

## Uma Metodologia de Apoio à Seleção de Softwares Educativos para o Ensino de Matemática

**Maria de Fátima C. de Souza; Mauro C. Pequeño;  
José Aires C. Filho y Cidcley T. de Souza**

*Universidade Federal do Ceará  
Centro Federal de Educação Tecnológica do Ceará*

Departamento de Computação  
60455-760. Fortaleza, Ce (Brasil)  
Email: fatimasouza@lia.ufc.br; mauro@vdl.ufc.br

Faculdade de Educação da Universidade Federal do Ceará  
60020-110. Fortaleza, Ce (Brasil)  
Email: j.castro@ufc.br

Centro Federal de Educação Tecnológica do Ceará  
60040-531. Fortaleza, Ce (Brasil)  
Email: cidcley@cefetce.br

**Resumo:** Em virtude ao intenso desenvolvimento de produtos de software educativo atualmente, a seleção desses produtos pelos professores tem se tornado uma tarefa cada vez mais difícil. Argumentamos nesse trabalho que mesmo existindo inúmeras estratégias de avaliação elas não são efetivamente aplicadas para realizar uma seleção baseada na comparação das características dos produtos de software avaliados. Nesse sentido, apresentamos uma nova abordagem para a avaliação de produtos de software educativo. Nessa abordagem propomos uma metodologia que tem como objetivo principal permitir a seleção de produtos de software educativo que possuam características funcionais semelhantes, através da utilização de métricas definidas pelo próprio avaliador e que são baseadas na teoria dos campos conceituais e formalizadas utilizando a técnica de pontos de casos de uso.

**Palavras chave:** Avaliação de Software, Ensino de Matemática, Software Educativo, Teoria dos Campos Conceituais, Pontos de Casos de Uso.

**Resumen:** En virtud del intenso desarrollo de productos de software educativos hoy, la selección de esos productos por los maestros ha se tornado una tarea cada vez más difícil. Argumentamos en este trabajo que mismo existiendo innúmeras estrategias de evaluación ellas no son efectivamente aplicadas para realizar una selección basada en la comparación de los rasgos de los productos de software evaluados. En este sentido, presentamos un nuevo abordaje para la evaluación de productos de software educativo. En este abordaje proponemos una metodología que tiene como objetivo principal permitir la selección de productos de software educativo que tengan rasgos funcionales semejantes, por medio del uso de métricas definidas por el propio evaluador y que son basadas en la teoría de los campos conceptuales y formalizadas utilizando la técnica de puntos de casos de uso.

**Palabras clave:** Evaluación de software, Enseñanza de Matemática, Software Educativo, Teoría de los Campos Conceptuales, Puntos de Casos de Uso.

---

## 1. Introdução

Tem-se observado, de forma cada vez mais intensa, o lançamento de produtos de software no mercado que, segundo seus fabricantes, poderiam auxiliar o trabalho de professores e facilitar a aprendizagem dos alunos. No entanto, grande parte destes programas é de baixa qualidade tanto técnica quando pedagógica, fato explicado pela dificuldade de se expressar conceitos pedagógicos na produção de software educativo (SE) (Tchounikine, 2002).

Desta forma, é fundamental que se faça uma avaliação sistemática da qualidade e dos efeitos de tais produtos de software antes de aplica-los na sala de aula. Porém, tal procedimento não ocorre com frequência, visto que muitas instituições de ensino adquirem programas que são utilizados pelos alunos sem uma avaliação prévia.

No entanto, pelo fato de existir uma grande variedade de SE disponíveis atualmente que podem ser utilizados para trabalhar um mesmo conteúdo, essa tarefa de avaliação torna-se bastante difícil, principalmente se considerarmos as técnicas de avaliação propostas atualmente na literatura. De fato, essas técnicas têm como ênfase a análise tanto de aspectos cognitivos como de aspectos de usabilidade, normalmente utilizando métricas simplificadas de forma a categorizar um determinado software ou mostrar que certos aspectos estão ou não presentes. Contudo, em virtude da expressividade limitada dessas métricas (por exemplo, pontuações que variam de 0 e

4), essas metodologias acabam não sendo eficazes no que tange a comparação de produtos de software<sup>1</sup>.

Nesse artigo, propomos uma nova metodologia de apoio à avaliação de SE. Nossa abordagem se baseia na idéia de formalizar tanto aspectos pedagógicos como técnicos de forma a possibilitar a realização de uma avaliação comparativa precisa entre SE. Nossa metodologia propõe a adoção e a especificação de métricas para aspectos relevantes definidas pelo próprio professor a serem aferidos nos SE que estão sendo avaliados.

Para alcançar esses objetivos, propomos nesse trabalho, a utilização da técnica de engenharia de requisitos de Pontos de Casos de Uso (Karner, 1993), largamente utilizada para estimar esforço de desenvolvimento de software, juntamente com a teoria dos campos conceituais de Vergnaud (Vergnaud, 1990), de modo a capturar, respectivamente, os requisitos funcionais e não funcionais, além de fatores pedagógicos, considerando uma perspectiva pedagógica construtivista.

## **2. Avaliação de Software Educativo e a Teoria dos Campos Conceituais de Vergnaud**

### **2.1. Modelos de Avaliação**

Existem vários modelos de avaliação de SE propostos na literatura. De uma maneira geral, podemos destacar os modelos de avaliação objetiva, formativa e os baseados em teorias da aprendizagem (Oliveira, Costa, & Moreira, 2001). Na avaliação objetiva uma equipe multidisciplinar analisa diferentes aspectos a serem considerados na qualidade do produto através de listas de critérios que representam a posição teórica de cada autor sobre a aprendizagem. Já a avaliação formativa é realizada durante a utilização do SE pelos próprios usuários através de entrevistas, questionários ou o acompanhamento de perto da interação do usuário com o software. Os modelos que utilizam teorias de aprendizagem se propõem a justificar a concepção do SE de acordo com alguma teoria de aprendizagem (Oliveira et al., 2001).

Contudo, tradicionalmente, os SE são analisados seguindo-se grades de categorias oriundas da engenharia de software que focalizam parâmetros gerais relativos à qualidade da interface, à coerência de apresentação dos conceitos e aos aspectos ergonômicos gerais dos sistemas (Gomes, Castro Filho, Gitirana, 2002). Esta avaliação é feita a partir da aplicação de tabelas de critérios nas quais aspectos como: consistência da representação, usabilidade, qualidade da interface, qualidade do feedback, etc., são

---

<sup>1</sup> Produto de software é um termo utilizado em Engenharia de Software para identificar um software que possa ser vendido/fornecido a um cliente.

considerados segundo uma escala de três ou quatro níveis (regular, bom, ótimo; ou regular, bom, muito bom e ótimo).

Como já citamos, as metodologias para avaliação de SE são basicamente voltadas à verificação da presença de determinados requisitos nos produtos de software avaliados. Em alguns casos há a utilização de algum tipo de pontuação simples onde esses pontos são levantados basicamente a partir de observações superficiais e informais. Por conta dessa informalidade, podemos concluir que essas metodologias não são tão eficazes quando o objetivo principal é comparar SE de mesmo domínio, principalmente no que tange a aspectos pedagógicos.

## 2.2. A Teoria dos Campos Conceituais

A teoria dos campos conceituais, proposta por Vergnaud (Vergnaud, 1990), é uma teoria cognitivista que oferece um referencial ao estudo do desenvolvimento cognitivo e da aprendizagem de competências complexas, particularmente aquelas implicadas nas ciências, levando em conta os próprios conteúdos do conhecimento e a análise conceitual de seu domínio. Embora Vergnaud esteja especialmente interessado nos campos conceituais das estruturas aditivas e das estruturas multiplicativas (Vergnaud, 1983), a teoria dos campos conceituais não é específica desses campos, nem da matemática. Vergnaud define o termo conceito como terna de conjuntos (Magina, Campos, Nunes & Gitirana, 2000) (S, I, R) onde:

- S é um conjunto de situações que dão sentido ao conceito;
- I é um conjunto de invariantes (objetos, propriedades e relações) sobre os quais repousa a operacionalidade do conceito, ou o conjunto de invariantes operatórios associados ao conceito, ou o conjunto de invariantes que podem ser reconhecidos e usados pelos sujeitos para analisar e dominar as situações do primeiro conjunto;
- R é um conjunto de representações simbólicas (linguagem natural, gráficos e diagramas, sentenças formais, etc.) que podem ser usadas para indicar e representar esses invariantes e, conseqüentemente, representar as situações e os procedimentos para lidar com elas.

Argumentamos que a definição de métricas precisas para a avaliação de requisitos funcionais e não funcionais de SE e a utilização dos conceitos de campos conceituais de Vergnaud para a elicitação de requisitos de domínio pode nos fornecer um mecanismo bastante eficiente para permitir a realização de uma escolha mais precisa de quais SE utilizar em sala de aula e qual o seu impacto pedagógico resultante da utilização desses SE relativos aos conceitos que estamos querendo ensinar.

### 3. Engenharia de Requisitos e os Pontos de Casos de Uso

Antes de apresentarmos a metodologia que propusemos é importante deixar claro quais as funções dos requisitos no processo de desenvolvimento de um software sob o ponto de vista formal da Engenharia de Requisitos. Além disso, ressaltaremos aqui a abordagem que utiliza pontos de casos de uso (Kraner, 1993) para a realização de estimativas de tempo e esforço de desenvolvimento de produtos de software baseado nas informações dos requisitos.

#### 3.1. Requisitos de Software

Segundo Sommerville (2004), podemos definir requisitos de um software da seguinte forma:

1. Requisitos Funcionais: São declarações de funções que o sistema deve fornecer, como o sistema deve agir a entradas específicas e como deve se comportar em determinadas situações. Em alguns casos, os requisitos funcionais podem também explicitamente declarar o que o sistema não deve fazer;
2. Requisitos Não funcionais: São restrições sobre os serviços ou as funções oferecidos pelo sistema. Entre eles destacam-se restrições sobre o processo de desenvolvimento, padrões, entre outros;
3. Requisitos de Domínio: São requisitos que definem funções específicas de determinados domínios de aplicação. Esses requisitos tanto podem ser funcionais como não funcionais.

Normalmente, os requisitos funcionais são verificados em um determinado software apenas pela sua presença ou ausência. Já os requisitos não funcionais, não podem ser avaliados da mesma forma que os funcionais. De fato, a verificação de requisitos não funcionais não é uma tarefa simples, pois muitas vezes esses requisitos são também tratados como funções a serem fornecidos pelo sistema, sendo essas funções difíceis de serem testadas nos produtos de software. Para resolver esses problemas, os requisitos não funcionais devem ser expressos quantitativamente (Sommerville, 2004), utilizando métricas que possam ser efetivamente testadas.

Para que possamos verificar a presença e a efetividade de um requisito funcional em um SE de matemática apenas a especificação de casos de uso não é suficiente. Nesse trabalho, propomos a utilização de pontos de casos de uso para extrair informações das especificações a serem usadas na avaliação de SE.

### 3.2. Pontos de Casos de Uso

Atualmente, a análise de sistemas orientados a objetos utiliza diagramas de casos de uso para descrever as funcionalidades do sistema de acordo com a forma de utilização por parte dos usuários. A técnica de análise de dimensão por casos de uso foi criada para permitir que seja possível estimar o tamanho de um software ainda na fase de levantamento de requisitos, utilizando-se dos próprios documentos gerados nesta fase de análise como subsídio para o cálculo dimensional. A técnica de estimativa por pontos de caso de uso foi proposta em 1993 por Gustav Karner (Karner, 1993). Essa técnica trata de estimar o tamanho de um software de acordo com o modo como os usuários o utilizarão, a complexidade de ações requerida por cada tipo de usuário e uma análise em alto nível dos passos necessários para a realização de cada tarefa. Os passos necessários para a geração da estimativa são:

1. Classificar os atores envolvidos em cada caso de uso, de forma a obter um somatório de pontos não-ajustado. O peso total dos atores do sistema (Unadjusted Actor Weight, ou UAW) é calculado pela soma dos produtos do número de atores de cada tipo pelo respectivo peso;
2. Calcular o peso bruto dos casos de uso (Unadjusted Use Case Weight, ou UUCW). O cálculo do UUCW é realizado como no cálculo de peso dos atores, somando-se os produtos da quantidade de casos de uso classificados em cada tipo pelo peso nominal do tipo em questão. O peso total não ajustado (Unadjusted Use Case Points, ou UUCP) é calculado pelo somatório entre os pesos de atores e casos de uso:  $UUCP = UAW + UUCW$ ;
3. Calcular os fatores de ajuste. O método de ajuste é constituído de duas partes - um cálculo de fatores técnicos (Technical Complexity Factor, ou TCF), cobrindo uma série de requisitos funcionais do sistema; e um cálculo de fatores de ambiente (Environment Factor, ou EF), requisitos não-funcionais associados ao processo de desenvolvimento; Esses fatores de ajuste são calculados utilizando as seguintes fórmulas:

$$TCF = 0.6 + (.01 \times TFactor)$$

$$EF = 1.4 + (-0.03 \times EFactor)$$

4. Finalmente, podemos calcular o valor total do sistema em (Use Case Points, ou UCP) utilizando-se da seguinte fórmula:

$$UCP = UUCP \times TFC \times EF$$

É válido ressaltar que os pesos utilizados para a definição da complexidade de atores, de casos de uso e para a definição de fatores técnicos e ambientais são apresentados na literatura de forma mais ou menos padronizada. Esses pesos são definidos de forma empírica de acordo com a complexidade de um caso de uso ou da importância de um ator.

#### **4. Avaliação Comparativa de Software Educativo**

O objetivo principal da nossa metodologia de avaliação é auxiliar o professor no processo de seleção de produtos de software para um determinado domínio da matemática, fornecendo um mecanismo mais detalhado de se aferir requisitos funcionais, não funcionais e pedagógicos-cognitivos que sejam relevantes para os produtos de software que estão sendo avaliados.

##### **4.1. Metodologia de Avaliação Comparativa**

Na Figura 1 apresentamos de forma simplificada a metodologia de avaliação que estamos propondo. Nesse diagrama podemos observar as atividades envolvidas no processo de avaliação.

1. As seguintes são atividades da metodologia:
2. Selecionar Software Candidato: Nessa primeira atividade o software a ser avaliado é escolhido;
3. Levantar Requisitos Funcionais: O levantamento dos requisitos funcionais deve ser realizado através da construção de diagramas de casos de uso considerando as funções/operações fornecidas pelo software que está sendo avaliado;

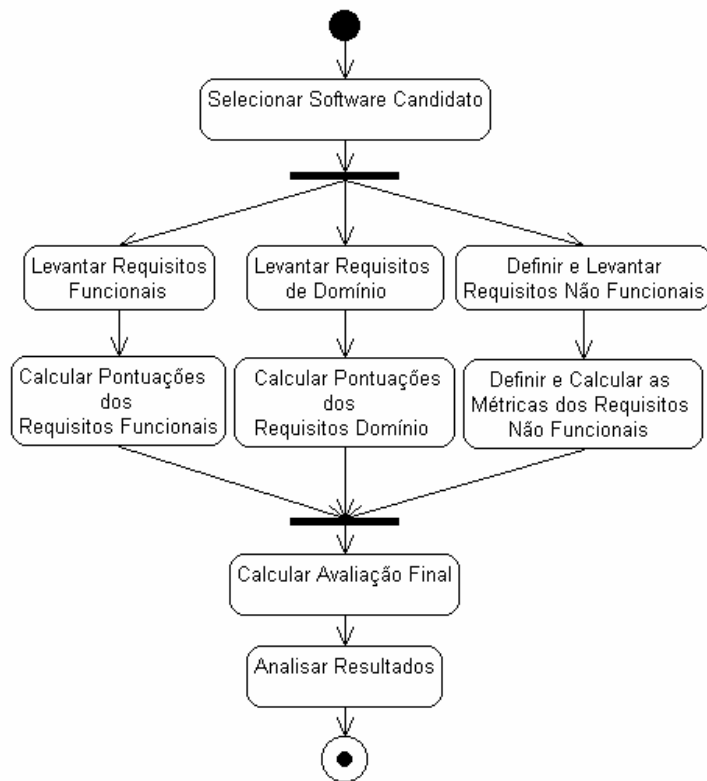


Figura 1. Atividades da Metodologia de Avaliação Comparativa

4. Levantar Requisitos de Domínio: Os requisitos de domínio dizem respeito aos aspectos pedagógicos/cognitivos importantes para o aprendizado do conteúdo abordado pelo software. Para realizar essa atividade utilizamos a teoria dos campos conceituais de Vergnaud, de modo a considerarmos mais precisamente os esquemas necessários para a aprendizagem dos conceitos matemáticos trabalhados no software. Essa atividade é realizada mediante o preenchimento de um formulário de avaliação de requisitos de domínio, que denominamos de Tabela de Avaliação de Requisitos de Domínio;
5. Definir e Levantar Requisitos Não Funcionais: Nessa etapa são identificados os requisitos não funcionais que devem ser utilizados para avaliar o software. Esses requisitos devem ser escolhidos se observando os objetivos definidos para o software no contexto dos conteúdos que estão sendo utilizadas na sala de aula.



6. Calcular Pontuações dos Requisitos Funcionais: Depois de devidamente especificados os casos de uso dos requisitos funcionais, realizamos o cálculo das pontuações desses requisitos. Esse cálculo é fornecido aplicando a técnica de pontos de casos de uso. Como resultado dessa atividade, teremos o valor funcional estimado do software;
7. Calcular Pontuações dos Requisitos Domínio: Os requisitos pedagógicos são calculados nessa atividade através de cálculos realizados baseado nos valores da Tabela de Avaliação de Requisitos de Domínio. Como resultado dessa atividade, teremos o valor pedagógico/cognitivo estimado do software;
8. Definir e Calcular as Métricas dos Requisitos Não Funcionais: Baseado nos requisitos não funcionais levantados na atividade 3 devemos realizar a definição e o cálculo das métricas utilizadas para aferir os requisitos não funcionais escolhidos;
9. Calcular Avaliação Final: O cálculo final da avaliação do SE é realizado utilizando os valores levantados nas atividades anteriores. Esse valor final irá ser composto pelos fatores funcionais, não funcionais e por requisitos pedagógicos, permitindo que seja realizada uma avaliação mais precisa por parte do professor;
10. Analisar Resultados: Com os resultados das avaliações de SE de mesmo domínio, o avaliador poderá escolher os produtos de software que venham melhor se adequar às métricas que foram aferidas.

#### **4.2. Especificação de Casos de Uso e Cálculo de Métricas Funcionais**

Nesse trabalho, para a especificação dos casos de usos dos SE de forma a capturar as ações dos usuários, algumas restrições devem ser observadas. Essas restrições tanto dizem respeito aos tipos de atores que podem fazer parte dos cenários quanto à definição da importância dos atores e dos casos de uso. Essa importância será utilizada para a definição de pontuações para o cálculo dos pontos de casos de uso para o software. Dessa forma, temos as seguintes restrições:

1. Devem ser utilizados apenas dois atores (Professor, Aluno), com pesos 2 e 3 respectivamente: essa restrição mostra que estamos valorizando apenas os papéis dos professores e alunos para o software que está sendo avaliado. Os pesos para cada ator são definidos segundo a importância da cada um no processo. Nesse caso propomos a definição de um peso mais alto para o aluno, de modo a priorizarmos a escolha de produtos de software que

ofereçam características funcionais direcionadas à esse ator. Contudo, também pontuamos positivamente a participação do professor como mediador na utilização de um SE, como apresentado em Oliveira, Gomes e Borges Neto (2001b).

2. Os casos de uso devem possuir seus pesos definidos de acordo com a sua importância para o tópico de estudo, variando de 0 a 5: para um software de ensino de equações para turmas iniciais, a apresentação dos passos do cálculo pode ser pontuado com 5. Já esse mesmo fator não é relevante para um software de nível superior, podendo ser pontuado com 1, ou até mesmo 0.
3. Desse modo, o cálculo final para as métricas dos requisitos funcionais dos pontos de casos de uso será composto apenas pelas Unadjusted Use Case Points (UUCP), seguindo a fórmula apresentada anteriormente:

$$UUCP = UAW + UUCW$$

#### 4.3. Levantamento e Cálculo de Pontuações de Requisitos de Domínio

Os requisitos de domínio, no caso desse trabalho, são requisitos relativos a fatores pedagógicos/cognitivos. Abordamos esse tema utilizando a teoria dos campos conceituais de Vergnaud. Como apresentado na seção 2, um campo conceitual pode ser definido como: um conjunto de situações que dão sentido a um conceito; um conjunto de invariantes e um conjunto de representações simbólicas para o conceito. Se utilizarmos o campo conceitual da álgebra, esses fatores podem ser representados da seguinte forma:

- Situações: aqui devem ser relacionados as situações e os problemas a resolver, visto que esses fatores é que dão sentido ao conceito que está sendo tratado. Por exemplo, calcular a área de um quarto com o propósito de revesti-lo;
- Invariantes: são os princípios lógicos subjacentes ao próprio conceito. No caso da álgebra, podem ser considerados como invariantes: incógnita, variável, equação, inequação, fórmula e função;
- Representações: é a forma como que os invariantes podem ser representados de modo a permitir a manipulação dos conceitos. Essas representações podem ser formais, tais como símbolos de igualdade "=", ou informais, tais como figuras e objetos (representação icônica).

Para a realização do levantamento dos requisitos de domínio, utilizamos nesse trabalho, uma tabela, denominada Tabela de Avaliação de Requisitos de Domínio. Nessa tabela observaremos os três aspectos definidos por Vergnaud para a construção de um conceito. Sendo que são pontuados positivamente os produtos de software que utilizarem essas representações. Além disso, pontuamos a contextualização das representações com relação ao perfil dos alunos que utilizarão o software e pela inter-relação entre os aspectos. A Tabela 1 deve ser utilizada para a realização do levantamento de requisitos de domínio para o campo conceitual da álgebra.

<i>Nível de Ensino Abordado</i>	<input type="checkbox"/> Ensino Infantil <input type="checkbox"/> Ensino Fundamental <input type="checkbox"/> Ensino Médio <input type="checkbox"/> Ensino Superior <input type="checkbox"/> Não se Aplica
<i>Situações Apresentadas</i>	<input type="checkbox"/> Problemas de Equações <input type="checkbox"/> Problemas de Inequações <input type="checkbox"/> Funções <input type="checkbox"/> Sistemas de Equações <input type="checkbox"/> Sistemas de Inequações <input type="checkbox"/> Outro. _____ <input type="checkbox"/> Não apresenta situação
<i>Invariantes Envolvidas</i>	<input type="checkbox"/> Variáveis <input type="checkbox"/> Incógnitas <input type="checkbox"/> Equações <input type="checkbox"/> Inequações <input type="checkbox"/> Funções <input type="checkbox"/> Outro. _____
<i>Representações Utilizadas</i>	1 - Formal    2 - Linguagem Natural    3 - Icônica Variáveis <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Incógnitas <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Equações <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Inequações <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Funções <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<i>Correlações</i>	Representações são compatíveis com o Nível de Ensino abordado? <input type="checkbox"/> Existem relações diretas observáveis entre as Representações e os Invariantes? <input type="checkbox"/>

Tabela 1. Requisitos de Domínio

Para realizar o cálculo de pontuações para os requisitos de domínio, consideraremos esses tipos de requisitos como fatores técnicos (*Technical Factors*)(Referência de UCP) dos pontos de casos de uso. Assim, utilizaremos a tabela de pesos (Tabela 2) para os fatores. Na Tabela 3, apresentamos a forma como se avaliar cada um dos fatores no software.

Fator	Peso
Situação	5
Invariantes	2
Representações	1.5
Correlações	5

Tabela 2. Pesos dos Fatores Técnicos

Fator	Cálculo dos Valores
Situação	Somar 1 ponto para cada situação selecionada na Tabela 1. Usar 0 se o item escolhido for "Não apresenta situação".
Invariantes	Somar 1 ponto para cada invariante selecionada na Tabela 1
Representações	Somar 1 ponto para cada Representação selecionada na Tabela 1
Correlações	Somar 2 pontos para cada Correlação selecionada na Tabela 1

Tabela 3 - Avaliação de Fatores Técnicos

O cálculo final dos fatores técnicos relacionados aos requisitos de domínio levantados a partir de uma perspectiva dos campos conceituais de Vergnaud deve ser realizado utilizando a seguinte fórmula:

$$TCF = 0.6 + (.01 \times TFactor)$$

Onde TCF é o Fator de Complexidade Técnica (*Technical Complexity Factor*) e Tfactor é o Fator Técnico Total (*Total Technical factor*) que é calculado pela soma dos produtos dos pesos de cada fator técnico pelo valor levantado pela Tabela 1.

Os valores definidos para os pesos na Tabela 2, segue a variação sugerida em (Kraner, 1993), que foi definida de 0 a 5. No nosso caso, como estamos realizando a

avaliação de requisitos de domínio e nesse caso o que é mais importante são os fatores pedagógicos subseqüentes a esse conceito, avaliamos com valores maiores os SE que ofereçam situações que forneçam o aprendizado de um conceito e os que possuem correlações entre essas situações e sua representações. Por esse motivo o fator Situação e Correlações possuem peso 5. Os demais valores

#### 4.4. Definição, Levantamento e Cálculo de Pontuações de Requisitos Não Funcionais

Na fase de definição dos requisitos não funcionais escolhemos que requisitos não funcionais são importantes para os produtos de software que estão sendo avaliados. Por questão de simplificação será utilizada nesse trabalho apenas a usabilidade como requisito não funcional a ser avaliado. Para o requisito de usabilidade devem ser definidas e especificadas as métricas a serem utilizadas. Particularmente para usabilidade podemos utilizar métricas como (Nielsen, 2001):

1. Tempo para completar uma tarefa;
2. Frequência da utilização do help ou da documentação;
3. Número de clicks do mouse;
4. Distância percorrida pelo mouse.

Para definir a importância dessas métricas em relação à avaliação que estamos realizando, devemos fornecer um peso para cada métrica. Na metodologia que estamos propondo, cada métrica de requisitos não funcionais está sendo tratada como um fator ambiental (*Environment Factor*) definido na técnica de pontos de casos de uso. Na Tabela 4, apresentamos as pontuações para esses requisitos:

Requisito Não Funcional	Peso
Tempo para completar uma tarefa	5
Frequência da utilização do <i>help</i> ou da documentação	2
Número de clicks do mouse	3
Distância percorrida pelo mouse	3

Tabela 4. Pesos dos Fatores Ambientais

Para realizar o levantamento dos requisitos não funcionais nos produtos de software que estão sendo avaliados temos que utilizar cada software e anotar as informações necessárias para cada requisito. Para automatizar essa tarefa

apresentamos em (Souza, Castro Filho & Pequeno, 2004) uma ferramenta para o suporte a extração de requisitos não funcionais de SE.

O cálculo do valor relativo aos requisitos não funcionais é realizado pela fórmula seguinte, que é utilizada para o cálculo de fatores ambientais dos pontos de casos de uso:

$$EF = 1.4 + (-0.03 \times EFactor)$$

Onde EF significa fator ambiental (*Environment Factor*) e EFactor significa Fator Ambiental Total (*Total Environment Factor*).

Os valores para cada requisito a ser elicitado nos produto de software, devem variar entre 0 e 5. Esses valores devem ser ponderados de acordo com os resultados mais elevados, e ajustados para que fiquem dentro da faixa de variação. Por exemplo, se for avaliado um requisito não funcional entre os produtos que estão sendo comparados tendo 20 como seu maior valor, esse deve ser considerado como 5 e, utilizando uma regra de três simples, todos os demais valores devem ser ajustados.

#### 4.5. Cálculo da Avaliação Final e Análise de Resultados

Depois de realizar todos os levantamentos de requisitos funcionais, de domínio e não funcionais, a atividade seguindo do processo de avaliação é calcular o valor final do software. Esse valor final, também utilizando uma abordagem baseada em pontos de casos de uso, é fornecido pela fórmula:

$$AUCP = UUCP \times TCF \times EF$$

Onde AUCP significa Pontos de Casos de Uso Ajustados (*Adjusted Use Case Points*). Utilizando esses valores calculados para cada software candidato, podemos ter uma avaliação precisa de diversos aspectos do software, desde fatores puramente operacionais até fatores pedagógicos/cognitivos.

Para validar a abordagem apresentada nesse trabalho, apresentamos um exemplo simplificado da avaliação comparativa de dois SE utilizados para auxiliar no ensino de álgebra.

#### 4.6. Seleção dos Produtos de Software Candidatos

Os produtos de software escolhidos para apresentar a nossa abordagem de avaliação comparativa são dois produtos de software que apresentam características funcionais e não funcionais bastante semelhantes, ou seja, a priori seria bastante difícil para o professor a decisão de qual software usar em suas aulas.

O primeiro software utilizado foi o Jogo da Balança (Luca, 1995). Esse software permite, através de uma balança de pratos, trabalhar com o conceito de equações e comparação entre quantidades. Desenvolvido em sete níveis de dificuldade, o software permite a utilização de representações gráficas de maçãs, carneiros, entre outras, para mostrar a igualdade entre quantidades nos dois pratos da balança. O aluno, através de operações de adição, subtração, multiplicação e divisão, pode manipular as quantidades nos pratos até conseguir o equilíbrio. Nos níveis mais avançados, podem ser utilizados pesos para manipular quantidades maiores.

O segundo software utilizado foi o Balança Interativa (Castro Filho, 2004). Esse software também trabalha com uma balança de pratos e, como no Jogo da Balança, trabalha com os conceitos de equações. Além disso, o software também trabalha com conceitos de inequações, permitindo a comparação entre quantidades diferentes e mostrando a noção de “maior que” e “menor que”. Desenvolvido em dez níveis diferentes, esse software permite a utilização de pesos com valores desconhecidos (representados por letras) e de pesos com valores conhecidos, sendo que o aluno deve, de um lado da balança, colocar os pesos desconhecidos e do outro os pesos conhecidos até que o equilíbrio ocorra. Nesse software, a expressão matemática referente à disposição de pesos poderá ser apresentada em qualquer tempo. Em níveis mais elevados essas expressões podem ser manipuladas diretamente.

#### 4.7. Especificação de Casos de Uso e Cálculo de Métricas Funcionais

Os dois produtos de software que estão sendo avaliados foram desenvolvidos para serem utilizados pelos alunos, com a mediação do professor, para o auxílio do desenvolvimento de aptidões algébricas. Contudo, para o cálculo do valor dos requisitos funcionais para esses produtos, apenas o ator Aluno será considerado, visto que o ator professor não irá realizar tarefas diretas no software.

Inicialmente, para o cálculo do valor dos requisitos funcionais do Jogo da Balança, podemos identificar os seguintes casos de uso com seus respectivos pesos:

Caso de Uso	Peso
Escolher Pesos	3
Exibir Expressão Algébrica	3
Mudar Nível	2

Já a avaliação de requisitos funcionais para o Balança Interativa foi considerada utilizando os seguintes casos de uso com seus respectivos pesos:

Caso de Uso	Peso
Escolher Pesos	5

Exibir Expressão Algébrica	5
Mudar Nível	2
Limpar Pratos	0

Na utilização do Jogo da Balança os pesos são gerados dos dois lados dos pratos. Em seguida é permitida a realização de operações de adição ou subtração de unidades em cada lado do prato. Então podemos verificar se os pratos estão equilibrados através de um botão que mostra a expressão matemática correspondente.

No Balança Interativa, os pesos podem ser escolhidos livremente dos dois lados dos pratos, sendo que é possível utilizar pesos com valores desconhecidos (representados por letras) ou pesos com valores conhecidos. Em seguida o aluno pode manipular esses pesos e trabalhar com equilíbrio entre os pratos, substituindo os pesos livremente. Um outro fator importante é a utilização de um botão que pode ser acionado em qualquer tempo para apresentar a expressão algébrica formal relativa à configuração da balança.

Por conta de uma maior flexibilidade na escolha dos pesos a Balança Interativa tem esse requisito com um valor maior que o correspondente no Jogo da Balança. Já pelo fato de ser permitida a verificação das expressões algébricas formais a qualquer tempo, o Balança Interativa também tem esse requisito mais bem avaliado que no Jogo da Balança. Desse modo, são calculados os pontos de casos de uso para cada software da seguinte forma:

$$\begin{aligned} & \text{Jogo da Balança:} \\ & \text{UAW} = 3 \times 1 = 3 \\ & \text{UUCW} = 2 \times 3 + 1 \times 3 = 9 \\ & \text{UUCP} = \text{UAW} + \text{UUCW} = 12 \end{aligned}$$

$$\begin{aligned} & \text{Balança Interativa:} \\ & \text{UAW} = 3 \times 1 = 3 \\ & \text{UUCW} = 2 \times 5 + 1 \times 2 + 1 \times 0 = 12 \\ & \text{UUCP} = \text{UAW} + \text{UUCW} = 15 \end{aligned}$$

Até aqui, podemos concluir que funcionalmente o Balança Interativa fornece mais recursos para o aluno. Isso implica dizer que o Balança Interativa realiza melhor as tarefas que elicitamos nos casos de uso de acordo com os valores de importância que demos para cada caso de uso.



#### 4.8. Levantamento e Cálculo de Pontuações de Requisitos de Domínio

Para um software de apoio ao aprendizado de álgebra, além dos fatores funcionais, também os fatores pedagógicos subjacentes aos conceitos devem ser tratados de alguma forma. Nesse sentido, abordamos a avaliação desse tipo de requisito, aqui tratado com requisito de domínio, utilizando a teoria dos campos conceituais de Vergnaud.

Aplicando a Tabela de Avaliação de Requisitos de Domínio, que apresentamos na seção 4, obtivemos os seguintes resultados:

*Jogo da Balança*

Fator	Peso	Valor Levantado	Resultado
Situação	5	1	5
Invariantes	2	1	2
Representações	1.5	2	3
Correlações	5	2	10
			<b>20</b>

*Balança Interativa*

Fator	Peso	Valor Levantado	Resultado
Situação	5	1	5
Invariantes	2	3	6
Representações	1.5	4	6
Correlações	5	4	20
			<b>37</b>

Na avaliação sobre Situação, ambos os produtos trabalham com a abstração de uma balança para trabalhar com os conceitos algébricos, dessa forma ambos foram avaliados da mesma maneira.

Com relação aos Invariantes os dois produtos trabalham com a noção de equações através da abstração do equilíbrio entre os pratos. O Balança Interativa utiliza também as noções de inequações, através da manipulação dos pesos enquanto a balança está em desequilíbrio, e de incógnitas, com a utilização de pesos de valores desconhecidos.

É utilizada uma representação baseada em figuras (balança, pesos, maçãs, tartarugas, etc.) para manipular as informações de Invariantes utilizadas nos dois produtos, bem como a representação formal para as equações. Adicionalmente, o Balança Interativa, faz uso de representações formais para incógnitas e inequações.

As correlações observadas nos dois produtos indicam que suas representações são compatíveis com os níveis educacionais onde eles são aplicados. Já o Balança Interativa apresenta uma melhor relação às representações e os invariantes, permitindo a visualização direta dos conceitos de inequações e incógnitas.

O cálculo final dos resultados da avaliação de requisitos de domínio é o seguinte:

Jogo da Balança:

$$TCF = 0.6 + (.01 \times 20) = 0.6 + 0.2 = 0.8$$

Balança Interativa:

$$TCF = 0.6 + (.01 \times 37) = 0.6 + 0.37 = 0.97$$

Podemos concluir com a avaliação dos requisitos de domínio que o Balança Interativa apresenta melhor uma avaliação com relação a fatores pedagógicos

#### **4.9. Definição, Levantamento e Cálculo de Pontuações de Requisitos Não Funcionais**

Para a avaliação dos requisitos não funcionais realizamos a tarefa de verificar uma equação simples (ex.:  $2 + X = 5$ ). No Jogo da Balança essa tarefa é realizada utilizando as operações de adição ou subtração para tentar igualar a quantidade de elementos nos pratos da balança. Assim, é escolhida uma operação e então é definida a quantidade de elementos que deverão ser adicionados ou subtraídos (dependendo da operação escolhida) clicando nas setas próximas à caixa de texto fornecida para este fim até, atingir o valor desejado.

No Balança Interativa existem várias incógnitas cujos valores devem ser descobertas, para essa avaliação utilizaremos apenas uma incógnita, de modo a ser possível realizar uma comparação mais aproximada com o Jogo da Balança. Para trabalhar uma equação simples, devemos inicialmente escolher um peso sem valor definido e coloca-lo em um dos pratos. Isso é feito clicando sobre o peso e arrastando para o prato desejado. Em seguida devemos selecionar os pesos com valores definidos e arrastar (ou retirar-los) para pratos até alcançar o equilíbrio.

Na definição das métricas não funcionais para a avaliação utilizamos três métricas simples. A primeira o número de clicks no mouse para a realização de uma única operação, cujo peso que definimos é 2, visto que esse é um critério relativamente relevante para a usabilidade de um software, principalmente para o público ao qual ele se destina.

A segunda métrica foi o tempo necessário para realizar uma operação simples, que recebeu peso 2, por ser esse um fator que envolve diretamente a motivação do aluno, visto que o tempo longo na realização de uma operação pode fazer o aluno perder a atenção no conteúdo.

A terceira métrica foi a distância percorrida pelo mouse na realização da tarefa. Essa métrica está ligada à dificuldade da operação de “arrastar-e-soltar” para pessoas que não utilizam computador freqüentemente ou com coordenação motora ainda em formação. Assim, esse critério foi pontuado com peso 3.

Vale ressaltar que existem diversas outras métricas que poderíamos ser utilizadas, contudo resolvemos colocar nesse trabalho apenas essas três métricas, que já são suficientes para exemplificar a aplicação da metodologia proposta.

Os resultados obtidos na avaliação foram os seguintes:

*Jogo da Balança*

<b>Requisito Não Funcional</b>	<b>Peso</b>	<b>Valor Levantado</b>	<b>Resultado</b>
Nº de clicks no mouse	2	5	10
Tempo de uma operação	2	3.75	7.5
Distancia percorrida pelo mouse	3	2.67	8.01
			<b>25.51</b>

*Balança Interativa*

<b>Requisito Não Funcional</b>	<b>Peso</b>	<b>Valor Levantado</b>	<b>Resultado</b>
Nº de clicks no mouse	2	0.71	1.42
Tempo de uma operação	2	5	10
Distancia percorrida pelo mouse	3	5	15
			<b>26.42</b>

Na avaliação do número de clicks do mouse para a realização de uma operação simples no Jogo da Balança, são necessários 1 click para se escolher a operação, mais uma quantidade de clicks definida pela número a ser adicionado ou subtraído de um dos pratos e um outro click para se verificar o resultado. Assim, temos um total de 2 + VI (Valor da Incógnita) que estamos procurando. Para realizar essa avaliação, vamos supor que o a incógnita tem valor 5, assim teremos um total de 7 clicks.

Para a avaliar o número de clicks realizados no Balança Interativa, não consideramos os clicks necessários para selecionar os pesos. Assim, consideramos apenas um click para apresentar o resultado da expressão algébrica formal quando atingimos o resultado desejado.

Realizando o ajuste dos valores para a faixa de aceitação (entre 0 e 5), temos que o valor 7 passa a ser 5 e o valor 1 passa a ser 0.71.

Com relação ao tempo necessário para realizar uma operação simples no Jogo da Balança podemos verificar facilmente que isso dependerá do conhecimento do aluno na realização de operações aritméticas simples. Assim, quanto melhor o domínio do aluno dessas operações, menor será o tempo necessário para equilibrar os pratos. Isso decorre do fato de que basta apenas adicionar ou subtrair valores aos pratos para realizar a operação. Para essa avaliação utilizamos um valor empírico de 15s.

Já para realizar a avaliação do tempo necessário para a realização de uma operação simples no Balança Interativa não podemos considerar somente o conhecimento em aritmética. Nesse software utilizamos incógnitas (representados por letras) que devemos tentar equilibrar usando os pesos com valores conhecidos. Assim, a sorte também é um fator importante, visto que o equilíbrio é conseguido se observando os valores dos pesos que escolhemos aleatoriamente. Também utilizamos um valor empírico médio<sup>2</sup> de 20s para avaliar esse requisito.

Realizando o ajuste nos valores considerando a escala de 0 a 5, temos que o valor 20 será avaliado como 5 e o valor 15 será avaliado como 3.75.

O terceiro e último critério utilizado, que é a distância percorrida pelo mouse para realizar uma operação, foi aferida no Jogo da Balança considerando as distâncias percorridas entre os botões de escolha das operações e de verificação de valor, e entre a caixa de texto de definição de quantidades. O valor médio, calculado da mesma forma que no requisito anterior, foi aferido nesse software como 24cm. No Balança Interativa, pelo fato de todos os pesos serem manipulados através de operações de “arrastar-e-soltar”, esse requisito teve seu valor médio aferido como 45cm.

Realizando o ajuste nos valores considerando a escala de 0 a 5, temos que o valor 45 será avaliado como 5 e o valor 24 será avaliado como 2.67.

O resultado final dos cálculos para os requisitos não funcionais é o seguinte:

Jogo da Balança:

---

<sup>2</sup> Essa média foi calculada pela média dos tempos de utilização do software para realizar 10 operações simples.

$$EF = 1.4 + (-0.03 \times 25.51) = 1.4 - 0.7653 = 0.6347$$

Balança Interativa:

$$EF = 1.4 + (-0.03 \times 26.42) = 1.4 - 0.7926 = 0.6074$$

Os resultados obtidos na avaliação de requisitos não funcionais nos levam a concluir que, com relação a esse tipo de requisito, o Jogo da Balança leva vantagem sobre o Balança Interativa.

É válido ressaltar que os valores não funcionais aferidos estão tecnicamente ligados ao perfil do avaliador. Contudo, como as avaliações que são apresentadas nesse exemplo foram realizadas por um mesmo avaliador, e como temos a intenção de comparar os produtos de software, as aproximações que fizemos não afetam diretamente os resultados.

##### 5. Cálculo da Avaliação Final e Análise de Resultados

Com base nos resultados dos requisitos funcionais, de domínio e não funcionais, podemos efetuar os cálculos necessários à obtenção dos resultados finais da avaliação para os dois produtos de software. Esses cálculos são realizados da seguinte maneira:

Jogo da Balança:

$$AUCP = 12 \times 0.8 \times 0.6347 = 6.09312$$

Balança Interativa:

$$AUCP = 15 \times 0.97 \times 0.6074 = 8.83767$$

Esses resultados nos mostram que de acordo com a nossa metodologia o produto de software Balança Interativa comparativamente é melhor que o produto de software Jogo da Balança.

Somente essa informação já seria útil para a realização da seleção desse software para ser utilizado em sala de aula. Contudo, a nossa metodologia permite uma visualização mais ampla dos aspectos relevantes à tarefa de seleção de um SE. De fato, se observarmos os valores finais calculados, podemos perceber que ele é uma composição de resultados. Cada um desses resultados já pode, por si só, fornecer informações importantes para a seleção do SE. Assim, podemos priorizar os requisitos de domínio (pedagógico-cognitivos) na seleção, sendo que, por exemplo, podemos selecionar um SE que seja rejeitado pelo cálculo final das pontuações e que tenha um valor de requisitos de domínio superior.

## 6. Conclusão

Para que todo o ferramental tecnológico disponível atualmente possa de fato fazer parte do cotidiano das salas de aula, não basta que os professores aprendam a utilizar o computador. Os inúmeros SE disponíveis atualmente fornecem mecanismos que podem auxiliar de modo efetivo a aprendizagem de conceitos, dos mais simples aos mais elaborados. Contudo, essa disponibilidade de SE tem causado a proliferação de produtos dos mais diversos tipos. Atualmente, podemos encontrar inúmeros SE que trabalham conceitos semelhantes, o que tem dificultado a seleção de um determinado produto de software a ser utilizado em sala de aula.

Nesse trabalho, argumentamos que é importante o tratamento preciso de aspectos funcionais e não funcionais como um modo de realizar uma avaliação voltada para a seleção de produtos de SE de mesmo domínio de aplicação. Além de ser necessário a consideração de aspectos pedagógicos/cognitivos nesse processo de avaliação. Dessa forma, apresentamos uma metodologia de suporte a avaliação comparativa de SE. Essa metodologia considera que o próprio avaliador deve escolher que requisitos são importantes para ele e, a partir daí, realizar a definição e especificação de métricas para serem aferidas nos produtos de software que estão sendo comparados.

Na metodologia que propomos, essas métricas são ajustadas para serem utilizadas como insumos para a aplicação da técnica de pontos de casos de uso, amplamente utilizada na realização de estimativas de esforço de desenvolvimento de produtos de software. Através dessa abordagem, conseguimos gerar uma pontuação para cada software, permitindo a realização de uma seleção mais refinada e que considera diversos aspectos relevantes aos SE que vão além da sua interface.

## 7. Agradecimento

A autora Maria de Fátima C. de Souza agradece o apoio financeiro concedido pela FUNCAP (Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico) para a realização dessa pesquisa.

## 8. Referências

- Castro Filho, J. A. (2004). Balança Interativa [Software].
- Gomes, A. S., Castro Filho, J. A., Gitirana, V, Spinillo, A. (2002). Avaliação de Software Educativo para o Ensino de Matemática. In WIE 2002 Workshop Brasileiro de Informática Educativa. Florianópolis: SBC.

- Magina, S., Campos, T. M. M., Nunes, T., & Gitirana, V. (2000). *Repensando Adição e Subtração: Contribuições da Teoria dos Campos Conceituais*. São Paulo: PROEM.
- Nielsen, J. (2001). *Usability Metrics*. [on-line]. <http://www.useit.com/alertbox/20010121.html> [29 de outubro de 2004].
- Oliveira, C. C., Costa, J. W., & Moreira, M. (2001). *Ambientes Informatizados de Aprendizagem: Produção e Avaliação de Software Educativo*. Campinas: Papirus.
- Oliveira, S. S., Gomes, A. S., & Borges Neto, H. (2001b). *Avaliação de Software Educativo para o Ensino de Matemática - O Caso das Estruturas Aditivas*. In *Encontro de Pesquisa Educacional do Nordeste: Educação, Desenvolvimento Humano e Cidadania*.
- Karner, G. (1993). *Metrics for Objectory*. (Diploma Thesis, University of Linköping, Sweden).
- Luca (1995). *Jogo da Balança [Software]*. Positivo Informática.
- Sommerville, I. (2004). *Software Engineering (7th ed)*. Harlow: Addison-Wesley.
- Souza, M. F. C., Castro Filho, J. A., & Pequeno, M. C. (2004). *Uma Abordagem Semi-Automática para a Avaliação Comparativa de Software Educacional de Matemática*. In *XV Simpósio Brasileiro de Informática na Educação - SBIE'2004*. Manaus:SBC.
- Tchounikine, P. (2002). *Pour une Ingénierie des Environnements Informatiques pour l'Apprentissage Humain*. In *Revue I3*, Vol. 2 (pp. 59-95).
- Vergnaud, G. (1990) *La Théorie des Champs Conceptuels*. *Recherches en Didactique des Mathématiques*.
- Vergnaud, G. (1983). *Multiplicative Structures*. In R. Lesh & M. Landau (Eds.), *Acquisition of Mathematical Concepts and Processes*. New York: Academic Press.