

## Avaliação de Softwares Educacionais: Critérios para seleção de Softwares educacionais para ensino de Matemática

Jeferson Fernando de Souza Wolff<sup>1,2</sup>



**RESUMO** - Este artigo apresenta o resultado da elaboração de uma ficha para a avaliação de *softwares* educacionais de Matemática, organizada pelo grupo de pesquisa *Softwares Livres: Uma rede para o ensino e aprendizagem de Matemática na Região Carbonífera* da Ulbra São Jerônimo - curso de Matemática. Este material está fundamentado na teoria de Aprendizagem Significativa de Ausubel. A análise de um *software* que tem como objetivo a sua utilização no ensino deverá levar em consideração, além dos aspectos técnicos, como por exemplo, sistema operacional e *hardwares* necessários para a utilização do *software* de maneira eficiente, mas também a análise pedagógica, a qual se destina principalmente a conhecer o *software* como uma ferramenta educacional e não apenas para realização de tarefas. Dentro deste foco pedagógico, descrevemos vários itens, os quais consideramos com fundamentais na análise de *software* educacional, que vão desde que tipo de *software* (tutorial, simulação...) até a identificação dos *subsunçores* que os alunos devem possuir para a utilização do material.

**Palavras-chave:** Matemática, Softwares educacionais, Aprendizagem Significativa.

**ABSTRACT** - This paper presents the development of a plug for the evaluation of educational software for mathematics, organized by the research group Free Softwares: A network for teaching and learning mathematics in the Coal Region Ulbra St. Jerome - Mathematics course. This material is based on the theory of Ausubel Meaningful Learning. The analysis of software that aims to be used in teaching should take into account both the technical aspects, such as operating system and hardware needed to use the software efficiently, but also the pedagogical analysis, which mainly intended to meet the software as an educational tool, not just for assignments. Within this pedagogical focus, describe multiple items, which we consider fundamental to the analysis of educational software, ranging from what kind of software (tutorial, simulation,...) to identify subsumers that students must possess to use the material .

**Keywords:** Mathematics, educational software, Meaningful Learning.

1 – Projeto Financiado pela Universidade Luterana do Brasil.

2 – Professor do Curso de Matemática da Universidade Luterana do Brasil – ULBRA/São Jerônimo, RS, Brasil.

**Dados para correspondência**

Jeferson Fernando de Souza Wolff  
Rua João Pessoa, nº 361, Centro,  
Triunfo, RS, Brasil.  
E-mail: jefersonwolff@terra.com.br

Recebido em: 10/01/2008.  
Revisado em: 19/03/2008.  
Aceito em: 27/06/2008.

**Área:** Metodologias e estratégias de ensino aprendizagens.

## INTRODUÇÃO

O computador hoje é um objeto de fascinação, tanto para alunos quanto para professores. O contágio, pela busca da utilização de computadores na educação, inclui todos os níveis de ensino, desde as séries iniciais até o nível superior, desde escolas particulares até as públicas. Em muitos casos a utilização da informática no ensino depende apenas do preparo e da qualificação do professor, pois laboratórios de informáticas existem.

Porém, antes de qualquer análise de softwares, é fundamental a reflexão sobre o uso da informática nas escolas: a sua influência no processo da aquisição do conhecimento sob o ponto de vista da Aprendizagem Significativa. Assim, para a implantação de aulas com a utilização da informática, antes de qualquer coisa o professor deve ter bem claro quais são os objetivos que pretende alcançar, para que o uso da mesma seja de forma responsável e com potencialidades pedagógicas verdadeiras, não sendo apenas utilizados como máquinas de programas agradáveis e divertidos.

Conforme VIEIRA, “o uso do computador na educação tem como objetivo promover a aprendizagem dos alunos e ajudar na construção do processo de conceituação e no desenvolvimento de habilidades importantes para que ele participe da sociedade do conhecimento e não simplesmente facilitar o seu processo de aprendizagem.”

Este artigo apresenta quais são os critérios que estão sendo utilizado pelo grupo de pesquisa do curso de Matemática ULBRA São Jerônimo, para análise de softwares educacionais. Esta análise consiste em relatar em uma ficha os aspectos técnicos e pedagógicos dos softwares, sob a luz da teoria da Aprendizagem Significativa.

Esta avaliação é a primeira etapa do projeto “*Software Livres: Uma rede para o ensino e aprendizagem de Matemática na Região Carbonífera*”, que se constituirá em analisar e classificar os softwares para ensino e aprendizagem de Matemática, sendo as demais etapas dirigidas à elaboração de material didático para a utilização dos softwares; e a última etapa será de oficinas destinadas à formação e capacitação de professores para que possam inserir a informática em suas aulas.

O que se pretende com a primeira etapa é uma análise criteriosa dos softwares selecionados, e que ela possa contribuir para apontar que tipo de proposta pedagógica o software analisado poderá ser melhor aproveitado. Além disso, esta ficha de avaliação serve como uma sugestão de análise de softwares educacionais para os professores que pretendem utilizar a informática como meio para a aprendizagem.

### 1. Softwares educacionais e a Aprendizagem Significativa

Todo o projeto, desde a avaliação até as oficinas para formação de professores, está baseado na Teoria da Aprendizagem Significativa de Ausubel. Esta teoria aborda que não há Aprendizagem Significativa se o conteúdo a ser aprendido não se relacionar com o *subsunçor* do aprendiz, ou seja, para que o novo conceito seja aprendido, este deverá se relacionar com os conhecimentos que o aluno já possui, desta

forma o professor deverá sempre verificar quais são os *subsunçores* necessários para que ocorra uma Aprendizagem Significativa. Caso não exista o *subsunçor* necessário para a aprendizagem desse novo conceito, o aluno terá o apenas o que Ausubel descreve como Aprendizagem Mecânica, uma aprendizagem só para o momento, sem relação nenhuma com os conceitos existentes no cognitivo do aluno.

Então, uma das etapas fundamentais na análise dos *softwares* é a identificação de quais são os *subsunçores* necessários que cada *software* exige para a sua utilização, para com isso obter o melhor aproveitamento. Caso não se realize esta análise, o que poderá ocorrer é que um aluno venha a utilizar um *software* e não consiga relacionar o conteúdo o qual o *software* se propõe a desenvolver, sendo desta forma em vão a utilização do mesmo. O que irá ocorrer, neste caso, é que o aluno apenas aprenderá a dar comandos exigidos pelo *software*, tendo talvez uma aprendizagem do *software*, de como este funciona e opera, mas sem que tenha conseguido relacionar os conteúdos Matemáticos apresentados pelo mesmo. Desta forma, em vão a tentativa de utilização de *software* para ensino de Matemática ou para qualquer outra disciplina; o professor apenas irá ensinar a utilizar o *software*, o qual o aluno poderá aprender ou não, mas em nada, ou quase nada, irá crescer quanto aos objetivos principais, que é a aprendizagem do conteúdo.

No caso de não haver relação do conteúdo abordado pelo *software* com o *subsunçor* do aluno, este poderá aprender a utilizar o *software* de maneira correta, conhecer os comandos e as ferramentas disponibilizados pelo *software*, porém, não saberá interpretar e analisar os resultados, ou seja, terá apenas uma Aprendizagem Mecânica.

Então, é de fundamental importância o planejamento, por parte do educador, de quais são os objetivos a serem atingidos com a utilização do software, para com isso analisar quais são os *subsunçores* necessários que o aluno deve possuir para que venha a ter uma Aprendizagem Significativa.

Desta forma, o que pretendemos com este artigo é discutir alguns critérios para a avaliação de *softwares* educacionais, à luz da Teoria da Aprendizagem Significativa.

Sabe-se que há diferentes maneiras de classificar os *softwares* educacionais, uma das quais, por exemplo, consistiria em categorizar de acordo com a natureza do *software* e suas propriedades. Uma outra forma seria classificar pela finalidade para a qual o *software* é utilizado no processo educacional, assim como informação e reforço, entre outros. A maneira através da qual pretendemos apresentar a classificação de *software* levará em conta as características técnicas, a finalidade e principalmente os objetivos educacionais.

## 2. Metodologia

Inicialmente foi realizada pesquisa sobre diversas formas e normatizações referentes à classificação de softwares, a qual serviu para que fosse possível apontar alguns caminhos, a fim de se ter um referencial de análise de *software* educacional.

Para avaliar a qualidade dos softwares, foi elaborada uma breve ficha de avaliação que está subdividida em quatro grupos: Identificação, Características Técnicas, Aspectos Pedagógicos e Resumo/Sugestões. A seguir passaremos a descrever cada um desses itens.

## 2.1 Identificação

O primeiro tópico abordado na ficha de avaliação é a identificação do *software*, e tem como objetivo conhecer a origem do mesmo. Nesse item é indicado que seja preenchido com os seguintes aspectos: o nome do *software*, o autor, a instituição ou empresa na qual o *software* foi desenvolvido, caso não possua vínculo com nenhuma instituição, apenas colocar sem vínculo, o ano da publicação do *software*, a versão, o local onde foi elaborado (País e Município), local de disponibilização (caso *internet* colocar o endereço), quais são os objetivos apresentados pelo autor quando da utilização do *software* e um breve resumo descrevendo as principais características do *software*.

## 2.2 Características Técnicas

Este tópico tem como objetivo conhecer as características técnicas referentes aos *softwares*. Ele está subdividido em 12 itens que abordam desde a classificação de *software* até aspectos necessários para a sua utilização, como, por exemplo, o sistema operacional exigido pelo programa. A seguir passaremos a descrever cada um dos itens.

O primeiro item abordado refere-se à classificação dos *softwares*, e está dividido em quatro grupos, de acordo com as características de licença, conforme apresentado a seguir:

*Software Livre*: denominamos como um programa que tem a liberdade de ser usado, copiado, modificado e redistribuído gratuitamente.

*Freeware*: *software* proprietário que é disponibilizado gratuitamente, mas não pode ser modificado.

*Shareware*: é o *software* disponibilizado gratuitamente por um período de tempo ou com algumas funções limitadas, como por exemplo, a impossibilidade de salvar o trabalho realizado, mas que implica no posterior pagamento para a sua licença.

*Demo*: é uma versão de demonstração do *software*. É possível usar o programa com apenas algumas funções disponíveis.

O segundo ponto abordado no tópico características técnicas é quanto ao uso do *software*, que pode ser classificado como genérico ou específico.

*Software* genérico é o que pode ser utilizável em qualquer disciplina, ou em outras atividades não educativas. São exemplificativos deste tipo de produto, os processadores de texto, as planilhas de cálculo etc.

*Software* específico. Como o nome indica, trata-se de um tipo de *software* concebido com a finalidade de ser usado no ensino, e nomeadamente na aprendizagem de temas concretos. São exemplos

deste tipo de produto os programas de simulação usados no ensino de temas de ciência e de exercícios de matemática etc.

O terceiro item refere-se se existe ou não *help*, de alguma ajuda para o usuário, durante a utilização do programa. Caso exista ajuda, é levantado o questionamento se essa pode ser acessada a qualquer momento, sem prejuízos de navegação. Além disso, analisa-se se o *help* aborda os itens fundamentais para a utilização, ou seja, se o *help* apresenta os itens básicos para a utilização.

Outro item extremamente importante é qual o sistema operacional que o *software* exige para a sua execução. Essa análise é imprescindível, pois podemos optar por um determinado *software* de acordo com seus objetivos e conteúdo abordados, mas se na escola onde se vai trabalhar com os alunos o sistema operacional for incompatível, a utilização dessa forma tornará impossível a aula prevista.

O quinto item aborda o idioma que o *software* pode ser utilizado. Essa análise se torna importante a medida que um aluno que esteja utilizando o *software* e pretende utilizar a Ajuda. Se esta for de idioma que não faz parte do *subsunçor* do aluno, em nada irá contribuir. O mesmo é válido para os comandos: se estes forem incompreensíveis pelos alunos, a única aprendizagem que terão será a Mecânica.

O sexto item está relacionado com a facilidade de leitura e clareza dos comandos. Se os comandos a serem executados não estiverem claros, a aprendizagem poderá se tornar comprometida, pois se aluno não sabe o significado de um determinado comando, ou seja, se não sabe o que está fazendo, provavelmente também não estará tendo uma Aprendizagem Significativa.

Os itens sete e oito abordam as características referentes às necessidades de *hardware* ou de algum *software* para a execução, ou seja, verificar se o mesmo é auto-executável. Essa análise, assim como o sistema operacional, é imprescindível na hora da utilização do *software*. Por exemplo, se o programa possui alguma animação e os computadores nos quais serão desenvolvidas a atividades não apresentarem placa de vídeo, será em vão a tentativa de realizar a atividade com os alunos.

No item nove a abordagem dada refere-se à facilidade da utilização do *software*. Nesse item objetiva-se verificar se a utilização é simples e se há possibilidade de desfazer alguma tarefa ou seguir a execução, sem que se tenha que reiniciar o programa.

O décimo item trata de analisar a existência de áudio e animações no *software*. Esse item é importante para dar suporte ao item sete, verificando a necessidade de *hardware* ou *software* especial para a execução do som ou animação. Além disso, nesse item temos a preocupação de verificar se não há excesso de informação, deixando o seu uso com poluição visual ou sonora, desvirtuando do objetivo principal: que é a aprendizagem do aluno.

No décimo primeiro item é verificado se existe a possibilidade da utilização do *software* em rede, o que poderá ser uma necessidade dependendo dos equipamentos que os professores possuem para realizar as suas aulas.

E o último item refere-se à possibilidade de reiniciar uma atividade a partir do ponto onde foi encerrada, ou seja, iniciar a utilização do *software* a partir de qualquer ponto de interesse do aluno. *Softwares*

que não possuem esta possibilidade, acabam por tornar-se cansativos, desmotivando os alunos durante a sua execução.

### 2.3 Características Pedagógicas

As características pedagógicas dizem respeito ao ambiente educacional, pertinência ao programa curricular e aos aspectos didáticos. A análise dos itens pedagógicos está baseado na Teoria da Aprendizagem Significativa de Ausubel. Ao total são doze itens, os quais passaremos a descrever a seguir.

O primeiro item refere-se ao tipo de *software*, que conforme Vieira pode ser classificado em algumas categorias de acordo com os objetivos pedagógicos. São eles: Tutoriais, exercícios e práticas, multimídia e internet, simulação e modelagem e jogos.

**Tutoriais:** Caracterizam-se por transmitir informações de forma organizada, como se fossem um livro animado, um vídeo interativo ou um professor eletrônico. A informação é apresentada ao aluno seguindo uma seqüência, em que pode ser escolhida a informação desejada. O computador assume o papel de uma máquina de ensinar, onde a interação com o aluno consiste apenas na leitura da tela ou execução de comandos fornecidos pelo material. A interação com o aluno é muito baixa. Em um tutorial o aluno é um mero receptor de informações.

**Exercícios e Práticas:** Enfatizam a apresentação das lições ou exercícios. A ação do aluno se restringe em virar a página de um livro eletrônico ou realizar exercícios, cujo resultado pode ser avaliado pelo próprio computador. As atividades exigem apenas o fazer, o memorizar informação, não importando a compreensão do que se está realizando. Com esse tipo de *software* o aluno já interage fornecendo respostas aos problemas propostos, mas o aluno não tem a possibilidade de criar situações problemas para serem resolvidas pelos computadores, ou seja, se limita aos problemas fornecidos pelo programa.

**Aplicativos:** São programas voltados para aplicações específicas, como processadores de texto, planilhas eletrônicas, e gerenciadores de banco de dados. Embora não tenham sido desenvolvidos para uso educacional, permitem interessantes usos em diferentes ramos do conhecimento. Estes *softwares* são classificados como genéricos, pois podem ser utilizados por outras disciplinas, ou seja, não foi criado com o objetivo específico para uma disciplina ou educacional.

**Multimídia e Internet:** O uso de multimídia oferece muitas possibilidades de combinações com textos, imagens e sons e a ação do aluno se resume em escolher opções oferecidas pelo *software*. Após a escolha, o computador apresenta a informação disponível e o aluno pode analisar a mesma. Às vezes o *software* pode oferecer ao aluno, também, a oportunidade de selecionar outras opções e navegar entre elas. É muito semelhante ao tutorial.

Dessa forma, o uso de multimídia pronta e *Internet* são atividades que auxiliam o aprendiz a adquirir informações, mas não a compreender ou construir conhecimentos com a informação obtida.

Em uma segunda situação o aluno possui a possibilidade de elaborar. Dessa forma, ele seleciona as informações em diferentes fontes e programas, construindo, assim, um sistema de multimídia que possibilita

ao aluno refletir sobre os resultados obtidos, compará-los com suas idéias iniciais e depurar em termos de qualidade, profundidade e significado da informação apresentada.

O tipo de execução do sistema de autoria se assemelha ao processador de texto, pois executa uma sucessão de informações, e não a própria informação. Ele também não registra o processo que o aluno usa para montar o *software* multimídia.

**Simulação e Modelagem:** Para que um fenômeno possa ser simulado no computador, basta que um modelo desse fenômeno seja implementado no equipamento. Assim, a escolha do fenômeno a ser desenvolvido é feito *a priori* e fornecido ao aluno.

A simulação pode ser fechada ou aberta. Fechada quando o fenômeno é previamente implementado no computador, não exigindo que o aluno desenvolva suas hipóteses, teste-as, analise os resultados e refine seus conceitos. Apenas simulará aquilo que já vem previamente definido pelo programa.

A simulação pode ser aberta quando fornece algumas situações previamente definidas e encoraja o aprendiz a elaborar suas hipóteses, que deverão ser validadas por intermédio do processo de simulação no computador. Nesse caso, é possibilitada a vivência de situações difíceis ou até perigosas de serem reproduzidas em aula, permitem desde a realização de experiências químicas ou de balística, dissecação de cadáveres, até a criação de planetas, que o próprio aluno pode montar conforme o seu interesse.

O modelo do fenômeno é montado pelo aluno, que utiliza recursos de um sistema computacional para implementar esse modelo no computador. Esse tipo de *software* exige um certo grau de envolvimento na definição e representação computacional do fenômeno e, portanto, cria uma situação bastante semelhante à atividade de programação.

Mas a diferença entre modelagem e programação está no nível de possibilidades de implementação da simulação. A modelagem vai depender do programa que se está utilizando, tendo as suas limitações. Já na programação o aluno pode implementar a simulação que desejar, dependendo somente da linguagem de programação utilizada.

**Jogos:** Normalmente são constituídos com a finalidade de desafiar e motivar o aluno, envolvendo-o em uma competição com a máquina ou colegas. Os jogos permitem interessantes usos educacionais, principalmente se integrados a outras atividades.

Porém, os jogos têm a função de envolver o aprendiz em uma competição, e essa competição pode dificultar o processo da aprendizagem, uma vez que, enquanto estiver jogando, o interesse do aprendiz está voltado para ganhar o jogo, e não em refletir sobre os processos e estratégias envolvidos no mesmo. Sem essa consciência é difícil uma transformação dos esquemas de ação em operação, alerta Valente.

O segundo tópico refere-se aos conteúdos matemáticos abordados. O objetivo desta análise é de conhecer quais são os conteúdos abordados, descrevendo-os. Essa análise torna-se fundamental por dois motivos: primeiro, para reconhecermos quais são os *subsunçores* necessários para a utilização da programa; e, segundo, o nível da abordagem do conteúdo.

Identificados os conteúdos, passamos para a análise seguinte, que se refere à interdisciplinaridade, ou seja: se os conteúdos abordados possuem relação com outras áreas do conhecimento.

Os itens quatro e cinco estão relacionados: o primeiro refere-se a qual faixa etária ou nível cognitivo o programa está destinado, e o segundo se a interface está adequada ao nível cognitivo. O objetivo desta análise é identificar se os conteúdos abordados estão adequados ao nível cognitivo indicado pelo fabricante/autor do *software*, para que não ocorra uma abordagem muito superficial ou fora da realidade do aluno, e se a interface é acessível ao aluno de acordo com o seu nível cognitivo.

O item seis refere-se à análise do vocabulário utilizado pelo programa. Conhecida a faixa etária para a qual o programa destina-se, o passo seguinte é verificar se o vocabulário está de acordo com nível do aluno, ou seja, se utiliza ou não vocabulários que não estão relacionados com o cotidiano do aluno. Caso se confirme esta utilização, deve-se identificar quais são os termos utilizados para que possam ser explicados aos alunos quando forem utilizar o programa, caso contrário, em vão será a tentativa de realizar a Aprendizagem Significativa.

Além do vocabulário, temos também que analisar se os objetivos e as etapas do programa são claros e se estão no mesmo nível em que o *software* está direcionado. Esse ponto é analisado no item sete da ficha de avaliação. O objetivo desse item é de verificar se o aluno terá facilidade em utilizar o programa sem que se perca ou não compreenda as etapas atingidas ou a serem atingidas, comprometendo dessa forma os objetivos educacionais.

Os itens oito e nove possuem certa relação, os dois referem a forma abordada pelo programa quando o aluno comete algum erro. No primeiro item é verificado se o programa incentiva aluno a continuar executando as tarefas caso cometa um erro, ou se desmotiva, causando constrangimento. Uma situação comum em apresentações é o resultado da resposta ser abordado em níveis: muito bom, bom, ruim ou péssimo. Quando o aluno recebe um *feedback* que o seu resultado foi péssimo ou ruim sem sequer serem apresentadas explicações referentes ao erro, isso pode causar uma desmotivação, uma sensação de incapacidade por parte do aluno. O item seguinte, nono, trata da situação de erro, quando é cometido, se apresenta um reforço referente ao conteúdo, ou informação clara de onde foi o equívoco do aluno. Esse ponto é importante, pois em muitas situações os alunos acham que sua resposta está correta, mas o programa apresenta resultado contrário, sem maiores explicações. Com isso, perde-se uma grande oportunidade de recuperar os conceitos que ainda não foram absorvidos pelo aluno.

No décimo item é analisado se o programa apresenta múltiplos caminhos para a mesma solução. Esse é outro ponto importante, pois para a mesma resposta ou elaboração de simulação cada aluno poderá interpretar e organizar suas idéias de formas diferentes, mas chegando ao mesmo resultado. Se não forem considerados esses múltiplos caminhos, o aluno poderá achar que cometeu algum erro durante a construção de sua resposta, prejudicando a aprendizagem.

O nível de aprendizagem é abordado no décimo primeiro item, e para o mesmo foram considerados três níveis: Sequencial, Relacional e Criativo, conforme Vieira.



Para o *software* que se apresenta no nível seqüencial, temos um aluno com aprendizado passivo, ou seja, mero receptor de informação. O foco principal deste nível está na preocupação de só transferir a informação; o objetivo do ensino é apresentar o conteúdo para o aprendiz e ele, por sua vez, deverá memorizá-lo e repeti-lo quando for solicitado.

Para o nível relacional, o *software* tem como objetivo a aquisição de determinadas habilidades, permitindo que o aluno faça relações com outros fatos ou outras fontes de informação. A ênfase é dada ao aluno e a aprendizagem se processa somente com a interação do aluno com a tecnologia.

Ao nível criativo está associada a criação de novos esquemas mentais, possibilitando a relação de pessoas e tecnologias compartilhando objetivos comuns.

O penúltimo item aborda o nível de interação. Busca conhecer o nível de interatividade com o aluno. Quanto maior o nível de interatividade, mais o aluno possui a possibilidade de construir seu conhecimento; sempre, claro, considerando quais são os seus *subsunçores*. Quanto menor o nível de interatividade, mais o aluno é um mero receptor de informação, ou seja, menor é a sua participação na construção do seu conhecimento.

E o último item da análise pedagógica consiste em descrever quais são os *subsunçores* necessários para que aluno tenha uma aprendizagem significativa, ou seja, conhecer quais são os conceitos necessários que o aluno deve possuir para que tenha uma Aprendizagem Significativa. Com esta análise o professor terá condições de verificar se o *software* está adequado aos conteúdos que deseja trabalhar, caso contrário o professor terá de trabalhar os subsunçores necessários ou optar por outro *software*.

## 2.4 Conclusões/Recomendações/Sugestões

Neste último tópico, o avaliador irá descrever as suas conclusões, recomendações e algumas sugestões de utilização do *software* no ensino. As recomendações e sugestões têm como objetivo apresentar algum item específico do programa que não foi descrito anteriormente e que o avaliador considera fundamental para a utilização do mesmo. A conclusão levará em consideração todos os itens analisados e, a partir da mesma, verificar a coerência para a utilização em aula. Claro que a utilização em aula irá depender de quais são os objetivos a serem atingidos pelo professor.

### 3. Ficha de avaliação

Conforme já descrito anteriormente, apresentaremos a nossa sugestão de ficha para a avaliação dos aspectos Técnicos e Pedagógica.

1 Identificação do Software
1.1 Nome do software:
1.2 Autor (s):
1.3 Empresa/Instituição que elaborou:
1.4 Ano:
1.5 Versão:
1.6 Local:
1.7 Local de disponibilização
1.8 Objetivo:
1.9 Resumo:

2. Características Técnicas do Software
2.1 O <i>software</i> se classifica melhor como:
2.1.1 <i>Software</i> livre
2.1.2 <i>Freeware</i>
2.1.3 <i>Shareware</i>
2.1.4 Demo
2.2 Classificação quanto ao uso:
2.2.1 Genérico
2.2.2 Específico
2.3. Possui ajuda ( <i>help</i> )?
2.3.1 Este é de fácil acessibilidade em qualquer momento de utilização do <i>software</i> ?
2.3.2 O <i>help</i> abrange os tópicos principais de utilização do programa?
2.4 Qual o sistema operacional utilizado pelo <i>software</i> ?
2.4.1 <i>Windows</i>
2.4.2 <i>Linux</i>
2.4.3 Outros. Quais?
2.5 Idioma do <i>software</i> ? Existe mais de uma possibilidade?
2.6 Facilidade de leitura e clareza dos comandos?
2.7 É auto-executável?
2.8 Especifica a necessidade de algum <i>hardware</i> ou <i>software</i> ? Qual?
2.9 É de fácil utilização?
2.10 Possui animações e som?
2.10.1 Caso haja a utilização de som e animação, há excesso de utilização dessas mídias, causando uma poluição sonora e visual?
2.11 É possível a sua utilização em rede?
2.12 Permite que “sessões” interrompidas sejam reiniciadas a partir do “ponto de parada”, se assim o desejar?

3. Aspectos Pedagógicos	
3.1 Tipo de <i>software</i>	
3.1.1 Tutorial	
3.1.2 Exercícios e prática	
3.1.3 Aplicativo? Qual?	
3.1.4 Multimídia – <i>internet</i> ?	
3.1.4.1 Pronto	
3.1.4.2 Sistema de autoria	
3.1.5 Simulação	
3.1.5.1 Aberto	
3.1.5.2 Fechado	
3.1.6 Modelagem	
3.1.7 Jogos	
3.2 Conteúdos abordados	
3.3 Possui relação com outros conceitos matemáticos ou de outras disciplinas	
3.4 Faixa etária aconselhável	
3.5 Interface adequada para a faixa etária.	
3.6 Vocabulário adequado para a compreensão do aluno	
3.7 Se os objetivos do <i>software</i> e as etapas a serem atingidas são claros e estão no nível de compreensão do aluno;	
3.8 Quando o aluno erra o <i>feedback</i> é agradável ou constrangedor.	
3.9 Caso o aluno erre em termos conceituais, o <i>software</i> apresenta reforço ou informação, buscando apresentar ao aluno quais foram os erros cometidos?	
3.10 Apresenta múltiplos caminhos para a solução do mesmo problema?	
3.11 Nível de aprendizagem	
3.11.1 Seqüencial	
3.11.2 Relacional	
3.11.3 Criativo	
3.12 Qual o nível de interação que <i>software</i> apresenta?	
3.12.1 Alta interatividade	
3.12.2 Média Interatividade	
3.12.3 Baixa Interatividade	
3.13 Quais são os <i>subsunçores</i> necessários para a aprendizagem significativa utilizando este <i>software</i> ?	

#### 4. Conclusões/ Recomendações/ Sugestões.

### 3. Considerações finais

Para a implementação da informática em aula, o primeiro passo a ser dado é o interesse do professor pela utilização dessa tecnologia em sua aula. Se este não considerar necessário a implementação de momentos que utilizem este recurso no seu fazer pedagógico, em vão é qualquer análise ou descrição de *softwares* educacionais. Mas para os professores que se sentem comprometidos com a utilização dessa tecnologia em sala de aula, é fundamental a análise do *software* que pretende utilizar.

Com a ficha de análise implementada a partir do projeto *Software Livres: Uma rede para o ensino e aprendizagem de Matemática na Região Carbonífera*, fica como sugestão para o professor que a utilize-a para auxiliar na adequação do *software* aos objetivos da sua aula.

Como segunda etapa do projeto está prevista a análise e a elaboração de sugestão de utilização dos *softwares* em atividades educacionais. Os resultados obtidos serão disponibilizados na Ulbra, Capus São Jerônimo.

## REFERÊNCIAS

- AUSUBEL, D. P.; NOVAK, J. D.; HANESIAN, H. *Psicologia educacional*. Rio de Janeiro: Interamericana, 1980. 625 p.
- CAMPOS, Gilda H. Bernardino de, Ana Regina Rocha – *Avaliação da qualidade de Software Educacional* – Em Aberto, Brasília, ano 12, n. 57, jan./mar. 1993.
- Gomes A. S., Castro Filho J. A., Gitirana V., Spinillo A., Alves M., Melo M., Ximenes J.: *Avaliação de software educativo para o ensino de matemática*, WIE'2002, Florianópolis (SC).
- MOREIRA, M. A. *Teorias de aprendizagem*. São Paulo: Ed. Pedagógica e Universitária, 1999. 195 p.
- MOREIRA, M. A. *Aprendizagem significativa*. Brasília: Editora Universidade de Brasília, 1999. 130 p.
- OLIVEIRA, Silvia Sales de; *Avaliação de Software Educativo para o Ensino de Matemática - O caso das Estruturas Aditivas* – <<http://www.cin.ufpe.br/~asg/producao/wie2002.pdf>> acesso em: 12 maio de 2008.
- SILVA, B.H.A.M. (1995). *Avaliação de Softwares Educacionais*. Atas do VII Congresso Internacional Logo, Porto Alegre, pp. 480-485.
- SILVA, Dirceu - *Informática e Ensino: visão crítica dos softwares educativos e discussão sobre as bases pedagógicas adequadas ao seu desenvolvimento* <<http://penta2.ufgrs.br/edu/edu3375/leciona.html>> acesso em: 10 abril de 2008
- VALENTE, J. Armando - *Análise dos diferentes tipos de Softwares usados na Educação* - NIED – UNICAMP.
- VIEIRA, Fábila Magali Santos - *Avaliação de Software Educativo: Reflexões para uma Análise Criteriosa*. <<http://edutec.net/Textos/Alia/MISC/edmagali2.htm>> acesso em: 12 maio de 2008.

Wolff, Jeferson Fernando de Souza. Avaliação de *Softwares* Educacionais: Critérios para seleção de *Softwares* educacionais para ensino de Matemática. Revista Ciência e Conhecimento, v.3, n.1, p.1-14, junho/2008.