3
Design de interação e Ambientes Virtuais de Aprendizagem (AVA)

Segundo Pereira et al. (2007), AVA (ambientes virtuais de aprendizagem) é a nomenclatura adotada no Brasil para designar os softwares que cumprem a função de mediar o ensino a distância veiculado através da internet ou espaço virtual. O AVA é composto por diversas mídias oriundas da evolução das Tecnologias de Comunicação e Informação (TIC´S) que permitem a emissão e a recepção de mensagens. Tem como principal função permitir a interação entre os "atores do processo educativo" (Pereira et. al, 2007, p.4). A qualidade deste ambiente depende do envolvimento de cada ator durante a execução deste processo.

Na literatura internacional, são encontrados os seguintes termos relacionados a AVA: "Web-based learning, online learning, learning management systems, virtual learning, enviroments, e-learning," entre outros. (Pereira et al., 2007, p.5).

Milligan (1999 apud Pereira et al., 2007) descreve que para um software educacional [8] ser considerado um AVA, ele deve apresentar algumas ferramentas fundamentais: - Controle de acesso; Administração para acompanhar os atores do processo educativo; Controle do tempo; Avaliação; Comunicação (síncrona e assíncrona); Espaço privativo; Gerenciamento de conteúdo; Apoio ou ajuda on-line sobre o ambiente; Manutenção. E Pereira et al. (2007) completam que além de apresentar estas ferramentas o AVA precisa moldar a proposta pedagógica de cada instituição que se encaixa dentro de quatro eixos fundamentais: gerenciamento pedagógico; gerenciamento administrativo; produção; comunicação, documentação e informação.

Pereira et al. (2007) citam que o processo evolutivo de um AVA ainda está em fase inicial e que é necessário a colaboração de pesquisas relacionadas

[8] Chavez descreve que não é fácil definir o que é realmente um software educacional. Para este autor, qualquer software utilizado para algum objetivo educacional ou pedagógico pode ser considerado um software educacional, independente da natureza ou finalidade para qual tenha sido criado.Disponível em:< http://edutec.net/Textos/Self/EDTECH/softedu.htm >. Acesso em: 12 ago. 2010

à interface, navegabilidade, adaptabilidade e usabilidade para que estes softwares sejam mais transparentes aos usuários. E que “a idealização, o desenvolvimento ou a customização de um AVA e a sua interface deve seguir aspectos pedagógicos, funcionais, ergonômicos e estéticos”. (Pereira et al., 2007, p.18)

O suporte à aplicação destes aspectos em um AVA pode ser fornecido através do design de interação. Preece et al. (2005) descrevem o design de interação como fundamental para pesquisas e projetos de sistemas baseados na interação humano-computador, pois investiga o uso da interface a partir de uma abordagem de desenvolvimento centrada no usuário. Destacam que o processo do design de interação envolve quatro atividades básicas:

1) Identificar necessidades e estabelecer requisitos
2) Desenvolver designs alternativos que atendam esses requisitos
3) Construir versões interativas dos designs
4) Avaliar o design construído

Estas atividades do design de interação estão inseridas dentro de um processo que possui outras atividades que interferem no desenvolvimento do AVA. Meister (2001) e Preece et al.(2005) pontuam que o designer que deseja trabalhar com desenvolvimento de software precisa conhecer as características específicas do processo para fornecer boas soluções para a usabilidade e para o design da interface.

Para compreender melhor este processo, a presente pesquisa apresenta nos itens a seguir algumas características destas atividades, cita algumas recomendações que auxiliam na construção interativa de design e descreve alguns métodos e técnicas, existentes no mercado atual, para avaliar o design construído.

3.1
Design no Processo de desenvolvimento de um AVA

O desenvolvimento de um AVA obedece as mesmas especificações de processo para o desenvolvimento de um software qualquer. Segundo Somerville (2007), o processo de desenvolvimento de software consiste na associação de atividades e resultados para a produção de um software. Estas atividades “são comuns a todos os processos de software” (Somerville, 2007, p. 6) e são categorizadas como: especificação, desenvolvimento, validação e evolução de software. A realização destas atividades é uma tarefa complexa que depende do

tipo de cada software, das pessoas e das organizações envolvidas, por isso esta pesquisa aborda resumidamente as características das atividades.

O software educativo, independente das suas características pedagógicas, se enquadra no processo básico descrito por Somerville. O modelo[9] de processo adotado para a elaboração deste software se modifica de acordo com os objetivos da empresa de desenvolvimento, da equipe pedagógica e anseios dos usuários (professor, aluno, conteudista, etc).

A etapa de especificação do software, também chamada de engenharia de requisitos, determina quais serão os objetivos e as restrições do processo. Nesta etapa, os custos, o tempo, os fatores impostos pelo cliente e pela tecnologia são analisados. Gomes et al.(2003) realizaram um estudo sobre os requisitos para projetos de software educativo e destacaram que estes requisitos devem englobar a aprendizagem dos alunos, a "mediação a ser promovida pelo professor" junto com o contexto de uso e a qualidade técnica. Estes autores também alertam que os usuários do software educativo já precisam ser consultados nesta etapa para captar melhor os requisitos do sistema, as necessidades dos usuários constituem as bases dos requisitos e determinam o desenvolvimento do design.

Campos et al.[10] (1996) pontuam que o método de desenvolvimento evolucionário[11] é o mais adequado para a construção de uma hipermídia educacional, pois permite a participação dos usuários em todas as etapas do processo de construção de software para que novos requisitos sejam acrescentados ao desenvolvimento do software. Somerville (2007) alerta que a presença de erros nesta etapa de especificação pode prejudicar todo o processo de desenvolvimento e implementação do software.

A etapa de desenvolvimento de software envolve os processos de projeto e de programação do software (Somerville, 2007). O processo de projeto possui atividades específicas e intercaladas que são comuns a qualquer tipo de software. As atividades são: projeto de arquitetura, especificação abstrata, projeto de interface, projeto de componente, projeto de estrutura de dados e projeto de algoritmo.

[9] Modelos são abstrações dos processos que são utilizadas para explicar diferentes enfoques para os desenvolvimentos de software. Alguns modelos são definidos como: cascata, evolucionário e engenharia de software baseada em componentes. Somerville (2007)

[10] CAMPOS, F.; CAMPOS, G.; ROCHA, A. R.. **Dez etapas para o desenvolvimento de software educacional do tipo hipermídia.** Rio de Janeiro, UFRJ Disponível em< http://www.c5.cl/ieinvestiga/actas/ribie96/ETAPAS.HTML>. Acesso em: 29 mai. 2010

Para o desenvolvimento do software educativo é fundamental definir qual será o método de ensino adotado, pois este guiará as tomadas de decisão sobre as atividades do projeto. Por exemplo, é no projeto de interface que as características do método de ensino são representadas e outros requisitos que foram definidos, mas para que isto aconteça, a interface deve ser construída com base nos resultados provenientes da avaliação da usabilidade realizada em diferentes contextos de uso com usuários específicos durante o desenvolvimento do software. O objetivo desta etapa é verificar se a interface permite analisar a aprendizagem de conceitos específicos (Gomes et al., 2003.).

Após a construção da interface, iniciam-se os projetos de componente, dados e algoritmo que se caracterizam de acordo com perfil da empresa e equipe de desenvolvedores. Reco[12] (2004) destaca que a linguagem de programação do software educativo deve ser fácil, ter bom nível de interatividade e flexível para permitir alterações no decorrer do processo.

A etapa de validação demonstra se o software está em conformidade com os objetivos que foram propostos na etapa de requisitos. Inclui validação da qualidade técnica, da interface do mesmo e, no caso de software educacional, validação da qualidade pedagógica. São realizadas inspeções e revisões que testam os componentes individuais desenvolvidos, o sistema integrado e a aceitação do software com os usuários finais. É o momento de confirmar se o software educacional solucionou o problema de ensino aprendizagem para qual foi desenvolvido (Campos et al.,1996).

Por fim, a etapa de evolução do software, ou manutenção, permite alterações no mesmo ao longo de sua vida útil (Somerville, 2007). Esta etapa é fundamental para o software educativo, pois permite o reparo de defeitos, a adaptação do software a um ambiente operacional diferente e, principalmente, adicionar ou modificar alguma funcionalidade para se adequar ao perfil dos usuários e à proposta de ensino e aprendizagem.

É importante destacar que o designer instrucional junto com o especialista em ergonomia devem assegurar que os padrões para uma boa usabilidade devem estar presentes em cada uma destas etapas.

---

[11] O desenvolvimento evolucionário baseia-se na ideia de desenvolvimento de uma implementação inicial, expondo os resultados aos comentários do usuário e refinando esse resultado por meio de várias versões até que seja desenvolvido um sistema adequado. (Sommerville, 2007, p. 45)

[12] REGO,B.. Concepção de Software Educativo no Ensino/aprendizagem das Línguas. Escola Superior de Educação, Instituto Politécnico de Viseu. Disponível em: http://www.esev.ipv.pt/servicos20042005/upload%5Cma%5C817%5CArtigosobreindividualiza%C3%A7%C3%A3oeinteractividade.pdf>. Acesso em: 25 mai 2010

3.2
Design de interface para AVA

Preece et al. (2005) citam que a atividade de desenvolvimento do design de interface se subdivide em duas partes: design conceitual e design físico. O design conceitual apresenta o modo de interação mais significativo para o produto, pode ser baseado em atividades (instruir, conversar, manipular, navegar, pesquisar, etc.) ou na manipulação de objetos já existentes no mercado. Baseada em Preece et al. (2005), Torrezan et al. (2008) fizeram um levantamento característico do modelo conceitual para o design de materiais educacionais e destacou a importância de uma equipe interdisciplinar (pedagogos, professores, designers, técnicos em informática, programadores e conteúdista) para construção do modelo conceitual, pois é fundamental refletir sobre a atividade pedagógica aliada ao planejamento técnico e gráfico.

O design físico surge da evolução do modelo conceitual, consiste em uma necessidade de concretizar as atividades, as decisões e escolhas selecionadas para o design de interação. Kalbach (2009, p. 291) trata o design físico de interfaces web como atividade multidisciplinar e esta envolve três áreas: design de informação, interação com a navegação e design gráfico.

Preece et al. (2005) sugerem que os designers devem se apoiar em princípios, regras e recomendações para a construção do design físico. Os problemas que podem surgir durante a construção do design físico podem ser solucionados ao observar designs semelhantes.

De acordo com Santos[13] [199?], a interface dos softwares educacionais para ambiente web não apresenta diferenças significativas das interfaces dos sites convencionais. Por isso, as diretrizes do projeto para interfaces educacionais seguem as mesmas diretrizes para design de interface de sites em geral.

A presente pesquisa apresenta a seguir algumas recomendações elaboradas por diversos autores para cada uma dessas áreas, mas destacando em alguns momentos, recomendações específicas para interfaces de softwares educacionais.

[13] Disponível em:< http://pt.scribd.com/doc/6795008/Neide-Santos-Design-de-Interfaces-de-Software-Educacional> Acesso em: 02 Fev. 2010

Áreas	Recomendações	Autores
Design da Informação	**Design da Interface:** • Evite elementos gráficos (linhas, pontos) desnecessários na página; • Manter uma diagramação básica e repetitiva para aplicação em todas as páginas do ambiente; • Crie páginas com medidas que se adaptam a qualquer resolução da tela (Design Líquido[14]); • O design da interface deve ser consistente; • Valorize os espaços em branco da página; • Utilize navegação em camadas para priorizar as informações mais importantes, crie uma hierarquia visual de elementos na página; • Titule o nome do curso em todas as telas. • Acrescente um link para retornar a página principal em todas as telas; • Assegure que a página tenha uma navegação fácil e consistente. E que contenha somente a informação necessária para que o aluno realize a tarefa; • Utilize CSS[15] para assegurar que	KALBACH, James (2009) WEINMAN, Lynda (1998) WILLIAMS, Robin e TOLLETT, John (2001) NIELSEN, Jacob (2000) TULLIS (1997) apud AGNER (2002) AGNER (2002) KRUG (2006)

[14] FINCK (1999, apud AGNER, 2002) considera o design líquido como o projeto visual para site que se adapta de acordo com as dimensões da tela do usuário;

	a página fique fácil de carregamento e acessível a todos os usuários; • Faça uso de listas drop-down para facilitar o acesso das informações; • Apresente uma cor diferente para os links visitados; • Realize pesquisa com os alunos para descobrir quais são as informações mais importantes; • Destaque as informações mais importantes para o aluno; Elementos mais requisitados devem aparecer no topo da tela; • Apresente categorias de informações de forma clara para os alunos; • Considere os diversos tipos de browser existentes; • Evite o uso de rolagem, caso não seja possível, utilize componentes para compactar o tamanho da página;	
	Para o texto: • O vocabulário deve ser simples e familiar ao aluno; • Utilize fonte sem serifa. Texto escrito em maiúscula é mais difícil para ler, utilize este texto em rótulos de menu de navegação;	

[15] CSS (Cascading Style Sheets) linguagem de estilo para definir a apresentação de documentos escritos em linguagem de marcação como HTML e XML. Disponível em: <http://pt.wikipedia.org/wiki/Cascading_Style_Sheets>. Acesso em: 01 Jun. 2010.

	• O tamanho da fonte deve ser maior que 10 pontos, recomendam-se utilizar fontes de 12 pontos. Vários tamanhos de fontes auxiliam na hierarquia visual da navegação; • Utilize negrito e itálico para destacar algum texto importante da página. Utilize fontes padronizadas para web (Arial, Verdana, Times New Roman, Courier, Helvética); • O alinhamento do texto à esquerda é mais fácil de ler; • Utilize contraste de cores do texto com o fundo da página; • Evite textos longos e distribuídos em várias telas; • Utilize imagens de texto somente para títulos e botões; • Quando não for possível utilize palavras completas, recomenda-se fazer o uso apropriado de abreviações;	
Interagindo com a navegação	• O AVA deve apresentar uma navegação fácil e com linguagem familiar ao aluno; • O aluno deve utilizar o mínimo de passos para realizar uma tarefa; • O AVA deve apresentar um conteúdo condizente com o material do curso; • O painel de menu deve ter informações claras com	TULLIS (1997) apud AGNER(2002) KALBACH, James (2009) PORTUGAL(2004) KRUG (2006)

	descrição para cada palavra ou expressão; • Organize as informações do menu, uma recomendação é listar as opções dos menus por ordem de freqüência de utilização; • Facilite a retro-navegação; • Proponha um retorno visual a seleção dos itens do menu; • Utilize o sublinhado para destacar algum link, não utilize sublinhado em todos os links da tela, pois pode atrapalhar a legibilidade da informação; • Não dependa de uma cor para comunicar a navegação ou qualquer elemento de importante de uma página; • Crie alvos maiores para os links; • Insira legendas em links e imagens; • As opções do menu devem ter sentenças curtas; • Minimize o esforço necessário do aluno para compreender e tomar decisões sobre as opções de navegação; • Evite a janela pop-up. Os recursos de navegação e funcionalidades devem ser a apresentados, se possível, em uma única tela; • Agrupe as páginas em seções lógicas; • Disponibilize trilhas (breadcrumb)	

	e mapa de navegação do ambiente virtual; • Evite o uso de barra de rolagem; • Reduza o tempo de resposta às ações do aluno; • Possibilite ao aluno editar, acrescentar e modificar funções específicas do sistema; • Insira ferramentas de desfazer a ação do aluno; • Insira ferramentas de ajuda; • O AVA deve apresentar objetivos claros de aprendizagem: o escopo e a seqüencia do conteúdo devem ser bem definidos; • O AVA deve guardar informações relevantes sobre a performance dos alunos.	
Design Gráfico	**Cor:** • Utilize cores hexadecimais para os textos; • Utilize cores para distinguir as áreas de navegação; • Não abuse no emprego das cores; • Utilize código de cores familiar e consistente; • Pode ser utilizada para transmitir informações qualitativas e quantitativas. E contribui para a memorização; • Utilize de três a sete cores para layout de um curso a distância; • A cor azul deve ser utilizada com cautela em textos;	WEINMAN, Lynda (1998) PIMENTA, Sofia (2007) MARCUS (1994) apud AGNER (2002) PORTUGAL(2004)

	• Utilize uma cor suave para o fundo da página. Não coloque uma cor escura de background com texto claro para páginas com opção de impressão; • Cores fortes devem ser utilizadas para atrair a atenção; • Utilize as cores para agrupar os elementos da página; • Use um código de cores para aplicações onde o aluno precisa distinguir várias categorias de dados; • Considere o contexto cultural do aluno para fazer a escolha da cor.	
	Ícones: • O uso de metáforas auxilia no aprendizado, mas o seu uso deve ser ponderado; • Use rótulos claros e sem abreviatura; • Devem ser consistentes; • Utilize métodos e técnicas ergonômicos para validá-los com os usuários.	NIELSEN, Jakob (2000)
	Imagens: • Devem conter um título (alt[16]) para serem acessíveis a todos os alunos; • Devem ser salvas em formatos	WEINMAN, Lynda (1998) WILLIAMS, Robin e TOLLETT, John

[16] Função ALT é utilizada na linguagem de HTML para titular as imagens. O código desta função é: <img src="nomedoarquivodaimagem" alt=nome da imagem">. KALBACH, James (2009)

	de arquivos para web: JPG[17], PNG[18], GIF[19] e SVG[20]; • Devem ter baixa resolução: 72dpi; • Devem ser convertidas para os modos de cores RGB ou Indexada; • As imagens utilizadas no conteúdo devem ser colocadas em tamanho pequeno com um link para uma foto maior que contenha maiores detalhes.	(2001) PORTUGAL (2004)
	Tabela: • Use um comprimento de linha moderado; • Utilize margens para definir a área de leitura; • Utilize componentes[21] para criar tabelas dinâmicas que permitem manipulação da estrutura e do conteúdo; • Utilize cores para destacar os textos apresentados nas tabelas; • Padronize o design das tabelas.	WEB STYLE GUIDE apud PORTUGAL (2004)
	Multimídia: Animação, Som e Vídeo: • Apresente animação para o estado ocioso do sistema. • As animações devem ter regras claras para facilitar a interação	MORAES , Márcia e MILENE Silveira (2007) FILATRO, Andrea (2008)

[17] JPEG – formato de arquivo de alta compactação para imagens. Utiliza 16,7 milhões de cores. WEINMAN, Lynda (1998);
[18] PNG – Portable Network Graphics, método de compactação sem perda de qualidade para imagens. Armazena até 32 bits, aceita transparência. WEINMAN, Lynda (1998);
[19] GIF – Graphic Interchange Format formato de arquivo de alta compactação para imagens. Utiliza 256 cores e aceita transparência e animação. WEINMAN, Lynda (1998)
[20] SVG - abreviatura de Scalable Vectorial Graphics, utiliza linguagem XML para descrever formato vetorial para imagens bidimensionais. VALENTINE, Chelsea et. Al (2001)
[21] Componente é uma unidade de software independente que pode ser composta por outros componentes de software, encapsula uma série de funcionalidades. Sommerville (2008)

	do aluno com o sistema; • A animação deve representar um agente usando um personagem ou uma figura humana que inspire confiança para que o estudante acredite nas suas recomendações, este agente deve apresentar personalidade e comportamentos condizentes com a cultura dos alunos; • O aluno deve ter controle (desfazer ações, parar e iniciar) sobre o agente animado; • A animação deve fornecer para o aluno o controle sobre o seu processo de aprendizagem; • A animação deve prevenir ações dos alunos que comprometam o seu funcionamento; • O aúdio deve ser utilizado para expressar: mensagens simples e curtas; Quando o sistema visual do aluno estiver sobrecarregado; E quando a mensagem exige uma ação imediata; • O conteúdo do áudio deve estar disponível em material impresso para acompanhamento do aluno; • O vídeo deve ser utilizado para mostrar a personalidade do tutor; • O vídeo pode ser utilizado para ilustrar/reforçar explicação de conteúdos muito complexos; • Deve dispor de uma boa qualidade de áudio; • O vídeo não deve ter um tempo	DEATHERAGE (1972) apud MENDEL (1997) AGNER (2002)

	muito longo de duração. Podem ser utilizados para apresentar a personalidade dos participantes e representar seqüências de maneira realista.	
	Formulário: • Coloque títulos compreensíveis: evitar terminologia da computação; • Permita aos usuários fácil navegação no formulário; • As instruções devem ser compreensíveis. • Agrupamento lógico e seqüenciamento de campos: os campos relacionados devem estar adjacentes e o seqüenciamento deve repetir padrões usuais; • Layout do formulário com apelo visual: atentar ao espaçamento entre os campos de entrada; se os usuários estão trabalhando a partir de um formulário em papel, assegurar-se que o *layout* da tela e do impresso sejam equivalentes; • Rótulos de campos familiares: use termos comuns; • Prevenção de erros quando possível; • Correção de erro para caracteres individuais e campos inteiros: deve ser permitido apagar e sobrescrever;	NORMAN (2003) apud SANTA-ROSA, José Guilherme e MORAES, Anamaria (2005) SHNEIDERMAN (1998) apud SANTA-ROSA, José Guilherme e MORAES, Anamaria (2005)

	• Botão de envio: deixar claro aos usuários, no local correto, qual a ação a ser tomada para demonstrar que terminaram de preencher o formulário.	

Tabela 6 - Recomendações para o design da interface

A criação de recomendações é um processo evolutivo que pode ser influenciado pelo perfil do usuário e pelo avanço tecnológico, uma recomendação aplicada atualmente não pode ter a mesma validade daqui a um determinado período de tempo. Exemplos como, recomendações que orientam ao uso de frames[22] e tamanho fixo para a resolução das imagens em tela já não fazem mais sentido ao serem aplicadas nas tecnologias atuais, pois estudos sobre acessibilidade indicam que o uso de frame impede os softwares de leitura de tela de apresentar o conteúdo aos usuários com deficiência visual. E a resolução fixa da imagem em tela impede que o site possa ser visualizado em qualquer tamanho de tela.

Nem sempre as recomendações apresentadas podem atender as necessidades de um perfil específico de um grupo de usuário, o ideal é investir em pesquisa de usabilidade para validar as recomendações sugeridas e propor novas recomendações.

3.3
Usabilidade e qualidade do Software Educacional

A qualidade de um software depende da combinação de diversos fatores que se modificam de acordo com o contexto de uso e os requisitos solicitados pelo cliente (Pressman, 1995). Segundo este mesmo autor, um fator de exemplo é a usabilidade que é “avaliada considerando os fatores humanos, a estética global, a consistência e a documentação”. (Pressman, 1995, p. 729).

A ISO[23] 9126, criada em 1991 para fornecer uma estrutura para avaliar a qualidade do software, é definida por um modelo composto de características ou

[22] Kalbach (2009, p.207) descreve que a recomendação atual é evitar o uso de frames e propõe explorar novas soluções.

[23] ISO (**Organização Internacional para Padronização)** organização não governamental e que coordena o trabalho de órgãos de 127 países membros para promover a padronização de normas técnicas em âmbito mundial. Disponível em:<

fatores gerais do software. A sua função é apresentar às empresas de desenvolvimento de software um modelo de qualidade a seguir que promete evitar a insatisfação do cliente, o retrabalho e propiciar a redução de custos. Esta ISO destaca a usabilidade como elemento fundamental em um conjunto composto pelas demais caracterísitcas: funcionalidade, confiabilidade, eficiência, manutenibilidade e portabilidade.

Estas características (tabela 7) se subdividem em outras subcaracterísticas e estas subcaracterísticas se subdividem em atributos[24]. Estes atributos são medidos durante o desenvolvimento do software. Segundo Somerville (2007), a criação e a importância dada a cada atributo depende do perfil da empresa de desenvolvimento e dos objetivos do processo de qualidade adotado. É mais comum especificar a usabilidade qualitativamente ao invés de usar métricas.

Características	**Subcaracterísticas**
• **Funcionalidade:** conjunto de atributos que devem garantir que as funções devem satisfazer necessidades explícitas ou implícitas.	Adequação, Acurácia, Interoperabilidade, Segurança de acesso, Conformidade.
• **Usabilidade:** conjunto de atributos que descrevem o esforço necessário para o uso;	Inteligibilidade, Apreensibilidade, Operacionalidade
• **Confiabilidade**: capacidade de um software para manter seu nível de desempenho por um período determinado.	Maturidade, Tolerância a falhas, Recuperabilidade
• **Eficiência:** conjunto de atributos que incidem sobre a relação entre o nível de desempenho do software e a quantidade de recursos utilizados	Comportamento em relação ao tempo, comportamento em relação aos recursos.
•**Manutenibilidade**: atributos que descrevem o esforço necessário para fazer alguma modificação no software;	Analisabilidade, Modificabilidade, Estabilidade, Testabilidade
• **Portabilidade:** capacidade do software de ser transferido de ambiente;	Adaptabilidade, capacidade para ser instalado, capacidade para substituir, conformidade

Tabela 7- Características e subcaracterísticas (CHUA e DYSON, 2004)

Segundo Antonellis et al. (2007), a norma ISO 9126 não especifica como são medidos os atributos e desconsidera fatores não técnicos como, gerencial, político e econômico que podem contribuir com o sucesso do software.

http://pt.wikipedia.org/wiki/Organiza%C3%A7%C3%A3o_Internacional_para_Padroniza%C3%A7%C3%A3o> Acesso em: 15 Ago. 2010.

[24] Atributos estão relacionados a métrica de software que podem ser expressos e medidos numericamente. SOMMERVILLE (2007)

Destaca que alguns atributos são considerados mais importantes que outros e este grau de importância depende dos anseios do usuário do software.

O usuário “gerente” está interessado na qualidade global, em características que podem interferir no atraso do cronograma de planejamento, nos prazos de execução e no excesso de custos. O usuário “desenvolvedor” está interessado na implantação dos requisitos, nas qualidades intermediárias e finais do software. O “usuário final” está interessado em usar com facilidade o software, verificar se o software dispõe as funções exigidas, se é confiável e eficiente.

A qualidade de um software educacional não está desvinculada das exigências da ISO 9126 e nem da engenharia de software, mas as diretrizes determinadas por estas disciplinas também não são suficientes para o processo de avaliação. CHUA e DYSON (2004) utilizaram a ISO 9126 para avaliar um software de educação a distância e concluíram que as subcaracterísticas são genéricas e deveriam oferecer atributos mais específicos. Estes autores destacaram que a característica usabilidade deveria incluir subcaracterísticas como coerência, simplicidade, consistência, legibilidade, uso de cores e sistema de ajuda. E eles propõem a inclusão da “satisfação do usuário” como uma característica global e concluem que a norma é útil para obter uma visão dos pontos fracos e fortes do software, mas a aplicação da norma deve ser acompanhada de atributos que apresentam critérios didático-pedagógicos. Para Godói (2009) a avaliação da qualidade de um software educacional exige a análise de 3 critérios: pedagógicos, ergonômicos e comunicacionais.

De acordo com Pressman (1995) não há como medir com precisão a qualidade de um software, pois este processo é subjetivo e exige a participação de especialistas para avaliar cada parte. Além dos usuários desenvolvedor e gerente, o software educacional possui outros usuários finais: aluno, professor e mantenedor. Esta variedade de usuários para o software educacional contribui muito para a complexidade da avaliação da sua qualidade.

No campo da avaliação da qualidade do software educacional é possível encontrar alguns métodos e ferramentas que avaliam o software de acordo com os fatores técnicos apresentados na ISO9126 acrescentados dos fatores didático-pedagógicos. Estes métodos e ferramentas são baseados em checklists, diretrizes, escalas de avaliação e questionários (Godoi, 2009). Alguns métodos e ferramentas encontrados na literatura atual e que enfatizam a usabilidade e ergonomia são descritos como:

• **Método Reeves:** apresenta critérios pedagógicos e sobre a usabilidade da interface. Estes critérios são demonstrados em uma escala de avaliação que vai do 0(zero) a 1(um). A análise gráfica da pontuação obtida em cada critério determina se o software é de maior ou menor qualidade.(Silva, 1999)

• **Método Ticese:** Técnica de Inspeção Ergonômica de Software Educacional. Segundo Silva (1998) este método é baseado em questões e apresenta três módulos (classificação, avaliação, contextualização) que são baseados em aspectos cognitivos, psicologia da aprendizagem, pedagogia e ergonomia.

• **Método Ergolist**[25]: propõe uma avaliação rápida da interface para descobrir problemas ergonômicos. É baseado em checklist composto por opções que determinam a ergonomia de uma interface humano-computador. As opções investigadas neste método são: presteza, agrupamento por localização, agrupamento por formato, feedback, legibilidade, concisão, ações mínimas, densidade informacional, ações explícitas, controle do usuário, flexibilidade, experiência do usuário, proteção contra erros, mensagem de erro, correção de erros, consistência, significados e compatibilidade. Além de apresentar questões relacionadas a estas opções, o método apresenta um conjunto de recomendações específico para cada opção. (Silva, 1999)

• **Técnica de Muchielli:** Segundo Silva (1998), esta técnica propõe um modelo de concepção do software educacional que aplica seis fases para o projeto e avaliação. Estas fases consistem em elaboração, preparação, concepção da maquete, desenvolvimento, experimentação e melhorias, e por último, avaliação pedagógica. Na avaliação pedagógica são analisados itens sobre a qualidade do modelo pedagógico, o conceito de concepção, a flexibilidade do software, o sistema de ajuda, a qualidade das telas e a avaliação contínua do produto. Para a coleta de dados, este método utiliza a observação das reações do usuário - alvo, avaliação das aquisições e impressões sobre a qualidade do software, questionários, entrevistas e ouvir um grupo de especialistas antes de fazer funcionar o software.

• **Modelo de avaliação Guedes.** Check-list direcionado aos profissionais de educação para avaliar a usabilidade e a qualidade didático-pedagógica de ambientes virtuais de aprendizagem. Apresenta 10 critérios de avaliação, sendo que cada critério é composto de 5 atributos: Apresentação geral da interface; Legibilidade da interface; Navegabilidades; Adaptação ao usuário; Orientação

[25] Disponível em:< http://www.labiutil.inf.ufsc.br/ergolist/index.html>. Acesso em: 10 Jun. 2010

adequada ao usuário; Proximidade e agrupamento; Alinhamento; Padronização e consistência; Mecanismos de avaliação; Princípios pedagógicos adotados. (Fernandes, 2010)

• **MAEP:** método de avaliação ergopedagógico. Consiste em uma ferramenta de avaliação composta por critérios pedagógicos e ergonômicos, baseia-se em Check-list e não exige do avaliador conhecimentos em ergonomia e pedagogia. (Godoi, 2009)

•**FASE**[26]: ferramenta de avaliação automática de software educativo direcionada para professores. Utiliza várias questões que avaliam a interface gráfica, a usabilidade, os requisitos funcionais e os fatores cognitivos/pedagógicos. (Souza, 2006)

• **PEDACTICE**[27]**:** avalia o software através de uma ficha descritiva que analisa o potencial de aprendizagem (natureza curricular, os objetivos da aprendizagem, interação social, formas de avaliação, etc.) e a apreciação global do produto (flexibilidade, versatilidade, qualidade do conteúdo, qualidade da interface gráfica, etc.). É direcionado aos professores e especialistas da educação que podem realizar a análise do software em sala de aula ou visualizar os fóruns e resultados sobre os produtos já avaliados. Este instrumento possibilita a criação de um banco de dados que demonstra a eficiência dos softwares avaliados. (Godoi, 2009)

• **MAQSE** - Metodologia para avaliação de qualidade de *software* educacional: orienta os desenvolvedores e usuários (professores, alunos, pedagogos) na escolha do software. Utiliza na avaliação formativa ou somativa técnicas variadas para a identificação dos problemas. (Godoi, 2009)

• **MAQSEI**[28]: - Metodologia de Avaliação de Qualidade de Software Educacional Infantil: engloba aspectos técnicos (usabilidade) e pedagógicos do software infantil. Pode ser utilizado por profissionais da área de educação, informática e afins que realizam avaliações somativas ou formativas. Consiste em várias fases: 1ª) Reconhecimento e proposta do software; 2ª) Planejamento dos testes; 3ª) Realização dos testes com usuários-alvo; 4ª) Análise dos dados e produção do relatório final. Nesta fase, realiza-se uma análise quantitativa do

[26] Disponível em: < http://www.niee.ufrgs.br/eventos/RIBIE/2006/ponencias/art082.pdf> . Acesso em: 07 Fev. 2010

[27] Projeto que envolve várias universidades, é baseado na interação entre práticas de sala de aula, experimentação, investigação e contribuição de produtores. Disponível em :<http://www2.fpce.ul.pt/projectos/pedactice/ >.Acesso em: 07 Fev. 2010

[28] Disponível em :<http://www.nce.ufrj.br/sbie2003/publicacoes/paper38.pdf >. Acesso em: 07 Fev. 2010

software em relação às heurísticas pedagógicas e de usabilidade. (Atayde et.al, 2003)

Godoi (2009) realizou uma pesquisa sobre as características de outros métodos, sistemas e ferramentas, além dos citados acima, e concluiu que, independente da escolha do método, todos os usuários do software (desenvolvedor, gerente, aluno, professor e mantenedor) devem estar envolvidos no processo de avaliação da qualidade. Godoi (2009) destaca que nem todos os métodos envolvem o usuário durante o processo e que é necessário elaborar instrumentos que avaliem a usabilidade. Para demonstrar os métodos que envolvem o usuário, Godoi elabora o quadro abaixo (Figura 1), onde as técnicas agrupadas no item “sim” envolvem a participação do usuário.

não	sim	
●		CSEI
	●	MAEP
●		PROINFO
●		TICESE
●		Cronje [1998]
●		Hanna et al. [1997]
●		Squires & Preece [1999]
	●	Reeves & Harmon [1996]
		ASE
●		ESEF
●		PCEM
●		SEF
●		SK
	●	MAQSE
	●	MAQSEI
	●	Modelo JIGSAW
	●	IAQSEM
●		PEDACTICE
●		CASE
●		MEMI
●		MEDA
	●	SASE
	●	SOFTMAT
14	**8**	**Total**

Figura 1 - Apresentação de exemplo de aplicação com usuários (GODOI, 2009)

Godoi (2009) descreve que estes métodos combinam critérios pedagógicos com os critérios ergonômicos. Os critérios ergonômicos mais freqüentes são: “controle do utilizador, documentação/ materiais de apoio, feedback imediato, flexibilidade, funcionalidade geral, gestão de erros, identificação do software, legibilidade, qualidade nas opções de ajuda e usabilidade da interface. Já os menos freqüentes são: ações explícitas do usuário, amenidade de diálogo, apresentação da informação, armazenamento de informação, conforto áudio/ visual, correspondência do software com o mundo real, desempenho do usuário, estética, fidelidade navegacional, organização da informação, ramificações de conteúdo, segurança, tempo de exposição de telas, trabalho do usuário e uso de ilustrações/ animações.” (Godoi, 2009, p.87)

A maior parte dos métodos que apresentam a ergonomia, usabilidade e design da interface gráfica como critérios de avaliação, indicam como técnica de inspeção o uso de checklist ou questionário, mas Preece et al. (2005) e Chapanis (1996) afirmam que a aplicação de questionário é uma técnica estabelecida para a coleta de dados e que o avaliador não deve ser fiel aos resultados obtidos, e as avaliações da usabilidade não devem se basear somente no uso de checklist. Somerville (2007) também chama a atenção para pesquisas que priorizam somente dados quantitativos, pois estes nem sempre podem ser considerados para descrever atividades humanas. Portanto, é recomendável uma combinação de técnicas para avaliar o software.

Preece et al. (2005) citam que existem muitas técnicas de avaliação e estas são categorizadas da seguinte forma: observar o usuário, solicitar opiniões dos usuários, solicitar opiniões dos especialistas, testes com usuários e modelagem do desempenho das tarefas realizadas por usuários.

Moraes (1992) afirma que observar o usuário significa identificar alguma informação sobre o comportamento do mesmo em relação a alguma situação ou acontecimento, mas é impossível “observar muitas coisas ao mesmo tempo. Por isso, uma das condições fundamentais para a observação é limitar e definir o que se deseja observar.” (Moraes, 1992, p. 318)

Para solicitar a opinião dos usuários sobre um produto, Preece et al. (2005) recomendam a utilização das técnicas entrevistas, questionários e grupos de foco.

Solicitar a opinião de especialista consiste em uma forma de avaliação preditiva[29], onde os especialistas realizam tarefas específicas do usuário-alvo e são guiados por heurísticas para identificar problemas na interface.

[29] PREECE, J. et al. Design de interação homem-computador. Porto Alegre: Bookman, 2005. p. 363

Os testes com usuários são realizados com usuário-alvo, o objetivo é coletar dados sobre o desempenho do mesmo. E por fim, a modelagem do desempenho das tarefas realizadas é utilizada para prever a eficácia de uma interface e análise do desempenho. Preece et al. (2005) destacam o uso das técnicas de modelo de GOMS[30] e Keystroke[31] para esta etapa da avaliação.

A avaliação da interface com o usuário não é um processo simples e não está a prova de falhas (Somerville, 2007). As empresas de desenvolvimento sabem da importância da aplicação da usabilidade para elaborar o software de qualidade, mas a desinformação, o custo e o tempo gasto para a execução são relativamente altos. Por isso, é cada vez mais freqüente encontrar métodos, técnicas, ferramentas, etc. que prometem levantar rapidamente os problemas de usabilidade de um software educacional. Mas, se os anseios do usuário com a interface não forem investigados e compreendidos não será possível garantir um software de qualidade.

[30] GOMS é o termo utilizado para objetivos (goals), operador (operator), métodos (methods) e regras de seleção (selection rules), este método foi uma tentativa de modelar o conhecimento e os processos cognitivos envolvidos quando o usuário interage com o sistema. (Preece et al.,2005. p. 471)

[31] Keystroke fornece previsões numéricas reais do desempenho do usuário. As tarefas podem ser comparadas no que diz respeito ao tempo levado para realizá-las. (Preece et al.,2005. p. 472)