrio para a Concentração (nível simbólico)
AproEqMi – Aproximação do Equilíbrio ao Nível Microscópico	RepEqCo – Representação do Equilíbrio para a Concentração (nível simbólico)

Estes resultados indicam uma clara superioridade do grupo E sobre o grupo C, pois apenas nas variáveis relacionadas ao conceito velocidade, AproEqVel e RepEqVel não houve diferenças entre os grupos E e C. Em relação à AproEqVel e RepEqVel, não foi detectada diferença entre a *performance* dos grupos. Desta forma, conclui-se que a atividade com o *software* foi bastante efetiva, surtindo efeitos no aumento da compreensão dos estudantes, para as variáveis analisadas, à exceção daquelas que se referem ao conceito de velocidade da reação.

4.3.3.6. Concepções alternativas detectadas no guia de simulação

A análise das respostas do guia permite fazer um mapeamento, ainda que com pouca precisão, das concepções alternativas apresentadas pelos estudantes. É importante lembrar que as perguntas do guia não focavam nenhuma concepção em particular, por isso os resultados não podem ser estendidos a toda a população de estudantes. Afinal, é possível que um estudante que escreveu pouco não tenha deixado transparecer uma determinada

incompreensão. Inicia-se, então a exposição abordando as concepções alternativas, detectadas diversas vezes. A Tabela 5 mostra algumas delas. Deve-se lembrar que havia 20 estudantes no grupo E, portanto eles estão numerados de 21 a 40.

A observação destes resultados permite concluir que algumas duplas (estudantes com identificações consecutivas tipo ímpar e par) compartilham a mesma concepção alternativa, e provavelmente a evolução conceitual, quando houver. Para ilustrar algumas destas concepções, transcreveram-se algumas das respostas que as identificam. O estudante 29, por exemplo, ao ser perguntado sobre o estado do sistema após o equilíbrio ser atingido, respondeu da seguinte forma:

- Agora as concentrações de produtos e reagentes se tornam iguais, havendo máxima produção de produtos.

Essa resposta denota presença de duas concepções alternativas a respeito do estado do sistema em equilíbrio. O estudante 26 mostra possuir outras duas concepções alternativas sobre o estado de equilíbrio ao afirmar:

- A reação ocorreu totalmente, e agora possuímos apenas moléculas de HI, ou seja, atingimos o equilíbrio químico da reação.

O estudante 34, por sua vez, demonstra não compreender de forma ampla o conceito de equilíbrio dinâmico ao afirmar, sobre t_1 :

- Parte do I_2 já reagiu com H_2 , mas ainda não o suficiente para alcançar o equilíbrio.

A afirmação do estudante 31, a respeito do caráter reversível do equilíbrio químico, deixa clara sua tentativa de conciliar suas concepções alternativas com conceitos científicos:

- A solução ficará incolor devido à cor do produto ser incolor. As moléculas diminuirão sua velocidade depois do equilíbrio, para voltarem a se agitar para mais tarde reverter o equilíbrio.

A Figura 48, com uma das respostas do estudante 30, dá um exemplo da concepção **reagentes se apresentam como íons**.

Tabela 5

Concepções apresentadas pelos estudantes na análise do guia de simulação (grupo E).

Concepção alternativa	Estudantes que apresentaram	Porcentagem da amostra
1- As concentrações de produtos e reagentes são iguais no equilíbrio.	21, 25, 29(1), 39, 40	25%
2- Há apenas produtos no equilíbrio.	23(1), 24(1), 26(1), 31	20%
3- Negação do equilíbrio dinâmico: “algumas moléculas não reagiram”.	22, 23, 34, 39	20%
4- Reagentes se apresentam como íons.	29(1), 30(1), 33, 34	20%
5- O Iodo é mais pesado, por isso é mais visível.	21, 23, 24	15%
6- Negação do equilíbrio reversível.	22(3), 31	10%
7- No equilíbrio ocorre máxima formação de produtos(2).	29, 30	10%
8- As moléculas é que atingem o equilíbrio.	21	5%
9- Há mais formação de produtos no equilíbrio(2).	35	5%
10- T_0 marca o início do movimento das moléculas, com progressivo aumento da velocidade de cada molécula.	31	5%
11- No equilíbrio as moléculas se agitam menos.	31	5%
12- Uso da palavra mistura como sinônimo para reação.	26	5%

(1) Estudantes que afirmaram concluir estar enganados sobre o assunto após o uso do *software*.

(2) Pode ser indicativa de duplo erro, pois a formação de produtos é mínima, e o estudante pode pensar que as taxas de formação e consumo não são iguais.

(3) Reação reversível apenas quando as duas velocidades são iguais.

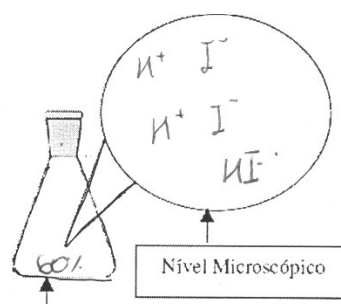


Figura 48: Representação dos reagentes como íons.

O próximo assunto abordado nesta análise de resultados refere-se a questões que podem ser levantadas em função dos dados encontrados. Ao iniciar as seções subseqüentes, o

objetivo da análise será explicitado, bem como o método utilizado para chegar aos resultados e à conclusão. A última seção apresenta uma particularidade da análise dos resultados.

4.3.3.7 Análise dos gráficos de concentração e velocidade feitos no pós-teste

Nesta seção apresenta-se a incidência de erros no desenho dos gráficos de concentração e velocidade. O objetivo é verificar se os gráficos dos estudantes dos grupos C e E apresentam uma quantidade de incorreções coerentes com o resultado do teste estatístico. Para isso, foram construídas as Tabela 6 e 7, que mostram a frequência das incorreções nos gráficos de concentração e velocidade, detectadas no guia de simulação.

Tabela 6

Incorreções no gráfico das concentrações de produtos e reagentes no equilíbrio.

	Grupo C	Grupo E	Soma
1- Gráficos com concentração de produtos e reagentes iguais no equilíbrio.	8	1	9
2- Gráficos de concentração de produtos e reagentes não se cruzam.	5	2	7
3- Gráficos sem sentido no contexto.	3	1	4
4- Gráficos invertidos.	0	0	0
5- Gráficos em branco	0	0	0
Soma	16	4	19

Tabela 7

Incorreções no gráfico para velocidades direta e inversa no equilíbrio.

	Grupo C	Grupo E	Soma
1- Gráfico invertido.	7	7	14
2- Gráfico com velocidade de produtos diferente de reagentes no equilíbrio.	1	1	2
3- Gráficos de concentração.	1	1	2
4- Gráficos sem sentido.	4	4	8
5- Gráficos de velocidade de produtos e reagentes com a mesma altura (1)	13	10	23
6- Nenhum gráfico.	1	0	1
Soma	27	23	50

(1) 5 estudantes do grupo E desenharam gráficos de velocidade com alturas marcadamente diferentes para as velocidades de produtos e reagentes. Isto, contudo, não significa que eles saibam o motivo de ser este o formato das curvas. Isto não foi premiado. Todas as demais ocorrências apresentadas foram penalizadas.

Em relação às concentrações (Tabela 6), observa-se que 80% dos estudantes do grupo C têm algum problema com no desenho do gráfico. Já no grupo E, 20% apresentam incorreções. Estes dados corroboram a análise estatística, que não acusou diferença entre grupos para as variáveis relativas à concentração. Merece destaque o fato de 8 estudantes do grupo C (40%) desenharem gráficos com concentração de produtos igual à dos reagentes, ao passo que apenas 1 estudante do grupo E (0,05%) o fez. Isto não chegou a favorecer os estudantes do grupo E por causa do pouco valor atribuído ao desenho do gráfico na análise das variáveis. A incorreção 2, **gráficos de concentração de produtos e reagentes não se cruzam**, muito freqüente entre os estudantes do grupo C, não foi penalizada. Isto poderia ter afetado o resultado estatístico a favor do grupo E, porém seria um exagero avaliar o formato do gráfico neste sentido. Para as velocidades (Tabela 7), a freqüência de incorreções observada é maior.

4.3.3.8 Sobre a análise dos desenhos no pós-teste (variáveis 2 a 4)¹⁸

Alguns estudantes desenharam apenas duas moléculas nas questões 1.5; 2.5 e 3.5. Isto pode ter prejudicado aqueles que tinham uma boa representação microscópica da reação, pois foram penalizados. O motivo para isto é que com duas representações não é possível avaliar a distribuição uniforme das moléculas, dado importante sobre a qualidade da representação microscópica (a reação em estudo é gasosa). Da mesma forma, isto dificulta a avaliação da compreensão da aproximação e da representação do equilíbrio químico. Caso fosse feita uma análise rigorosa, por exemplo, um estudante apenas teria chances de ser classificado como nível 4 para a representação microscópica de um sistema em equilíbrio (RepEqMi), caso desenhasse pelo menos 6 moléculas (4 de produto e uma de cada reagente). Donde se conclui que o enunciado da questão deveria ter incentivado no mínimo seis desenhos.

4.4 CONSIDERAÇÕES SOBRE O PROCESSO

Este relato das ações tomadas para produzir o *software* de acordo com a metodologia proposta deixa evidentes as diversas dificuldades que se colocaram frente à equipe. Ficaram

¹⁸ 2: Representação Atomística (RepAtom). 3: Representação Microscópica para a Aproximação do Equilíbrio (AproEqMi). 4: Representação Microscópica de um Sistema em Equilíbrio (RepEqMi).

expostas também as diversas decisões que tiveram que ser tomadas, em todas as três etapas. Espera-se que a descrição desta experiência aproxime as dimensões da teoria e da prática, servindo de fonte de referência para aqueles que estiverem desenvolvendo projetos similares.

Com esta intenção, são apresentados no próximo capítulo questionamentos que se colocaram frente à equipe, e que não puderam ser respondidos de forma conclusiva. Tais questões concernem à configuração da metodologia e a priorização de recursos.

5. DISCUSSÃO

Embora a metodologia aplicada tenha se mostrado adequada para responder às necessidades de integração de profissionais com os usuários e da estruturação das atividades de elaboração do projeto educacional, desenvolvimento e avaliação, algumas questões e pareceres devem ser colocados.

O primeiro deles é em relação à elaboração do primeiro conjunto de requisitos, que compõem o projeto educacional. Conforme item 3.1 (página 61), foi uma opção da equipe utilizar um modelo pouco formalizado para a elaboração do projeto. Todavia, ferramentas que permitam sistematizar a construção da estratégia educacional podem ser especialmente úteis, principalmente, para times que não disponham de um especialista na área educacional, ou que não estejam certos sobre quais as dificuldades enfrentadas pelos estudantes na compreensão domínio estudado.

Dentre as ferramentas que se julga adequada para esse fim, cita-se:

- a. a lista de questionamentos para orientar a elaboração do projeto educacional, descrita por Van der Mast (1995);
- b. JAD (*Joint Application Development*), utilizada principalmente por equipes ágeis. Neste processo de grupo, os participantes interagem livremente, o que substitui a técnica de empregar entrevistas com usuários para determinar requisitos de um sistema;
- c. *Design Macroergonômico – DM* – proposto por Fogliatto e Guimarães (1999). Esta ferramenta, segundo os autores, foi projetada para incorporar as necessidades dos usuários em produtos e postos de trabalho. Porém, acredita-se que pode ser útil para a tarefa de coleta de requisitos em projetos de *software*.

Não foram encontrados trabalhos relacionando os custos e benefícios do uso destas três ferramentas. Todavia, pela natureza de cada uma delas, pode-se sugerir que a escolha seja feita considerando a experiência no ensino do domínio e o conhecimento das dificuldades dos estudantes. Quanto mais segura a equipe estiver, menor a necessidade de formalização. Por exemplo, no caso do *software* desenvolvido, como a equipe estava segura de ter feito um levantamento de requisitos satisfatório, não foram utilizadas nenhuma das ferramentas supra citadas.

Ainda sobre o levantamento do primeiro conjunto de requisitos (e conseqüente definição do planejamento educacional), o parecer é que tal perfil livre da condução mostrou-se adequado, pois se conseguiu extrair o conjunto de objetivos e estratégias educacionais utilizadas no programa, cobrindo os tópicos sugeridos dentro do currículo do ensino de nível médio brasileiro. Entretanto, alguns comentários devem ser feitos, sendo o primeiro a respeito da existência de informações confiáveis sobre as dificuldades dos estudantes com o tema equilíbrio químico. A quantidade de pesquisas científicas com foco neste assunto é vasta o suficiente, para permitir que se extraia delas o conjunto inicial de requisitos para o projeto. Estas respondem às importantes perguntas: o que os estudantes sabem a respeito do assunto e quais as dificuldades que eles têm. Nem todos os projetos poderão se beneficiar disto. O segundo comentário é sobre o acesso a esta informação. No caso presente, ele foi possível porque fez parte da equipe um professor com vasta formação e experiência em Ensino de Ciências, que sabia da existência da informação e onde ela poderia ser encontrada. Deve-se ainda considerar que o fato de este ser um projeto acadêmico, realizado dentro de uma universidade federal facilitou muito o acesso à estas informações. O terceiro comentário é que, o fato de a hipótese que sustentou as escolhas para a interface da primeira versão ter se mostrado ligeiramente equivocada, não desvaloriza a forma como foi feita o levantamento dos requisitos, tanto para a primeira quanto para a segunda versão. Acredita-se que esta seja uma dinâmica própria de projetos de pesquisa, que se propõem solucionar um problema através de investigação científica.

A segunda questão levantada pela aplicação da metodologia proposta remete à entrada no ciclo de iterações de *design*, e diz respeito à inclusão de uma etapa que não foi realizada, mas que talvez diminuísse a quantidade de ciclos e de protótipos construídos. A etapa que se sugere acrescentar é a modelagem, logo após a conclusão do projeto educacional. A escolha da linguagem utilizada para realizar esta modelagem cabe aos profissionais responsáveis pela programação, dependendo de fatores afetivos (gostar da linguagem, conhecimento, segurança etc.) e práticos (por exemplo, não adotar UML¹⁹ caso não seja utilizada uma linguagem orientada a objetos), e por esses motivos, não serão sugeridas opções. O parecer em relação ao modelo de desenvolvimento (prototipação), é que este mostrou-se adequado para o caso. Esse modelo foi adotado desde o início da construção, permitindo que sempre houvesse o debate acerca de ferramentas, funcionalidades e aparência do *software*. A possibilidade de realizar

¹⁹ *Unified Modelling Language*

iterações de *design* deu flexibilidade ao projeto do *software*, de modo que foi possível responder aos novos requisitos que se impunham. Deve ser ressaltado que todos os protótipos eram avaliados sempre pelos dois membros da equipe e, diversas vezes, por pelo menos um especialista no domínio. O motivo desta ressalva é que não se pode concluir sobre o fator que mais colaborou para o sucesso deste esquema de trabalho. Pode ter sido tanto o próprio modo de trabalhar, cuja eficiência é defendida, principalmente, pelos adeptos da adoção de *design* participativo (BODKER; IVERSEN, 2002; GRØNBÆK et al., 1997; MULLER, 2001; MULLER, 1991; MULLER et al., 1993; SPINUZZI, 2002), como pode ter sido a presença de especialistas.

A terceira questão que merece destaque é a sugestão de uso de técnicas de *design* participativo com regras explícitas, que podem ser bastante úteis em dois casos principais: quando não há *designers* profissionais na equipe e quando não é conhecido o domínio a ser explorado. Estas técnicas fornecem estratégias para criar e para decidir entre diversas idéias e propostas de interfaces, além de permitirem que todos expressem suas opiniões de forma democrática. É ressaltado que seu uso é potencializado se guiado por princípios de *design*, conforme foi feito neste trabalho. Para projetos em que a interação com o usuário representa um desafio, é oportuno lançar mão de meios estruturados para realizar a avaliação das interfaces criadas. Estes meios estão descritos na literatura de interação homem computador, e alguns dos métodos são: avaliações heurísticas; uso de *checklists* e modelagem do comportamento do usuário. Cada uma destas abordagens tem vantagens e desvantagens, e sua aplicação deve ser decidida caso a caso. Para orientar esta decisão é sugerido consultar a Revisão Bibliográfica desta pesquisa. O parecer desta dissertação em relação ao uso feito de tais técnicas é que elas foram úteis e adequadas. Retomando a ordem de aplicação durante a construção do *software*:

- a. Após a construção de cada protótipo, seguiram-se atividades de avaliação. No início dos trabalhos, estas avaliações estavam mais preocupadas em averiguar se o programa funcionava de acordo com as leis do Equilíbrio químico. Tais avaliações eram informais e não estruturadas, ou seja, não havia um protocolo a seguir.
- b. Depois que este estágio foi ultrapassado, foi feita uma inspeção com a TICESE (GAMEZ, 1998), que apontou problemas pontuais da interface. A utilização de um *checklist* desta natureza se revelou bastante apropriado, pois apontou erros causados pelo esquecimento. Por exemplo, ao projetar uma interface, sabe-se que algumas diretrizes devem ser seguidas, por exemplo: assegurar boa legibilidade dos desenhos e textos, distribuir os componentes de

forma harmônica, agrupar por similaridade, etc. Estas *guidelines* (ver página 32 para exemplos) são seguidas sem que se faça esforço, pois são aprendidas durante a formação acadêmica do *designer*. No entanto, isso não significa a certeza de uma interface clara e livre de erros de *design*. Por este motivo, o uso de *checklists* se revelou oportuno, pois é uma maneira rápida, fácil e barata de encontrar problemas. Porém, é importante compreender que existe uma classe de problemas que não pode ser detectada por *checklists*, que são aqueles relacionados ao domínio. Isso não significa que se deve ter uma ferramenta que procure por erros, e sim uma estratégia que sirva não apenas para detecção de erros, mas também para captação de novas propostas. Parte-se do princípio que pode haver formas mais criativas e eficazes de lidar com ou apresentar um determinado assunto, acreditando-se que a entrevista seja o método mais apropriado para este fim.

c. Por isso, conforme relatado no Capítulo 4, foram realizadas entrevistas com 5 professores, na qual se pediu por sugestões e se expôs o programa às críticas. Diversas sugestões foram apresentadas, e muitas delas foram incorporadas. Houve, inclusive, o caso de criação de um novo componente, exclusivamente a partir de sugestões extraídas das entrevistas. Devido à importância das sugestões e críticas feitas e ao fato de que a entrevista era o único meio de chegar a elas, confirma-se que este é um método importante de especificação de requisitos de um *software* educacional. O posicionamento das entrevistas no ciclo de atividades propostas também foi considerado adequado. Realizá-las ao final das iterações permitiu que os professores focassem apenas os aspectos educacionais do programa, sem se deter na qualidade da apresentação dos componentes da interface. Todavia, destaca-se que a qualidade das sugestões feitas pode estar relacionada com a experiência dos professores no ensino de química, o que traz novamente à tona a questão da importância da participação de especialistas durante as atividades da metodologia.

d. Após esse processo de submissão da interface à avaliação de professores, foi feito um experimento com uma turma do curso técnico de Química. As questões propostas foram:

1. Os estudantes que utilizaram o *software* e realizaram a atividade educacional apresentam notas melhores que os que não realizaram?
2. A compreensão de algum dos três tipos de representação (microscópico, macroscópico e simbólico) é favorecida pelo uso do *software*?
3. Algum grupo de conceitos não foi afetado pelo uso do *software*?

A resposta à primeira questão é afirmativa, afinal, a atividade educacional com o uso do *software* parece ter ajudado os estudantes a melhorar sua compreensão sobre equilíbrio químico em nível simbólico. Os estudantes do grupo E apresentaram, em relação aos do grupo C, melhor compreensão da evolução da reação e do estado de equilíbrio.

A análise das variáveis em estudo permite concluir sobre a segunda questão. Observa-se que os estudantes do grupo E apresentam melhor compreensão atomística, na qual a representação de átomos e moléculas, pelo menos em sistemas gasosos, foi muito favorecida. Em menor escala, também com efeitos positivos, está a compreensão da representação macroscópica da reação, ou seja, a mudança de cor. Estendendo estes resultados para toda a classe de *softwares* de simulação, percebe-se seu grande potencial como ferramentas de ensino de química e ciências em geral. Não foram observadas, contudo, diferenças em relação ao nível simbólico.

Sobre a terceira questão, acredita-se que há evidências que o conceito **velocidade** foi o menos suscetível a mudanças. Acredita-se que isso se deve ao fato de ele ser bastante mais abstrato que os demais, e difícil de ser compreendido com uso de representações não-simbólicas.

De uma forma geral, esses resultados foram coerentes com a literatura. É importante frisar que não se conhece os efeitos da inclusão deste *software* a longo prazo. Também são coerentes com a experiência dos professores de química que participaram da avaliação e do projeto deste *software*. Todos afirmaram que os estudantes não compreendem gráficos, sendo que um deles inclusive sugeriu que o módulo numérico fosse escondido.

Desta forma, modifica-se a metodologia, através da adição de ferramentas específicas para cada etapa, cuja escolha varia de caso para caso. A Figura 49 mostra as alterações, com as ferramentas sugeridas nesta discussão na cor azul.

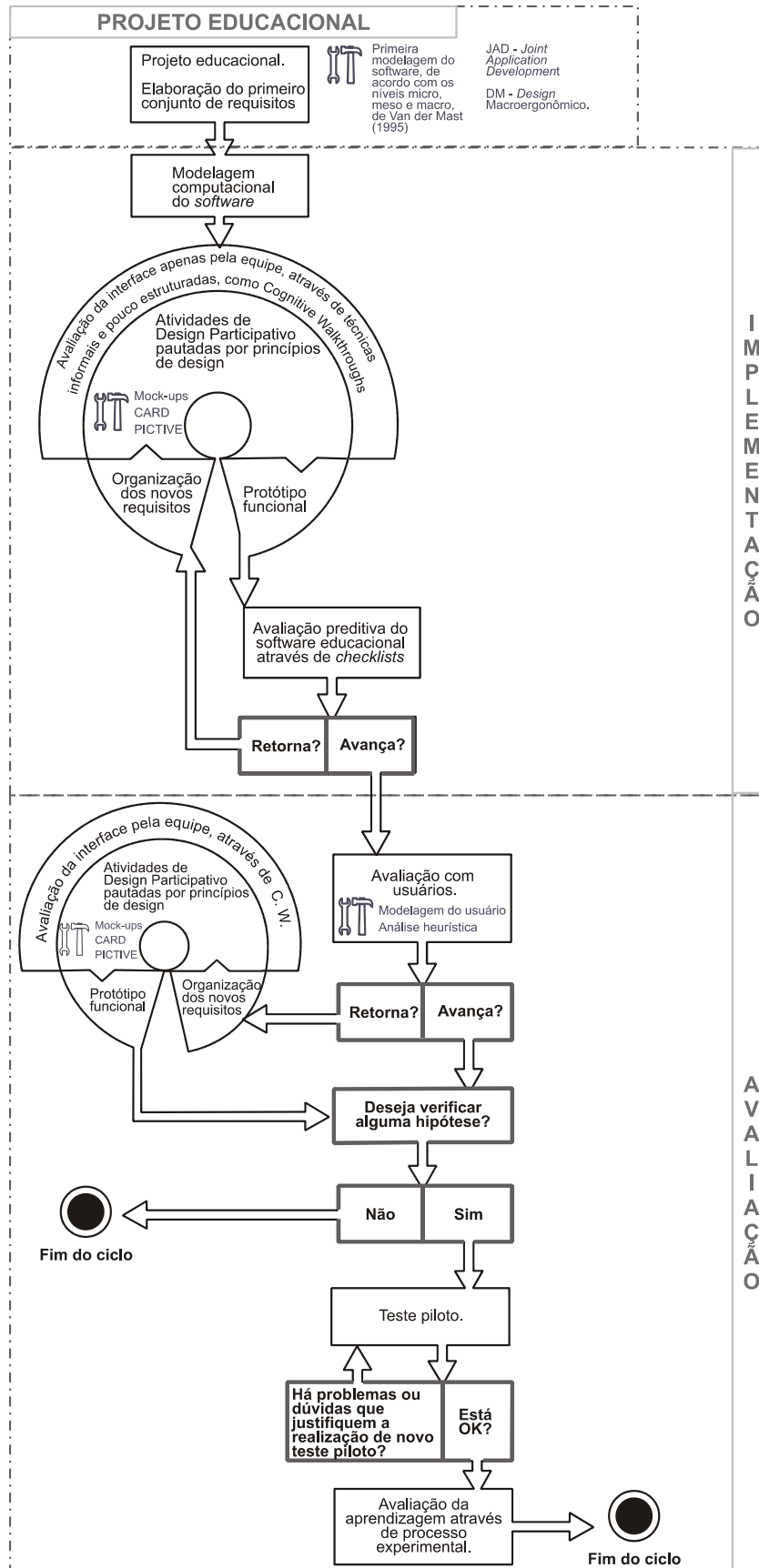


Figura 49: Modificações estruturais na metodologia e acréscimo de ferramentas.

A partir deste ponto é questionada a eficiência da metodologia em si. Idealmente, se desejava que a prática com as ferramentas na seqüência de etapas sugeridas nesta pesquisa levasse com segurança à qualidade do *software* educacional. No entanto, não foi possível determinar, neste trabalho, a força da relação entre estes dois fatores. É certo que, revisando a história da engenharia de *software*, se percebe que tais propostas surgiram da necessidade de responder a problemas que desafiavam as equipes de desenvolvimento. Porém, nenhuma delas oferece garantia de sucesso.

Assim, é importante refletir sobre quais fatores influenciaram de forma crítica o sucesso do produto (entenda-se, neste trabalho, sucesso como adequação pedagógica). Para responder esta questão, foram formuladas duas hipóteses. A primeira é que as diferentes atividades, realizadas durante a produção, têm impacto diferente no produto final. Tomando a interface gráfica como referência para julgamento, deseja-se saber como cada atividade colaborou para sua configuração atual. A segunda hipótese levantada é que a presença de especialistas leva à qualidade. Contudo, antes de iniciar o desenvolvimento da argumentação deve-se esclarecer que nenhuma destas duas questões poderá ser respondida de forma conclusiva.

O impacto das decisões tomadas, em cada fase, não pode ser quantificado, ou seja, não há como estabelecer relação de importância para as etapas (por exemplo: o projeto educacional é duas vezes mais importante que o teste piloto). Uma possibilidade de medir a influência de cada fase seria estabelecer uma escala ordinal de importância para cada componente da interface e relacioná-lo com a(s) fase(s) que participaram de sua implementação. Assim, por exemplo, poderia ser solicitado a um grupo de professores que arbitrasse um valor para cada componente e gerar, desta forma, a escala imaginada. Tendo posse da documentação do histórico do desenvolvimento, tal relacionamento poderia ser feito. A partir daí, seria possível, ainda, estabelecer marcos de referência para gerenciar projetos, que poderiam auxiliar a equipe a tomar vários tipos de decisões sobre: a disponibilidade de tempo para desenvolver um módulo de inteligência artificial e a necessidade de modificar a forma de interação em um programa que gera formas e calcula áreas, etc. No entanto, este trabalho não será feito em razão do tempo disponível para a conclusão da pesquisa. Por este motivo, tal questão será abordada de forma subjetiva, utilizando o julgamento pessoal de um dos membros da equipe de criação do *software* Equil. Para isso, é apresentado um mapa das duas telas da versão 2 do *software*, sendo atribuída uma cor para cada componente. É então julgada a importância de cada fase na construção de cada componente, e, de acordo com este

juízo, ele é pintado. Alguns componentes foram considerados *default* para um programa de equilíbrio químico. Estes são pintados de preto. O resultado deste procedimento é mostrado na Figura 50. Deve ser ressaltado que não é simples, em alguns casos, determinar com precisão o que é um componente, pois a maioria deles está articulada semanticamente com outros. Além disso, o mapa não deve ser interpretado pela área ocupada por cada cor, e sim pela predominância de cada cor em cada componente.

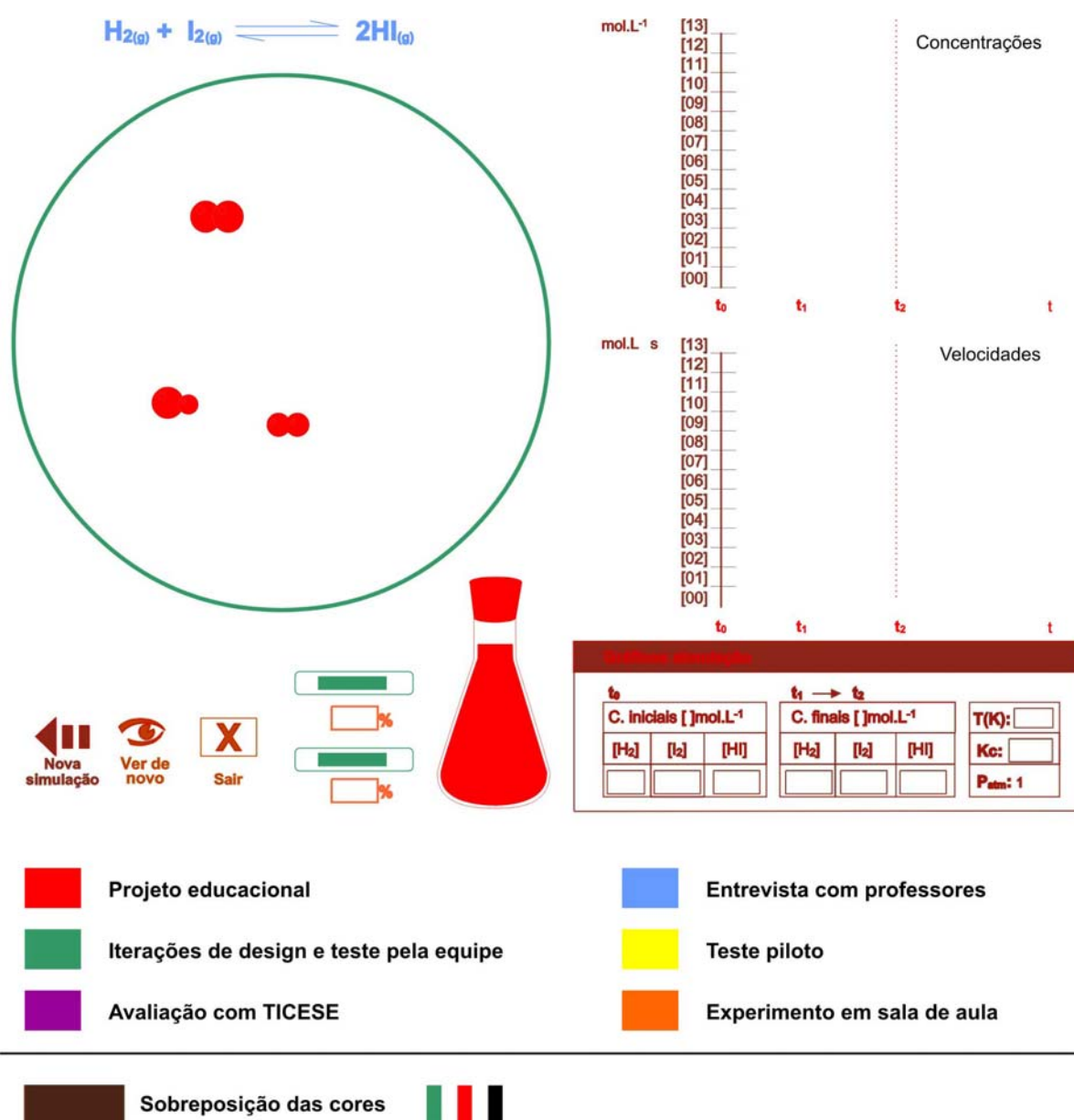


Figura 50: Mapa de cores da importância de cada etapa por componente.

Apesar de – como frisado diversas vezes – esta figura não ser conclusiva, alguns aspectos do processo de criação do *software* ficaram evidentes. O primeiro deles está relacionado com a complexidade do projeto. Nota-se poucos componentes pintados de preto (aqueles que se espera encontrar em qualquer programa sobre equilíbrio químico), o que indica a grande quantidade de idéias e de decisões. Caso se admita que, para cada componente não marcado de preto (isto é, para cada componente inédito) houve pelo menos mais de uma proposta, percebe-se uma intensa atividade de *design*, envolvendo criação, avaliação e modificação. Além disso, todos os componentes pintados de preto apresentam as tonalidades verde e vermelho, o que indica que houve decisões relacionadas ao *design* e ao projeto educacional em cada um deles. Todavia, fica claro que todos os componentes passaram por algum tipo de avaliação.

Outra questão fica realçada pela Figura 50, a da importância das etapas. Percebe-se que as etapas de *design* e projeto educacional competem pelo posto de maior influência. A presença das duas etapas é bastante equilibrada. Alguns componentes, apesar de tingidos com ambas as cores (verde e vermelho), devem sua aparência e funcionamento mais a uma etapa que a outra. Por exemplo, os componentes do nível simbólico (gráficos e módulo numérico), foram mais influenciados por decisões de *design* do que do projeto educacional. De fato, o projeto educacional deste *software* atribui menor importância a estes componentes do nível representacional simbólico, pois parte do princípio que eles não são utilizados pelos estudantes para raciocinar sobre equilíbrio. Isto não significa, de forma alguma, que sua importância para a modelagem do equilíbrio químico seja questionada. Por outro lado, nota-se que dois componentes (o frasco e as moléculas) estão tingidos apenas de vermelho. Pode parecer pouco em termos de área, porém, eles representam as bases sobre o qual o projeto educacional deste *software* foi feito: a importância da articulação entre os três níveis de representação e da boa visualização do nível microscópico para compreensão de equilíbrio químico. Na verdade, não é exagero afirmar que todas as demais decisões partiram deste princípio. Por isto, acredita-se que estes fatos constituem evidência de que o projeto educacional é a etapa mais determinante do processo, no sentido que partem daí as primeiras direções. Porém, um bom projeto educacional não é garantia de um bom *software*, pois a importância das decisões de *design* fica patente.

Uma vez aceitas estas conclusões, convém questionar a importância das etapas de avaliação. Afinal, a observação da Figura 50 pode induzir à idéia de que tais etapas podem ser eliminadas. Em face desta constatação podem ser feitas duas ponderações. A primeira é em

favor da etapa de entrevistas com professores, inteiramente responsável pelo componente de animação do tamanho da dupla seta, um item absolutamente fundamental do *software*. A segunda ponderação é em favor da realização de um teste piloto e do experimento. Caso não se saiba avaliar a qualidade educacional do *software*, a tarefa de atribuir importância a uma ou outra etapa do processo de desenvolvimento perde o sentido. Afinal, isto deveria ser feito separando *softwares* com qualidade dos sem qualidade.

A partir deste ponto, será elaborada a argumentação para verificar a segunda hipótese levantada pela aplicação desta metodologia: a da importância de especialistas na equipe. Para este fim, foi elaborado um balanço, mostrando que atividades consideradas críticas que não poderiam ter sido realizadas sem o auxílio de especialistas na área. A Figura 51 mostra a relação entre a atividade realizada com a presença de cada especialista. A importância foi, mais uma vez, atribuída de forma subjetiva, com base na experiência adquirida com a construção do programa. O nível de especialização considerado foi o dos participantes do projeto, ou seja, o especialista em educação possui doutorado em Ensino de Ciências, o *designer* possui bacharelado em *design* e mestrado em Ergonomia, o programador não possui experiência nem educação formal e os professores entrevistados possuem no mínimo mestrado. O papel de *designer* e programador foi acumulado pela mesma pessoa, e o papel de especialista no domínio foi dividido entre o especialista em educação e o *designer*/programador. O projeto de experimento foi feito pelo especialista em educação e a avaliação dos resultados foi feita pelo *designer*/programador.

Atividade	Necessidade de especialistas em			
	Educação	<i>Design</i>	Programação	Domínio (Equilíbrio químico)
Elaboração do primeiro conjunto de requisitos: projeto educacional	Muito necessário	Medianamente necessário		Muito necessário
<i>Design</i> da interface	Muito necessário	Muito necessário	Medianamente necessário	Muito necessário
Avaliação da interface com <i>checklists</i>	Pode ser realizada por qualquer um dos membros da equipe			
Professores entrevistados	Muito necessário			Muito necessário
Projeto do experimento	Muito necessária a presença de um especialista em estatística e do especialista na área que se deseja realizar o teste de hipótese.			
Condução do experimento	Medianamente necessário			Medianamente necessário
Avaliação dos resultados do experimento	Medianamente necessário			Muito necessário

Figura 51: Relacionamento entre área de especialidade e tarefa.

Esta pode ser comparada com a de Van der Mast (1995), mostrada na Figura 52.

	Domínio	Tecnologia educacional	Tecnologia de mídias	Engenharia de <i>software</i>	Gerenciamento de projeto
Seleção do problema educacional e definição dos requisitos para a solução.	X				X
Análise do contexto educacional e da natureza do assunto. Definição dos principais objetivos e de como testar o sucesso.	X	X			
Estudo da viabilidade de soluções alternativas via prototipagem.	X	X	X	X	X
Finalização do <i>design</i> macro.	X				
Projeto pedagógico.	X	X			
Seleção da mídia e <i>design</i> .	X	X	X		
<i>Design</i> conceitual da interação.	X	X	X	X	
Modelagem da simulação.	X		X	X	
Modelagem da interface.			X	X	
<i>Design</i> técnico.				X	
Integração.			X	X	
Procedimentos de revisão, melhoramento e aceitação.	X	X	X	X	X
Entrega do projeto.					X

Figura 52: Relacionamento entre as fases do jeito de trabalhar e as especialidades necessárias em cada uma. Extraída de Van der Mast (1995, p. 96). Um X indica que a especialidade está diretamente envolvida na atividade.

A comparação das Figuras 51 e 52 mostra diferenças na atribuição de importância a cada uma das especialidades. A razão para isto é que dois dos projetos investigados por Van der Mast eram *softwares* educacionais para treinamento em indústria, enquanto apenas um era para educação formal (ensino de línguas). Acredita-se seja possível concluir que quanto maior o grau de conhecimento implícito/tácito para adquirir *expertise* no assunto e quanto maior sua complexidade, maior a necessidade de especialista do domínio. No caso do *software* para Equilíbrio químico, devido à extensiva regulamentação da atividade de ensino (o que é necessário saber para considerar um sujeito hábil no domínio), houve uma necessidade menor de especialistas do domínio, em comparação a Van der Mast (1995). A importância dos sujeitos especialista em educação e especialista em tecnologia educacional é semelhante nas

duas figuras. No entanto, não é possível dizer se ambos têm a mesma qualificação, pois Van der Mast não especificou o grau de especialização dos sujeitos que pesquisou. A atribuição de importância aos especialistas em tecnologia (mídia e engenharia) é maior em Van de Mast por causa da maior complexidade dos projetos por ele estudados. Desta forma, acredita-se que seja direta a relação entre nível de especialização, quantidade de diferentes especialistas e qualidade do produto.

Finalizando este capítulo de discussões, comenta-se a respeito das diferenças entre os *softwares* Equil e Le Chat, que tratam do mesmo assunto, ou seja, equilíbrio químico. Em Paiva (2000), não se encontram referências explícitas à forma como foi conduzida a construção do Le Chat, ainda que o projeto educacional que apóia as escolhas tenha sido largamente documentado. Assim, de acordo com Paiva (2000, p.133), o objetivo do *software* seria:

visualizar as alterações produzidas em sistemas químicos gasosos por alterações de concentrações de reagentes ou produtos, temperatura do sistema ou pressão (volume) a que o sistema está sujeito, em conformidade com o princípio de Le Chatelier. [...] Do programa faz parte um conjunto de roteiros de exploração que poderão ser usados pelo aluno, nos respectivos níveis de ensino, com a colaboração ou não do professor. Estes roteiros estão disponíveis e são executáveis dentro do próprio programa.”

Em relação ao tratamento das concepções alternativas, Paiva (2000, p.144) afirma que:

O programa Le Chat 2.0 não tem uma relação de causa-efeito com as concepções alternativas, quer dizer, não é um conjunto explícito de ferramentas para desmontá-las sistemática e diretamente. Certo é, porém, que este *software* educativo é um instrumento útil para fazer face a algumas delas.

Parece haver dois fatores que determinam as diferenças observadas entre a forma como o tema equilíbrio químico foi tratado, sendo o primeiro o fato de o Le Chat não possuir uma estratégia educacional propriamente dita, voltada especificamente ao tema. O segundo fator seria o fato de o processo de construção do *software* não estar no centro das atenções do autor. Como evidência disto pode-se citar o fato de “incluir melhoramentos estéticos” (p.133) ser um dos objetivos a alcançar na segunda versão do Le Chat. Encerra-se esta discussão chamando atenção para o fato que o tratamento dado à interface, bem como sua integração às demais tarefas de implementação de um *software* (educacional ou não) não pode ser tratado

meramente dentro do plano estético. Por esta razão, a definição e o estudo de seus componentes e funcionalidades não deve ser uma etapa à parte, e sim uma atividade contínua dentro do ciclo de desenvolvimento.

CONCLUSÃO

Considera-se, sobre o conjunto de atividades que compõem a metodologia, que sua aplicação respondeu de forma satisfatória às duas questões norteadoras da pesquisa: à da integração e à da estruturação das atividades.

Os objetivos específicos também foram atendidos de forma satisfatória. Estes eram: (1) promover a integração da equipe de desenvolvimento com os professores e estudantes, durante o projeto, a implementação e a avaliação; (2) produzir um *software* educacional de acordo com esta proposta metodológica e (3) realizar uma avaliação experimental da aprendizagem, através de uma atividade educacional que envolva o uso deste *software*, em ambiente de ensino presencial.

Como indicação para futuros estudos, sugere-se que seja determinado o impacto de cada etapa da metodologia e a presença de cada tipo de profissional em relação: (1) ao processo de desenvolvimento e (2) à qualidade educacional do *software*. A resposta para a primeira questão seria especialmente importante para equipes de desenvolvimento que devem elaborar orçamentos e fixar prazos. A resposta à segunda questão seria importante tanto para o fornecedor do *software* quanto para seus clientes, pois, no caso de não haver recursos disponíveis na quantidade ideal, seria possível priorizar estes recursos e as atividades a ser realizadas.

REFERÊNCIAS

- ANDRADE, P. F. **Modelo Brasileiro de Informática na Educação**. In: RIBIE'94 - 2º Congresso Iberoamericano de Informática Educativa, Lisboa. Disponível em <<http://www.c5.cl/ieinvestiga/ribie94.htm>>. Acesso em 04 ago 2004.
- BAILEY, G.. In: INTERCHI, 1993, Amsterdam: ACM. **Iterative Methodology and Designer Training in Human Computer Interface Design**.
- BARNEA, N., DORI, Y. J. Computerized Molecular Modeling as a Tool to Improve Chemistry Teaching. **Journal of chemical information and Computer Science**, v. 36, p. 629-636, 1996.
- _____. Computerized Molecular Modeling the New Technology for Enhancing Model Perception among Chemistry Educators and Learners. **Chemistry Education: Research and Practice in Europe**, v. 1 n. 1, 109-120, 2000.
- _____. A Computer Aided Instruction Module on Polymers. **Journal of Chemical Information and Computer Sciences**, v. 33 n. 3, p. 325-331, 1993.
- BASTIEN, J. M. C.; SCAPIN, D. L.: **Ergonomic Criteria for the Evaluation of Human-Computer Interfaces**. Relatório de Pesquisa n° 0156. INRIA – Institut National de Recherche en Informatique et en Automatique. Rocquencourt, França, 1993. Disponível em: <<http://www.inria.fr/rrrt/rt-0156.html>>. Acesso em: 15 mar 2004.
- BATISTA, J.; FIGUEIREDO, A. D.: **Desenvolvimento de programas educativos por prototipagem continuamente evolutiva**. In: 2º Simpósio Investigação e Desenvolvimento de *Software* Educativo, 1997, Coimbra. Disponível em: <<http://lsm.dei.uc.pt/simposio/pdfs/c07.pdf>>. Acesso em 10 nov. 2004.
- BECK, K. et al.. **Manifesto for agile software development**, 2001. Disponível em: <<http://agilemanifesto.org>>. Acesso em 10 de jun. 2004.
- BERGQUIST, W.; HEIKKINEN, H. Student Ideas Regarding Chemical Equilibrium. **Journal of Chemical Education**, v. 67, p. 1000-1003, 1990.
- BJERKNER, G.: Some P.D. Advice. **Communications of the ACM**, v. 36 n. 4, p. 39, 1993.
- BOEHM, B. A Spiral Model of *Software* Development and Enhancement. **IEEE Computer**, v. 21 n. 5, p. 61-72. 1988.

- BODKER S.; IVERSEN, O. S. Staging a Professional Participatory *Design* Practice – Moving PD Beyond the Initial Fascination of User Involvement. In: NordiCHI'02, Arhus: ACM, p. 11-18, 2002.
- BORGHOLM, T.; MADSEN, K. H. Cooperative usability practices. **Communications of the ACM**, v. 42 n. 5, p. 91-97, 1999.
- BROWN, T. L.; LeMAY, H. E.; BURSTEN, B. E. **Química: Ciência Central**. Rio de Janeiro: LTC – Livros Técnicos e Científicos Editora S.A, 1994.
- BURD, L. **Desenvolvimento de Softwares para Atividades Educacionais**. Campinas: UNICAMP, 1999. 241 p. Dissertação (mestrado) – NIED – Núcleo de Informática na Educação, Universidade de Campinas, 1999.
- CAMACHO, M.; GOOD, R. Problem Solving and Chemical Equilibrium: Successful Versus Unsuccessful Performance. **Journal of Research in Science Teaching**, v. 26 n. 3, p. 251-272, 1989.
- CAMPBELL, D. T.; STANLEY, J.C, **Experimental and Quasi-Experimental Designs for Research on Teaching**. In N. L. Gage (ed.), Handbook of Research on Teaching. Chicago: Rand McNally, 1963.
- CAMPOS, F. C.; ROCHA, A. R. C. **Dez Etapas para o Desenvolvimento de Software Educacional Multimídia**. In: 3º Congresso Iberoamericano de Informática Educativa – RIBIE, 1996, Barranquillas.
- CARD, S. K.; MORAN, T. P.; NEWELL, A. The Keystroke-Level Model for User Performance Time With Interactive Systems. **Communications of the ACM**, v. 23, n. 7, 396-410, 1980.
- CARMEL, E.; WHITAKER, R. D.; GEORGE, J. F. PD and Joint Application *Design*: a Transatlantic Comparison. **Communications of the ACM**, v. 36 n. 6, p.40-48, 1993.
- CARRAHER, D. O Papel do Computador na Aprendizagem. **Acesso Revista de Educação e Informática**, São Paulo, v. 3. n. 5, p. 21-30, janeiro 1992.
- CASTRO, G. C. M.; AGUIAR, T. C. **Engenharia de Software no Processo de Desenvolvimento de Software Educacional Multimídia**. In: XXV Conferência Latino Americana de Informática - CLEI'00, 2000, Assunção.

COSMANN, R. The Evolution of Educational Computer *Software*. **Education** v. 116, p. 619-641. 1996.

CROSIER, J. K.; COBB, S.; WILSON, J. R.: Key Lessons for the *Design* and Integration of Virtual Environments in Secondary Science. **Computers & Education**, v. 38, p. 77-94, 2002.

CYBIS, W. A. **Engenharia de Usabilidade: Uma Visão Ergonômica**. Apostila, Maio 2003. Florianópolis, Maio 2003. Disponível em: <www.labiutil.inf.ufsc.br/Apostila_nvVersao.pdf>. Acesso em: 05 Mar 2004.

DORREGO, E. **Modelo Para la Producción y Evaluacion Formativa de Medios Instruccionales, Aplicado al Video y al Software**. In: RIBIE'94 - 2º Congresso Iberoamericano de Informática Educativa, Lisboa. Disponível em <<http://www.c5.cl/ieinvestiga/ribie94.htm>>. Acesso em 04 ago 2004.

DRIVER, R.; EASLEY, J. Pupils and Paradigms: a Review of Literature Related to Concept Development in Adolescent Science Students. **Studies in Science Education**, v. 5. p. 61-84, 1978.

FOGLIATTO, F. S.; GUIMARÃES, L. B. M.: *Design* Macroergonômico: uma Proposta Metodológica para Projeto de Produto. **Produto & Produção**, 3 (3), 1999.

GABEL, D. Use of the Particulate Nature of Matter in Developing Conceptual Understanding. **Journal of Chemical Education**, v. 70 n. 3, p. 193-194, 1993.

GAMEZ, L. **TICESE: Técnica de Inspeção de Conformidades Ergonômicas em Software Educacional**. Dissertação (mestrado). Universidade de Minho, 1998.

GIL, V. M. S.; PAIVA, J. M. C. **Programa Lechat - Simulações em Equilíbrio Químico (Princípio de Chatelier)**. In: RIBIE'94 - 2º Congresso Iberoamericano de Informática Educativa, Lisboa. Disponível em <<http://www.c5.cl/ieinvestiga/ribie94.htm>>. Acesso em 04 ago 2004.

GOMES, A. S.; WANDERLEY, E. G. **Elicitando Requisitos em Projetos de Software Educativo**. In: WIE 2003 – Workshop Brasileiro de Informática Educativa, 2003, Convergências Tecnológicas – Redesenhando as Fronteiras da Ciência e da Educação. Anais. Campinas: SBC, 2003, v. v.

GRØNBÆK, K.; KYNG, M.; MOGENSEN, P. Toward a cooperative experimental system development approach. In: Kyng, M., & Mathiassen, L. (Eds.), **Computers and design in**

context, p. 201-238, 1997. Cambridge, MA: The MIT Press. Disponível em <<http://www.daimi.au.dk/PB/412/PB-412.pdf>> Acesso em 10 abr 2004.

GRÜNBERG, J. A. **Profesores Y Computadores: Una Investigacion Sobre Los Factores Que Afectan El Uso De Computadores En Colegios Secundarios**. In: RIBIE'94 - 2º Congresso Iberoamericano de Informática Educativa, Lisboa. Disponível em <<http://www.c5.cl/ieinvestiga/ribie94.htm>>. Acesso em 04 ago 2004.

HACKLING, M. W.; GARNETT, P. J. Misconceptions of Chemical Equilibrium. **European Journal of Science Education**, v. 7 n. 2, p. 205-214, 1985.

HOCHSTEIN, L. **GOMS**, 2002. Disponível em: <<http://www.cs.umd.edu/class/fall2002/cmssc838s/tichi/goms.html>>. Acesso em 10 abr 2004.

KOZMA, R. B.; RUSSEL, J. Multimedia and Understanding: Expert and Novice Responses to Different Representations of Chemical Phenomena. **Journal of Research in Science Teaching**, v. 34 n. 9, p. 949-968, 1997.

LA TAILLE, Y. **Ensaio sobre o Lugar do Computador na Educação**. São Paulo: Iglu Editora, 1989.

LEVY, P. **Cibercultura**. São Paulo: Editora 34, 1999. 260 p.

LEWIS, C.; RIEMAN, J. **Task-Centered User Interface Design: A Practical Introduction**, 1994. Disponível em <<http://hcibib.org/tcuid/>> Acesso em 02 fev 2004.

MOLICH, R.; NIELSEN, J. Improving a Human-Computer Dialogue: What *designers* Know About Traditional Interface *Design*. **Communications of the ACM**, v. 33, p. 338-342, 1990.

MOREIRA, Marco Antônio. **Ensino e Aprendizagem: Enfoques Teóricos**. São Paulo: Editora Moraes, 1985.

MOREIRA, Marco Antônio. **Pesquisa em Ensino: o Vê Epistemológico de Gowin**. São Paulo: Editora Pedagógica e Universitária, 1990.

MULLER, M. J.: Layered Participatory Analysis: New Developments in the CARD Technique. In: **CHI'01 - Conference on Human Factors in Computer Systems**. New York: ACM, 2001, pp. 90-97.

_____: PICTIVE: an Exploration in Participatory *Design*. In: **CHI'91 - Conference on Human Factors in Computer Systems**. New Orleans: ACM, 1991, p. 225-231.

- MULLER, M. J.; WILDMAN, D. M.; WHITE, E. A. Taxonomy of PD Practices: a Brief Practitioner's Guide. *Communications of the ACM*, v. 36 n. 6, p. 26-27, 1993.
- MUÑOZ, P. A. Educación y Nuevas Tecnologías de la Información y la Comunicación en América Latina. **Tecnología y Comunicación Educativas**, v. 33, p. 4-27, jan/jun. 2001.
- MYERS, B. A. Challenges of HCI *Design* and Implementation. **Interactions**, p. 73-83, jan. 1994.
- NIAZ, M. Relationship Between Student Performance on Conceptual and Computational Problem of Chemical Equilibrium. **International Journal of Science Education**, v. 17 n. 3, p. 343-355, 1995.
- NIELSEN, J., LANDAUER, Thomas. **A Mathematical Model of the Finding of Usability Problems**. In: INTERCHI, p. 206-213, 1993, Amsterdam: ACM.
- NIELSEN, Jakob. **Finding Usability Problems Through Heuristic Evaluation**. In: CHI, p. 373-383, 1992, Monterey: ACM.
- ORLANDI, C. C. **Um Estudo Sobre a Utilização de Simulações Computacionais no Ensino de Equilíbrio químico**. Canoas: ULBRA, 2004. 271 p. Dissertação (mestrado) – Programa de Pós Graduação em Ensino de Ciências e Matemática, Universidade Luterana do Brasil, Canoas, 2004.
- PAIVA, J. C. **Ensino do Equilíbrio químico: Subtilezas e Simulações Computacionais**. Aveiro: Universidade de Aveiro, 2000. Tese (Doutorado) – Departamento de Química da Universidade de Aveiro. Portugal. Disponível em <www.fis.uc.pt/~jcpaiva/td> Acesso em 25 ago 2004.
- PAPERT, S. **Mindstorms: Children, Computers, and Powerful Ideas**. Basic Books: New York, 1982.
- PERNIN, J. **M.A.R.S.. Un Modèle Opérationnel de Conception de Simulations Pédagogiques**. Grenoble : Université Joseph Fourier, 1996. 276 p. Tese (Doutorado) - Laboratoire CLIPS – IMAG, Université Joseph Fourier, 1996.
- PRESSMAN, R.S. **Engenharia de Software**. São Paulo : Makron Books do Brasil, 1995, 3. ed, 1055 p.
- RATTNER, H. Os Impactos das Novas Tecnologias na Educação. **Acesso Revista de Educação e Informática**, São Paulo, v. 3. n. 5, p. 31-33, janeiro 1992.

ROYCE, W. **Managing the Development of Large Software Systems**. Proceedings of IEEE WESCON, August 1970.

RUBIN, J. **Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests**. New York : John Wiley, 1994.

SCHNEIDERMAN, B. **Designing the User Interface: Strategies for Effective Human Computer Interaction**. Addison-Wesley. 1998.

SOLOMONIDOU, C.; STAVRIDOU, H. *Design and Development of a Computer Learning Environment on the Basis of Student's Initial Conceptions and Learning Difficulties About Chemical Equilibrium*. **Education and information Technologies**, v. 6 n. 1, p. 5-27, 2001.

SPINUZZI, C.: A Scandinavian Challenge, a US Response: Methodological Assumptions in Scandinavians and US Prototyping Approaches. In: SIG'02. Ontario – Canada, p. 208-215.

SQUIRES, D.; McDOUGALL, A.: **Como Elegir y Utilizar Software Educativo: Guia para el Profesorado**, 1994, s/ed. Ed. Morata - Madrid. 174 pgs. Coleção: Pedagogia: Educación Crítica. Traduzido por Pablo Manzano, do original *Choosing and Using Educational Software: a Teachers' Guide*, ed. Taylor & Francis.

SQUIRES, D.; PREECE, J. Usability and Learning: Evaluating the Potential of Educational Software. **Computers and Education**, v. 17, n.1, p. 15-22. 1996.

VALENTE, J. A.; ALMEIDA, F. J. Visão Analítica da Informática na Educação no Brasil: A Questão da Formação do Professor. **Revista brasileira de Informática na Educação**, São Paulo, v. 1, setembro 1997. Disponível em <<http://www.inf.ufsc.br/sbc-ie/revista/nr1/valente.html>>. Acesso em 25 set 2004.

VAN DER MAST, C. A. P. G.: **Developing Educational Software: Integrating Disciplines and Media**. Delft: Technische Universiteit Delft. 276 p. Tese (Doutorado). Technische Universiteit Delft, 1995.

WU, H. K.; KRAJCIK, J. S.; SOLLOWAY, E. Promoting Understanding of Chemical Representations: Student's Use of a Visualization Tool in the Classroom. **Journal of Research in Science Teaching**, v 38 n. 7, p. 821-842, 2001.

ZAMBRANO, J. **Enseñanza Asistida por Computador y Producción de Software Educativo (PROSDOS)**. Caracas, Venezuela: Imprenta Universitaria Universidad Central de Venezuela, 1995.

APÊNDICE A - TERMO DE CONSENTIMENTO LIVRE INFORMADO

TERMO DE CONSENTIMENTO LIVRE INFORMADO

Eu, _____ declaro estar devidamente informado(a) e de acordo em participar do estudo sobre uma “Metodologia de Desenvolvimento Participativa de Software Educacional”, com o objetivo de colaborar com a pesquisa da mestrandia Gabriela Trindade Perry do Curso de pós-graduação em Engenharia de Produção da Universidade Federal do Rio Grande do Sul, com ênfase em Ergonomia.

Declaro estar ciente de que os dados coletados para esta pesquisa são de caráter sigiloso e não haverá publicação e nem identificação do(s) colaborador(es) participantes desta pesquisa.

Porto Alegre, ____ de _____ de 2005.

Estudante

Professora

Pesquisadora

APÊNDICE B - GUIA DE SIMULAÇÃO

Nome(s): _____

PROJETO DE PESQUISA

UTILIZAÇÃO DE EXPERIMENTOS VIRTUAIS NO ENSINO DE EQUILÍBRIO QUÍMICO GUIA PARA UTILIZAÇÃO DO SIMULADOR “EQUIL”.

Prezado estudante

Vamos estudar Equilíbrio Químico de forma integrada, utilizando um enfoque que aborda um fenômeno químico ocorrendo simultaneamente em três níveis:

- **Macroscópico:** São os fatores perceptíveis de uma reação química, como as cores e a aparência.
- **Microscópico:** É tudo o que não podemos observar a olho nu, como por exemplo, os átomos e moléculas, bem como as suas transformações.
- **Simbólico:** São os símbolos utilizados pelos químicos para representar os fenômenos químicos, como as equações, os gráficos, os coeficientes e outros.

Utilizaremos simulações computacionais. Assim, nossa atividade consiste em:

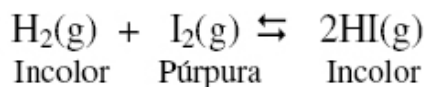
1º PREVISÃO: Sem iniciar a simulação, você irá descrever livremente o que você pensa sobre a questão, justificando sua previsão.

2º OBSERVAÇÃO: Você irá realizar a simulação, observar o fenômeno e comparar com a sua resposta da predição.

3º EXPLICAÇÃO: Você irá descrever possíveis semelhanças e/ou diferenças entre a sua resposta da predição com a resposta fornecida pelo computador.

Na folha seguinte está descrito o sistema que iremos estudar.

Um estudante investiga certa reação química que, após um determinado tempo, permanece em equilíbrio químico. Este estudante, no laboratório de química, misturou 1 mol de moléculas de $\text{H}_2(\text{g})$ e 1 mol de moléculas de $\text{I}_2(\text{g})$. Simbolicamente, o estudante escreve corretamente a equação química para esta reação:



Este estudante deseja estudar esta reação em três tempos:

T_0 : ESTADO INICIAL – logo no instante em que começa a reação.

T_1 : FASE INTERMEDIÁRIA – a reação já iniciada e antes de atingir o Equilíbrio.

T_2 : FASE FINAL – a reação no Equilíbrio Químico.

E em relação a três níveis de observação:

Nível macroscópico: A cor da mistura.

Nível microscópico: Os tipos de átomos utilizados, o comportamento destes átomos, o número de átomos e/ou moléculas que você está utilizando, o tamanho, como reagem, e como você imagina estes átomos dentro de um recipiente.

Nível Simbólico: O gráfico das concentrações dos reagentes e dos produtos em função do tempo.

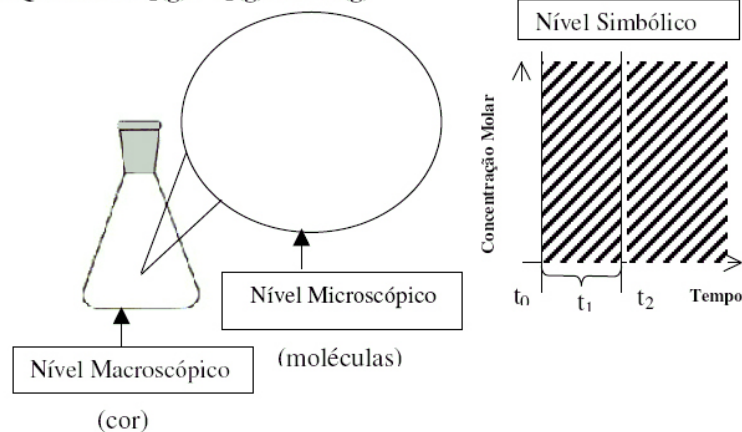
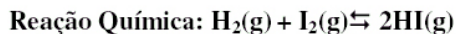
Ajude este estudante a responder as questões a seguir, de maneira que você utilize a sua **predição** sobre o que acontece com esta reação nos tempos t_0 , t_1 e t_2 :

ETAPA DA PREVISÃO

1.1 T₀ - O estado inicial da Reação.

Desenhe o gráfico das concentrações em função do tempo para t_0 apenas e, desenhe como você imagina o comportamento dos átomos e/ou moléculas, prevendo como fica a cor do frasco.

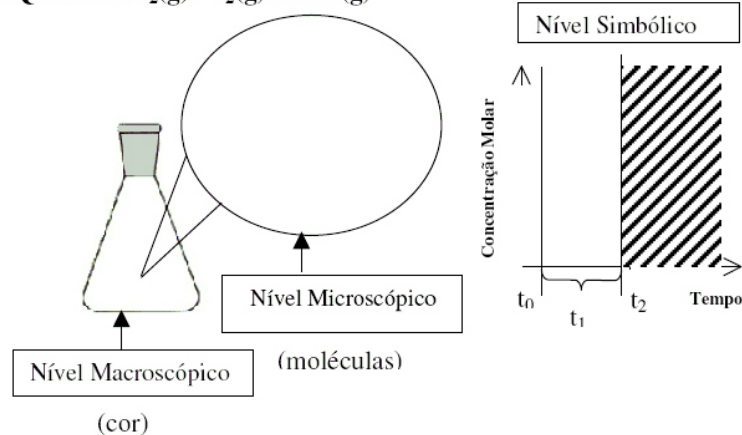
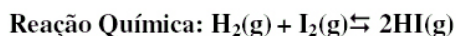
Justifique sua previsão:

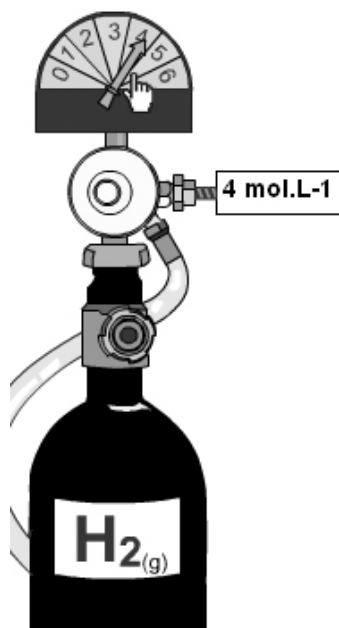


1.2 T₁ - A Reação já iniciada e antes de atingir o Equilíbrio.

O que acontece no recipiente utilizado pelo estudante, em nível macroscópico, microscópico e simbólico? Desenhe o gráfico das concentrações em função do tempo para t_0 e t_1 , desenhe como você imagina o comportamento dos átomos e/ou moléculas e a cor da mistura no frasco.

Justifique sua previsão:

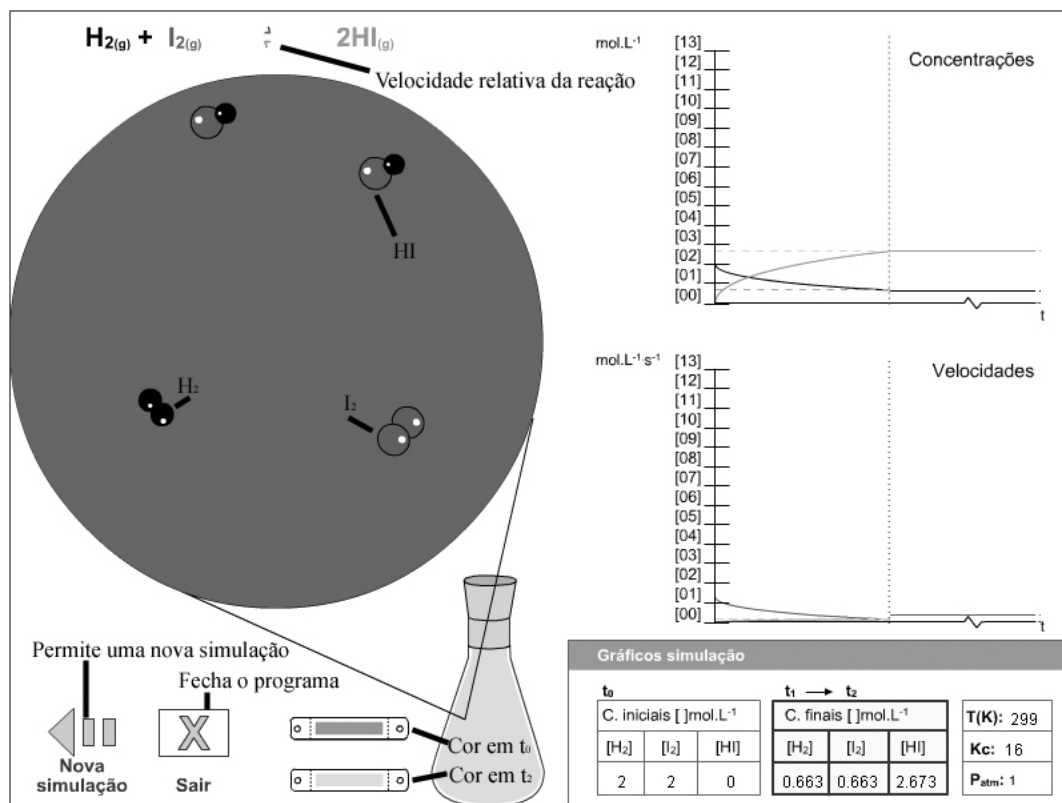




Em seguida, clique no botão abaixo para iniciar a simulação.



A figura abaixo mostra a tela do programa depois de pressionado este botão



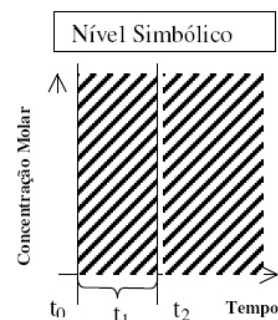
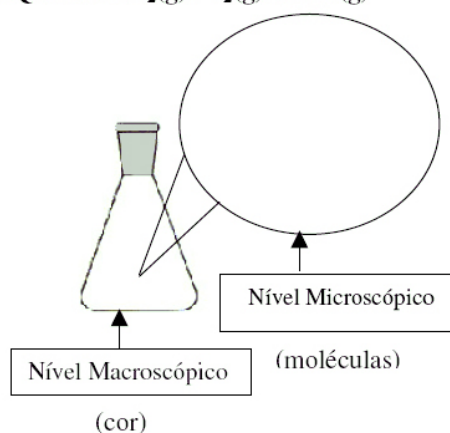
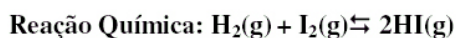
Agora, você irá observar a simulação, conforme os 3 tempos citados anteriormente (t_0 , t_1 e t_2). Veja o que acontece no estado inicial, logo no instante em que começa a reação (t_0), a reação já iniciada e antes de atingir o Equilíbrio (t_1) e no Equilíbrio Químico (t_2). Observe também que a simulação ocorre nos três níveis de representação, o macroscópico, o microscópico e o simbólico.

ETAPA DA OBSERVAÇÃO

2.1 T_0 - O estado inicial da Reação.

A partir da simulação, registre a sua observação.

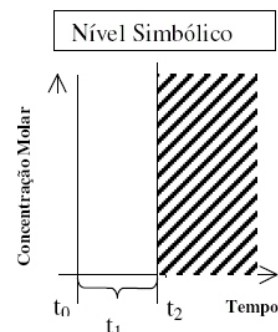
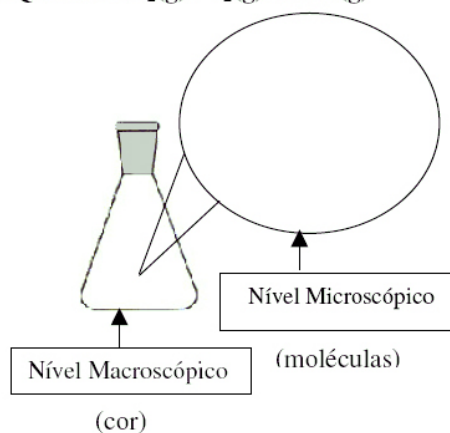
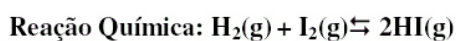
Justifique sua previsão:



2.2 T_1 - A Reação já iniciada e antes de atingir o Equilíbrio.

A partir da simulação, registre a sua observação.

Justifique sua previsão:



Você previu corretamente a cor do recipiente para o instante t_0 ? _____

—

3.2 T₁ - A Reação já iniciada e antes de atingir o Equilíbrio.

3.2.1 Você previu corretamente o estado atual (t_1) das concentrações dos reagentes e dos produtos? _____

E a tendência para a reação? _____

3.2.2 Você previu corretamente quais eram as moléculas existentes no frasco, para esta fase intermediária da reação? _____

3.3 T₂ - A Reação no Equilíbrio Químico.

3.3.1 Você previu corretamente o estado atual (t_2) das concentrações dos reagentes e dos produtos? _____

E a tendência para a reação? _____

3.3.2 Você previu corretamente quais eram as moléculas existentes no frasco, para esta fase?

Quais são as suas conclusões? _____

—

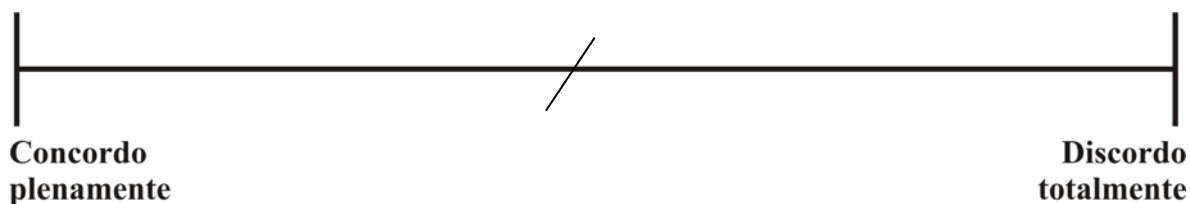
—

—

QUESTIONÁRIO DE AVALIAÇÃO REFERENTE À SIMULAÇÕES COMPUTACIONAIS

Procure responder o questionário livremente, sem se preocupar em agradar ou não o professor, desta forma você realmente estará contribuindo para uma avaliação satisfatória da atividade desenvolvida.


Nas questões que seguem, você deve marcar na linha conforme sua opinião, como no exemplo abaixo, onde o estudante está indeciso ou não tem opinião.



Leia com atenção cada afirmativa antes de expressar sua opinião.


Sinta-se à vontade para fazer qualquer comentário adicional, utilizando para isso o verso da folha de respostas.

1. A utilização de simulações durante as aulas me ajudou a refletir sobre a articulação entre os níveis de representação, macroscópico, microscópico e simbólico de um fenômeno químico.



**Concordo
plenamente** **Discordo
totalmente**

2. As atividades desenvolvidas durante as aulas de Química Geral II, como o pré-teste, a simulação computacional e o pós-teste permitiram que eu assimilasse melhor a teoria, os conceitos e as idéias sobre Equilíbrio Químico.




**Concordo
plenamente** **Discordo
totalmente**

3. Não gostei do método de ensino.



**Concordo
plenamente** **Discordo
totalmente**

4. Os conteúdos Equilíbrio Químico não despertam meu interesse



**Concordo
plenamente** **Discordo
totalmente**

5. Equilíbrio Químico é um assunto de difícil compreensão.

Concordo plenamente **Discordo totalmente**

6. Tenho a impressão de que aprendi bastante depois de utilizar, na atividade de simulação o método de ensino, predizer, observar e explicar.

Concordo plenamente **Discordo totalmente**

7. Respondi o questionário com atenção.

Concordo plenamente **Discordo totalmente**

9. A utilização de simulações computacionais me ajudou a entender o conceito de Equilíbrio Químico dinâmico.

Concordo plenamente **Discordo totalmente**

10. Sei utilizar bem diversos programas de informática (windows, Word, Excell, etc.)

**Concordo
plenamente** **Discordo
totalmente**

11. A utilização do software em duplas ou grupos atrapalhou meu desenvolvimento.

**Concordo
plenamente** **Discordo
totalmente**

—

Muito obrigada pela sua participação! Sua colaboração foi extremamente importante para esta pesquisa. Caso você tenha interesse em acompanhar sua evolução, os dados serão publicados em www.gabriela.trindade.nom.br onde serão também disponibilizadas as diferentes versões e eventuais atualizações do software.

APÊNDICE C - PÓS-TESTE

SOBRE ESTE INSTRUMENTO

Você responderá questões em relação a Equilíbrio Químico. Estas questões referem-se à: concentração molar, velocidade da reação, comportamento das moléculas e aparência do sistema. Há questões de múltiplas escolha e questões em que você deve desenhar moléculas e gráficos. Se você achar necessário, escreva na prova. Por favor, faça todas as suas anotações na prova, não utilize materiais de rascunho.

Os tópicos acima citados serão abordados em relação a três momentos, descritos abaixo:

t_0 => instante inicial, o momento da mistura,

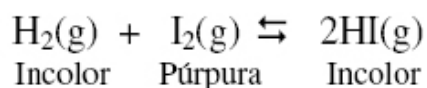
t_1 => antes da reação atingir o equilíbrio e

t_2 => depois que a reação atingiu o equilíbrio.

Se você tiver alguma dúvida em relação à interpretação das questões, peça auxílio ao seu professor ou ao pesquisador que está aplicando o teste. Dúvidas relacionadas à Química não poderão ser respondidas.

SITUAÇÃO PROPOSTA - EQUILÍBRIO QUÍMICO

No laboratório, um químico mistura, em um recipiente de 1L, 1 mol de gás Hidrogênio (H_2) e um 1 mol de vapor de Iodo (I_2), à temperatura de $458^\circ C$, conforme a equação:



1. RESPONDA AO BLOCO DE QUESTÕES ABAIXO, LEVANDO CONSIDERANDO O INSTANTE INICIAL DA REAÇÃO, OU SEJA t_0 .

1.1. CONCENTRAÇÃO DOS PRODUTOS (HI)

QUAL O VALOR ATUAL (t_0) DA CONCENTRAÇÃO?

- Máximo
 Mínimo
 Valor intermediário

QUAL A TENDÊNCIA DA CONCENTRAÇÃO?

- Aumentar
 Diminuir
 Permanecer a mesma

1.2. CONCENTRAÇÃO DOS REAGENTES (H_2) E (I_2)

QUAL O VALOR ATUAL (t_0) DA CONCENTRAÇÃO?

- Máximo
 Mínimo
 Valor intermediário

QUAL A TENDÊNCIA DA CONCENTRAÇÃO?

- Aumentar
 Diminuir
 Permanecer a mesma

1.3. VELOCIDADE DA REAÇÃO DIRETA (FORMAÇÃO DE HI)

QUAL O VALOR ATUAL (t_0) DA VELOCIDADE?

- Máximo
 Mínimo
 Valor intermediário

QUAL A TENDÊNCIA DA VELOCIDADE?

- Aumentar
 Diminuir
 Permanecer a mesma

1.4. VELOCIDADE DA REAÇÃO INVERSA (FORMAÇÃO DE H_2 E I_2)

QUAL O VALOR ATUAL (t_0) DA VELOCIDADE?

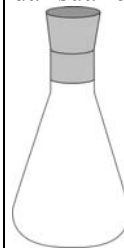
- Máximo
 Mínimo
 Valor intermediário

QUAL A TENDÊNCIA DA VELOCIDADE?

- Aumentar
 Diminuir
 Permanecer a mesma

1.5. Desenhe o comportamento dos átomos/moléculas no instante inicial da reação. Identifique os símbolos, caso você use algum.

1.6. Qual a cor do recipiente no momento da mistura? Use o código de cores fornecido para dar sua resposta.



2. RESPONDA AO BLOCO DE QUESTÕES ABAIXO, LEVANDO CONSIDERANDO QUE A REAÇÃO JÁ INICIOU, PORÉM AINDA NÃO ATINGIU O EQUILÍBRIO, OU SEJA T1.

2.1. CONCENTRAÇÃO DOS PRODUTOS (HI)

QUAL O VALOR ATUAL (t1) DA CONCENTRAÇÃO?

- Máximo
 Mínimo
 Valor intermediário

QUAL A TENDÊNCIA DA CONCENTRAÇÃO?

- Aumentar
 Diminuir
 Permanecer a mesma

2.2. CONCENTRAÇÃO DOS REAGENTES (H₂ E I₂)

QUAL O VALOR ATUAL (t1) DA CONCENTRAÇÃO?

- Máximo
 Mínimo
 Valor intermediário

QUAL A TENDÊNCIA DA CONCENTRAÇÃO?

- Aumentar
 Diminuir
 Permanecer a mesma

2.3. VELOCIDADE DA REAÇÃO DIRETA (FORMAÇÃO DE HI)

QUAL O VALOR ATUAL (t1) DA VELOCIDADE?

- Máximo
 Mínimo
 Valor intermediário

QUAL A TENDÊNCIA DA VELOCIDADE?

- Aumentar
 Diminuir
 Permanecer a mesma

2.4. VELOCIDADE DA REAÇÃO INVERSA (FORMAÇÃO DE H₂ E I₂)

QUAL O VALOR ATUAL (t1) DA VELOCIDADE?

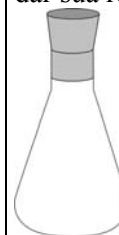
- Máximo
 Mínimo
 Valor intermediário

QUAL A TENDÊNCIA DA VELOCIDADE?

- Aumentar
 Diminuir
 Permanecer a mesma

2.5. Desenhe o comportamento dos átomos/moléculas no instante inicial da reação. Identifique os símbolos, caso você use algum.

2.6. Qual a cor do recipiente no momento da mistura? Use o código de cores fornecido para dar sua resposta.



RESPONDA AO BLOCO DE QUESTÕES ABAIXO, LEVANDO CONSIDERANDO QUE A REAÇÃO JÁ ATINGIU O EQUILÍBRIO, OU SEJA T₂.

CONCENTRAÇÃO DOS PRODUTOS (HI)

QUAL O VALOR ATUAL (t_0) DA CONCENTRAÇÃO?

- Máximo
- Mínimo
- Valor intermediário

QUAL A TENDÊNCIA DA CONCENTRAÇÃO?

- Aumentar
- Diminuir
- Permanecer a mesma

CONCENTRAÇÃO DOS REAGENTES (H₂ E I₂)

QUAL O VALOR ATUAL (t_0) DA CONCENTRAÇÃO?

- Máximo
- Mínimo
- Valor intermediário

QUAL A TENDÊNCIA DA CONCENTRAÇÃO?

- Aumentar
- Diminuir
- Permanecer a mesma

VELOCIDADE DA REAÇÃO DIRETA (FORMAÇÃO DE HI)

QUAL O VALOR ATUAL (t_0) DA VELOCIDADE?

- Máximo
- Mínimo
- Valor intermediário

QUAL A TENDÊNCIA DA VELOCIDADE?

- Aumentar
- Diminuir
- Permanecer a mesma

VELOCIDADE DA REAÇÃO INVERSA (FORMAÇÃO DE H₂ E I₂)

QUAL O VALOR ATUAL (t_0) DA VELOCIDADE?

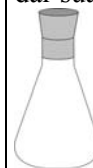
- Máximo
- Mínimo
- Valor intermediário

QUAL A TENDÊNCIA DA VELOCIDADE?

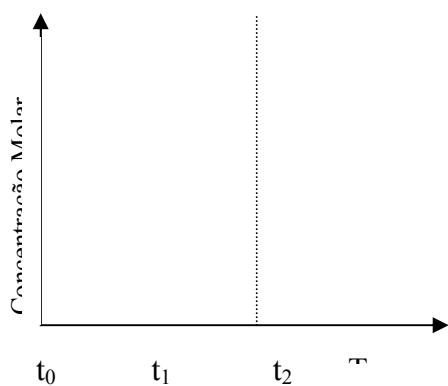
- Aumentar
- Diminuir
- Permanecer a mesma

Desenhe o comportamento dos átomos/moléculas no instante inicial da reação. Identifique os símbolos, caso você use algum.

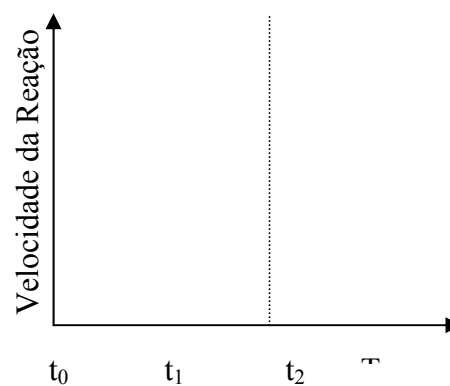
Qual a cor do recipiente no momento da mistura? Use o código de cores fornecido para dar sua resposta.



Agora, desenhe os gráficos das concentrações e da velocidade, em função do tempo.



Não esqueça de desenhar um gráfico para os produtos e outro para os reagentes. Identifique os gráficos, utilizando P para produtos e R para reagentes.



Não esqueça de desenhar um gráfico para a velocidade direta e outro para a velocidade inversa. Identifique os gráficos, utilizando D para velocidade direta e I para velocidade inversa.

Muito obrigada por sua participação! Tenha certeza de que todas as suas respostas foram muito importantes para este trabalho!

APÊNDICE D - MANUAL PARA INTERPRETAÇÃO DAS RESPOSTAS DOS ESTUDANTES DO PÓS-TESTE

1. RepMac – Representação Macroscópica de um sistema, desde o instante inicial até o Equilíbrio químico. Perguntas 1.6, 2.6 e 3.6.

4	3	2	1
Respostas corretas	Ordem certa, porém duas respostas consecutivas iguais.	Inversão.	Seqüência incoerente.
	Ordem certa, porém começa com 100%.	Demais casos que não se enquadram em 3 ou 1.	

2. ReAtom – Representação Atomística. Perguntas 1.5; 2.5 e 3.5.

	Diferencia átomos e moléculas.	Diferenciação dos raios atômicos.	Identifica espécies.	Distribuição uniforme*.	Representa 3 ou mais espécies*.	Desenha a mesma quantidade de espécies em todas as questões
4	Todos “sim”					Não avaliado
3	Um “não”					
2	Dois “não”					
1	Três ou mais “não”					

* Caso satisfaça ambas as condições ao mesmo tempo em qualquer uma das perguntas, considera-se um “sim” para estes quesitos.

3. AproEqMi – Aproximação do Equilíbrio no Nível Representacional Microscópico. Perguntas 1.5; 2.5.

	$t_0(1)$	Qtd. em t_1	$t_1 (n \geq 4)$	$t_1 (n=3)$
4	P=0	4+	P≤R, senão nível 3.	-
3		3	-	P e R
2(2)				Apenas P
1		$n \leq 2$	Não avaliado	

(1) Se há produtos em t_0 , ao final da avaliação subtrai-se 1 do nível final, para os que se enquadram nos níveis 3 ou 4.

(2) Se há compartimentalização classifica como 2, não importa o resto ou se não há evolução de t_0 para t_1 .

Obs.: **P** significa produtos e **R** reagentes.

4. RepEqMi – Representação do Equilíbrio Químico no Nível Representacional Microscópico. Pergunta 3.5.

	Quantidade t_2	R X P
4	4 ou mais	$P_{t_2} > P_{t_1}$ SE $P_{t_1} = P_{t_0} = 0$, nível 3(1)
3	3 ou menos	$P_{t_2} = P_{t_1}$ SE $P = 3$ ou $P_{t_2} < P_{t_1}$, nível 2.
2(2)		Apenas P.
1	Desconexo	

(1) P pode ser menor que R.

(2) Se satisfaz pelo menos o nível 2 e apresenta compartimentalização.

Obs.: P significa produtos e R reagentes.

5. AproEqCo – Aproximação do Equilíbrio para a Concentração para o Nível Representacional Simbólico. Perguntas 1.1; 1.2; 2.1 e 2.2.

	t_0		t_1	Gráfico
	R e P	Tendência		
4	Certo	Certo	Certo	Certo. Se invertido desce um nível.
3	Certo	Permanecer	Igual a t_0 ou um erro que possa ser atribuído à compreensão da pergunta. Se incoerente nível 2.	
2	Outros		Coerente com a inversão. Se desconexo nível 1.	Esboço correto, senão nível 1.
1	Desconexo ou respostas iguais para R e P.			Não avaliado

Obs.: P significa produtos e R reagentes.

6. RepEqCo – Concentração no Equilíbrio para o Nível Representacional Simbólico. Perguntas 3.1 e 3.2.

	P e R	Tendência	Gráficos	
	$P = \text{máx} / R = \text{mín}$	P e R permanecem?	$P = R$	$P \neq R$
4	Sim	Sim	-	Sim
3	Sim*		Sim	-
2	Não avalia		Se $\theta \neq 180^\circ$ nível 1	
1	Desconexo			

Obs.: P significa produtos e R reagentes.

7. AproEqVel – Aproximação do Equilíbrio para as Velocidades no Nível Representacional Simbólico. Perguntas 1.3; 1.4; 2.3 e 2.4.

	t_0		t_1	Gráfico	
	R e P	Tendência		Correto	Invertido
4	Certo	Certo	Certo	Certo	-
3	Certo. Se invertido, nível 2.	Certo	Igual a t_0 ou um erro que possa ser atribuído à compreensão da pergunta. Se incoerente nível 2.	-	Invertido
2	Certo	Não avalia	Coerente com a inversão ou igual a t_0 . Se desconexo nível 1.	Esboço correto, senão nível 1.	
1	Desconexo ou respostas iguais para R e P.			Não avaliado	

Obs.: **P** significa produtos e **R** reagentes.

8. VelocEqui- Velocidade no Equilíbrio no Nível Representacional Simbólico. Perguntas 3.3 e 3.4.

	P e R	Tendência	Gráficos
	P=mín. / R = máx.	P e R permanecem	P = R
4	Sim	Sim	Sim. Senão desce um nível.
3	P = R(1)		
2(2)	Não avalia		Se $\theta \neq 180^\circ$ nível 1
1	Desconexo		

(1) Apenas se gráfico e respostas anteriores coerentes com a não aceitação de que a velocidade direta diminui ao longo do processo.

(2) Gráficos de concentração ou velocidades diferentes no gráfico.

Obs.: **P** significa produtos e **R** reagentes.

APÊNDICE E – TABELAS DO χ^2 SIGNIFICATIVO A 95%

Tabela 1

Resultados das avaliações para a variável RepMac.

		RepMac			
Grupo		Notas 0, 1 e 2	Notas 3 e 4	Total	p
Controle	Frequência	12,00	8,00	20,00	
	% Grupo	60,00	40,00	100,00	
	% RepMac	80,00	32,00	50,00	
	Resíduo Ajustado	2,94	-2,94		
Equil	Frequência	3,00	17,00	20,00	
	% Grupo	15,00	85,00	100,00	0,00
	% RepMac	20,00	68,00	50,00	
	Resíduo Ajustado	-2,94	2,94		
Total	Frequência	15,00	25,00	40,00	
	% Grupo	37,50	62,50	100,00	
	% RepMac	100,00	100,00	100,00	

Tabela 2

Resultados das avaliações para a variável ReAtom.

		ReAtom			
Grupo		Notas 0, 1 e 2	Notas 3 e 4	Total	p
Controle	Frequência	17,00	3,00	20,00	
	% Grupo	85,00	15,00	100,00	
	% ReAtom	80,95	15,79	50,00	
	Resíduo Ajustado	4,12	-4,12		
Equil	Frequência	4,00	16,00	20,00	
	% Grupo	20,00	80,00	100,00	0,00
	% ReAtom	19,05	84,21	50,00	
	Resíduo Ajustado	-4,12	4,12		
Total	Frequência	21,00	19,00	40,00	
	% Grupo	52,50	47,50	100,00	
	% ReAtom	100,00	100,00	100,00	

Tabela 3

Resultados das avaliações para a variável AproEqMi.

		AproEqMi			
Grupo		Notas 0, 1 e 2	Notas 3 e 4	Total	p
Controle	Frequência	15,00	5,00	20,00	
	% Grupo	75,00	25,00	100,00	
	% AproEqMi	78,95	23,81	50,00	
	Resíduo Ajustado	3,48	-3,48		
Equil	Frequência	4,00	16,00	20,00	
	% Grupo	20,00	80,00	100,00	0,00
	% AproEqMi	21,05	76,19	50,00	
	Resíduo Ajustado	-3,48	3,48		
Total	Frequência	19,00	21,00	40,00	
	% Grupo	47,50	52,50	100,00	
	% AproEqMi	100,00	100,00	100,00	

Tabela 4

Resultados das avaliações para a variável RepEqMi.

		RepEqMi			
Grupo		Notas 0, 1 e 2	Notas 3 e 4	Total	p
Controle	Frequência	16,00	4,00	20,00	
	% Grupo	80,00	20,00	100,00	
	% RepEqMi	94,12	17,39	50,00	
	Resíduo Ajustado	4,80	-4,80		
Equil	Frequência	1,00	19,00	20,00	
	% Grupo	5,00	95,00	100,00	0,00
	% RepEqMi	5,88	82,61	50,00	
	Resíduo Ajustado	-4,80	4,80		
Total	Frequência	17,00	23,00	40,00	
	% Grupo	42,50	57,50	100,00	
	% RepEqMi	100,00	100,00	100,00	

Tabela 5

Resultados das avaliações para a variável AproEqCo.

		AproEqCo			
Grupo		Notas 0, 1 e 2	Notas 3 e 4	Total	p
Controle	Frequência	9,00	11,00	20,00	
	% Grupo	45,00	55,00	100,00	
	% AproEqCo	75,00	39,29	50,00	
	Resíduo Ajustado	2,07	-2,07		
Equil	Frequência	3,00	17,00	20,00	
	% Grupo	15,00	85,00	100,00	0,04
	% AproEqCo	25,00	60,71	50,00	
	Resíduo Ajustado	-2,07	2,07		
Total	Frequência	12,00	28,00	40,00	
	% Grupo	30,00	70,00	100,00	
	% AproEqCo	100,00	100,00	100,00	

Tabela 6

Resultados das avaliações para a variável RepEqCo.

		RepEqCo			
Grupo		Notas 0, 1 e 2	Notas 3 e 4	Total	p
Controle	Frequência	10,00	10,00	20,00	
	% Grupo	50,00	50,00	100,00	
	% RepEqCo	71,43	38,46	50,00	
	Resíduo Ajustado	1,99	-1,99		
Equil	Frequência	4,00	16,00	20,00	
	% Grupo	20,00	80,00	100,00	0,05
	% RepEqCo	28,57	61,54	50,00	
	Resíduo Ajustado	-1,99	1,99		
Total	Frequência	14,00	26,00	40,00	
	% Grupo	35,00	65,00	100,00	
	% RepEqCo	100,00	100,00	100,00	