

Capítulo 3 | Softwares educativos

3.1 | Software educativo e seu papel na informática educativa

3.2 | Breve histórico sobre softwares educativos

3.3 | Características e classificações dos softwares educativos

3.4 | Considerações sobre usuários de softwares educativos

3.5 | Aprendizagem colaborativa (CSCL) e Trabalho colaborativo (CSCW)

3.6 | Elementos e princípios de design em softwares educativos.

3.7 | Lições aprendidas

O capítulo 3 desta dissertação visou abordar o estado da arte dos softwares educativos, buscando apresentar além de algumas definições, um panorama histórico da evolução destas ferramentas. Foram estudadas ainda, as características e classificações desta categoria de softwares, bem como alguns aspectos relativos à interface e a forma como os princípios de design, educação e ergonomia nela se apresentam. Através da identificação dos usuários de softwares educativos, pretendeu-se entender também a importância do diálogo entre estes e a relevância desta relação usuário-software no contexto educacional, sem deixar de ponderar seus pontos positivos e negativos.

São apresentadas também as definições de CSCL (*Computer Suported Collaborative Learning*) e CSCW (*Computer suported Cooperative Work*), temas cujas pesquisas estão diretamente relacionadas ao desenvolvimento e avaliação de softwares educativos, discutidas por autores especializados que buscaram pesquisar os processos colaborativos para entender melhor sua aplicação neste tipo de software. Mesmo sabendo que o objeto de estudo desta pesquisa foi o de estabelecer uma investigação sobre métodos de avaliação de usabilidade, estamos cientes de que para avaliar estes métodos é preciso aplicá-los a um software educativo, pois, a princípio, esta é a forma mais lógica de testá-los. Logo, ao escolher esta categoria de software como suporte para realização dos testes, torna-se imprescindível estudar suas definições e esclarecer como se deu o surgimento desses artefatos, verificando as condições nas quais eles se desenvolveram e como chegaram a fazer parte da rotina das escolas entre alunos das mais variadas idades.

3.1 | Software educativo e seu papel na informática educativa

Segundo Cox (2003), softwares educativos são programas voltados especificamente para atividades de educação escolar desenvolvidas em salas de aula: histórias interativas, enciclopédias, dicionários, tutoriais, exercícios práticos, autoria e diversas outras ações como, por exemplo, softwares de simulação e jogos educacionais.

A definição sugerida por Cox (2003) não considera ainda que estes softwares possam ser usados em casa (ou outros ambientes), com a presença de um pai ou um outro adulto que faça o papel do educador, papel este que na escola seria do professor (ou tutor) da criança que estará utilizando o software. Também não considera, ainda, a questão de que existem softwares para auxiliar pessoas no aprendizado junto a um educador e outros cujos objetivos dizem respeito a auxiliar pessoas no processo de “auto-educar-se”.

Observando agora o conceito dado por Oliveira, Costa e Moreira (2001), *software* educacional é todo aquele utilizado em contextos de ensino-aprendizagem. Olhando por este ângulo, todo programa de computador poderia ser considerado como software educativo sempre que fosse usado com funções de aprendizado escolar, mesmo que esta não seja sua principal função.

Acredita-se que esta última definição seja um pouco mais abrangente que a de Cox (2003) e Garcia (2001) compartilha da mesma idéia, ao afirmar que software educativo é todo aquele que pode ser usado para algum objetivo educacional, por docentes ou alunos, qualquer que seja a natureza ou finalidade para a qual tenha sido criado.

Existem ainda autores que apresentam definições semelhantes ou complementares. Alguns como Galvis (1992), Basili (1994) e Vieira (1998), definem software educativo com a preocupação maior de avaliá-lo de forma global. Enquanto outros autores como Carraher (1990), Hisnostroza (1994) e Druin (1997), mostram definições de software educativo dentro de um contexto mais amplo, onde estão envolvidas também as questões sobre o uso das ferramentas e os papéis dos atores desse contexto: professores e alunos.

Assim como na pesquisa desenvolvida por Clunie (2000), neste estudo também se considerou software educativo como um programa computacional com características estruturais e funcionais, pedagogicamente sustentáveis, que se utiliza como **auxiliar no ensino**, na aprendizagem e na gerência educacional. Esse caráter auxiliar nos faz refletir sobre a importância da presença do professor, ou um outro educador, que possa estar por perto quando a criança estiver utilizando o software.

Sobre esta questão, da presença de um adulto (pai ou professor) junto à criança durante a utilização do software, Clunie (op. cit.) afirma ainda que o fato de se ter no mercado uma grande quantidade de softwares educativos à disposição das instituições de ensino, torna evidente a urgente necessidade de que o educador e o pessoal técnico de apoio educacional conheçam e saibam aplicar instrumentos para seleção e avaliação do software educacional.

De acordo com Tardif (2000), softwares educativos devem ser concebidos com base em princípios de cognição e de teorias construtivistas, visto que para o educador é importante identificar no software essas características psico-pedagógicas onde ele conseguirá estabelecer suas relações entre o conteúdo oferecido pelo material e o modo como ele pode aplicar o mesmo.

Tardif (op. cit.) analisa também da necessidade de uma organização dos conhecimentos prévios e estratégias de ensino do professor somados à sua habilidade de aliar estes conhecimentos à utilização do software educativo no contexto de aula. Complementando estas idéias, encontramos um estudo anterior ao de Tardif, onde Carraher (1990) já explicava que um software educativo por si só, não é um “estimulante” para o aprendizado, pois seu uso precisa estar apoiado numa integração entre os envolvidos e as demais tarefas do contexto.

A partir dos estudos feitos por autores como, Carraher (1990), Clunie (2000) e Tardif (2000), entendemos que essa crescente oferta de softwares educativos leva a sociedade a ter acesso rápido a inúmeras possibilidades de aquisição e uso desses artefatos. Tem-se observado que as pessoas buscam e utilizam estes softwares como alternativas para reforço da informação vista em sala de aula na escola e, deste modo, entende-se que este uso tende a ser interpretado como uma intensificação da aprendizagem sobre um conteúdo.

Mas a partir do momento em que a utilização destes softwares ocorre como parte integrante do processo de ensino-aprendizagem, como, por exemplo, quando são usados em sala de aula, acredita-se então que eles não serão mais vistos como um mero reforço escolar, eles começariam a assumir sua posição na relação entre o aluno e o professor, bem como se mostram como um fator importante no diálogo que ocorre entre eles na busca pela construção de conhecimento.

Sobre este diálogo entre os usuários de softwares educativos, autores como Netto (2001) e também Peres e Meira (2002) alertam que o diálogo ocorre mediante um entendimento do conteúdo que será estudado, dos conceitos pertinentes ao tema e da situação contextual em que o software está sendo utilizado por estes usuários. Mais adiante, no tópico 3.3, sobre perfil dos usuários de softwares educativos, foram discutidos estes diálogos com maior aprofundamento, mas por hora, é mais importante abordar o papel do software educativo na escola.

É sabido que há muito se fala na importância do computador e do software educativo como parte do contexto escolar, a prova disto são estudos como os de Alava et al (2001), Dowbor (2001) e Kensi (2003). Estes autores discutem a respeito da utilização destas ferramentas apontando para necessidade de assumir o potencial pedagógico e manter a cautela no que se refere a “supervalorização” do software em detrimento de outras ferramentas mais “tradicionais” ou “convencionais”, tais como ferramentas analógicas: jogos educativos feitos de madeira e etc.

Conforme dito no capítulo anterior, existe uma discussão científica sobre a melhor forma de aliar tecnologia e educação. Contudo, não se pode dizer que a inserção do computador em sala de aula causou grandes mudanças na área da educação e trouxe novas reflexões para pais, educadores e também para designers (e outros desenvolvedores de softwares educativos). O ponto mais importante das reflexões, propostas pelos pesquisadores de software educativo, é a busca por um uso deste instrumento que possa contribuir de forma efetiva para o processo educacional, sem deixar que os usuários (crianças e adultos) sejam tomados por um sentimento de “admiração” ou “encantamento” que ofusque o objetivo principal do computador em sala de aula, que é o de auxiliar no processo de ensino-aprendizagem.

Sobre esta discussão, viu-se que há autores que defendem os pontos positivos do uso de computadores e softwares nas escolas, como Tapscott (1998), Tajra (2002) e Valente (2003) e há muitos outros que enxergam mais pontos negativos do que positivos, como Setzer (1988), Weiss (1998) e Armstrong & Casement (2001). O fato é que, mesmo que esse uso traga pontos positivos e negativos, ele traz também uma preocupação do governo brasileiro em conseguir superar as dificuldades, a fim de investir mais na educação do país, colocando a informática como parte da rotina escolar. Esta preocupação se mostra com tamanha importância, que já foram desenvolvidos inúmeros programas de governo voltados para discussão da informática educativa nas escolas brasileiras, assim como o tema encontra-se em discussão através de outras nomenclaturas, conforme a área de pesquisa em que está sendo pesquisado.

Segundo Santos (2003), antes do PROINFO (Programa de Informática na Educação), existiram três outros projetos que apresentavam suas propostas a respeito da informática educativa, contudo, segundo o autor, não eram convenientes à realidade do novo milênio, em um cenário onde a palavra de ordem era globalização. Os programas citados por Santos (2003), são Educom, Eureka e Gênese, que foram iniciados, respectivamente, em 1984, 1990 e 1992; O PROINFO nasceu em 1997 e foi posto em prática em 1998.

Uma questão que deve ser mencionada é que, enquanto o governo discutia o papel da informática em escolas públicas, nas escolas particulares já se via em prática a utilização dos computadores em sala de aula. Entretanto, esta é uma outra discussão que não é pertinente a esta pesquisa, embora se saiba que alguns autores, como Moraes (2002), asseguram que a inconsistência e fragilidade desses programas têm origem em políticas educacionais que valorizam as técnicas e métodos em

detrimento da educação em si, trazendo à tona o que a autora chama de educação tecnicista, baseada em “pensamentos sistemáticos e sequenciais do conteúdo”. Em contrapartida a esta idéia de educação tecnicista, encontramos a opinião de Valente (2004) afirmando que:

“O termo ‘Informática na Educação’ significa a inserção do computador no processo de aprendizagem dos conteúdos curriculares de todos os níveis e modalidades de educação. Para tanto, o professor da disciplina deve ter conhecimento sobre os potenciais educacionais do computador e ser capaz de alternar adequadamente atividades tradicionais de ensino-aprendizagem e atividades que usam o computador. (...) Portanto, os desafios na implementação do computador na escola objetivando uma mudança educacional são enormes. No entanto, se eles não forem atacados corremos o risco de perpetuarmos uma escola que já é obsoleta. Só que agora, ela será obsoleta, porém, usando a informática.”

Compreendeu-se que a intenção de Valente é mostrar que a informática educativa, assim como o uso de softwares educativos, deve fazer parte de uma rotina que valorize primeiramente a educação. Deste modo, não se deve deixar que a mesma torne-se secundária, na medida em que um professor precise adaptar o conteúdo educacional à ferramenta quando, na verdade, ele deveria descobrir qual o potencial pedagógico que aquela ferramenta tem, além de como ela pode adaptar-se à sua prática docente.

No que concerne às outras nomenclaturas utilizadas quando se discute o tema informática educativa e softwares educativos, é importante salientar que existem estudos sobre educação à distância, como os de Alava et al (2001) e Pallof e Pratt (2002), que se dedicam à questão dos softwares educativos dentro de um contexto mais amplo. Conseqüentemente, na visão dos autores, estas ferramentas passam a ser parte de um conjunto de inúmeros artefatos auxiliares do processo de ensino aprendizagem que incluem não só softwares, como também uso da Internet e a formação de comunidades para educação à distância.

Sendo assim, a título de complementação, vale frisar que outros estudos relacionados ao uso de softwares educativos também podem ser encontrados com nomenclaturas, ou siglas, conforme as que aparecem nos livros de Galvis (1997) e Filatro (2002) e que podem se relacionar, direta ou indiretamente, com o uso softwares educativos em diferentes contextos, como por exemplo:

- DIC = Design Instrucional Contextualizado;
- CAI = Instrução Assistida Pelo Computador;
- PEI = Produtos Educacionais Informatizados;
- EAD = Educação À Distância;
- AVA = Ambientes Virtuais De Aprendizagem;
- TIC = Tecnologia De Informação E Comunicação;
- MEC = Materiais Educativos Computadorizados;
- CSCL (*Computer Suported Collaborative Learning*);
- CSCW (*Computer Suported Cooperative Work*).

O que há de comum entre todas essas nomenclaturas é que, em regra geral, elas tratam de quatro pontos principais na construção de saberes: (i) o indivíduo (aluno e/ou professor), (ii) o conteúdo (também chamado de saber ou de conhecimento), (iii) a mediação (também citada como interação) e (iv) a ferramenta (sistemas computadorizados) que para os efeitos deste estudo, restringiu-se o termo ferramentas a softwares educativos.

Para exemplificar as relações entre estes quatro pontos, são apresentados a seguir a figura 3.1 que mostra como Alava et al (2001) apresentam as ligações existentes nesta relação que se está interpretando como sendo a relação interação-usuário-software-educativo:

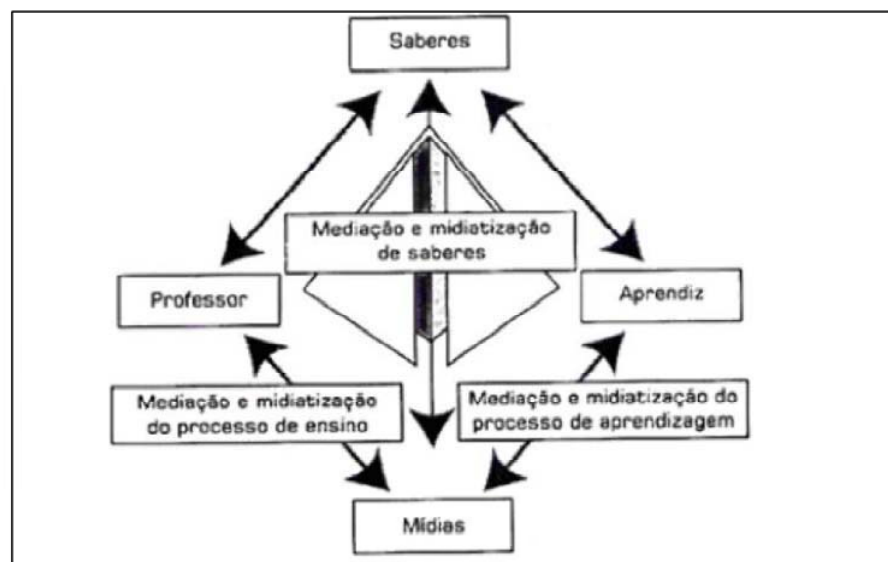


Figura 3.1: Ligações entre saberes, indivíduos, conteúdos e mídias (ALAVA et al, 2001).

A figura mostra que o uso de mídias (que para os fins desta são os softwares educativos) deve ser encarado como mais um ponto de apoio nas relações de ensino-aprendizagem entre professores e aprendizes. Devem ser observadas também as “pontes” (ligações) estabelecidas entre mídias, professores, alunos e saberes (ou conhecimentos), pois através delas é permitido aprender que as mediações existentes entre esses elementos influenciam no processo de ensino do professor, bem como no processo de aprendizagem do aluno.

Desta forma, para concluir nossas reflexões sobre o conceito de software educativo e seu papel na informática educativa, foi aceito, para os fins desta pesquisa, que um software é educativo quando seu foco está direcionado para um contexto de ensino apoiado por bases pedagógicas. Sua importância na informática educativa se dá no momento em que ele consegue oferecer “pontes” entres os elementos e usuários envolvidos no processo de ensino-aprendizagem, isto é, fazer parte das relações entre professores, alunos, conteúdo e contexto, sem “sobrepôr” nenhuma destas relações.

3.2 | Breve histórico sobre softwares educativos

Para apresentar um panorama geral sobre a história dos softwares educativos, tomou-se como ponto de partida algumas das informações fornecidas por Gomes (2005) em suas aulas sobre engenharia de software educativo, ministradas na UFPE (Universidade Federal de Pernambuco). O professor explicou que durante o período pós-guerra (meados de 1950) houve uma notável necessidade de desenvolver meios de comunicação em massa para promover a formação de novos profissionais da área de comunicação, entre outras necessidades sócio-econômicas e político-culturais da época.

Foi um período onde se falou muito em pesquisas da área de psicologia e alguns estudiosos, como Skinner (1938), discutiam suas teorias a respeito do Behaviorismo. No capítulo anterior, foi visto que esta teoria caracterizava-se pela defesa de que os processos cognitivos poderiam ser investigados a partir do ensino programado, ou seja, uma automatização e individualização do indivíduo durante o processo de ensino-aprendizagem.

Segundo esta premissa, os professores das escolas criavam condições para que os elementos fossem trabalhados através da idéia de “reforço” das respostas positivas e da “repreensão” das respostas negativas, no sentido de alertar o aluno para fato de que deveria prestar atenção nos acertos e não repetir os mesmo erros. Acredita-se, assim como os autores Carneiro (2003) e Cox (2003), que talvez esta não seja a forma ideal para educar indivíduos, mas é sabido que ainda é aplicada em diversas escolas e serve também como fundamentação para softwares educativos.

A importância de comentar novamente essas teorias de educação reside no fato de que muitas delas influenciaram, e ainda influenciam, o desenvolvimento de diversos softwares educativos, segundo estudos da professora Campos (1997). Dessa forma, é preciso entender parte das discussões sociais e educacionais (até mesmo psicológicas) ocorridas desde a década de 50, pois só assim é possível compreender o porquê da aplicação dessas teorias no desenvolvimento dos softwares.

Nos anos 20 (antes deste período pós-guerra referido anteriormente), encontram-se algumas ações que já apontavam para uma “queda” destas correntes positivistas em educação, nas quais era possível perceber que o comportamento e aprendizagem deveriam ser julgados considerando o contexto, e não somente fatos isolados, de acordo com estudos sobre interdisciplinaridade (FAZENDA, 1999). Neste período, foi criada a máquina de Pressey (a Drum Tutor) baseada no princípio do psicólogo Thorndike, onde havia uma única questão com uma sequência de respostas (do tipo múltipla escolha) para o aluno decidir pela certa. De acordo com Almeida (2001), em sua tese de doutorado, esta máquina foi criada em 1924 e nela o aluno não poderia ir para a instrução seguinte sem que tivesse sucesso prévio, pois a máquina continha um teclado com quatro chaves para busca de respostas (múltipla escolha) e uma única janela (onde eram colocadas as questões) atuava como prevenção para que o aluno não seguisse adiante sem sucesso na questão anterior.

Retornando à sequência histórica, foi observado que este paradigma de ensino através de uma lógica, onde o acerto significa “siga adiante” e o erro significa “não, faça de novo”, começa a mudar a partir dos estudos de Piaget, que começa a propor o Construtivismo entre os anos 30 e 40, embora suas obras só começassem a ser traduzidas para outras línguas nas décadas de 50 e 60 (FLAVELL, 1992).

A respeito da obra de Piaget, Flavell (1992) e Ausubel (1983) explicam que o desenvolvimento cognitivo do indivíduo durante a sua evolução, passa por fases ou períodos de desenvolvimento, onde são identificados estágios como o sensório-motor, o pré-operatório, o operatório-concreto e o formal, que se caracterizam por apresentar estruturas de funcionamento qualitativamente diferentes e, conseqüentemente, lógicas diferenciadas no aprendizado de adultos e crianças.

Por volta dos anos 60, encontram-se as primeiras traduções das pesquisas de Vygotsky, onde a união entre psicologia e educação passava pela investigação a respeito do contexto onde o indivíduo estava situado. Piaget já apresentava sua teoria construtivista, assim como Vygotsky apresentava sua teoria sócio-construtivista e ambos acreditavam no conhecimento prévio do indivíduo e na influência do meio social sobre o conhecimento apresentado por este. Estas idéias influenciavam decisivamente na educação e, posteriormente, influenciariam também no desenvolvimento de softwares educativos (CAMPOS, 1997).

Esta visão sugeria um aprendizado estruturado e adaptado, de modo que o aluno pudesse ser estimulado a refletir e responder corretamente aos questionamentos colocados pelo professor. Esta forma de incentivar o processo de aprendizagem é encontrada até hoje em muitos softwares educativos que não seguem um modelo linear (contínuo) de ensino (como a máquina de Pressey), pelo contrário, sabe-se que eles buscam “ensinar” algo à medida que o aluno sente-se pronto para seguir adiante.

Um dos primeiros estudiosos a propor um modelo não linear foi Crowder (1959). Este pesquisador desenvolveu uma nova proposta de ensino onde a interface de navegação oferecia “conexões” ao aluno, isto é, oferecia um modelo não-linear, onde o aluno poderia passar para tarefa seguinte de acordo com a resposta resultante da tarefa anterior. Ele não precisava ficar limitado a uma sequência de telas com respostas do tipo “sim” ou “não”, essa nova organização possibilitava sempre seguir adiante e continuar o aprendizado. Depois de Crowder (1959), houve uma evolução quanto à natureza “cooperativa” da máquina, a partir da proposta de Pask (1959), onde havia um programa tutor que orientava o aluno a seguir adiante quando obtivesse um acerto, pois o programa analisava o aprendizado do aluno antes de oferecer uma nova opção de tarefa. Tanto a proposta de Crowder, como a de Pask, fez parte do que os autores consideram como ensino programado, assim como os exemplos de programas desse período, *Plato*, *Cerl* e *Ticcit*, entre outros também citados por Almeida (2001).

Durante a década de 50, foram criados programas que atuavam como “treinadores” ou “simuladores”. Estes eram usados em situações fictícias, de modo que o aluno precisava aprender como proceder e quais medidas tomar, caso aquela situação acontecesse na vida real, como no caso de simuladores de vôo ou de navegação, por exemplo: *The Whirlwind*. Nessa mesma época, a empresa americana IBM criou o *Coursewriter* para o ensino de matemática binária, que mais tarde foi aperfeiçoado e transformado no *Excheck* para ensino de Lógica e teoria dos conjuntos (ALMEIDA, 2001).

Mas foi em meados dos anos 60 que começaram a surgir os primeiros programas influenciados pelo Construtivismo e, neste momento, diversos estudiosos (Demogli, Piaget, Morin) lançaram a proposta de que a aprendizagem não deve ser vista como uma sucessão de “entradas e saídas”, mas sim, como uma construção do conhecimento a partir de atividades humanas (ação, interação e colaboração), de modo que o indivíduo não poderia ser tratado como um mecanismo de trabalho que, simplesmente, responde perguntas.

Neste período (anos 60 e 70) foram encontrados textos de Seymour Papert, realizando suas pesquisas sobre uso de computadores por crianças no MIT (*Massachusetts Institute of Technology*). Foi uma fase onde se assistiu o surgimento da linguagem LOGO, termo utilizado para designar simultaneamente uma teoria de aprendizagem, uma linguagem de programação, um material que permite ao indivíduo demonstrar os processos mentais empregados na resolução de problemas num contexto de ação sobre o mundo exterior, considerada como o primeiro exemplo de interface gráfica educativa, de fato.

Nesta época, Papert foi até a suíça e estudou junto com Piaget, sendo assim, o LOGO foi discutido como proposta de uma nova “filosofia” de educação por meio da auto aprendizagem. Segundo relatos de Chaves (2005) em sua contribuição para o livro de Setzer (1988), comenta-se inclusive este encontro:

“ A filosofia emergiu dos contatos de Papert, de um lado, com a obra do psicólogo e epistemólogo suíço Jean Piaget, e, de outro lado, com as pesquisas, do MIT e de outros centros de estudo, sobre o problema da inteligência artificial. Fundamentando-a em uma filosofia da educação, não é de admirar que os criadores da linguagem de programação LOGO tenham resolvido torná-la um instrumento, o mais adequado possível, para aplicações na área educacional”

Chaves (2005) explica que Papert criou o que seria a primeira interface educativa baseada em idéias construtivistas, pois o LOGO era um programa onde havia uma tartaruga que desenhava um caminho pelo chão conforme as diretrizes e comandos indicados pelos alunos. Deste modo, o *feedback* (resposta) do programa seria de acordo com a ação ordenada a ele e caso o aluno errasse nesse comando, ele não seria punido pelo erro, mas sim, poderia continuar a realizar sua tarefa, sem ônus.

Segundo Gomes (2005) um software educativo deve ser uma projeção de um modelo de ensino aprendizagem em função de um perfil de usuário, projeção esta que pode ter relação, ou não, com seu mundo real. Sendo assim, acredita-se que, a partir da colocação do professor, fica mais fácil explicar como surgiram os “micromundos”, que são definidos por este autor, como metáforas de um mundo real. A figura 3.2 a seguir, que foi apresentada durante as aulas de Gomes (2005) através de seu material expositivo, ajuda a compreender esta definição de micromundos:

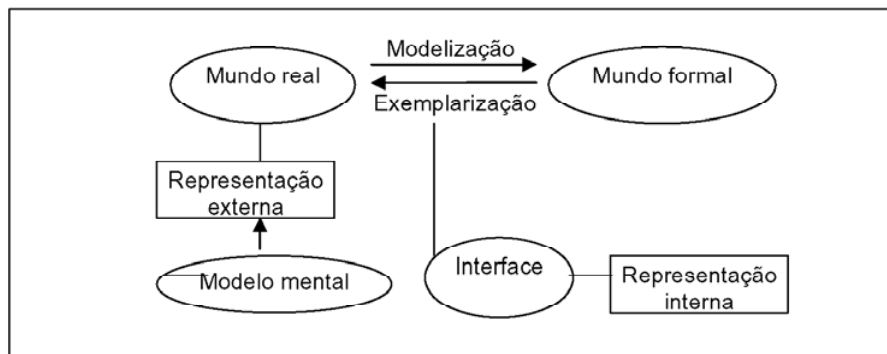


Figura 3.2: Definição de micromundos como metáforas do mundo real (GOMES, 2005).

De acordo com a figura 3.2, é possível entender que as imagens as quais o indivíduo têm de seu mundo real são transformadas em representações transmitidas através da interface de um sistema, convertendo-se assim em um micromundo.

Por volta dos anos 70, os micromundos e programas do tipo tutores, já começavam a ser estudados como programas baseados em modelos de aprendizado do aluno, combinado aos modelos de aprendizado do professor e um modelo de educação que desse embasamento para seu conteúdo. Almeida (2001) cita como exemplos o *Scholar*, o *Sophie* e *West*.

Contudo, os micromundos continuaram sendo bastante estudados por Papert (1991) uma vez que mudavam o paradigma dos softwares educativos, além de estarem deixando sua condição de ensino programado e passando a assumir um conceito que estes estudiosos chamavam de “Construcionismo” baseado em teorias construtivistas.

No início dos anos 80, os programas foram se aperfeiçoando graficamente e suas interfaces ficaram mais atrativas, foram criados exemplos como o *Zoombinis*, *Squeakland* e *Croquet*. Estes programas ultrapassam a questão do contexto, proposta pelo construtivismo. Isso acontecia porque eles começavam a exigir do aluno criatividade e interação, de maneira que eles precisavam compreender qual tarefa estava sendo proposta na interface e qual a melhor solução para respondê-la, dispondo agora de recursos de áudio, em programas como tutoriais e de exercício-e-prática, conforme ressalta o professor Valente (2005):

“O aparecimento dos microcomputadores, principalmente o Apple, no início dos anos 80 permitiu uma grande disseminação dos microcomputadores nas escolas. Essa conquista incentivou uma enorme produção e diversificação de CAIs, como tutoriais, programas de demonstração, exercício-e-prática, avaliação do aprendizado, jogos educacionais e simulação.”

A década de 90 caracteriza-se um pouco mais pela busca de uma integração entre os softwares educativos e a Internet, onde surgiram discussões sobre educação à distância e ambientes virtuais de ensino como ambientes Virtus e AulaNet. Também neste período, surgiram jogos em rede (via Internet) como *Symcity*, *RPG* e outros, não necessariamente educativos, mas que propiciam um tipo de interação desejada pelos softwares educativos, conforme Guerra (2000).

Desde o período em que começou a haver esta integração entre softwares educativos e Internet, muitos softwares foram se aperfeiçoando e oferecendo recursos e atividades que poderiam ser utilizadas tão logo que o software fosse instalado no computador. Outros recursos podem ser adicionados ao software, caso o usuário queira se conectar a Internet e solicitar esses arquivos no site do fabricante, como no caso do software Mundo da criança (suporte dos testes realizados nesta dissertação).

Começa também a ser colocada em foco a questão de aprendizagem colaborativa apoiada por computadores, conhecida como CSCL, conforme visto mais adiante. Alguns pesquisadores, como Wan (1994) e Campos et al (2002), investigam continuamente como os softwares educativos e os sites de educação podem auxiliar para essa aprendizagem colaborativa e como será possível medir o quanto se aprendeu através da utilização destes sistemas.

Desde as máquinas de ensinar, passando pelo tutores e micromundos, chegando aos softwares integrados com Internet e softwares de aprendizagem colaborativa, a investigação sempre tende a buscar novas aplicações que auxiliem professores e alunos. Atualmente, existem inúmeras pesquisas sobre inteligência artificial que criam novas possibilidades educacionais, como por exemplo, em sistemas, como os robôs Elisa e Pixelbot, que “conversam” com seus usuários. Sendo que esse último, utilizado no ambiente Virtus, é capaz de realizar um “diálogo” com o usuário a fim de ajudá-lo na busca de informações pela internet.

Observando esse rápido panorama histórico, conclui-se que a evolução dos softwares educativos contribui para despertar nos pesquisadores um interesse maior pelo tema educação, aliando-se ao tema tecnologia. Ao longo de várias décadas eles puderam observar que sempre houve e, possivelmente, sempre haverá essa vontade de usar novas ferramentas em sala de aula e, assim, extrair delas algo novo para o contexto de ensino, mesmo que seja superado rapidamente, poderá ter sido útil como experiência vivida por alunos e professores.

3.3 | Características e classificações dos softwares educativos

Estudar softwares educativos traz à tona a necessidade de conceituá-los, como foi feito no início do capítulo, e caracterizá-los, como foi desenvolvido neste tópico. A princípio, é preciso dizer que se aceitou as afirmações de Clunie (2000), quanto ao fator que caracteriza um software educativo como sendo um programa computacional com características pedagógicas, e utilizado como **auxiliar no ensino de seus usuários**.

Sendo assim, acreditou-se que podem existir softwares que não atendam às características sugeridas por estudiosos, ou talvez algumas das listas de características não tenham base teórica suficiente para se “classificar” um software, mas, ainda assim, entendemos que os softwares terão seu teor educativo em acordo, conforme o modo com que forem usados, e as listas de características terão seu valor conforme as necessidades de “parâmetros” de orientação dos professores interessados. De acordo com Sette (2004), almeja-se que um software educativo apresente as seguintes características:

- Explore a criatividade, a iniciativa e a interatividade, propiciando ao aluno a postura ativa;
- Desperte a curiosidade e incentive o trabalho cooperativo e interdisciplinar;
- Estimule ou não a competitividade (de acordo com a linha pedagógica adotada);
- Estimule a reflexão, o raciocínio e a compreensão de conceitos;
- Ressalte a importância do processo em vez do resultado obtido (certo ou errado);
- Encoraje o alcance dos objetivos, evitando-se as tentativas irrefletidas sobre o processo;
- Provoque mudanças desejáveis no processo de ensino-aprendizagem;
- Propicie a construção do conhecimento pelos seus usuários;
- Contemple aspectos de linguagem (faixa etária, gênero, ambiente de uso...);
- Considere aspectos sócio-culturais, éticos, pedagógicos, etc;
- Estimule o aluno a propor e resolver problemas.

Assim como Sette (*op. cit.*) descreveu as características desejáveis, encontra-se uma descrição semelhante em uma revista da área de educação que, embora não seja uma revista científica, foi considerado seu valor uma vez que a mesma remeteu a uma lista de requisitos proposta no ano de 1987. O que despertou a atenção é o fato de que 17 anos antes de Sette (2004), já eram observadas algumas características pertinentes a softwares educativos semelhantes àquelas descritas pela autora e, por isso, julgou-se ser interessante apresentá-las neste estudo.

Segundo a Revista Educação (1987), um software educativo deve atender aos seguintes requisitos:

- Propor e/ou permitir que o usuário complemente e interfira no produto e nas respostas;
- Permitir múltiplos caminhos de pesquisa e de soluções dos problemas;
- Ter coerência com o princípio de coletividade para uma melhor qualidade de vida;
- Facilitar a dimensão ética e pedagógica do processo de aprendizagem;
- Possibilitar o raciocínio e a reflexão, para a produção de novas e melhores ações;
- Professor e aluno devem poder registrar e refletir sobre o processo e o seu conhecimento;
- Ser instigante, provocando no aluno a busca de novas informações;
- Permitir o desafio possibilitando ao educando buscar, construir e valorizar sua produção.
- Possibilitar a descrição dos procedimentos, de forma clara e objetiva;
- Ter o erro trabalhado e que a partir de um *feedback* o aluno possa aprender por meio dele;
- Desafiar o aprendiz na busca da exploração do conhecimento de forma prazerosa.
- Dar condições para que o estudante prossiga de forma cooperativa.

O que há de comum entre as características vistas por Sette (2004) e as outras vistas pela Revista Educação (1987) é o fato de que, em ambas, existe uma preocupação com a parte pedagógica do software e em nenhum momento são consideradas questões de usabilidade. No estudo de Sette (*op. cit.*) ainda há uma menção aos aspectos tecnológicos, mas não necessariamente a questões de usabilidade, é natural que isso aconteça, visto que, sendo ela uma professora com formação na área de informática, não é primordial que seus conhecimentos englobem questões particulares à área de usabilidade, portanto, suas colocações são pertinentes e precisariam ser apenas complementadas por outras, oriundas de profissionais de outras áreas.

Esta preocupação com as características pedagógicas é mostrada em alguns estudos da área de ciência da computação, considerando a importância de avaliar corretamente um software educativo. Pesquisadores, como Gladcheff (2001), apresentam suas considerações sobre os aspectos educativos característicos de um software dessa natureza, antes de se colocar quanto aos tecnológicos, como tópicos listados a seguir:

- Aspectos pedagógicos: programas de ensino, objetivos, formas de avaliação, etc
- Aspectos cognitivos: forma de aquisição e memorização do conhecimento
- Aspectos psicopedagógicos: motivação, individualização da aprendizagem, etc
- Aspectos lúdicos: referentes ao caráter de jogos, brinquedos e divertimentos.
- Aspectos socioculturais: oportunidade de uso do computador, intercâmbio cultural, questões associadas à cultura...

Existe ainda um adendo que deve ser feito quanto ao uso de animações, sons e vídeos em interfaces educativas, pois, embora não sejam características determinantes de um software educativo, deveriam ser citadas como uma forma de ratificar que estes recursos podem ser parte das características de softwares que estão sendo utilizados em contexto educativo.

Para esclarecer a afirmação contida no parágrafo anterior, pode-se tomar como exemplo o uso do software Microsoft Power Point para criação de uma apresentação de slides, com sons e animações, sobre determinado conteúdo. Mesmo sabendo que este software não é, exatamente, um software educativo, uma vez que se supõe que este esteja sendo utilizado para fins educativos, o professor e seus alunos poderiam dispor de seus recursos de sons e animações para identificar novas possibilidades de apresentação de um tema.

No exemplo citado, entendeu-se que os alunos estariam lidando com características e recursos tecnológicos (como aqueles já mencionados). Entretanto, o professor e seus alunos, provavelmente, estariam mais ocupados com outras características do software, características que facilitassem o aprendizado de ambos e que pudessem suprir a necessidade de apresentação de um tema, pois, deste modo, conseguiriam priorizar as informações aprendidas e a realização de suas tarefas.

Muitas das “tecnologias” (sons e animações) encontradas em softwares educativos, assim como as listas de características encontradas para tais softwares, têm sido estudadas sob uma ótica mais empírica. Logo, torna-se relevante mencionar que os itens, os quais aparecem nas listas apresentadas neste tópico, não devem ser vistos como “características imprescindíveis”, pois em muitos casos, os softwares podem atender aos seus objetivos pedagógicos, mesmo sem dispor destes recursos e características.

Foi considerado que as características citadas por Sette, pela Revista educação e por Gladcheff (op. cit.), apresentam informações que influenciam nas escolhas dos usuários, portanto, embora não sejam decisivas para caracterização de um software, também não podem ser consideradas irrelevantes. Sabe-se que, para muitos professores, elas servem como “guias de orientação”, assim como os itens listados e os recursos tecnológicos nem sempre são vistos por eles apenas como elementos meramente “decorativos” e/ou “atrativos” do software.

No que se refere à classificação dos softwares educativos, de acordo com Escobar (1989), eles podem ser de forma aberta (construtivista / heurística) ou de forma fechada (instrucionista / algorítmica). Entretanto, Campos (1994) explica que podem ser categorizados quanto à sua base pedagógica, podendo ser construtivistas ou behavioristas.

De acordo com Valente (1998), para o professor, avaliar softwares educativos pode ser um exercício interessante, esta ação pode auxiliar na compreensão do papel do computador e na influência que este exerce no processo de construção do conhecimento.

Considerou-se então que, saber identificar e classificar os softwares, também se mostra como fator decisivo no momento de escolher qual o melhor software para ser utilizado em sala de aula, assim como, no momento de decidir qual a melhor forma de avaliar este software e o aprendizado proporcionado por ele. Valente (1998) apresenta a seguinte classificação (tabela 3 a seguir):

Tabela 3.1: Classificação de Valente (1998) para softwares educativos.

Programação:	Oferece ao aluno a possibilidade de solucionar problemas através da construção de estratégias próprias de aprendizagem. O software de programação permite que o professor tome conhecimento do processo que o aluno desenvolveu para resolver o problema apresentado, assim como possibilita ao próprio aluno a reflexão sobre suas ações e idéias, tornando-o consciente dos conhecimentos que possui.
Tutorial:	Disponibiliza informações previamente definidas e limitadas pelo próprio software, oferecendo ao aluno apenas a possibilidade de escolha ou de definição da sequência pedagógica na qual estas informações serão apresentadas. O tutorial não fornece recursos para acompanhamento do processamento da informação pelo aluno, destinando ao professor a tarefa de verificar e avaliar se houve aprendizagem.
Jogos:	Podem ter características de um tutorial ou apresentarem algumas possibilidades de interferência através da descrição e complementação de situações previamente definidas pelo programa. São raros os jogos que permitem a alteração ou a criação de regras, contudo, são muitos os que envolvem o aprendiz num espírito de competição. Na utilização deste tipo de software, faz-se necessária a presença do professor para discutir e propiciar condições para que o aluno compreenda o que está fazendo.
Processador de texto:	A interação do aluno com o computador é mediada pelo idioma natural e pelos comandos que ele fornece ao programa para a formatação do texto. Os recursos oferecidos por este software limitam-se a estilos de fonte e formatação, não apresentando características que possibilitam o aluno refletir sobre o processo. Contudo, o fato do texto não precisar ser re-escrito para correções, pode estabelecer uma relação diferente entre o aluno e o texto. Como no caso dos tutoriais, no processador de texto o professor também deve criar meios de verificação e avaliação próprios, visto que o programa não oferece um feedback fiel do processo.
Simulador:	Possibilita a visualização virtual de situações reais, onde os alunos podem participar através da realização de testes e experiências. Pode ser classificado como aberto ou fechado. Para que ocorra a aprendizagem, é necessário que sejam criadas condições complementares ao uso do software, permitindo que o aprendiz desenvolva um processo de reflexão e atue com envolvimento na experiência ou fenômeno simulado.
Multimídia e internet:	Apesar da ação do aprendiz resumir-se a escolha das opções que são oferecidas pelo software, ele tem a oportunidade de refletir sobre as informações que são apresentadas. São programas que auxiliam o aluno na busca de informação, mas que não garantem a construção do conhecimento. Através deste tipo de software, o aluno também dispõe de recursos que proporcionam a criação e elaboração de mídias (construção de páginas na Internet, apresentação personalizada de trabalhos, desenvolvimento de animações, etc.), fato que o torna autor de todo um processo.

A classificação de Valente (op. cit.) é utilizada em métodos de avaliação de usabilidade como o CASE (Comunidade para Avaliação de Software Educativo), conforme veremos no capítulo 5. Contudo, salientamos que em classificações anteriores, como a de Escobar (1989), são mencionados os chamados sistemas abertos, onde já se identificava a possibilidade do educando pode utilizar mais livremente as opções que o software oferece, como acontece nos sistemas multimídia-internet, citados por Valente (op. cit.).

A seguir, serão apresentadas outras classificações de softwares educativos que poderão expandir as possibilidades de “enquadramento” dos mesmos e esclarecer se eles podem ser classificados em mais de um item, conforme observa-se a seguir, na classificação por novéis de aprendizagem, sugerida por Vieira (1998):

- **Seqüencial** - A preocupação é só transferir a informação: o objetivo do ensino é apresentar o conteúdo para o aprendiz e ele por sua vez deverá memorizá-la e repeti-la quando for solicitado. Esse nível de aprendizado leva a um aprendiz passivo.
- **Relacional** - Objetiva a aquisição de determinadas habilidades, permitindo que o aprendiz faça relações com outros fatos ou outras fontes de informação. A ênfase é dada à busca por um certo nível de aprendizagem e esse contexto leva a um “aprendiz isolado”.
- **Criativo** - Associado à criação de novos esquemas mentais possibilita a interação entre pessoas e tecnologias compartilhando objetivos comuns. Esse nível de aprendizado leva a um aprendiz participativo.

É possível perceber que a classificação apresentada por Vieira (op. cit) mostra uma nova forma de relacionar essas três categorias com as bases pedagógicas do software, tendo em vista que classificar um software como “seqüencial”, seria muito semelhante a classificá-lo como “fechado” ou “instrucionista”. Do mesmo modo, seria possível interpretar que ao classificar um software como “criativo” e, conseqüentemente, estaríamos caminhando para classificá-lo como “construtivista”. Acredita-se que nenhuma destas situações estaria errada, seriam apenas olhares sobre o mesmo material, ou somente modelos diferentes de categorização. Na classificação apresentada por Campos (2005), foi encontrada a seguinte lista, conforme tabela 4:

Tabela 3.2: Classificação de Campos (2005) para softwares educativos.

Softwares aplicativos	Usados para tratamento de texto, planilhas, base de dados, software gráfico, software para cálculo etc. Podem ser considerados como abertos do ponto de vista da informática.
Micro-mundos	São sistemas informatizados abertos onde o aluno deve explorar um domínio com um mínimo de ajuda do sistema. O aluno aprende a aprender utilizando o ambiente para espelhar seus conhecimentos e construir novos objetos.
Coursewares	São produtos de software educacional clássicos que, a partir de uma situação interativa entre o aluno e um problema, leva o aluno a resolvê-la. A gama de atividades possíveis é vasta, mas cada sequência é fechada por respostas interpretáveis pelo programa.
Tutores inteligentes	São como alguns <i>coursewares</i> onde à característica de resolução de problemas acrescentou-se o componente tutorial onde são representados o modelo do aluno, o conhecimento do professor e a especialização do conhecimento a ser ensinado.
Hipertexto	É comumente definido como uma forma não linear de armazenamento e recuperação de informações. Isto significa que a informação pode ser examinada em qualquer ordem, através da seleção de tópicos de interesse. Desta forma, um hipertexto tem como principal característica a capacidade de interligar pedaços de textos ou outros tipos de informação entre si através do uso de palavras-chave.
Ambientes inteligentes de aprendizagem	Permitem ultrapassar a oposição simplista entre os defensores da aprendizagem por indução (atividades exploratórias do sujeito) e os defensores dos tutoriais. Estes sistemas devem ser capazes de favorecer a aquisição de conceitos e procedimentos associados a um domínio do conhecimento, permitindo ao aluno transformar suas experiências em conhecimento organizado.

Diante destas três classificações, cabe-se tentar classificar o software Mundo da criança, que já foi apresentado de forma sucinta no capítulo 1. Logo, para efeitos de contextualização, este software pode se “enquadrar” na categoria “multimídia e Internet”, proposta por Valente, a medida em que ele apresenta conteúdos que despertam no aluno a necessidade de criar seus próprios desenhos ou montar suas próprias estorinhas, onde o software auxilia apenas oferecendo botões de controle para girar, recortar ou imprimir o que foi criado. Ainda seria possível situar o Mundo da Criança como um software criativo, seguindo a classificação de Vieira (1998), pois sua interface faz com que o aluno seja participativo, à medida que as personagens oferecem comandos de voz e a criança que não sabe ler ainda pode se guiar por eles e executar ações para que a tarefa seja completada, do mesmo modo que o produto oferece a possibilidade de manipular joguinhos e imprimir resultados de tarefas realizadas.

Para finalizar, o software também poderia ser classificado como micromundo, pela classificação de Campos (1994). Esta classificação advém por considerar que o software apresenta como “seções” os diversos “mundos” e “planetas” que são, na verdade, metáforas de situações que a criança compreende, naturalmente. Toma-se como exemplo o “mundo dos bichos”, onde a seção traz diversas atividades sobre reino animal e dessa forma a criança entende que aqueles elementos possuem relação com algo que ela já vivenciou ou já viu ou, pelo menos, conhece de alguma forma o que está sendo apresentado.

Conhecer tais classificações permite adquirir fundamentos para avaliar melhor o software utilizado, seja como usuários diretos ou indiretos. É importante entender que os usuários, de modo geral, podem não se interessar por estas classificações, mas, talvez elas os ajudassem a escolher um software que atendesse uma necessidade de aprender história, por exemplo. Já no caso de usuários indiretos, como profissionais que desenvolvem o sistema (designers e outros), classificações como estas se tornam interessantes para entender a aplicação dos elementos e princípios de design.

Para concluir, poderia ser lançada uma reflexão sobre “qual seria a real utilidade destas classificações?” Não queremos diminuir a importância de tais classificações, mas acreditamos que o conceito de software educativo engloba, por exemplo, sistemas que funcionam como tutoriais, onde também estão presentes as simulações de uma sala de aula, contendo elementos gráficos que representam a relação entre educador e educando no processo de aquisição de conhecimento. Exemplos desse tipo e com estas características levam a acreditar que existam softwares que se “encaixam” em mais de um item dessas classificações, assim como outros podem não se encaixar em nenhum item, logo, a utilidade delas parece estar ligada diretamente à necessidade de classificação de quem as utiliza.

3.4 | Considerações sobre usuários de softwares educativos

A importância de se identificar quem são os usuários de softwares educativos está relacionada ao fato de que o computador utilizado como um recurso pedagógico adquire a importante função de auxiliar no processo de ensino e aprendizagem. Segundo Ersching (2002), o computador oferece aos usuários, alunos e professores, a possibilidade de transformar sua expressão criativa através de novas formas de (re) construção do conhecimento, como, por exemplo, através do uso de softwares.

Essa preocupação a respeito da inserção de computadores e softwares educativos, bem como, a respeito dos atores envolvidos nesse processo tem sido foco de projetos educacionais nacionais e também é encontrada nos estudos realizados por Vieira (1998), Fagundes (1998) e Valente (1998). Esses autores afirmam, inclusive, que uma das tarefas do Núcleo de Tecnologia Educacional do PROINFO - MEC é avaliar criteriosamente os softwares de uso educativo, pois são eles que determinam as possibilidades de uso dos computadores na educação, através da utilização feita pelos professores e alunos das escolas.

Para desenvolver um trabalho de avaliação de software educativo, que envolve critérios ergonômicos e pedagógicos, torna-se necessária a reflexão sobre o contexto em que essas ferramentas são utilizadas e quais os atores envolvidos, isto é, quem são os usuários “potenciais” de um software educativo. Alguns autores, como Druin (2001) e Gagné et al (1998), demonstram essa preocupação a respeito dos perfis de usuários que interagem com ambientes de aprendizagem, como os softwares educativos, seja na utilização, na mediação ou no desenvolvimento.

Para os fins desta pesquisa, assegurou-se que, além das crianças, as quais são vistas como público alvo por parte dos fabricantes destes softwares, pais e professores atuam indiretamente como mediadores entre a criança e o software, assim como existem os desenvolvedores (designers e outros profissionais) que atuam criando e testando estas ferramentas, assumindo posturas de usuários diretos e indiretos do sistema, dependendo da situação.

A importância de identificar como usuários de softwares esses três atores – criança – professor – designer – está no fato de que para avaliar um software educativo, é preciso conhecer melhor quem o utiliza, para que possam ser projetadas interfaces que atendam às necessidades desses usuários. Sobre esta questão de avaliação de software educativo, os autores Oliveira, Costa & Moreira (2001) afirmam que avaliar significa estabelecer um processo que tem início antes mesmo da sua criação, através da escolha da equipe que produzirá o software, em seguida, com o produto criado, os usuários julgarão se é viável a utilização deste na rede escolar.

Quanto ao professor como usuário de software educativo, não há dúvida de que as intenções do Ministério da Educação em equipar as escolas com computadores e softwares educativos despertou atenção de professores desde a rede particular à rede pública e do ensino infantil ao ensino superior (Cox, 2003). É indiscutível o poder de “encantamento” das máquinas sobre alunos e professores. Mas, independente do “deslumbramento” (ou da “rejeição”) durante a utilização dessa ferramenta, os professores devem estar atentos no sentido de garantir que o software educativo seja usado de uma forma responsável e com potencialidades pedagógicas verdadeiras, não sendo utilizado apenas como um brinquedo que contém joguinhos divertidos e agradáveis.

Uma idéia muito defendida, em relação ao papel do software educativo junto a professores e alunos, é que o computador facilita o processo ensino-aprendizagem. Essa idéia está ligada à generalização do fato de que o computador entrou na vida do homem para facilitar, como, por exemplo, no desenvolvimento de trabalhos escolares. Uma outra idéia é assumir o uso do computador como um dispositivo programado para resolver questionamentos acerca de um conteúdo debatido pelo professor em sala de aula. Nesse sentido, ele pode até complicar a vida do aprendiz, pois este teria que descrever para o computador, através de uma linguagem de programação, todos os passos para a resolução de um problema, e, se os resultados não corresponderem ao desejado, o aluno tem que adquirir informações necessárias e incorporá-las ao programa ao repetir suas operações.

O que está sendo exposto é que a forma de utilização de computadores e softwares educativos depende da interação professor-aluno-software-conteúdo, conforme já foi dito por Valente (1998). Portanto, o uso de softwares educativos na educação tem como objetivo promover a aprendizagem dos alunos e ajudar na construção do processo de ensino do professor e propiciar uma construção de conhecimento baseada na integração de que fala Valente (*op. cit.*). O autor enfatiza que o professor por si só não pode “fazer milagres”, é preciso que lhe seja dada uma capacitação para que ele possa dispor da ferramenta conforme suas necessidades, desenvolvendo habilidades importantes para que

faça do software educativo um artefato de apoio ao processo de ensino e não o veja como um 'inimigo' que poderá tomar sua função na escola.

No que se refere às crianças como usuários de softwares educativos, a integração delas com o professor e com os softwares educativos (tecnologia), pode ser vista no esquema gráfico identificado como "integração professor-aluno-tecnologia-conteúdo" que se encontra no livro sobre comunidades de aprendizagem, de Pallof e Pratt (2002). Mesmo sabendo que a nomenclatura usada pelos autores é "rede de aprendizagem", certamente, porque o livro trata mais especificamente de comunidades virtuais de aprendizagem, achou-se por bem, apresentar o esquema por acreditar que ele tem relação com as idéias que estão sendo discutidas sobre os usuários de softwares educativos, conforme aparece na figura 3.3 a seguir:

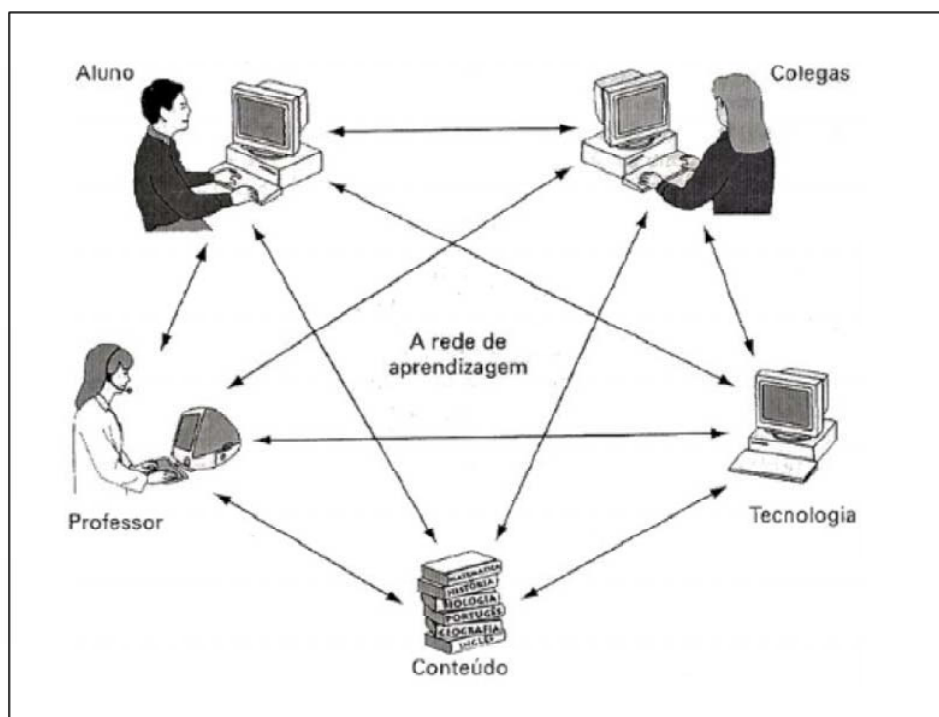


Figura 3.3: Integração professor-aluno-tecnologia-conteúdo. Pallof e Pratt (2002).

A figura 14 mostra a possível rede que se forma entre professores, alunos computadores conteúdo e tecnologia, salientando que no presente estudo a tecnologia usada são os softwares educativos. Essas redes (não, necessariamente, redes de computadores) parecem ser formações comuns entre indivíduos e ferramentas. Entretanto, mesmo nos dias de hoje, quando se fala no uso de computadores e aplicação de sua utilização junto às crianças em fase de alfabetização, pode-se nos deparar com crenças, mitos e resistências, não das crianças, mas, sim, de seus responsáveis.

De acordo com Napolitano e Batista (2005), através da junção dos conteúdos de aprendizagem presentes em softwares educativos, com os as atividades baseadas em teorias da educação, elaboram-se estratégias para que os atores, principalmente, as crianças, possam trabalhar com o software quebrando mitos e curando medos que existam, mesmo que os atores não dominem essa tecnologia educacional.

De acordo com Weiss e Cruz (2001), é possível sugerir softwares mais adequados de acordo com a idade das crianças. As autoras não esperam estabelecer indicações como se fossem “regras” ou “leis” as quais regem que software é indicado para que idade. O que elas sugerem é uma tabela comparativa, onde são apresentadas as faixas etárias, suas características comportamentais e uma sugestão sobre que tipo de software poderia agradar as crianças nessas condições. A tabela 3.3 a seguir foi transcrita dos textos de Weiss e Cruz (*op. cit.*) e apresenta sugestões de uso de sistemas computadorizados de acordo com a idade das crianças que irão utilizá-los.

Tabela 3.3: Sugestões de uso de sistemas de acordo com a idade das crianças. Weiss e Cruz (2001)

Idade	Linguagem e lógica	Softwares	socialização
6 anos	Gosta de histórias com várias cenas e personagens; Seu pensamento ainda se mistura com realidade e fantasia; Interessa-se por linguagem escrita.	Editores de desenho (que permitam também o desenho livre); Elaboração de livros de histórias e de histórias em quadrinhos;	Começa a compreender regras e ajuda a elaborá-las; É possível iniciar as atividades em grupo, com todos nos computadores.
7 a 12 anos	O pensamento lógico assume o lugar do egocentrismo intelectual, a criança adota uma postura crítica; Realiza operações mentais: as ações podem ser intelectualizadas; Organiza suas idéias em séries e classe lógicas.	Jogos educativos (com letras, por exemplo); Editores de textos.	Interage efetivamente com crianças e adultos

No que se refere à identificação de professores como usuários de softwares educativos, segundo Lyra et al (2003), afirmam que estas ferramentas existem há mais de 20 anos, mas para grande maioria dos profissionais de ensino o uso dessas tecnologias implica numa necessidade de uma longa formação. Apesar de existirem a tanto tempo, os softwares educativos ainda não foram utilizados em todo seu potencial na prática docente. A adoção sistemática de softwares educativos nas escolas é mínima. Isso se deve, em parte, à carência de sistemas de qualidade e falta de informações sobre aspectos técnicos e pedagógicos de bons sistemas existentes. Os professores não dispõem de uma grande quantidade de softwares de qualidade.

Além disso, os autores acrescentam que as informações disponíveis sobre os softwares são limitadas, sendo os mesmos avaliados inadequadamente. Sendo assim, observa-se que poucas são as divulgações de relatos de experiências vividas por profissionais da área. Mesmo após encontrar um software que julga adequado, o professor pode ter dificuldades para discernir se o mesmo é de boa qualidade para o ensino de determinado conteúdo, passando ainda pela situação de contar com pouca orientação para a escolha de softwares destinados ao uso contínuo na sua escola, por exemplo.

Para esclarecer um pouco mais o papel do professor como usuário de software educativo e seu papel junto ao aluno, propõe-se a leitura de uma outra figura proposta por Weiss e Cruz (2001). Tal figura apresenta a forma como participação do professor começa a se transformar em uma função de mediador, onde ele deve ouvir mais do que falar. As autoras explicam que dessa forma o aluno tornar-se-á mais participativo no uso das ferramentas de informática, como os softwares educativos, conforme ilustra a figura 3.4 a seguir:

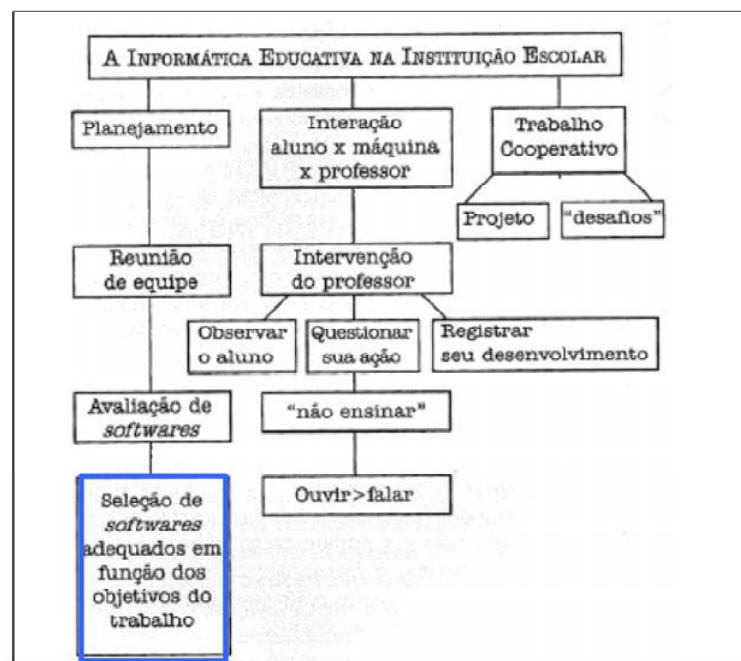


Figura 3.4: Informática educativa na instituição escolar

A figura mostra ainda a questão da seleção de softwares e é nesse momento que entra em cena o professor como "especialista" no domínio de conteúdo. É ele quem vai decidir se o software é adequado para o conteúdo que vai ser discutido com os seus alunos. Neste momento também, foi colocado em foco o terceiro usuário / ator envolvido no processo: o designer (ou outro desenvolvedor, como um programador de informática), que deveria ter conhecimento das necessidades gerais de um professor em sua prática docente..

No que diz respeito ao designer como ator deste cenário, uma capacidade de análise ergonômico-pedagógica é requerida numa avaliação de software educativo, para que possam ser avaliados critérios de funcionalidade, usabilidade, assim como critérios de aprendizagem e comunicação.

Pesquisadores, como Hanna et al (1997), Gamez (1998), Gladcheff (2001) e Atayde (2002), já demonstram em suas pesquisas que, de alguma maneira, é preciso haver uma aceitação da idéia de que um software deve ser avaliado por todos os perfis de usuários que usam o software educativo.

Neste sentido, o designer entra para o quadro de perfis de usuários educativos, seja como usuário direto (quando testa o sistema) ou como usuário indireto (quando desenvolve o sistema). No capítulo sobre avaliação de usabilidade, os papéis destes atores foram discutidos novamente com foco na avaliação participativa de um software educativo, onde foram vistas metodologias apropriadas para cada um deles individualmente, ou para grupos de perfis, como em métodos participativos.

3.5 | Aprendizagem colaborativa (CSCL) e Trabalho colaborativo (CSCW)

De acordo com Santoro, Borges e Santos (1998), entende-se por Aprendizagem Colaborativa Apoiada por Computador, ou *Compute Supported Collaborative e Learning* (CSCL), uma área de estudo que trata de formas pelas quais a tecnologia pode apoiar os processos de aprendizagem promovidos através de esforços colaborativos entre estudantes trabalhando em uma dada tarefa.

Quanto ao conceito de CSCW como Trabalho cooperativo Apoiado por Computador, ou *Computer Suported Coooperative Work*, Campos et (2003) explicam que esta é uma área de pesquisa que estuda o projeto, adoção e uso de sistemas colaborativos, ou seja, tecnologia computacional para apoio ao trabalho de grupos de indivíduos por meio de aplicações chamadas “groupwares”. O ponto importante para nossa discussão é entender a diferença entre CSCL e CSCW. Segundo **Campos et (op. cit)**, CSCL estuda as interações entre as pessoas num ambiente de aprendizagem, enquanto que CSCW, estuda o produto dessas interações, conforme nos informa a figura 3.5:

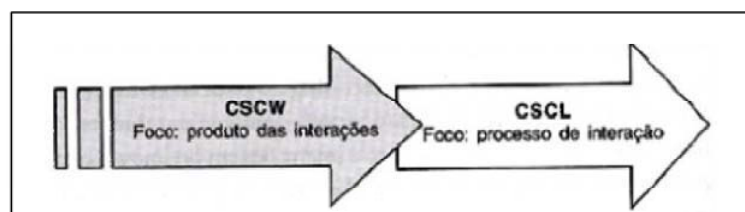


Figura 3.5: Aplicação de CSCW em CSCL.

A relevância de estudar estes conceitos junto ao tema “software educativo” é que as pesquisas sobre ambientes de aprendizagem e, principalmente, sobre a avaliação destes ambientes têm demonstrado que cada vez mais as pessoas estão fazendo a opção de formar grupos para aprender algo novo, ou simplesmente para trabalharem juntas avaliando um mesmo projeto. Muitas vezes, estas situações de aprendizagem e avaliação ocorrem por meio de softwares educativos, ou ferramentas semelhantes a eles. Estudiosos, como Cockburn e Greenberg (1993), têm dedicado atenção especial aos fatores que promovem a eficiência de ambientes CSCL e CSCW com ênfase no estudo sobre os usuários que utilizam estes ambientes e com que meios de comunicação.

O fato de ter um grupo que opta por se reunir para aprender em conjunto se revela como uma forma mais de uso das tecnologias educacionais. No que se refere a softwares educativos, este tipo de ação tem mostrado que existe aplicabilidade direta destes sistemas de aprendizagem, para integração de pessoas em busca de reflexões sobre um mesmo conteúdo. Essa aplicabilidade se mostra através da criação de sites como <www.euro-cscl.org> (2005) que é especializado em oferecer opções de softwares para esta finalidade, como é o caso do *Synergeia*, cujo propósito é facilitar a integração de pessoas com os mesmos objetivos colaborativos.

Em se tratando de CSCL, Campos et al (2003) afirmam que estes ambientes valorizam o aprendizado colaborativo a medida em que criam um espaço para trabalho em conjunto, mas que uma das dificuldades é a implementação de posições teóricas no contexto escolar. A autora salienta ainda, que os alunos assumem uma responsabilidade sobre a aprendizagem, desenvolvendo habilidades cognitivas para monitorar e dirigir sua própria aprendizagem e seu desempenho, como, por exemplo, em softwares educativos que apresentam ambientes colaborativos de aprendizagem.

O software utilizado nesta dissertação, Mundo da criança, não é um sistema de aprendizagem colaborativo, no entanto, ele agrega atividades que exigem do aluno uma postura participativa, de modo que ele precise buscar determinados conteúdos, clicando em opções que irão conectá-lo a Internet para ter acesso a tais conteúdos. Esta característica é muito comum em ambientes de aprendizagem colaborativa, onde o um aluno precisa ter iniciativa para ir buscar mais sobre determinado tema. A diferença do Mundo da criança para softwares específicos de aprendizagem colaborativa, é que nestes últimos, o aluno pode contar com a ajuda de outros indivíduos, presentes no mesmo ambiente virtual. Já num software não-colaborativo, o educando precisará buscar num ambiente físico (como a sala de aula) estes indivíduos que possam ajudá-lo em suas descobertas.

Pallof e Pratt (2002) apresentam uma representação gráfica (figura 3.6) que ajuda a entender como ocorrem essas interações entre os indivíduos que utilizam um sistema CSCL. Nesta representação, inferimos que o conjunto formado por quatro “peças” principais (trabalho em equipe + aprendizagem colaborativa + resultados planejados + criação de conhecimento) é o que forma um ambiente CSCL.

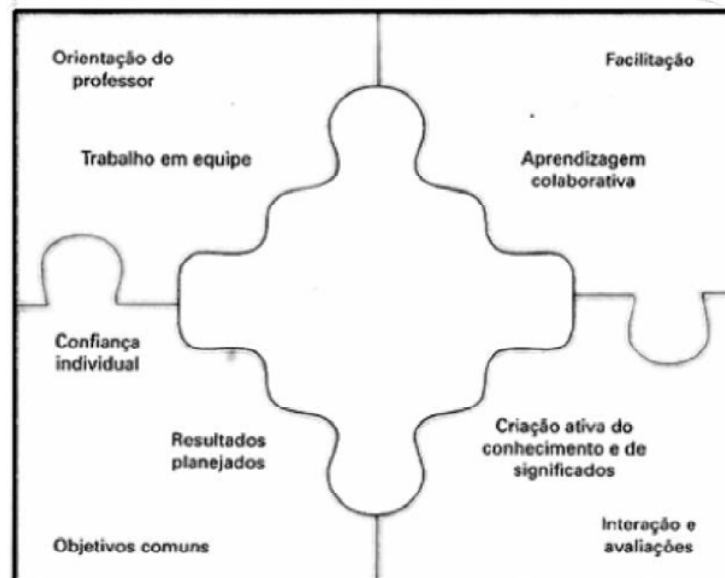


Figura 3.6: Interações entre os indivíduos que utilizam um sistema CSCL

No que se refere a CSCW, Campos et al (2003), apresentam a idéia de *groupwares*, não somente como um grupo de pessoas integradas num projeto, mas sim, como uma forma de trabalho que envolve tecnologias de software e hardware. As autoras defendem, principalmente, uma postura multidisciplinar por parte dos indivíduos, que precisam conhecer conceitos de interface-humano-computador junto com outros conceitos relacionados às ciências humanas, como educação e psicologia, para poder fazer uso de um ambiente CSCW.

Para esta pesquisa, a relevância de estudar CSCW está no fato de que algumas ferramentas de avaliação de softwares educativos, como o ambiente CASE, oferecem esse suporte para que vários usuários trabalhem juntos avaliando um mesmo software. Assim como o CASE, existem outras ferramentas / técnicas, e até métodos. O MAQSEI (2003), por exemplo, defende uma avaliação por vários perfis de usuários, seja de forma síncrona ou assíncrona. Algumas destas metodologias têm relação com métodos de design participativo e design centrado no usuário, entretanto, não propõem reunir, no mesmo ambiente, usuários com perfis e idades bastante diferentes.

A partir destas definições de CSCL e CSCW, conclui-se que existem softwares educativos que se apresentam com a função de reunir estudantes para interagir (trabalhar) numa mesma tarefa a fim de desenvolver conhecimentos originados dessas interações (CSCL). Na mesma dimensão dos CSCL, existem os ambientes de trabalho colaborativo (CSCW), que também podem ser softwares desenvolvidos para esta finalidade, onde as pessoas se reúnem para avaliar ou investigar essa aprendizagem, ou o produto de uma aprendizagem.

Existem, inclusive, pesquisas que envolvem CSCL e CSCW ao mesmo tempo, como as que são publicadas pelo grupo Palo Alto, cujos estudos evidenciam que empresas de tecnologia também estão investigando o tema com um enfoque comercial e não somente científico. Como exemplo, podemos citar as pesquisas realizadas por Bly (1988) que busca entender as relações entre pessoas que trabalham no mesmo ambiente colaborativo em três situações diferentes: (i) estando presentes face a face, (ii) estando separadas geograficamente somente interligadas através de áudio e vídeo e (iii) estando separadas geograficamente e interligadas apenas através de telefone.

Tendo em vista o avanço das tecnologias educacionais presentes em softwares educativos do tipo CSCL, e em ambientes de trabalho colaborativo do tipo CSCW, o design e a usabilidade presentes em suas interfaces representam um fator importante para a comunicação e realização das tarefas executadas entre os indivíduos dos *groupwares*. É preciso continuar investigando quais são as influências recebidas por um aluno que está fazendo parte de um ambiente convencional de educação, com sala de aula e professor (presentes fisicamente), em relação a um aluno em ambiente CSCL, onde estão presentes outros indivíduos que podem ter características físicas e culturais diferentes da sua e que, possivelmente, só serão conhecidas através de contato “virtual”.

Segundo a pesquisadora Paas (1999), a diferença entre o modelo convencional e um modelo (virtual) colaborativo com ênfase num design educacional está representado na figura 3.7. Esta figura foi escolhida para representar o modo de interpretação do que seria a diferença entre um modelo convencional de educação e um modelo que utiliza softwares educativos, do tipo CSCL, conforme vemos na figura abaixo:

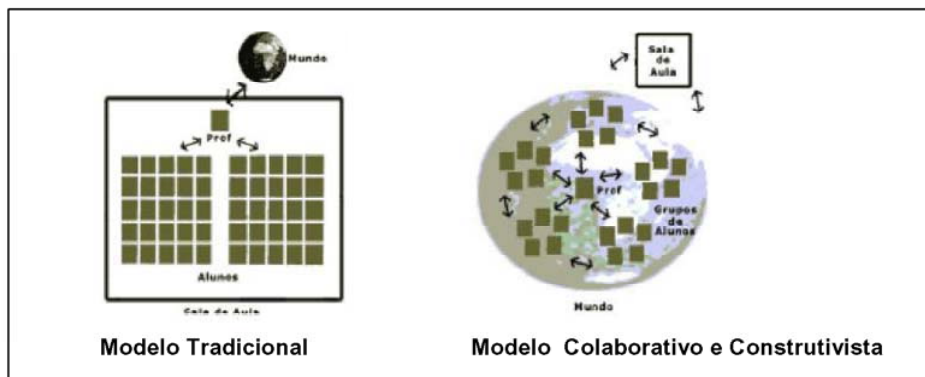


Figura 3.7: Diferença entre o modelo convencional e um modelo colaborativo e construtivista.

O desafio que fica para pesquisadores da área de design, assim como pesquisadores de outras áreas, é sobre qual a melhor forma de estudar a aprendizagem ocorrida nesses ambientes, pois mesmo sabendo da existência de métodos participativos (conforme já mencionamos) acreditamos que é tarefa complexa lidar com softwares de características multidisciplinares, como é o caso dos softwares educativos, principalmente, aqueles que são também ambientes CSCL. As autoras Campos et al (2003) apresenta uma tabela contendo as possíveis diferenças nos enfoques de aprendizagem e

no modo como deveriam ser avaliados, contudo, entendemos que esta tabela está tendo seu direcionamento voltado mais para área pedagógica, conforme mostra a tabela 6 a seguir:

Tabela 3.4: Enfoques de aprendizagem e forma de avaliação.

Enfoque:	Quem avalia:
Tradicional	Utilização de verificações e prazo mais longo; reforço negativo (punições, notas baixas) e reforço positivo (classificação).
Tecnicista	Avaliação de comportamento observáveis e mensuráveis; controle de comportamento face a objetos preestabelecidos.
Libertadora	A verificação direta da aprendizagem é desnecessária; avaliação da prática vivenciada entre educador / educando; auto –avaliação em termos de compromisso assumido com a prática social.
Progressistas	A avaliação é realizada a qualquer momento, pois sua preocupação é diagnosticar falas; observação do desempenho; valorização de outros instrumentos que não a “prova”.

Para concluir, deixa-se o questionamento sobre como deveriam ser avaliadas as questões de usabilidade em ambientes CSCL e como seriam feitas essas avaliações em ambientes CSCW, quais as formas de comunicação entre eles, ou ainda, como deveriam ser projetadas suas interfaces de comunicação. Acredita-se que estas podem ser algumas das lacunas existentes na relação CSCL x CSCW. Embora não seja o objetivo deste trabalho responder a este questionamento, sabe-se que entender esses *groupwares* ajuda na decisão de selecionar melhor que métodos seriam mais aconselháveis para avaliação de softwares educativos em testes com diferentes perfis de usuários.

3.6 | Elementos e princípios de design em softwares educativos.

De modo geral, os elementos e princípios de design que estão presentes em interfaces digitais de sistemas computadorizados podem deter características visuais semelhantes àqueles encontrados em sistemas de informações cotidianos, como se fossem metáforas de outros sistemas de informações que usamos em nossa rotina.

Segundo Rocha e Baranauskas (2003), mesmo não funcionando sempre, o seu uso crescente, especialmente em interfaces gráficas, favoreceu (ou “forçou”) a expansão do domínio da área de design de interfaces, com contribuições mais que relevantes de outras especialidades como design gráfico, industrial, lingüística, psicologia, educação e, sobretudo, da área de ergonomia. Nesta última, encontra-se diversos estudos realizados por Moraes (1998); Freire e Soares (2001); Soares (2003); e Freire, Soares e Padovani (2005), onde as questões de ergonomia em interfaces digitais aparecem com destaque sendo tratadas sob os enfoques das áreas de design e ergonomia.

Os micromundos (metáforas do mundo real) encontrados nas interfaces de softwares educativos são construídos a partir de elementos e princípios de design, tais como contraste, legibilidade, etc. Estes princípios estão baseados em estudos que exploram a melhor forma de apresentar um conteúdo ao usuário, no entanto, este campo de pesquisas envolve muitos estudos correlatos conforme explicam as autoras Rocha e Baranauskas (*op. cit.*). Estas autoras mencionam que a contribuição maior de pesquisas sobre metáforas e micromundos, elementos e princípios de design é ter feito com que o estudo de interfaces se tornasse uma preocupação “inter-multi-transdisciplinar”.

Para fins da dissertação ora apresentada, esses elementos e princípios foram estudados sob o ponto de vista de design. Entretanto, cumpre salientar que, algumas das abordagens do presente estudo, a respeito de design de interfaces, incluem também princípios de ergonomia e usabilidade, considerando que o tema de nosso trabalho são os métodos de avaliação de usabilidade, direcionados para softwares educativos e, deste modo, não há como “separar” as relações existentes entre as disciplinas citadas.

Segundo autores como Dondis (1997), Gomes Filho (2000), Lauer e Pentak (1999), existem elementos e princípios básicos que formam toda a base da comunicação visual, seja através de impressos ou através de interfaces digitais. Estes elementos são as cores, texturas, formas, tipografias e imagens, assim como os princípios básicos são formados por contraste, legibilidade, harmonia, equilíbrio, proporção e simetria.

Embora, saiba-se que essas nomenclaturas também mudam conforme as áreas de pesquisa, de modo geral, acredita-se que estes são os elementos e princípios encontrados também em interfaces de softwares educativos. Para entender melhor como esses elementos e princípios se relacionam, foi proposta a apresentação da figura 3.8, onde constam no primeiro quadro a lista de elementos básicos de design e no segundo quadro, a lista de princípios que indicam o modo como os elementos devem ser aplicados em projetos gráficos ou digitais.

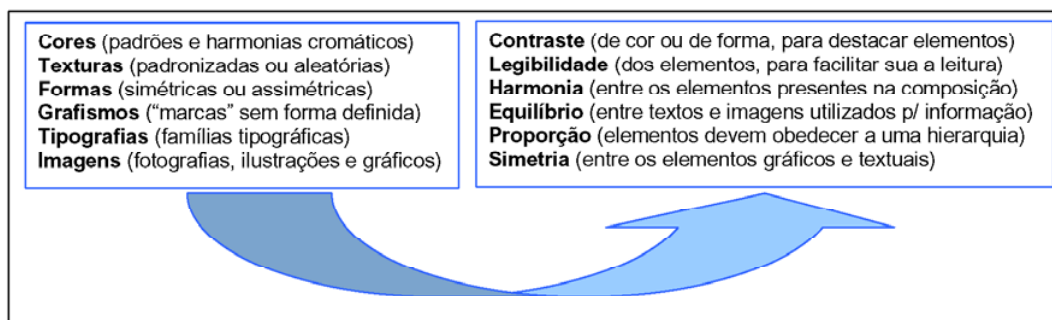


Figura 3.8: Elementos de design aplicados através de princípios básicos de design.

Segundo Milheim e Lavix, (1992) apud Fernandes (1996) o *design* de interfaces de sistemas educativos pode ajudar a melhorar a aprendizagem assim como a capacidade de um aluno em navegar através de um programa e interagir com ele. Entende-se que esta relação ocorre no momento em que o usuário visualiza a interface e procura elementos familiares para que possa identificar que tarefa ele poderá realizar naquele software educativo.

Fernandes e Machado (1996) afirmam ainda que se pode dizer, genericamente, que o *design* de interfaces em softwares educativos diz respeito à organização dos elementos do *design* numa interface. No entanto, não abundam na literatura definições precisas do que é o *design* de interfaces num contexto educativo.

Acredita-se que esta colocação se deve ao fato de que o design de interfaces em softwares educativos possui características peculiares a esta classe de software, onde é preciso concentrar na mesma interface elemento de design capazes de comunicar conceitos relativos à aprendizagem e à navegação, como, por exemplo, apresentando ícones com significados como siga adiante quando acertar a resposta de uma determinada pergunta. Contudo, se refere aos princípios de design, existem ainda outras considerações a serem feitas quando se considera estes princípios, do ponto de vista da área de ergonomia.

Segundo Soares (1993) a ergonomia oferece ao pesquisador parâmetros para investigação de aspectos físicos, cognitivos e organizacionais, principalmente, em ambientes educativos, onde estes aspectos são influenciados diretamente pelos comportamentos, conscientes ou inconscientes, dos usuários que se encontram concentrados em atividades de ensino-aprendizagem. Rocha e Baranauskas (2003) explicam que design e ergonomia são ciências que primam pelo conforto do usuário e através das pesquisas destas autoras (*op. cit*) encontra-se uma outra lista de princípios básicos de um bom design, conforme os que são citados também por Norman (1998):

- **Visibilidade e percepção do objeto e de sua aplicação (*affordance*):** indicam o mapeamento entre as ações pretendidas e as ações reais, bem como distinções importantes a serem operadas. A falta dessa visibilidade é o que torna muitos dispositivos controlados pelo computador difíceis de serem operados. Segundo as autoras (*op. cit*), quando se tem a percepção do objeto e de sua aplicação (*affordance*), o usuário sabe o que deve fazer apenas olhando, sem que sejam necessárias figuras ou rótulos ou instruções.
- **Bom modelo conceitual:** permite prever o efeito das ações e, sem um bom modelo conceitual, opera-se cegamente sob o comando, efetuando-se operações “receitadas” sem saber que efeitos esperar ou quais atitudes devem ser tomadas, caso as ações não funcionarem. Conforme as coisas vão dando certo, aprende-se a operar. Agora, quando as coisas dão errado ou quando se depara com situações novas necessita-se de um maior entendimento ou de um bom modelo.

- **Bons mapeamentos:** mapeamento é o termo técnico utilizado para denotar o relacionamento entre duas entidades. No caso de interfaces, indica o relacionamento entre os controles e seus movimentos e os resultados no mundo. Mapeamentos naturais são aqueles que aproveitam analogias físicas e padrões culturais, levando ao entendimento imediato. Um objeto é fácil de ser usado quando existe um conjunto visível de ações possíveis e os controles exploram mapeamentos naturais.
- **Feedback:** é um conceito conhecido da teoria da informação e controle, viria a representar o retorno das informações ao usuário sobre as ações que forem feitas, quais os resultados obtidos. Exemplificando com interfaces computacionais, quando se manda imprimir um documento em impressoras de rede, sempre se depara com problemas clássicos, como: “qual foi mesmo a impressora para qual enviei?” - o *feedback* deveria aparecer neste tipo de situação.

A importância dos princípios de design se mostra ainda na opinião de Milheim & Lavix (1992) apud Fernandes (1996), que afirma que as interfaces de softwares educativos com bom design devem permitir a máxima aprendizagem, mas acrescentam que devem fornecer ao aluno o controle apropriado do processo dessa aprendizagem.

Para complementar esta idéia, Fernandes (op. cit) apresenta uma lista citada por Galitz (1989) dos que os usuários esperam ver nas interfaces de softwares educativos: (i) uma apresentação de elementos ordenada, limpa e pouco confusa; (ii) uma indicação óbvia do que está sendo mostrado e do que deve fazer com ele; (iii) que a informação surja num local certo, através de uma linguagem clara e simples; (iv) uma indicação clara do que se relaciona com quê (dados, instruções, opções, etc.); (v) um processo simples de encontrar o que está no programa e de como lá chegar.

Observa-se que, em regra geral, no mercado de softwares, o design de interface vem sendo guiado, através de processos de desenvolvimento e avaliação, que estão baseados em diretrizes de empresas como a Microsoft, que produziu o “The Windows Interface: Guidelines for Software Design” (Microsoft, 1995) e a Apple, empresa que produziu o “Macintosh Human Interface Guidelines” (Apple, 1992). Estas empresas possuem equipes multidisciplinares que realizam estudos empíricos sobre o que os usuários esperam das interfaces, indicando, mais uma vez, que esta é uma preocupação de várias disciplinas e não somente das áreas de design da informação e ergonomia.

Para entender melhor o design presente em interfaces de softwares educativos poderiam ser feitos recortes diferentes de pesquisa, como por exemplo, estudá-lo somente pela ótica da ergonomia cognitiva, ou somente pelas questões de engenharia de software. Entretanto, acredita-se que o ponto chave que existe no diálogo entre o usuário e a interface diz respeito a uma análise que une conceitos presentes em design da informação, ergonomia informacional e educação, áreas que se completam e possibilitam uma avaliação deste diálogo com foco na visão do usuário.

Apenas a título de complementação, cabe uma rápida diferenciação sobre estas duas áreas e a forma como são vistas suas aplicações no estudo de interfaces de softwares educativos. Soares (2004), explica que nas pesquisas em design da informação tem-se uma análise que vai, geralmente da fase de concepção de um produto / sistema até a fase de prototipagem e testes do mesmo. Já em ergonomia informacional, está explícito um processo mais completo e mais cíclico, onde deve ser estudada a forma com que o usuário executa seu trabalho e daí então, trabalhar partindo destas informações, a fim de criar um sistema que atenda às suas necessidades de forma mais objetiva.

Observando esta diferenciação, algumas questões despertam atenção: “a ergonomia informacional não englobaria a definição de design da informação?” Acredita-se que, a princípio, englobaria, porém seria mais coerente dizer que elas são complementares, pois ambas podem estudar os mesmo problemas de usabilidade através de parâmetros diferentes. Segundo Soares (2002) a pesquisa ergonômica fornece dados fundamentais sobre design de um produto / sistema, assim como os testes ergonômicos fornecem um método para avaliar a adequação de designs específicos, através da investigação de aspectos físicos, cognitivos e organizacionais referentes à utilização do mesmo.

Conclui-se este tópico propondo uma reflexão: se os elementos e princípios de design estão sujeitos a “diretrizes” de orientação sobre como devem se apresentar em interfaces de sistemas educativos, então, até que ponto seriam necessárias adaptações nesta forma de apresentação para que os usuários de softwares educativos compreendessem melhor as informações contidas neles?

Embora responder a esta pergunta não seja o objetivo deste trabalho, considera-se como sendo o tema central softwares educativos e, portanto, deve existir uma dedicação à pesquisa sobre como organizar os elementos e princípios de design a fim de que sejam projetadas interfaces que comuniquem as informações com mais clareza para seus usuários, sejam eles adultos ou crianças.

3.7 | Lições aprendidas

Descobriu-se que estudar software educativo é tarefa muito complexa para ser tratada somente sob a ótica de design, isto porque envolve conceitos multidisciplinares, que podem ser interpretados de inúmeras maneiras de acordo com o campo de pesquisa em que está sendo investigados. Como, exemplo, as questões relativas à interface são apresentadas, onde se entende que a definição formulada por um educador poderia ser completamente diferente da definição formulada por um designer, ou por um outro profissional que lide com desenvolvimento softwares educativos.

Neste capítulo, foi possível compreender que os softwares educativos assumem seu caráter **educativo** à medida que são utilizados com esta finalidade, mas percebeu-se também que existem softwares cujo conteúdo foi pensado (desenvolvido) exatamente para dar suporte a uma determinada área de estudo, como um software de história da arte, por exemplo.

Desta forma, entendeu-se que existem também softwares que, embora não tenham sido direcionados para “fins educativos”, podem se prestar a este papel, dependendo da forma com que são utilizados pelos atores envolvidos no contexto de uso, como é o caso do Microsoft Word usado, a princípio, para digitação de textos.

Como conclusões parciais deste estudo, está o fato de que essas ferramentas foram evoluindo ao longo dos anos, tendo suas bases pedagógicas fortemente influenciadas pelas teorias de educação que foram estudadas no capítulo 2 desta dissertação. Deste modo, ao estudar softwares educativos é preciso ter consciência de que esta é uma área de pesquisa que envolve, no mínimo, três grandes áreas: educação, ergonomia e design. Numa perspectiva mais ampla, conforme visto no capítulo anterior, envolve também psicologia, sociologia, filosofia, informática e engenharia de software.

Através de um breve histórico, foi possível relacionar as teorias de educação com as bases pedagógicas encontradas em softwares educativos, onde muitos dos conceitos são transmitidos aos usuários através de uma interface gráfica que reúne elementos e princípios de design a fim de oferecer a possibilidade de “diálogo” usuário-software. Esses elementos e princípios foram sendo estudados, principalmente por pesquisadores de ergonomia, informática e design da informação, a fim de encontrar caminhos para aumentar a satisfação do usuário ao lidar com ferramentas como softwares educativos.

Assim como se observa nas conclusões dos capítulos anteriores, novamente, buscou-se construir uma representação gráfica das descobertas a respeito das intersecções entre as áreas de pesquisa relacionadas a estudos sobre softwares educativos, sendo agora o nosso foco nas áreas de ergonomia, design e educação. Busca-se estabelecer relações entre as três disciplinas citadas a fim de mostrar nossa compreensão dessas conexões. Partiu-se dos nomes das grandes áreas de pesquisa e foram selecionados alguns tópicos principais, os quais acreditou-se que deveriam ser discutidos em outras pesquisas sobre softwares educativos, conforme observamos na figura 3.9:

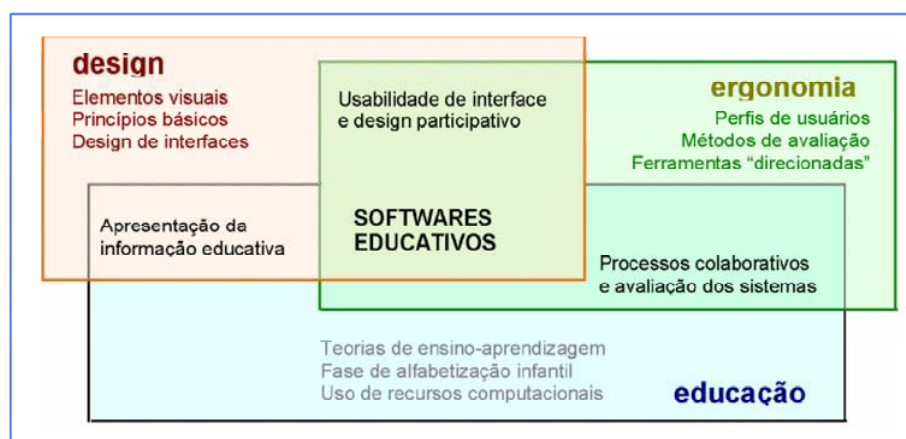


Figura 3.9: Áreas de pesquisa e tópicos envolvidos na avaliação de softwares educativos.

A figura 3.9 apresenta graficamente a reflexão que foi feita sobre os temas e subtemas relacionados a softwares educativos. Entende-se que da área de design, é essencial considerar a forma com que os elementos e princípios estão presentes em interfaces de softwares educativos. A área de ergonomia contribui com conceitos sobre métodos e ferramentas adequados para avaliação de interfaces, assim, ajuda a entender a importância do papel do usuário em todo processo de avaliação de um software dessa natureza. A área de educação, por sua vez, traz ao pesquisador as informações que orientam sua busca pelo conhecimento a respeito do usuário, como por exemplo: como se dá a fase de alfabetização escolar, como as pessoas se relacionam com recursos computacionais, etc. Sendo assim, não se poderia deixar de mencionar as intersecções entre as áreas, tais como: design participativo, conteúdo educativo através de interface digital e processos colaborativos através para avaliação de sistema.

Muitos destes temas já foram discutidos neste capítulo, no entanto, alguns serão reapresentados no capítulo sobre usabilidade, pois se acredita ser relevante entender como estes temas podem contribuir para melhores avaliações de usabilidade em softwares educativos, sejam elas fundamentadas em critérios de design, de educação ou de ergonomia, todas buscam melhorias para esta categoria de produto.

Estudou-se também quem são os atores envolvidos na utilização de softwares educativos: os alunos (ex: crianças), os educadores (ex: professores) e os desenvolvedores (ex: designers). Embora já se saiba que existam softwares educativos direcionados para adultos, ou ainda para educadores que não sejam os professores, acredita-se que para o recorte desta pesquisa, estes seriam os atores principais, isto porque eles foram considerados como usuários “potenciais” de softwares educativos, tendo em vista o papel que cada um exerce nessa utilização da ferramenta.

Outra conclusão oriunda dos estudos feitos para este capítulo é sobre a importância de investigar e valorizar as ações do educador e de seus alunos enquanto utilizam um software educativo. Aceita-se a idéia de que comportamento deles, ou suas dúvidas e sugestões podem ser convertidas em informações valiosas para o designer, uma vez que este estará desenvolvendo, ou avaliando, um software educativo que têm estas pessoas como seus usuários potenciais.

Foi visto também que, ao caracterizar e classificar os softwares com base nos autores pesquisados, foi possível caracterizar e classificar o software utilizado no estudo experimental desta dissertação. Para tanto poderia-se, talvez, especular sobre futuras classificações, pois acredita-se que com a evolução dos softwares educativos, muito em breve, tais classificações apresentadas ao longo do capítulo, podem não atender às necessidades dos pesquisadores. Entretanto, para o trabalho ora apresentado, estas características e classificações mostram-se satisfatórias e coerentes, considerando que já foram utilizadas por outros pesquisadores.

Para concluir, foi percebido que a evolução dos softwares educativos ao longo dos anos despertou a atenção de muitos estudiosos de diferentes áreas, mas observa-se que nem sempre houve uma “integração” entre eles. O que faz crer que, aparentemente, as pesquisas sobre este tema estivessem acontecendo de forma um tanto isolada, como se, por exemplo, um pesquisador de design investigasse um software educativo somente do ponto de vista de design, isto é, sem perceber o quanto outras ciências (como educação e psicologia) teriam a contribuir para compreensão do tema.

Sendo assim, a melhor descoberta ao longo do estudo, foi identificar que existem centenas de pesquisas sobre softwares educativos, com inúmeras abordagens, e todas elas têm seu devido valor. No entanto, essa descoberta também nos leva a repensar sobre o “recorte do tema” escolhido para a presente pesquisa, pois percebeu-se que, assim como o “olhar” do pesquisador está baseado em seu repertório de conhecimento e na sua formação científica, tal pesquisador deverá também conhecer “outros olhares” para que sua pesquisa não se torne relato limitado a um único foco.

Analisando a evolução dos softwares, os papéis dos usuários “potenciais” e outros subtemas deste capítulo, conclui-se que a “fronteira” que separa as pesquisas sobre software educativo pode ser vista como um “obstáculo” para um pesquisador que não queira aceitar o desafio de buscar outras visões do tema, ou pode ser vista como uma “recompensa” para quem quer correr o risco de aprender mais sobre o mesmo tema, contando com a colaboração multidisciplinar de outras áreas de conhecimento.