

**UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
CENTRO INTERDISCIPLINAR DE NOVAS TECNOLOGIAS NA EDUCAÇÃO
CURSO DE ESPECIALIZAÇÃO EM MÍDIAS NA EDUCAÇÃO**

ALEXANDRE CAVAGNI

**UM ESTUDO DE CASO SOBRE SOFTWARES VOLTADOS
PARA O ENSINO DA GEOMETRIA**

**Porto Alegre
2010**

ALEXANDRE CAVAGNI

**Um estudo de caso sobre softwares voltados para o ensino
da geometria**

Trabalho de Conclusão de Curso,
apresentado como requisito parcial para a
obtenção do grau de Especialista em
Mídias na Educação, pelo Centro
Interdisciplinar de Novas Tecnologias na
Educação da Universidade Federal do Rio
Grande do Sul – CINTED/UFRGS.

**Orientador(a):
BARBARA AVILA**

**Porto Alegre
2010**

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitor de Pós-Graduação: Prof. Aldo Bolten Lucion

Diretora do Centro Interdisciplinar de Novas Tecnologias na Educação: Profa. Rosa Maria Vicari

Coordenador(as) do curso de Especialização em Mídias na Educação: Profas. Rosa Vicari e Liane Margarida Rockenbach Tarouco

Bibliotecária-Chefe da Faculdade de Educação:

DEDICATÓRIA

Dedico esse estudo a todos os alunos do Colégio Estadual Alcebíades Azeredo dos Santos e a Direção da escola que muito contribuíram para a realização do trabalho.

AGRADECIMENTOS

Agradecimentos especiais à diretora Naira Veiga do Colégio Estadual Alcebíades Azeredo dos Santos.

RESUMO

Este trabalho apresenta Metodologia para Avaliação sobre a usabilidade de *Softwares* Educacionais no estudo de matemática do ensino médio. Neste contexto, os *softwares* avaliados foram: GEOGEBRA, POLY e CABRI, e as heurísticas que fundamentam a avaliação foram extraídas de Nielsen (1993). O estudo envolveu conhecimentos advindos dos anos de prática do autor, que vem atuando como professor no ensino médio e envolve também uma pesquisa realizada pelo mesmo sobre conceitos relacionados à usabilidade. Além disso, utilizou-se como subsídio a ISO 9241-11, onde se estabelece que a usabilidade corresponde à qualidade de uso que orienta e fornece parâmetros metodológicos visando a melhoria de processos de usabilidade de *software* educacional. A metodologia e as heurísticas foram validadas por meio de demonstrações e comparações entre os *softwares* de matemática em foco a geometria como conteúdo. Elas podem ser utilizadas em avaliações formativas, colaborando para que o desenvolvedor descubra aspectos negativos e positivos referentes aos *softwares*, permitindo ao professor uma visão geral sobre em que ponto a ferramenta pode se adequar aos seus objetivos pedagógicos.

Palavras-chave: Softwares de geometria, tecnologias educacionais, aprendizagem autônoma.

LISTA DE ABREVIATURAS E SIGLAS

ISO	Organização Internacional de Padronização
TIC	Tecnologias da Informação e Comunicação
UFRGS	Universidade Federal do Rio Grande do Sul

LISTA DE FIGURAS

Figura 1: Fonte: BARRETO (1999) p. 20

Figura 2: Esquema do conceito de usabilidade: Cláudia Dias (2007) p.25

Figura 3: Traçado de uma reta usando dois pontos no Software Geogebra. p.34

Figura 4: Traçado de uma reta usando vários pontos no Software Geogebra p.34

Figura 5: Traçado de segmentos usando vários pontos no Software Geogebra p.34

Figura 6: Criando uma esfera no software Poly p.38

Figura 7: Visualização dos comandos da página principal do Cabri p.41

Figura 8: Construção de vetores do Cabri p.41

LISTA DE TABELAS

Tabela 1: Características de qualidade de software da ISO/IEC FCD 9126-1(1998) p. 24

Tabela 2: Conceitos da ISO 9241-11(1998) p. 25

Tabela 3: Tabela 3 - Conceitos de Nielsen sobre Usabilidade p.

Tabela 4: Heurísticas de Nielsen (1998) p. 29

Tabela 5: Regras de Ouro de Shneiderman (1998) p. 30

Tabela 6: Adaptado da tese de dissertação de Hauber (2010) p. 43

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	7
LISTA DE FIGURAS	8
LISTA DE TABELAS	9
1 INTRODUÇÃO	11
1.1 Problemas	13
1.2 Objetivos.....	13
2 SOFTWARES EDUCACIONAIS	14
2.1 O que é Software Educacional.....	14
2.2 Conceito de Software Educacional.....	15
2.3 Tipos e Classificação de Alguns Softwares Educativos.....	16
2.4 Métodos de Avaliação de Softwares Educativos	18
2.5 O Uso das Tecnologias de Softwares Aplicadas no processo de aprender.....	20
3 USABILIDADE	22
3.1 Princípios que suportam a Usabilidade.....	22
3.2 Normas de Usabilidade.....	23
3.3 Características de Qualidade de Software da ISSO/IEC FCD 9126-1	24
4 HEURISTICAS DE NIELSEN E REGRAS DE OURO DE SHNEIDERMAN .	29
5 METODOLOGIA	32
5.1 Geogebra	33
5.2 Poly	36
5.3 Cabril 3D V2	39
5.4 Endreço Eletrônico de Alguns Softwares Educativos.....	45
6 CONSIDERAÇÕES FINAIS	47
REFERÊNCIAS.....	49

1– INTRODUÇÃO

As novas tecnologias que vêm sendo introduzidas nas escolas estaduais de ensino médio, como computadores, acesso à internet e *softwares* educacionais, têm levantado diversas questões a serem discutidas. Mas a que consideramos mais preocupante está relacionada com as novas dinâmicas da sala de aula, ao novo papel: do professor e do aluno. O estudo de *softwares* para micro computador como uma ferramenta didática e pedagógica na sala de aula como afirmam Lins e Gimenez (1977):

Essa concepção deve fazer parte da base para uma proposta para a educação matemática: álgebra, aritmética e geometria, vistas não como conteúdo justificado por sua própria existência, mas como instrumentos que participam da organização da atividade humana. Desta perspectiva, o estudo da matemática desprendido temporariamente de quaisquer problemas fora da matemática passa a ter um sentido diferente, o de estudar e aprimorar ferramentas de que dispõe, e nesse processo a matemática torna-se objeto e não mais ferramenta. (p.28)

Esta concepção de ensinar matemática sugerida propõe que os professores de matemática apresentem novas formas de aprendizado demonstrando novos conceitos e métodos de ensinar. Isso nos traz uma nova concepção e visualização dos conteúdos de geometria tornando esses mais acessíveis ao aluno, buscando uma real compreensão por parte do mesmo.

O aprendizado da Geometria e das Funções mostra-se dificultoso para muitos estudantes de matemática do ensino médio. Isso se deve, em parte, à dificuldade de visualizar os conceitos. Com o surgimento dos *softwares* educacionais, esta dificuldade foi amenizada (também em parte), uma vez que os *softwares* educacionais podem ser comparados ou vistos como materiais “concretos virtuais”, e ao mesmo tempo o aluno pode ter como dificuldade a visualização em uma terceira dimensão. (ALBUQUERQUE, 1999)

Entretanto, será que podemos afirmar que o material concreto ou jogos pedagógicos são realmente indispensáveis para que ocorra uma efetiva aprendizagem da matemática?

Pode parecer, à primeira vista, que todos concordem e respondam sim à pergunta. Mas isto não é tão simples de se responder. Um exemplo de uma posição divergente é colocada por Carraher & Schilemann (1988), ao afirmarem com base em suas pesquisas que:

não precisamos de objetos na sala de aula, mas de objetivos na sala de aula, mas de situações em que a resolução de um problema implique a utilização dos princípios lógico-matemáticos a serem ensinados" (p. 179). "Isto porque o material" "apesar de ser formado por objetivos, pode ser considerado como um conjunto de objetos 'abstratos' porque esses objetos existem apenas na escola, para a finalidade de ensino, e não tem qualquer conexão com o mundo do aluno (p. 180).

Para o aluno o concreto não significa os materiais manipulativos, mas as situações que ele tem que enfrentar socialmente. As colocações de Carraher & Schilemann (1998) nos servem de alerta: não podemos responder sim àquelas questões sem antes fazermos uma reflexão mais aprofundada sobre o assunto.

Os recursos de animação podem construir mover e permitir a observação sob vários ângulos das figuras geométricas e das funções os invariantes que aí aparecem e que correspondem às propriedades geométricas. Existem vários programas na internet de domínio livre aos quais os professores de matemática estão tendo acesso.

Os conceitos *softwares* educativos devem contemplar as propriedades, o que requer um preparo pedagógico e didático da proposta de trabalho.

Caso não aconteça essa proposta o aluno pode apresentar maior dificuldade no aprendizado da matemática, fazendo confusão entre os conteúdos. Neste sentido, a intenção deste estudo é de promover uma reflexão para que o professor de matemática repense o seu trabalho cotidiano, envolvendo práticas de ensino sobre geometria e funções. À medida que for usando os programas e se familiarizando com os mesmos poderá elaborar

suas próprias atividades a partir das sugestões aqui apresentadas, sempre zelando para não distorcer conceitos e propriedades do conteúdo por erros de interpretação da linguagem do programa, erro muito frequente. Devemos estar conscientes de que o programa é uma das ferramentas de mediação, e que em determinados momentos pode não ser a mais apropriada.

De acordo com Moraes (2000, pág. 74)

A realidade da pedagogia dos meios modernos, cuja interação professor-aluno-informação deverá levar o indivíduo a aprender a pensar, a aprender a antecipar, a aprender a cultivar o espírito crítico e criativo, para que ele possa sobreviver num mundo onde inúmeras informações estarão disponíveis, e que precisam ser criticamente avaliados, para serem transformadas em conhecimento.

Após ter observado alguns dos erros mais frequentes o professor poderá optar pela vantagem e a desvantagem do *software*. O professor deverá estabelecer quando é mais vantajoso para o ensino dos conteúdos e qual a dinâmica mais adequada.

1.1 – Problema

Os *softwares* Poly, Cabri e Geogebra atendem devidamente requisitos de usabilidade?

1.2 – Objetivos

- Analisar a usabilidade de *softwares* existentes com foco no ensino de geometria.
- Analisar e relatar aspectos positivos e negativos dos *softwares* em questão.

2 - SOFTWARES EDUCACIONAIS

Segundo Vesce (2008) um conjunto de instruções ordenadas que são entendidas e executadas pelo computador é chamado de *Software*. Os tipos principais de *softwares* são:

- ✓ Os sistemas operacionais (É um *software* básico que controla o funcionamento físico e lógico do computador).
- ✓ Os *softwares* aplicativos (É um *software* que executa os comandos solicitados pelo usuário).

Tipos diferentes de *software* contêm elementos dos *softwares* básicos e dos *softwares* aplicativos, mas são distintos e são definidos como *softwares* de rede, que permitem a comunicação entre si dos computadores e as linguagens de programação.

2.1 - O que é *Software* Educacional?

São várias as definições do que constitui um *software* educacional, que em certos momentos podem trazer dúvidas ao leitor. Visando esclarecer essa dúvida traremos aqui algumas perguntas sobre esse desse conceito.

Poderíamos citar que uma linguagem de programação¹ pode ser um *software* educacional?

- ✓ Um jogo pedagógico pode ser considerado um *software* educacional?
- ✓ Um programa utilizado pelos professores de matemática em suas aulas é um *software* educacional?

Segundo Valente (1999) não podemos responder com precisão a essas perguntas, pois isso decorre da falta de clareza sobre o que realmente é um

¹ Citamos os exemplos (visual basic, turbo c++)

software educacional, ou seja, se esse *software* está sendo utilizado para o fim pedagógico. O *software* pode não ter fim específico para a educação, mas pode ser utilizado na educação?

Quais são os critérios para que um determinado *software* seja considerado educacional?

- ✓ Que ele tenha sido feito sob a orientação da educação para desenvolver algum objetivo educacional, por exemplo, um estudo de geometria espacial.
- ✓ Que ele seja usado para algum objetivo educacional ou pedagogicamente defensável, qualquer que seja a finalidade com que tenha sido criado.
- ✓ Que tenha este um objetivo educacional e pedagógico fazendo interação entre educador e aluno.
- ✓ Que possa desenvolver o auto aprendizado do aluno
- ✓ Possa promover a criação pela recreação, percepção global do mundo.

Podemos afirmar que esses critérios aplicam-se não só para softwares educacionais, mas também para softwares comerciais. Cabe ao professor fazer essa diferença. (ALVES, 2010)

Analisando as questões acima podemos considerar *software* educacional aquele que puder ser usado para algum objetivo educacional ou pedagógico e didático.

Se vamos responder sim à questão sobre a introdução do computador na educação, temos que colocar computadores e softwares aplicativos à disposição dos professores e dos alunos nas escolas e que lhes sejam oferecidas formações e material de apoio para que descubram como eles podem lhes ser úteis.

2.2 - Conceito de Software Educacional

Podemos ter o *Software* Educacional, como uma ferramenta que que contém propriedades pedagógicas. Ainda cabe salientar que a aprendizagem a partir desse tipo de aplicativo pode se tornar prazerosa e de fácil de entendimento, quando trabalhada com critérios objetivos. Levando em conta

todos os conceitos de Usabilidade e sendo estes aplicados em um *Software* Educacional, o “efeito aprendizagem, entendimento” tende a aumentar. (TEIXEIRA e BRANDÃO, 2003)

O computador representa as diversas ferramentas da informática e juntamente com os *softwares* educativos torna-se cada vez mais um amplificador de potencialidades na capacitação e aperfeiçoamento de alunos e professores. Visto que hoje não somente as instituições de ensino representam o sistema do conhecimento mas a internet já faz parte deste sistema.

Os *softwares* podem ser considerados educacionais a partir do momento em que atendam uma metodologia que os contextualize no processo ensino-aprendizagem. Esclarecemos aqui como já dito acima que o *software* pode não ter sido construído com o fim educacional mas pode ele ser usado para esse fim. (VESCE, 2008)

A etapa primordial para a produção de um *Software* Educacional é definir a concepção pedagógica daqueles que estão envolvidos no seu desenvolvimento e implementação. É preciso ter uma equipe docente envolvida no desenvolvimento de *softwares* educativos, pois se estes não estiverem envolvidos a ferramenta pode ter um papel meramente teórico e sem funcionalidade educacional.

2.3 - Tipos e classificação de alguns *Softwares* Educativos

Afirma Valente (1999), estudioso na área da informática educativa, que os *softwares* educativos podem ser classificados de acordo com a maneira que o conhecimento é manipulado. A modalidade pode ser caracterizada como uma versão computadorizada dos métodos tradicionais de ensino. Sendo as categorias mais comuns desta modalidade os tutoriais, exercício, prática, jogos e simulação. Há também o caso de *softwares* de uso comercial que estão sendo utilizados também para usos educativos. Os *softwares* de cálculo de estruturas, muito utilizados na engenharia, são ferramentas comerciais mas também exercem função educativa.

Ao todo, Valente (1999), classifica os *softwares* existentes dentro das seguintes categorias:

- ✓ **Tutorial:** Procuram ensinar controlando processo de aprendizagem. É a forma de dar ao aluno um tutor individual. Em softwares como os tutoriais o aluno escolhe o que deseja estudar. Estes geralmente costumam ser ricos em inovações tecnológicas (hipertextos, interface com sons, imagens, animações, etc.) e o seu conteúdo é predefinido, com isso tendo que escolher entre as opções existentes, o que muitas vezes limita o aluno ou professor em seu estudo.
- ✓ **Práticas e Exercícios:** *software* educativo que utiliza perguntas e respostas, utilizadas normalmente para revisar material já estudado colabora para a linguagem visual do aluno.
- ✓ **Programação:** *softwares* utilizados pelo aluno para programa o computador (como a linguagem LOGO, por exemplo).
- ✓ **Aplicativos:** incluem processadores de texto, planilhas eletrônicas, etc.
- ✓ **Multimídia e Internet:** misturam som, imagem e texto, colaboram para estudos dinâmicos.
- ✓ **Simulação:** utilizados para simular situações reais, que sem o uso do computador dificilmente poderiam ser vistos pelos alunos, com a mesma qualidade e realismo nas formas tradicionais de ensino.
- ✓ **Jogos:** programado para entreter, possui grande valor pedagógico quando utilizado para ensinar contextos com a realidade. É defendido na educação que o aluno aprende melhor quando é livre para descobrir ele próprio as relações existentes em um dado contexto.
- ✓ **Ferramentas para resolução de problemas:** o aluno deve produzir qual problema quer solucionar. Muitas vezes não compreendido pelos alunos devido à forma como o *software* foi produzido, atende quase todas as disciplinas, tanto no conhecimento como no interesse, geralmente são softwares abertos que permitem ao professor constantemente descobrir novas formas de planejar atividades que atendam o contexto da disciplina.

Há também os de prática e exercícios, que são mais utilizados no estudo de matemática onde Gagné (1992) propõe duas fases de aprendizagem:

- ✓ A aplicação
- ✓ A retroalimentação,

São utilizados para a memorização de algum conteúdo já estudado pelo aluno e também utilizados para reforço de conteúdo pelo professor.

Embora muitas pessoas associem o termo *software* aos programas de computadores, Amaral e Guedes (2009), em um artigo intitulado "**Análise de construção de *Software* educativo com qualidade: Sugestão de ficha para registro e avaliação de *software* educativo**", explicam que *software* não é apenas o programa, mas toda a documentação associada e os dados de configuração necessários para fazer com que os programas operem corretamente.

Podemos associar a qualidade dos *softwares* como um plano de características que devem ser alcançadas para que o produto atenda as necessidades dos usuários.

Nos *softwares* educativos, a qualidade do processo de desenvolvimento é essencial para a obtenção da qualidade do produto e orientação pedagógica que é tratada. A norma ISO/IEC 12 119 que enfoca os requisitos de qualidade aos mesmos descreve que *softwares* educativos devem apresentar fatores que facilitam a atuação do professor e aumento do desempenho do aluno.

2.4 - Métodos de Avaliação de Softwares Educativos

Para Oliveira, Menezes & Moreira, (1987):

Avaliar é um processo de classificar situações específicas em função de parâmetros pré-estabelecidos... todo software educativo reflete necessariamente, uma concepção de ensino e aprendizagem, resultante de uma visão filosófica da relação sujeito-objeto.

A avaliação de *Software* Educativo inspira-se, numa avaliação tradicional do mesmo, no processo de avaliação dos professores e no processo de avaliação dos alunos (VIEIRA, 2010).

A avaliação tradicional tem como objetivo os aspectos técnicos e funcionais e as aplicações não considerando o conteúdo e as teorias pedagógicas.

Os professores têm na avaliação privilégio como elementos decisivos no processo que embora incorpore alguns aspectos pedagógicos mantém as preocupações de conceito técnico.

Os alunos devem ser o foco na avaliação dos professores, pois nesta avaliação têm suma importância os destinatários principais da ferramenta. Não esquecer que o aluno é o usuário final do produto.

A avaliação de *software* educativos, segundo Ramos (1998), tem objetivos a serem alcançados pelo mesmo: orientar as escolas e os professores na seleção e uso do *software* educativo. Dessa forma, busca-se:

- Discutir características do *software* educativo como potencial didático e pedagógico.
- Produzir informações úteis aos produtores de *software* educativo.
- Identificar eventuais erros tanto de usabilidade como o pedagógico.
- Contribuir para uma base de conhecimento científico-pedagógico a toda as comunidades.
- Estimular as novas práticas pedagógicas nas escolas.

Segundo Tsukumo (1997) no Brasil alguns métodos já foram difundidos levando em conta:

1. **Usabilidade** - Evidencia a facilidade de utilização do *software*.
2. **Confiabilidade** - Evidencia que o desempenho se mantém ao longo do tempo em condições estabelecidas.
3. **Eficiência** - Os recursos e os tempos envolvidos são compatíveis com o nível de desempenho requerido para o produto.
4. **Manutenibilidade** - Há facilidade para correções, atualizações e alterações.

5. **Portabilidade** - É possível utilizar o produto em diversas plataformas com pequeno esforço de adaptação.

Assim, os *softwares* educativos devem ser de fácil uso, confiáveis, eficientes e principalmente ter portabilidade para que possam ser utilizados em diversas plataformas que são acessadas por professores e alunos.

2.5 - O uso das tecnologias de *software* aplicadas no processo de aprender

O computador é a ferramenta adequada que auxilia o aluno no processo de aprender e compreender o conteúdo a ele apresentado servindo como um grande aliado, nas novas pedagogias aplicadas ao ensino.

Barreto (1999) mostra, na Figura 1 disposta logo abaixo, elementos existentes em uma determinada sociedade. O computador, que aqui representa as diversas tecnologias de informática usadas na educação e pode ser usado como mediador, amplificador de potencialidades na capacitação dos alunos, professores e da instituição de ensino.

O computador no ensino: um mediador e amplificador de capacidades.

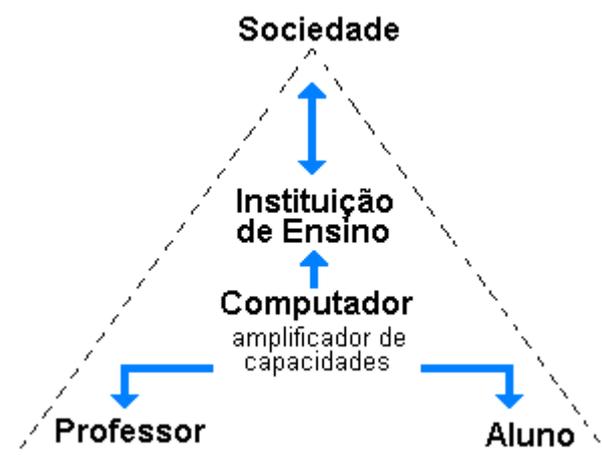


Figura 1 - Fonte: BARRETO (1999, p.216).

Percebem-se as diferentes motivações no uso do computador nas diferentes áreas da educação. Podemos afirmar que em algumas áreas, o uso

do computador é o único meio viável de utilizar certas aplicações. Olhando pela óptica de melhoria da educação ou da forma de educar, o computador oferece aos educadores uma diversidade de possibilidades ao educador e ao educando. (BARRETO, 1999)

Quando questiona-se um software qualquer, podemos dizer que se contribui para o processo de aprender não esquecendo que devemos investigar se cabe de fato o seu uso em determinado processo educativo.

3 – USABILIDADE

Usabilidade é a "capacidade de um produto ser usado por usuários específicos para atingir objetivos específicos com eficácia, eficiência e satisfação em um contexto específico de uso". ISO 9241-11 (1998)

O termo usabilidade começou a ser usado no início da década de 80, principalmente nas áreas de Psicologia e Ergonomia, como um substituto da expressão "user friendly" (traduzido para o português como "amigável"), a qual era considerada vaga e subjetiva. Isso porque no caso do *hardware*, remete a algo físico, que nos une à máquina. No caso do *software*, temos a ligação virtual, ou seja, a interface é o que demonstra a aparência, como aplicação e ferramenta. Além disso, usuários diferentes têm necessidades, de maneira que um sistema pode ser amigável para uma pessoa e não tão amigável para outra. (RAUBER, 2010)

Ainda conta Dias (2007, p. 25) que o termo usabilidade foi definido primeiramente pela ISO/IEC 9126 (1991) como forma de abordar a qualidade de um *software*. Essa abordagem é claramente orientada a produto e usuário e considera a usabilidade como "um conjunto de atributos de *software* relacionado ao esforço necessário para seu uso e para o julgamento individual de tal uso por determinado conjunto de usuários."

Assim, este termo ultrapassou os limites do ambiente acadêmico da, universidade passando a fazer parte do vocabulário técnico de outras áreas do conhecimento, sendo logicamente inserido na Tecnologia da Informação tendo sido traduzido literalmente para diversos idiomas. Em 1991, foi fundada a *Usability Professionals Association* (UPA), constituída por uma comunidade de profissionais, pesquisadores e empresas com participação ativa em estudos, pesquisas e testes de usabilidade. (DIAS, 2007)

Este conceito ainda sofreu uma evolução, sendo redefinido na parte 1 da norma

ISO/IEC *Final Committee Draft* (FCD) 9126-1, em 1998, que incluiu as necessidades do usuário. A partir daí, definem ainda outras características de qualidade de *software*, como funcionalidade, confiabilidade, eficiência, possibilidade de manutenção e portabilidade. (DIAS, 2007)

3.1 Princípios que suportam a usabilidade

Dias (2007) fornece alguns princípios que suportam as questões de usabilidade, sobre os quais nos apropriamos no presente trabalho.

- ✓ Eficácia e eficiência de uso
- ✓ Satisfação subjetiva
- ✓ Facilidade de aprendizado
- ✓ Facilidade de memorização
- ✓ Baixa taxa de erros
- ✓ Consistência
- ✓ Flexibilidade

Neste trabalho usaremos ainda três definições de usabilidade propostas por Dias (2007).

Definições orientadas ao usuário - relacionadas ao esforço mental ou atitude do usuário frente ao produto;

Definições baseadas no desempenho do usuário - associadas à forma de interação do usuário, com ênfase na facilidade de uso e no grau de aceitação do produto;

Definições orientadas ao contexto de uso - relacionadas às tarefas específicas realizadas por usuários específicos do produto, em determinado ambiente de trabalho.

3.2 Características de qualidade de software da ISO/IEC FCD 9126-1(1998)

Funcionalidade - capacidade do <i>software</i> de prover funções que atendem a necessidades expressas e implícitas, quando usado nas condições especificadas.
Confiabilidade - capacidade do <i>software</i> de manter seu nível de desempenho, quando usado nas condições especificadas.
Usabilidade - capacidade do <i>software</i> de ser compreendido, aprendido, usado e apreciado pelo usuário, quando usado nas condições especificadas.
Eficiência - capacidade do <i>software</i> de operar no nível de desempenho requerido, em relação à quantidade de recursos empregados, quando usado nas condições especificadas.
Possibilidade de manutenção - capacidade do <i>software</i> de ser modificado. Modificações podem abranger correções, melhorias ou adaptações do software, mudanças de ambiente ou nas especificações funcionais e de requisitos.
Portabilidade - possibilidades do <i>software</i> de ser transferido de um ambiente a outro.

Tabela 1 - Características de qualidade de software da ISO/IEC FCD 9126-1(1998)

A norma ISO 9241-11 Guidance on Usability (1998) definiu usabilidade como "a capacidade de um produto ser usado por usuários específicos para atingir objetivos específicos com eficácia, eficiência e satisfação em um contexto específico de uso."

Conceitos da ISO 9241-11(1998)

Usuário - pessoa que interage com o produto.
Contexto de uso - usuários, tarefas, equipamentos (<i>hardware</i> , <i>software</i> e materiais), ambiente físico e social em que o produto é usado.
Eficácia - precisão com que os usuários atingem objetivos específicos, acessando a informação correta ou gerando os resultados esperados. A precisão é uma característica associada à correspondência entre a qualidade do resultado.
Eficiência - precisão com que os usuários atingem seus objetivos, em relação à quantidade de recursos gastos.
Satisfação - conforto e aceitabilidade do produto, medidos por meio de métodos subjetivos e/ou objetivos. As medidas objetivas de satisfação podem se basear na observação do comportamento do usuário (postura e movimento corporal) ou no monitoramento de suas respostas fisiológicas. As medidas subjetivas, por sua vez, são produzidas pela quantificação das reações, atitudes e opiniões expressas subjetivamente pelos usuários.

Tabela 2 - Conceitos da ISO 9241-11(1998) - Dias (2007 p., 27).

Podemos considerar um *software* eficaz quando são atingidos os objetivos dos usuários. Se um *software* é fácil de usar, fácil de aprender e mesmo agradável ao aluno, mas não atende a objetivos específicos do usuário, ele não será usado, **mesmo que seja oferecido gratuitamente**. (DIAS, 2007)

Ainda, segundo a autora, a eficiência é definida por tempo de resposta, tempo total para realizar uma tarefa específica. Isto é geralmente utilizado para responder se o *software* pode ou não ser utilizado por professores em suas aulas.

A satisfação do usuário, terceiro elemento refere-se geralmente a opiniões dos alunos e professores a respeito de um sistema, normalmente descritos a partir de questionamentos feitos aos próprios alunos e professores. Há sistema que pode ser excelente para algumas pessoas e totalmente inadequado para outras (DIAS, 2007).

A usabilidade depende das tarefas específicas que os professores realizam com o *software*, não adianta querer uma tarefa que o *software* não foi programado para realizar, isto em termos matemáticos seria tentar colocar um quadrado dentro de uma circunferência sendo o raio da circunferência igual à aresta da base do quadrado. Portanto, a usabilidade é uma qualidade de uso de um *software*, associada diretamente ao seu contexto operacional as tarefas solicitadas.

Uma outra ênfase de eficácia e eficiência da norma ISO 9241-11 (1998) de usabilidade conceitua desempenho do usuário. Como a habilidade do usuário é realizar tarefas para as quais o sistema foi projetado, esse conceito é aplicado no sistema para avaliar o contexto em que o sistema será usado. A medida de satisfação do usuário é importante quanto ao seu desempenho ou ao produto final, ou seja, um sistema, mesmo sendo eficiente e capaz de gerar resultados eficazes.

O desempenho do usuário é avaliado à medida que os objetivos de uso do sistema são atingidos (eficácia) e os recursos (tempo, dinheiro e esforço mental) são gastos para atingir tais objetivos (eficiência). Dias (2000, p. 29)

Usabilidade pode ser definida como uma medida da experiência do usuário ao interagir com algum problema um aplicativo de *software* tradicional ou outro dispositivo que o usuário possa operar de alguma forma que envolva o mesmo problema.

Nielsen descreve cinco atributos da usabilidade Em seu livro **Usability Engineering (1993, p. 26): facilidade de aprendizado, eficiência de uso, facilidade de memorização, baixa taxa de erros e satisfação subjetiva.**

Facilidade de aprendizado - o sistema deve ser fácil de aprender de tal forma que o usuário consiga rapidamente explorá-lo e realizar suas tarefas com ele.
Eficiência de uso - o sistema deve ser eficiente a tal ponto de permitir que o usuário, tendo aprendido a interagir com ele, atinja níveis altos de produtividade na realização de suas tarefas.
Baixa taxa de erros - em um sistema com baixa taxa de erros, o usuário é capaz de realizar tarefas sem maiores transtornos, recuperando erros, caso ocorram.
Facilidade de memorização - após um certo período sem utilizá-lo, o usuário não frequente é capaz de retornar ao sistema e realizar suas tarefas sem a necessidade de reaprender como interagir com ele.
Satisfação subjetiva - o usuário considera agradável a interação com o sistema e se sente subjetivamente satisfeito com ele.

Tabela 3 - Conceitos de Nielsen sobre Usabilidade) - Dias (2007 p., 27).

Para um sistema de melhor qualidade tem que se dar atenção à sua eficácia, facilidade de aprendizado, eficiência de uso, facilidade de uso, facilidade de memorização baixa taxa de erros, satisfação de uso, facilidade de manutenção, baixa taxa de erros, e ainda a outras propriedades associadas a essas, como previsibilidade, familiaridade, poder de generalização, compatibilidade com o contexto de uso, tempo de resposta, flexibilidade e recuperação de erro. Com os atributos de Nielsen podemos descrever a avaliação de heurísticas de usabilidade, análise e interface de *softwares*. Juntamente com as regras de Ouro de **Shneiderman (1998)** a análise dos *softwares* se torna mais eficaz.

As heurísticas de Nielsen e regras de ouro de Sheinderman

Nielsen com a colaboração de Molich em 1990 desenvolveu as heurísticas para o método de avaliação heurística. Em 1994 em análise de fatores observados em um conjunto de 249 problemas de usabilidade detectados em estudos empíricos, Nielsen (1993, p. 3) simplificou esses problemas em dez heurísticas de usabilidade. Tais como as heurísticas de Nielsen, Shneiderman (1998, p. 74) estabeleceu as oito “regras de ouro”.

Heurísticas de Nielsen	
1.	visibilidade do estado atual do sistema - o sistema deve sempre manter informados os usuários a respeito do que está acontecendo, por meio de feedback apropriado em tempo razoável;
2.	correlação entre o sistema e o mundo real - o sistema deve falar a linguagem do usuário, com palavras, frases e conceitos familiares, ao invés de utilizar termos técnicos. As convenções do mundo real devem ser seguidas, fazendo com que as informações apareçam em uma ordem lógica e natural ao usuário;
3.	controle e liberdade do usuário - os usuários costumam escolher, por engano, funções do sistema, e precisam encontrar uma maneira de sair da situação ou estado indesejado, sem maiores problemas. Deve ser possível ao usuário desfazer ou refazer operações;
4.	consistência e padrões - os usuários não devem ter que adivinhar que palavras, situações ou ações diferentes significam a mesma coisa;
5.	prevenção de erros - melhor do que boas mensagens de erro é um projeto cuidadoso que previna, em primeiro lugar, a ocorrência de erros;
6.	reconhecimento ao invés de memorização - objetos, ações e opções devem ser visíveis. O usuário não deve ser obrigado a lembrar de informações ao passar de um diálogo a outro. As instruções de uso do sistema devem estar visíveis ou facilmente acessíveis quando necessário;
7.	flexibilidade e eficiência de uso - deve ser permitido ao usuário personalizar ou programar ações frequentes. Devem ser implementados aceleradores para serem adotados por usuários experientes;
8.	projeto estético e minimalista - os diálogos não devem conter informação irrelevante ou raramente necessária. Cada unidade extra de informação em um diálogo compete com unidades relevantes de informação e diminuem sua visibilidade relativa;
9.	suporte aos usuários no reconhecimento, diagnóstico e recuperação de erros - as mensagens de erro devem ser expressas em linguagem clara, sem códigos, indicando precisamente o problema e sugerindo soluções;
10.	informações de ajuda e documentação - a documentação do sistema deve sempre estar disponível ao usuário, mesmo que o sistema seja fácil de usar. A documentação de auxílio ao usuário deve ser fácil de pesquisar, focada nas tarefas que o usuário costuma realizar com o sistema e não muito longa.

Tabela 4 - Heurísticas de Nielsen (1993)

Regras de Ouro de Shneiderman	
1.	consistência - sequências de ações similares para situações similares; a mesma terminologia em menus e telas de ajuda ao usuário; padrão de cores, leiaute, fontes;
2.	atalhos para usuários frequentes - teclas especiais, macros e navegação simplificada são exemplos de atalhos que facilitam e agilizam a interação dos usuários mais experientes que usam o sistema com frequência, eliminando telas ou passos desnecessários;
3.	feedback informativo - toda ação do usuário requer uma resposta do sistema, a qual será mais ou menos detalhada ou informativa, dependendo do tipo de ação executada;
4.	diálogos que indiquem término da ação - as sequências de ações do sistema devem ser organizadas de tal forma que o usuário seja capaz de identificar quando cada grupo de ações foi completado com sucesso
5.	prevenção e tratamento de erros - o sistema deve ser projetado de tal forma que os usuários não consigam cometer erros de alta severidade e ainda recebam instruções adequadas para o tratamento dos erros que porventura ocorrerem;
6.	reversão de ações - tanto quanto possível, as ações devem ser reversíveis, aliviando, assim, a ansiedade dos usuários e encorajando-os a explorar o sistema;
7.	controle - os usuários mais experientes desejam ter a sensação de que detêm o controle sobre o processamento e que o sistema responde a suas ações, e não o contrário.
8.	baixa carga de memorização - a capacidade humana de memorização requer que a tela do sistema seja simples, consistente em relação às outras telas do conjunto e que a frequência de movimentos em cada tela seja reduzida.

Tabela 5 - Regras de Ouro de Shneiderman (1998)

4 - METODOLOGIA

Para avaliar os *softwares* GEOGEBRA, POLY E CABRI utilizaram-se as dez heurísticas de Nielsen juntamente com as regras de ouro de Shneiderman. Ao utilizar todos os critérios necessitou-se definir algumas características relacionadas ao potencial pedagógico e didático do produto, estabelecendo um olhar sobre as necessidades dos professores e alunos de apropriarem-se da ferramenta, para que se envolvam num processo de exploração sobre as suas potencialidades de adequação ao currículo ou de promoção da aprendizagem (STABLES, 1997; GÁLVIS, 1997; CASTAÑON, 1997; DEKKER, 1996).

Foi dada ênfase às características evidenciadoras das potenciais possibilidades dos alunos e professores construir, a partir da usabilidade do *software*, situações de aprendizagem.

Ao avaliar um produto necessita-se estabelecer critérios, utilizar o paradigma de qualidade, com origem na Engenharia de *Software*, segundo a norma ISO/IEC 9126-1, que a define como "a totalidade das características de um produto de *software* que lhe confere a capacidade de satisfazer necessidades explícitas e implícitas". As necessidades explícitas são expressas na definição de requisitos propostos por pedagogos ao produtor e as necessidades implícitas são aquelas que podem não estar expressas nos documentos do produto, mas que são necessárias aos professores e alunos.

Aplicaram-se dez características, dentro do conjunto estabelecido pelas heurísticas de Nielsen: *visibilidade do estado atual do sistema, correlação entre o sistema e o mundo real, controle e liberdade do usuário, consistência e padrões, prevenção de erros, reconhecimento ao invés de memorização, flexibilidade e eficiência de uso, projeto estético e minimalista, suporte aos usuários no reconhecimento, diagnóstico e recuperação de erros, informações de ajuda e documentação.*

Ao final, obtemos uma representação tabular da qualidade do produto descrito na tabela 5.

Utilizamos alguns exemplos aplicando os conceitos de usabilidade para que os leitores vejam mais claramente a metodologia de análise aplicada.

5 - ANÁLISES REALIZADAS

As análises foram feitas sobre um conjunto de *softwares* de matemática, analisando aspectos positivos e falhos de sua usabilidade. A análise se concentrou em três *softwares*, o geogebra, poly e cabri. O propósito da análise é verificar os problemas mais preponderantes observados nas ferramentas no que se refere à sua usabilidade e que devem ser observados por professores de matemática que virão a se utilizar das mesmas.

5.1 Geogebra

Começamos o trabalho com o programa Geogebra, estudo de caso um. Esse programa muito utilizado por professores de matemática do ensino médio possui as seguintes características:

É um programa leve com 15.274 Kb, criado para ser usado na sala de aula. Com ele se pode fazer construções com pontos, segmentos, retas, vetores, seções cônicas e funções. Também podem ser incluídas equações e coordenadas diretamente, assim se é capaz de lidar com variáveis para números oferecendo comandos para encontrar raízes e pontos extremos de uma função.

Quanto à precisão, é possível no menu opções determinar a quantidade de casas decimais. Pode-se optar por até quinze cinco casas decimais, sendo necessário converter para minutos e segundos se assim desejar, pois a parte decimal refere-se aos graus. Sendo assim, quando temos $25,5^\circ$ no sistema decimal e convertemos para o sistema sexagesimal obtemos $25^\circ 30'$. Porém, ao considerar apenas a unidade de medida inteira, ângulos congruentes, por motivos de arredondamento automático, aparecerão com medidas diferentes.

Aqui podemos notar falta de flexibilidade, consistência e satisfação subjetiva do usuário, pois a falta desses itens podem desestimular o usuário deixando de ser atrativo o *software*, até mesmo para os professores que dominam o conteúdo fica pouco atrativo.

Outra limitação é a relação de pertinência entre os objetos, que faz com que segmentos de reta fiquem dependentes da reta suporte a que pertencem. Para exemplificar:

- Traçamos uma reta “a” qualquer passando por dois pontos



Figura 3 – Traçado de uma reta usando dois pontos no Software Geogebra

Marcamos alguns pontos pertencentes a esta reta, consideramos alguns segmentos desta reta, no exemplo abaixo, os segmentos b, c e d.



Figura 4 - Traçado de uma reta usando vários pontos no Software Geogebra

Após, escondemos a reta suporte a utilizando a ferramenta  exibir objeto



Figura 5 - Traçado de segmentos usando vários pontos no Software Geogebra

Quando movemos os pontos A e B destes segmentos, movemos também a reta suporte a, e todos os segmentos se movem e alteram suas medidas. Este fato faz com que não seja viável utilizar o programa para classificação de segmentos colineares. Nesse caso é mais apropriado utilizar outras ferramentas tais como material manipulável e o livro didático.

A maioria das limitações foram observadas durante o estudo sobre o uso do *software* no ensino de Geometria Plana. No *software*, a marcação é de ciclo completo, ou seja, trata-se de geometria orientada. No seu uso de forma a não confundir o aluno serão necessários alguns cuidados, nos quais tomamos por base as heurísticas propostas por Nielsen (1993).

Realmente é necessário um preparo criterioso dos planos de aula para utilizar o programa GeoGebra como meio de mediação, pois é relevante analisar sua estruturação em relação ao conteúdo proposto, percorrer os possíveis caminhos que o aluno buscará para então definir como usá-lo. Este preparo necessita de um tempo maior, ficando muitas vezes inviável, devido à quantidade de horas disponíveis para isso. Desenvolver um pequeno manual também seria de grande ajuda para o aluno.

Segundo a heurística 10 de Nielsen “informações de ajuda e documentação” - a documentação do sistema deve sempre estar disponível ao usuário, mesmo que o sistema seja fácil de usar. A documentação de auxílio ao usuário deve ser fácil de pesquisar, focada nas tarefas que o usuário costuma realizar com o sistema e não muito longa. A falha nesta heurística no *software* faz com que ele deixe de ser utilizado nesse conteúdo prejudicando sua funcionalidade e seu objetivo.

5.2 - Poly

No estudo de caso dois vamos apresentar outro *software* muito utilizado pelos professores de matemática: o Poly, que é um programa para explorar e construir poliedros. Trata-se de um programa leve com 1,4 Mb.

O Poly é uma criação, que permite a investigação de sólidos tridimensionalmente, bidimensionalmente e de vista topológica. Possui uma grande coleção de sólidos platônicos e arquimedianos, entre outros. Esta análise pode ser feita de três modos diferentes: imagens tridimensionais, rede bidimensional e planificação. As imagens podem ser dinamicamente colocadas em movimento. O programa é uma preciosa ferramenta no estudo da geometria, permite uma rápida visualização de poliedros e pode construir uma base de para trabalhos de natureza investigativa na qual os alunos desenvolvem outras competências matemáticas (ex: argumentação e prova).

O programa é de fácil aprendizagem e utilização. Os alunos mais novos aprendem a utilizar o programa de uma forma intuitiva.

Apresenta os sólidos bem coloridos (havendo opção de mudar a cor dos mesmos), colaborando para atrair a atenção do usuário.

Ele permite a visualização de sólidos “fechados” sendo gradativamente “abertos”, até que sejam obtidas as planificações dos mesmos;

Permite também a visualização de sólidos não muito comuns de serem encontrados concretamente e nem mesmo representados nos livros como por exemplo, sólidos de Arquimedes e domos geodésicos.

O programa proporciona essencialmente a visualização gráfica de vários conjuntos de sólidos geométricos.

O Poly, porém, apresenta certos problemas nos aspectos descritos a seguir:

- Não ha versão em português;
- Não apresenta determinados sólidos comuns de serem estudados em Geometria Espacial, como por exemplo: pirâmide triangular e hexagonal;

- No geral, é muito superficial, não oferecendo recursos para que o aluno formule hipóteses e aplique-as;
- Oferece pouco estímulo à criatividade, podendo tornar-se facilmente cansativo.

Trata-se de um programa é simples no que se refere a comandos que podem ser usados por alunos muito novos ou com poucos conhecimentos de informática. Permite fazer a visualização de poliedros de três modos diferentes, o utilizador pode operar sobre as estruturas, mudando as cores, mudando a posição, alterando a dimensão e colocando-as em movimento. Estas possibilidades permitem trabalhos de natureza exploratória e investigativa que em termos pedagógicos vão muito além dos tradicionais modelos estáticos. Existe a possibilidade de copiar os poliedros para outro programa ou imprimir as suas imagens diretamente (ex: planificações). Globalmente o seu interesse educativo reside na visualização, construção (quando impressos), exploração e investigação.

Não são necessários conhecimentos específicos para a utilização deste programa. Pode ser utilizado pelos alunos nas aulas ou em trabalho autónomo em casa. Nas aulas, as propostas de trabalho devem ser naturalmente adequadas ao nível de escolaridade dos alunos. É possível transferir os poliedros em estudos para outros programas. O Poly Pro vai ao encontro de um estilo de aprendizagem em que os alunos são os construtores do seu saber e o professor é um orientador dos alunos. Os professores podem utilizar o programa na elaboração de materiais de trabalho para a sala de aula.

Pelas Heurísticas de Nielsen, notamos certas limitações no *software* reconhecimento ao invés de memorização - objetos, ações e opções devem ser visíveis. O usuário não deve ser obrigado a lembrar de informações ao passar de um diálogo a outro. As instruções de uso do sistema devem estar visíveis ou facilmente acessíveis quando necessário.

No dodecaedro, as telas ao lado do sólido pode levar o usuário a ter que decorar os sólidos platônicos, visto que a tela do *software* continua apresentando na tela direto sólidos como pirâmides, forçando uma memorização dos solidos platônicos.

No gráfico o aluno tem uma infinidade de recursos para visualizar poliedros, multicores e de vários ângulos.

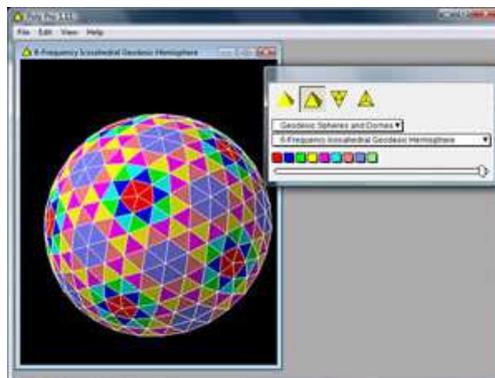


Figura 6 – Criando uma esfera no software Poly

Pelas regras de ouro de Shneiderman (1998) baixa carga de memorização - a capacidade humana de memorização requer que a tela do sistema seja simples, consistente em relação às outras telas do conjunto e que a frequência de movimentos em cada tela seja reduzida. No dodecaedro a frequência de movimento é muito rápida, deixando o usuário confuso no movimento do sólido.

Também pelas regras de ouro de Shneiderman (1998) diálogos que indiquem término da ação - as sequências de ações do sistema devem ser organizadas de tal forma que o usuário seja capaz de identificar quando cada grupo de ações foi completado com sucesso, não é levada em conta no *software*, não demonstra quando a ação foi completada.

Quanto aos conceitos da ISO 9241-11(1998) – confiabilidade, usabilidade, eficiência o *software* está muito bem orientado. Já na possibilidade de manutenção e portabilidade, o *software* não possui boa capacidade para atender seus usuários provocando travamento do sistema e até mesmo perda dos dados.

O programa permite essencialmente a visualização gráfica de vários conjuntos de sólidos geométricos.

Para podermos entender os aspectos operacionais do Poly devemos apresentar os sólidos apresentados pelo *software* que são:

- Sólidos Platônicos

- Sólidos de Arquimedes
- Prisma e anti-prisma
- Sólido de Johnson
- Deltaedros
- Sólidos de Catalan
- Dpirâmides e Deltoedros
- Esferas e Domos Geodesicos
- Os sólidos Platônicos

No *software* existem algumas limitações que podemos referenciar nas heurísticas de Nielsen (1993).

5.3 - Cabri 3D V2

No estudo de caso três vamos apresentar o *software* CABRI 3D pouco utilizado pelos professores de matemática da rede estadual por ser um programa com um custo razoável.

O Cabri 3D é um programa que, pode construir, visualizar e manipular rapidamente todo o tipo de objetos tridimensionais: retas, planos, cones, esferas, poliedros. Com ele, podemos fazer construções dinâmicas, das mais simples às mais complexas. Pode-se medir objetos, integrar dados numéricos e mesmo repetir o processo de construção de uma figura. Trata-se também de um programa leve, com 3,66 Mb.

Os desenhos de objetos geométricos são feitos a partir das propriedades que os definem. Mas não é só isto que ele nos oferece, e aqui está a riqueza do *software*: através de deslocamentos aplicados aos elementos que compõem o desenho, este se transforma, mantendo as relações geométricas que caracterizam a situação. Assim, para um dado objeto temos associada uma coleção de "desenhos em movimento", e as características invariantes que aí aparecem correspondem as propriedades geométricas do objeto. Do ponto de

vista de construção de conhecimento o que significa este recurso? Muito! Vejamos:

- Se sob ação de movimento, o desenho não corresponde ao que queremos duas são as possibilidades. Ou o objeto foi mal construído, isto é, as propriedades que caracterizam o objeto não foram bem utilizadas. Ou nossa imagem visual do objeto é que não é adequada, isto é, a construção foi feita corretamente, mas nossa visualização é que está não está de acordo com o esperado. Em qualquer um dos casos, o recurso do "desenho em movimento" leva ao ajuste entre componente conceitual e componente visual do objeto.
- Configurações clássicas passam a ter múltiplos aspectos visuais, e com isto passam a ser identificadas em situações não prototípicas.
- Os "desenhos em movimento" criam naturalmente um ambiente de investigação; os invariantes se destacam e isto nos intriga.

O programa é de fácil manipulação desde que o manual do usuário seja consultado.

O mecanismo implícito de criação de objetos simplifica consideravelmente a construção das figuras. Um ponto de intersecção entre uma linha e um plano de construção podem assim ser utilizados sem ter de criar antecipadamente. Muitos dos atributos gráficos (cor, tamanho, textura) podem ser aplicados a objetos para criar ainda mais e compreensíveis figuras atraentes. O formato de arquivos Cabri 3D é baseado no padrão XML (CabriML), para que qualquer usuário possa entender e modificar arquivos Cabri 3D. Combinado com o uso interno do padrão Unicode para a representação dos personagens, Cabri 3D pode ser usado para criar e ler dados em todas as línguas. O Cabri 3D plug-in permite que figuras de geometria dinâmica a ser publicado na Internet, e também no processamento de documentos de outra palavra.

Existem alguns problemas de orientação com Cabri 3D. O efeito é que, por vezes, os objetos de repente mudam de orientação em determinadas configurações. Para obter um exemplo desse problema, abra o arquivo "swing" e arraste o mesmo por cima da figura, verá que a orientação mudou mas o desenho não. Ainda não se sabe se esses problemas são evitáveis ou dependem da visualização do 3D, orientação e clareza.

O *software* apresenta uma caixa de ferramentas simples e compreensível.

Aqui estão as caixas de ferramentas possíveis:

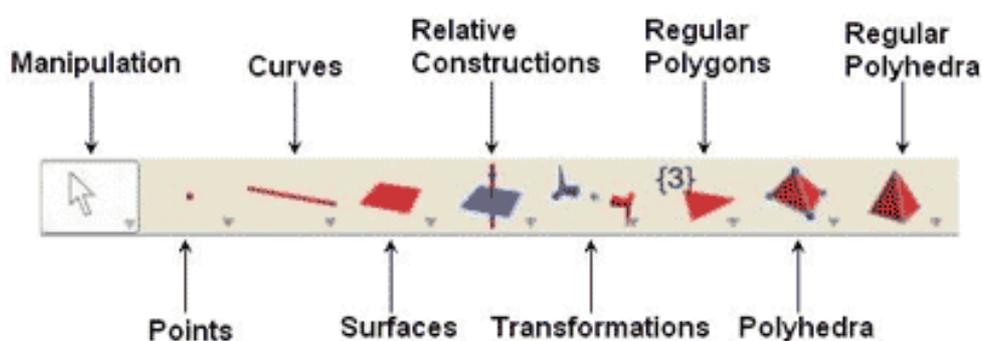


Figura 7 – Visualização dos comandos da página principal do Cabri

Cada caixa contém uma série de ferramentas. Por exemplo, aqui é a caixa de ferramentas de curvas, com a ferramenta vetor selecionada. Para usar uma ferramenta, primeiro selecione-a clicando na caixa de ferramentas se o que você quer é o que é visível clique, segure e arraste para alterar a ferramenta visível. Em seguida, selecione os objetos aos quais você vai aplicar a ferramenta, ou clique para criar pontos em que a ferramenta será aplicada.

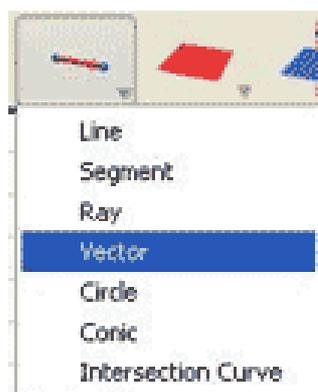


Figura 8 – Construção de vetores do Cabri

Com apenas alguns cliques, os alunos podem construir figuras em 2-D e figuras em 3-D, do mais simples ao mais complexo, combinando fundamentais objetos geométricos como pontos, ângulos, segmentos, círculos, planos, sólidos e transformações. Criar expressões algébricas utilizando os conceitos fundamentais, tais como números, variáveis e operações. Ligue geometria e álgebra de medição do comprimento, ângulos, área e volume e, em seguida, anexar esses valores numéricos diretamente à figura de usá-los em cálculos ou expressões algébricas. Explora um número de propriedades através da manipulação de seus elementos variáveis. Observar os efeitos das transformações dinâmicas, como redução e ampliação. Fazer conjecturas sobre as propriedades algébricas e geométricas, e em seguida, verificar a relação entre as várias partes de uma figura.

Relacionado com as Heurísticas de Nielsen (1993), observamos que o *software* possui restrições e certas limitações, como o não reconhecimento de alguns objetos, ações e opções que devam ser visíveis. O usuário não deve ser obrigado a lembrar de informações ao passar de um diálogo a outro. As instruções de uso do sistema devem estar visíveis ou facilmente acessíveis quando necessário. Também quanto à prevenção de erros o *software* não atende a heurística 5 de Nielsen.

Também pelas regras de ouro de Shneiderman diálogos que indiquem término da ação - as sequências de ações do sistema devem ser organizadas de tal forma que o usuário seja capaz de identificar quando cada grupo de ações foi completado com sucesso: o *software* não demonstra quando a ação foi completada ou terminada.

Quanto aos Conceitos da ISO 9241-11(1998) – confiabilidade, usabilidade, eficiência, o *software* está orientado de forma bem organizada. Já na possibilidade de manutenção e portabilidade, o *software* não possui boa capacidade para atender seus usuários provocando travamento do sistema e ate mesmo perda dos dados.

AVALIAÇÃO DA USABILIDADE NO GEOGEBRA, POLY E CABRI

	Heurísticas de Nielsen	GEOGEBRA	POLY	CABRI	DESCRIÇÃO DOS PROBLEMAS DE USABILIDADE IDENTIFICADOS
Heurística 1	visibilidade do estado atual do sistema - o sistema deve sempre manter informados os usuários a respeito do que está acontecendo, por meio de feedback apropriado em tempo razoável;				Possui visibilidade e mantém o usuário informado sobre o que são as etapas do processo, mas não possuem descrição das imagens e referência dos textos
Heurística 2	correlação entre o sistema e o mundo real - o sistema deve falar a linguagem do usuário, com palavras, frases e conceitos familiares, ao invés de utilizar termos técnicos. As convenções do mundo real devem ser seguidas, fazendo com que as informações apareçam em uma ordem lógica e natural ao usuário;				O software possui concordância com o mundo real, mas a linguagem é difícil para alguns leitores principalmente leitores com pouca noção de inglês.
Heurística 3	controle e liberdade do usuário - os usuários costumam escolher, por engano, funções do sistema, e precisam encontrar uma maneira de sair da situação ou estado indesejado, sem maiores problemas. Deve ser possível ao usuário desfazer ou refazer operações;				Pouca interação, o sistema não permite controle ao usuário, pois não possibilita editar o conteúdo, somente excluir para realizar novamente o comando.
Heurística 4	consistência e padrões - os usuários não devem ter que adivinhar que palavras, situações ou ações diferentes significam a mesma coisa;				A padronização é falha sobre os recados das versões, que não se repetem e que se referem a recados do perfil que o usuário s estão visitando.
Heurística 5	prevenção de erros - melhor do que boas mensagens de erro é um projeto cuidadoso que previna, em primeiro lugar, a ocorrência de erros;				As mensagens de erro são de fácil interpretação pelos usuários não provocando questionamentos pelos usuários, que em contrapartida não solucionam o problema inteiramente.
Heurística 6	reconhecimento ao invés de memorização - objetos, ações e opções devem ser visíveis. O usuário não deve ser obrigado a lembrar de informações ao passar de um diálogo a outro. As instruções de uso do sistema devem estar visíveis ou facilmente acessíveis quando necessário;				Possui comandos de teclado, mas só informa isso aos usuários na interface da ajuda.

Heurística 7	flexibilidade e eficiência de uso - deve ser permitido ao usuário personalizar ou programar ações frequentes. Devem ser implementados aceleradores para serem adotados por usuários experientes;				De um modo geral, os softwares não permitem ser adequados ao nível do usuário. Ou seja, não há muitas opções de aceleradores para usuários mais experientes.
Heurística 8	projeto estético e minimalista - os diálogos não devem conter informação irrelevante ou raramente necessária. Cada unidade extra de informação em um diálogo compete com unidades relevantes de informação e diminuem sua visibilidade relativa;				O <i>design</i> é condiz com a aplicação do software, mas traz informações que não ajudam o usuário.
Heurística 9	suporte aos usuários no reconhecimento, diagnóstico e recuperação de erros - mensagens de erro devem ser expressas em linguagem clara, sem códigos, indicando precisamente o problema e sugerindo soluções;				Os erros de inclusão ou digitação do usuário podem ocorrer na entrada de dados. O software reconhece esses erros e comunica o usuário sobre o erro, não permitindo a inclusão de novos dados.
Heurística 10	informações de ajuda e documentação - a documentação do sistema deve sempre estar disponível ao usuário, mesmo que o sistema seja fácil de usar. A documentação de auxílio ao usuário deve ser fácil de pesquisar, focada nas tarefas que o usuário costuma realizar com o sistema e não muito longa.				O (<i>Help</i>) de ajuda encontra-se disponível no rodapé, ou no topo do software, mas nenhum outro item tem ajuda incluída. A informação encontra-se somente em inglês.

Tabela 6 – Adaptado do trabalho de dissertação de Rauber – (2010)

1 O item  representa que não foi localizado problema de usabilidade, enquanto que o item  representa que um problema de usabilidade foi identificado.

6 – CONSIDERAÇÕES FINAIS

No desenvolvimento deste trabalho, foram abordadas e relacionadas entre si inúmeras questões referentes a *softwares* para o desenvolvimento do conteúdo de geometria no ensino médio, aprofundando-se ainda em conceitos de *softwares* educativos, e sua usabilidade. Desse modo, a título de considerações finais, procurar-se-á, a partir dos aspectos discutidos, apontar para possíveis desdobramentos e implicações referentes à usabilidade dos *softwares* para o ensino de matemática em particular a geometria.

Cabe ressaltar que o presente trabalho teve como objetivo verificar a usabilidade dos *softwares* Geogebra, Poly e Cabri, a fim de propor e aprimorar recomendações para a construção e adaptação de professores e alunos nos *softwares* estudados.

Para isso, partiu-se da hipótese de que os *softwares* estudados não seguem as recomendações ou proporcionam eficácia e eficiência para que alunos e professores utilizem-nas de forma plena. A partir das constatações obtidas durante o desenvolvimento deste estudo, com atenção aos resultados da avaliação de usabilidade, realizada através de diferentes métodos, foi possível entender certas peculiaridades dos *softwares*.

Refletindo sobre esses questionamentos importantes e, tomando como fundamentação teórica as abordagens de diversos pesquisadores citados, esclarece-se que a escolha de um *software* ou de um ambiente computacional merece reflexões e estudos a esse respeito. Na concepção do autor deste estudo, tal escolha deve estar vinculada a uma filosofia educacional em que se acredita, a uma metodologia educacional e ainda aos objetivos que se quer alcançar no novo estudo da matemática.

Como já mencionado acima, a escolha de um ambiente computacional relaciona-se com diversos aspectos teóricos e metodológicos, porém um dos aspectos fundamentais consiste na mediação do professor, a qual é fundamental no processo de aprendizado. O ambiente, por mais rico e

construtivo que seja não é suficiente para promover a construção do conhecimento (VALENTE 1993).

REFERÊNCIAS

ALVES Maria Felipe Alves (1987) "Análise de Software Educacional". Belo Horizonte, Brasil

DIAS, Cláudia. **Usabilidade na web: Criando portais mais acessíveis. 2. ed. Rio de Janeiro:** Alta Books, 2007.

CAMPOS, Gilda Helena Bemardino de, ROCHA, Ana Regina O da. Manual para a avaliação da qualidade de software educacional: relatório técnico do Programa de Engenharia de Sistemas e Computação. Rio de Janeiro: UFRJ, COPPE, 1990.

CARDOSO, Rogério Nesi Pereira. Predição, estimativa e medição da confiabilidade durante o ciclo de vida do software. Rio de Janeiro, 1990. Dissertação (Mestrado) — Programa de Engenharia de Sistemas e Computação, COPPE, UFRJ.

FREITAS, Afonso C, BARGUT, Maurício F., ROCHA, Ana Regina C. Características de qualidade de programas: relatório técnico do Programa de Engenharia de Sistemas e Computação. Rio de Janeiro: UFRJ, COPPE, 1985.

ISO/IEC 9126. Software product evaluation: Quality characteristics and guidelines for their use. 1991

ISO 9241 Part 11. Ergonomic requirements for office work with visual display terminals, Part 11: Guidance on usability. 1998

ISO 9241-11. Ergonomic requirements for office work with visual display terminals(VDTs). Disponível em http://www.iso.org/iso/catalogue_detail.htm?csnumber=16883

MORAES, Raquel de Almeida **Informática na educação.** 2 ed. Rio de Janeiro: DP&A, 2000.

NIELSEN, Jakob. **Usability Engineering.** San Francisco: Morgan Kaufman, 1993.

NIELSEN, Jakob; LORANGER, Hoa. **Usabilidade na Web**. Rio de Janeiro: Elsevier, 2007.

OLIVEIRA, Celina O, MENEZES, Eliane, MOREIRA, Mercia. Avaliação de software educacional. Documento base para discussão apresentado na XVIII Reunião da ABT. Belo Horizonte, 1986.

PRATES, R.O.; BARBOSA, S.D.J. Avaliação de Interfaces de Usuário: conceitos e métodos. In: Jornadas de Atualização em Informática, Congresso da Sociedade Brasileira de Computação, 2003. Disponível em: <http://www.inf.pucrio.br/~inf1403/docs/JAI2003_PratesBarbosa_avaliacao.pdf> . Acesso em: 11 NOV.. 2010.

RAUBER, Luis Henrique. **Usabilidade Das Interfaces De Aplicações De Mídias Sociais Para Deficientes Visuais: TWITTER E ORKUT**. Novo Hamburgo, RS, 2010.

RAMOS J.L. **A criação e utilização de micromundos de aprendizagem como estratégia de integração do computador no currículo do ensino secundário**. Dissertação de doutoramento. Universidade de Évora, 1998.

ROCHA, Ana Regina O, PASSOS, Maria Cristina J. Fonseca. **Critérios para avaliação de software para pecuária do leite: relatório técnico do Programa de Engenharia de Sistemas e Computação**. Rio de Janeiro:UFRJ, COPPE, 1990.

SANTA ROSA, Jose Guilherme, MORAES Anamaria: **Avaliação e projeto no design de Interface**. 1 ed. Rio de Janeiro: 2AB , 2008.

SPYER, Juliano. **Para entender a internet: noções básicas e desafios da comunicação em rede**. 2009. Disponível em <http://para-entender-a-internet.softonic.com.br/>. Acesso em: 18 nov. 2010.

TAKAHASHI, T. (Org.). **Livro Verde - Sociedade da Informação no Brasil**. Brasília: MCT, 2000.

TSUKUMO, A.N., et al (1997) "Qualidade de Software: Visões de Produto e Processo de Software". In: VIII CITS - CONFERÊNCIA INTERNACIONAL DE TECNOLOGIA DE SOFTWARE:QUALIDADE DE SOFTWARE, Curitiba, Paraná, Brasil. Anais

VALENTE, J. A. & Almeida, F.J. **Visão Analítica da Informática na Educação: a questão da formação do professor.** Revista Brasileira de Informática na Educação, Sociedade Brasileira de Informática na Educação, nº 1, pg. 45-60. (1997). VALENTE, José Armando. **Computadores e Conhecimento: repensando a educação. Por que o computador na educação.** Gráfica central da Unicamp, Campinas-SP, 1993.

WERNECK, Claudia. **Ninguém mais vai ser bonzinho na sociedade inclusiva.** Rio de Janeiro: WVA, 1997.

YIN, Robert K. **Estudo de caso: planejamento e métodos.** 3. ed. Porto Alegre, RS: Bookman, 2005.

ANEXO

Endereço Eletrônico de alguns softwares Educativos

- http://www.edumatec.mat.ufrgs.br/software/soft_geometria.php

Na página eletrônica acima podemos encontrar os seguintes softwares

- ✓  CABRI-GEOMETRY
- ✓  CINDERELLA
- ✓  CURVE EXPERT
- ✓  DR GEO
- ✓  EUKLID
- ✓  GEOMETRIA DESCRITIVA
- ✓  GEOPLAN
- ✓  GEOSPACE
- ✓  GREAT STELLA
- ✓  POLY
- ✓  RÉGUA E COMPASSO
- ✓  SHAPARI
- ✓  SKETCHPAD

✓  S-LOGO

✓  WINGEOM

• http://www.edumatec.mat.ufrgs.br/software/soft_funcoes.ph

Na página eletrônica acima podemos encontrar os seguintes softwares

✓  GRAPHEQUATION

✓  GRAPHMATICA

✓  MATHGV

✓  MODELLUS

✓  RATOS

✓  VRUM - VRUM

✓  WINPLOT

• http://www.edumatec.mat.ufrgs.br/software/soft_algebra.php

Na página eletrônica acima podemos encontrar os seguintes softwares

✓  WINMAT

Outros links de softwares Educativos para o ensino da matemática.

- <http://www.gregosetroianos.mat.br/software.asp>
- <http://www.somatematica.com.br/softOnline.php>
- <http://www.somatematica.com.br/>

- <http://www.edumat.com.br/software-matematicos/>