

# SISTEMA TUTOR DE COMPUTAÇÃO GRÁFICA

Bruno Pereira Gibicoski, Dalton Solano dos Reis – Orientador

Curso de Bacharel em Ciência da Computação

Departamento de Sistemas e Computação

Universidade Regional de Blumenau (FURB) – Blumenau, SC – Brasil

bgibicoski@furb.br, daltonreis@furb.br

**Resumo:** O resumo é uma apresentação concisa dos pontos relevantes de um texto. Informa suficientemente ao leitor, para que este possa decidir sobre a conveniência da leitura do texto inteiro. Deve conter OBRIGATORIAMENTE o OBJETIVO, METODOLOGIA, RESULTADOS e CONCLUSÕES. O resumo não deve ultrapassar 10 linhas e deve ser composto de uma sequência corrente de frases concisas e não de uma enumeração de tópicos. O resumo deve ser escrito em um único texto corrido (sem parágrafos). Deve-se usar a terceira pessoa do singular. As palavras-chave, a seguir, são separadas por ponto, com a primeira letra maiúscula. Caso uma palavra-chave seja composta por mais de uma palavra, somente a primeira deve ser escrita com letra maiúscula, sendo que as demais iniciam com letra minúscula, desde que não sejam nomes próprios.]

**Palavras-chave:** Ciência da computação. Computação gráfica. Ensino e aprendizagem. Unity.

## 1 INTRODUÇÃO

Você sabe matemática básica? De acordo com Fernandes (2016), o número de alunos brasileiros sem o conhecimento de matemática básica tem diminuído consideravelmente. Contudo, a posição do Brasil comparada ao resto do mundo é ainda considerada inferior. Idoeta (2018) critica o sistema de ensino brasileiro, afirmando que os alunos de hoje em dia não aprendem conceitos matemáticos, mas aprendem como decorar fórmulas.

As questões do Exame Nacional do Ensino Médio (ENEM) que são consideradas difíceis por sua complexidade, mas que apresentam resolução por fórmula tem um índice de acerto maior do que questões consideradas fáceis e dependem de conceitos de matemática básica (IDOETA, 2018).

Estes alunos que não desenvolveram uma boa base de matemática, acabam encontrando dificuldades no seu dia a dia, inclusive, durante um curso superior. Um exemplo de tal caso é a matéria Computação Gráfica do curso de Ciências da Computação que, para realizar simples interações com os objetos gráficos, é necessário o conhecimento de matrizes, circunferências, funções, entre outros.

Para facilitar o entendimento dos estudantes na matéria de Computação Gráfica, este trabalho pretende desenvolver uma aplicação que auxilie na aprendizagem de Computação Gráfica, explicando os conceitos matemáticos utilizados nesta matéria de uma maneira simplificada.

## 2 FUNDAMENTAÇÃO TEÓRICA

Nesse capítulo será apresentado a fundamentação teórica, no qual o item 2.1 irá tratar dos conceitos utilizados no desenvolvimento desse trabalho e no item 2.2 apresentará os trabalhos correlatos com o trabalho relatado nesse artigo.

### 2.1 CONCEITOS

Este capítulo descreve brevemente os assuntos que fundamentarão o estudo a ser realizado: sistemas tutores, Computação Gráfica e os conceitos matemáticos utilizados em seu meio.

#### 2.1.1 SISTEMAS TUTORES

Toledo (2015) afirma que um software tutorial é um sistema que contém informações organizadas em uma ordem pedagógica e é composto por instruções programadas, em que a interação entre aluno e computador é realizada através da leitura, audição ou escrita de informações. Segundo Toledo (2015) a principal proposta de um software educacional é dar suporte ao processo de ensino-aprendizagem nos diferentes conteúdos ministrados, tornando o aluno um ser ativo na construção do conhecimento.

O processo de ensino-aprendizagem com o uso de aplicações tecnológicas, de acordo com Toledo (2015), acontece de forma mais interativa e dinâmica, entretanto, para que tal processo aconteça de forma satisfatória, é necessário que as ferramentas utilizadas possuam princípios de qualidade em sua interface. A eficiência do uso de recursos computacionais, auxiliando na melhoria do ensino, deve acontecer pela avaliação dos resultados obtidos com o uso das ferramentas tecnológicas disponibilizadas para alunos e professores da instituição.

### 2.1.2 COMPUTAÇÃO GRÁFICA

Gomes (2003) diz que a Computação Gráfica pode ser descrita como o estudo da síntese, do processamento e da análise de objetos gráficos. Um objeto gráfico é normalmente representado através de polígonos que, posteriormente, costumam passar pelo processo de triangulação, no qual o objeto gráfico é dividido em vários triângulos menores para facilitar seu processamento. Para interagir com tais objetos gráficos, é útil que se entenda sobre as propriedades e características dos triângulos que o formam.

### 2.1.3 EQUAÇÃO DA RETA

Gomes (2003) diz que a Computação Gráfica pode ser descrita como o estudo da síntese, do processamento e da análise de objetos gráficos. Um objeto gráfico é normalmente representado através de polígonos que, posteriormente, costumam passar pelo processo de triangulação

### 2.1.4 SENO, COSSENO TANGENTE

Segundo Silva (1990), seno, cosseno e tangente de um ângulo são os resultados das relações entre os ângulos e as medidas dos catetos e da hipotenusa de um triângulo retângulo, sendo também chamadas de razões trigonométricas. Cada ângulo apresenta um valor de seno, cosseno e tangente.

### 2.1.5 MATRIZES

Silva (1990) afirma que uma matriz é uma tabela formada por  $m$  linhas e  $n$  colunas, podendo ser chamada de matriz de ordem  $m \times n$ . Persiano (1988) afirma que matrizes são necessárias em computação gráfica para se calcular os movimentos de transformação (translação, escala e rotação) de um objeto gráfico. Estes movimentos de transformação são operações consideradas usuais e são muito utilizadas em Computação Gráfica.

## 2.2 TRABALHOS CORRELATOS

Dentre os trabalhos correlatos, encontra-se o Visedu-Mat 2.0 (MACHADO, 2014), por sua proposta semelhante a esse trabalho, no qual é possível visualizar funções matemáticas em 2D e 3D. Nele também é possível a visualização de múltiplos objetos gráficos ao mesmo tempo. O segundo é o Geogebra (HOHENWARTER *et al.*, 2019) que além de permitir a visualização de funções em 2D e 3D, ele trabalha com outras áreas da matemática, como a probabilidade, estatística, entre outros. Ele permite a visualização de diversos objetos gráficos simultâneos quando se estiver tratando de um assunto que tenha objetos gráficos, como no modo de geometria ou no modo de gráficos. O último é o Pat2Math + Handwriting (MORAIS; JAQUES, 2017) por ser uma aplicação focada no ensino de matemática. Ele é um Sistema Tutor Inteligente (STI) que tutoria o aluno através de exercícios diversos de matemática.

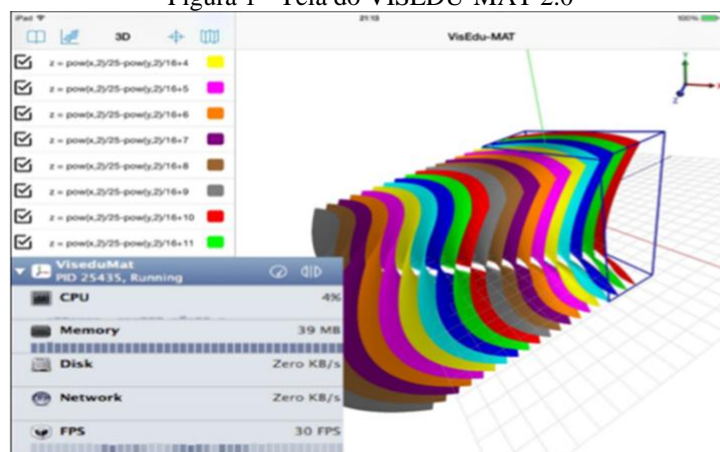
Quadro 1 – VISEDU-MAT 2.0

Referência	Machado (2014)
Objetivos	Permitir a visualização de funções matemáticas para facilitar o entendimento delas.
Principais funcionalidades	Visualização de funções matemáticas em um ambiente 2D e 3D. Criação de múltiplos objetos gráficos simultâneos.
Ferramentas de desenvolvimento	A aplicação foi desenvolvida para o iPad em OpenGL ES.
Resultados e conclusões	Foram realizados diversos testes de performance em sua aplicação. Todos os testes foram realizados com o Visedu-Mat 2.0 no modo 3D, utilizando a ferramenta Instruments 6.1 (56160) que vem instalado juntamente com o Xcode 6.1 (6A1052d) para exibir valores de dados como CPU, memória, disco, rede e Frames Per Second (FPS). Machado (2014) conclui que, embora tivesse encontrado dificuldades com a linguagem e com a validação das funções, obteve um resultado positivo, cumprindo todos os objetivos propostos.

Fonte: elaborado pelo autor.

A Figura 1 apresenta a tela do VISEDU-MAT 2.0, na qual é feito um teste de performance, onde na sub-tela na esquerda são criadas as funções matemáticas que aparecem representadas de maneira tridimensional na tela principal na direita. No canto inferior esquerdo é demonstrado o consumo de memória da aplicação por meio de uma aplicação externa.

Figura 1 - Tela do VISEDU-MAT 2.0



Fonte: Machado (2014).

#### Quadro 2 – GEOGEBRA

Referência	Hohenwarter <i>et al.</i> (2019)
Objetivos	Facilitar o processo de aprendizagem na área de matemática.
Principais funcionalidades	Apresenta representação visual para diversos temas da área da matemática, como geometria, álgebra, gráficos, probabilidade, estatística, entre outros.
Ferramentas de desenvolvimento	A aplicação foi desenvolvida em Java.
Resultados e conclusões	Ele é um software multiplataforma mundialmente conhecido e utilizado diariamente por diversos educadores para facilitar o processo de aprendizagem da área de matemática, o que resultou em diversos prêmios para a aplicação com o passar dos anos. O GeoGebra possui uma vasta comunidade de usuários em praticamente todos os países e com uma faixa etária diversa.

Fonte: elaborado pelo autor.

#### Quadro 3 – PAT2MATH + HANDWRITING

Referência	Morais e Jaques (2017)
Objetivos	Facilitar a utilização do Pat2math e consequentemente melhorar o desempenho dos usuários.
Principais funcionalidades	Todas as funcionalidades anteriores do Pat2Math além da inserção de dados por meio de desenho em smartphones e tablets.
Ferramentas de desenvolvimento	O próprio Pat2Math e o kit de desenvolvimento web MyScript Math.
Resultados e conclusões	Após os testes, não foi identificado um nível de aprendizado diferenciado, entretanto, foi observado um maior nível de engajamento e entretenimento por parte dos usuários.

Fonte: elaborado pelo autor.

#### Quadro 4 – VISEDU

Referência	
Objetivos	
Principais funcionalidades	
Ferramentas de desenvolvimento	
Resultados e conclusões	

Fonte: elaborado pelo autor.

### 3 DESCRIÇÃO

Neste capítulo serão descritos detalhes sobre a implementação do software, na seção 3.1 serão descritos os requisitos e objetivos da ferramenta, na seção 3.2 tem-se um detalhamento sobre a implementação do aplicativo além de softwares utilizados na implementação e na seção 3.3, uma visão geral da ferramenta.

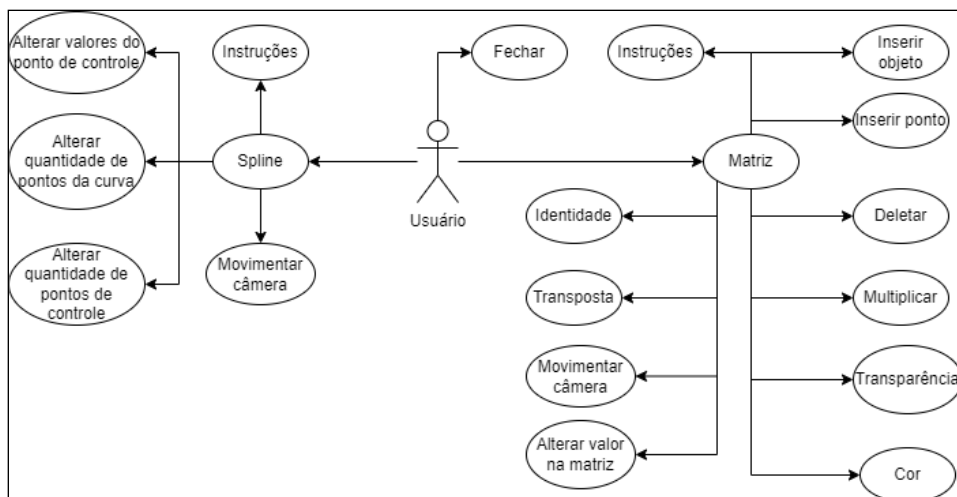
### 3.1 ESPECIFICAÇÃO

O objetivo dessa ferramenta é auxiliar no ensino de conceitos utilizados na matéria de Computação Gráfica, permitindo visualização do resultado das funções utilizadas e diversas interações com os objetos 3d resultado.

O desenvolvimento do protótipo seguiu os principais Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF) destacados a seguir:

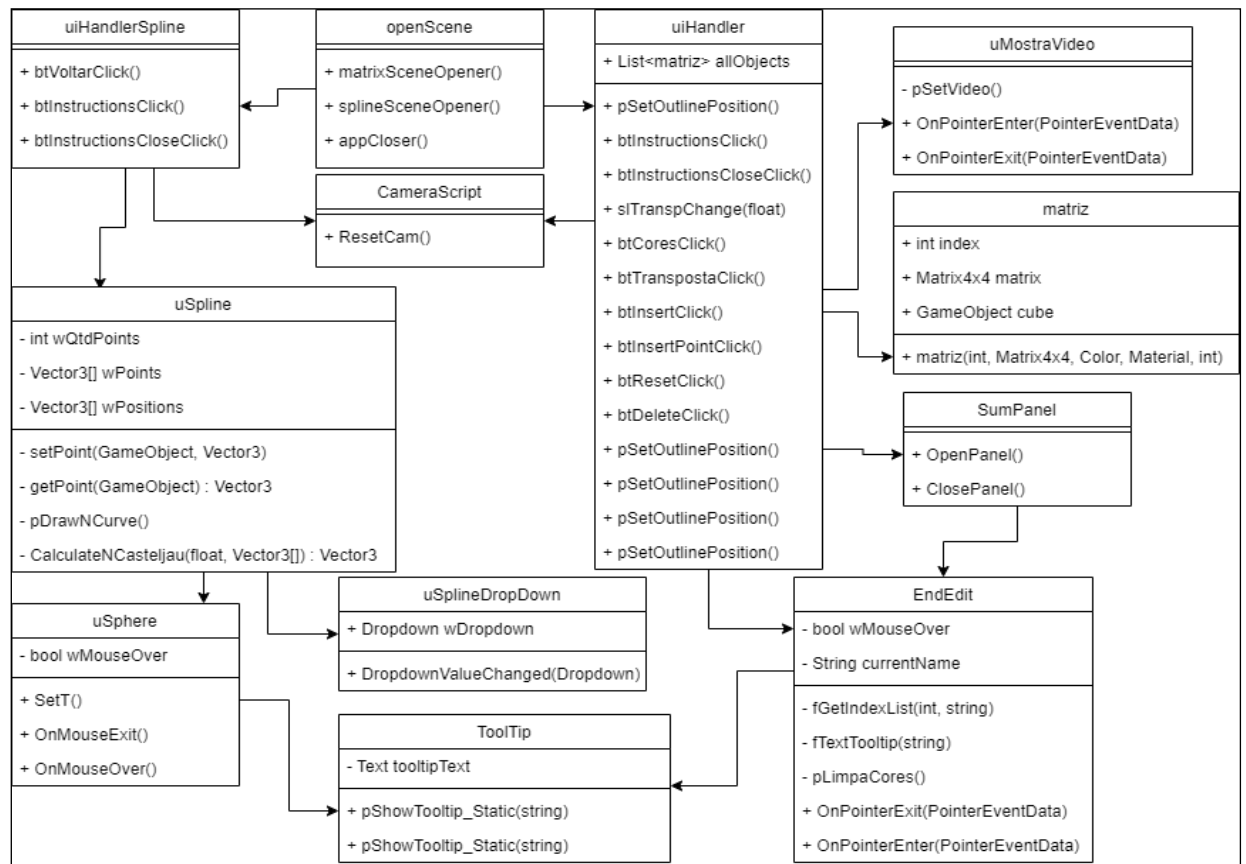
- permitir visualização de objetos gráficos em um ambiente 3D (Requisito Funcional - RF);
- permitir mais de um objeto gráfico simultâneo (RF);
- aplicar operações de transformação no objeto gráfico gerado ao alterar valores na função (RF);
- permitir movimentação livre da câmera em torno do objeto 3D gerado (RF);
- permitir alteração de valores como cor e transparência do objeto 3D gerado (RF);
- possibilitar pular e assistir novamente aos tutoriais (RF);
- ser desenvolvido em Unity (Requisito Não Funcional - RNF);
- ser visualizável em desktop, tablet, smartphone e web (RNF).

Figura 2 – Diagrama de caso de uso



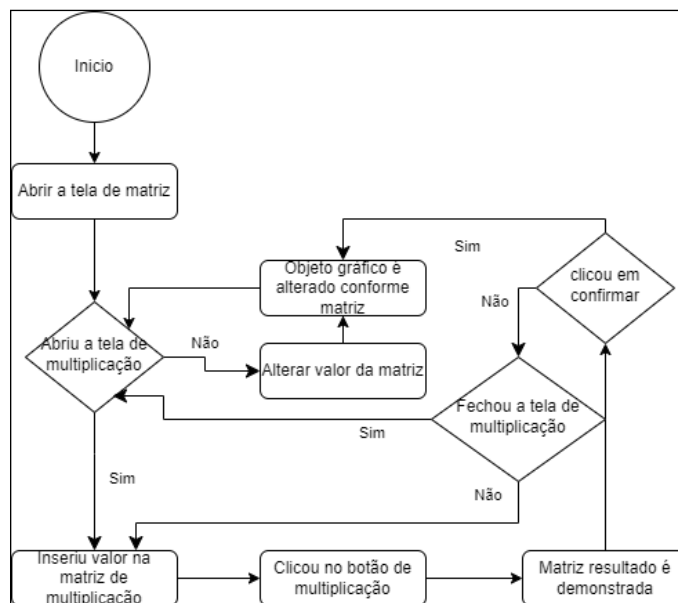
Fonte: elaborado pelo autor.

**Figura 3** - Diagrama de classes



Fonte: elaborado pelo autor.

**Figura 4 - Fluxograma** da multiplicação



Fonte: elaborado pelo autor.

### 3.2 IMPLEMENTAÇÃO

O projeto foi desenvolvido em **Unity**, utilizando a linguagem C# na programação. Também foi utilizado o **Blender** para a **modelação** do asset das setas de referência e os vídeos foram gravados utilizando o software **Screencastify**.

Na engine **Unity**, são utilizadas de cenas para realizar suas operações, nessa ferramenta foram utilizadas de três cenas para melhor organizar e demonstrar as funções da ferramenta. A primeira cena, **MainScene**, é utilizada como tela inicial apenas, por ela é possível acessar as duas outras cenas principais. A segunda cena, **MatrixScene**, é utilizada como

ambiente para todas as funções realizadas na matriz. Nessa cena, a maioria das funções foram feitas no código fonte `uiHandler`. Todo objeto gráfico criado na tela de matrizes contém a classe `matriz`, que pode ser observada na [figura 5](#), na qual apresenta o construtor e as propriedades necessárias para realizar as funções disponíveis na aplicação. Os objetos gráficos criados podem ser o cubo e o ponto, sendo que ambos são reconhecidos a partir de suas `tags` atribuídas pelo construtor no momento da criação. Cada objeto tem o seu objeto `matriz` e guarda suas transformações no mesmo.

Figura 5 - Classe matriz

```

8  public class matriz
9  {
10     7 references
11     public int index;
12     6 references
13     public Matrix4x4 matrix;
14     48 references
15     public GameObject cube;
16     3 references
17     public matriz(int pIndex, Matrix4x4 pMatrix, Color pCor, Material pMaterial, int prType = 1)
18     {
19         //Objeto matriz e suas propriedades.
20         index = pIndex;
21         matrix = pMatrix;
22         if (prType == 1)
23         {
24             cube = GameObject.CreatePrimitive(PrimitiveType.Cube);
25             cube.tag = "Cube";
26         }
27         else
28         {
29             cube = GameObject.CreatePrimitive(PrimitiveType.Sphere);
30             cube.tag = "Point";
31             cube.transform.localScale = new Vector3(0.3f, 0.3f, 0.3f);
32         }
33         cube.GetComponent<Renderer>().material = pMaterial;
34         cube.GetComponent<Renderer>().material.color = pCor;
35         cube.name = index.ToString();
36     }
37 }

```

Fonte: elaborado pelo autor.

Conforme as alterações são feitas nos objetos gráficos a rotina `update`, que fica em um loop constante conforme a aplicação está em execução, atualiza o `ui` com a matriz do objeto atualmente focado e desenha a `outline` no objeto. Conforme objetos são inseridos e removidos da cena através dos botões `inserir` e `remover` presentes, a lista de objetos `allobjects` é atualizada e o `index` do objeto atual é transferido para a variável de `foco wCubeIndex`, como pode ser observado na [figura 6](#). Além das funções principais também foi necessário criar rotinas de conversão que são utilizadas em diversas partes da aplicação, principalmente nas partes onde valores são atribuídos ou resgatados de componentes como `inputfields`, através das rotinas `strToFloat` e `strToInt`. Também foi criada uma classe `tooltip`, que apresenta qualquer texto em qualquer componente no formato de `hint`, essa rotina foi majoritariamente utilizada por componentes `inputfields` tanto na cena de `matrizes quanto` na cena de `splines`.

Figura 6 - Método de inserção

```

322 void btInsertClick()
323 {
324     //Rotina que insere mais um polígono.
325     wCubeIndex = allobjects.Count;
326     btCoresClick();
327     btResetClick();
328     matriz m = new matriz(wCubeIndex, matrix, cor, mtCube);
329     allobjects.Add(m);
330     xyzShow.text = "X=" + allobjects[wCubeIndex].cube.transform.position.x +
331                  " Y=" + allobjects[wCubeIndex].cube.transform.position.y +
332                  " Z=" + allobjects[wCubeIndex].cube.transform.position.z +
333                  " W=1";
334 }

```

Fonte: elaborado pelo autor.



A última cena da aplicação é a `splineScene`, na qual está presente todas as funções de spline. Nesta cena sem dúvidas a rotina mais importante é a `calculateNCasteljau`, que utiliza do algoritmo de `casteljau` para calcular cada ponto presente na linha de splines. Inicialmente foi utilizado o algoritmo de bezier para realizar os cálculos, mas ao desenvolver este algoritmo, obtive erros ao tentar criar splines com mais de 5 pontos de controle. Esta rotina retorna um ponto dependendo dos pontos de controle e da variável `t`, que apresenta um valor decimal entre 0 e 1, no qual o 0 marca o ponto inicial da spline e o 1 marca o ponto final.

Figura 7 - Método Casteljau

```

201 private Vector3 CalculateNCasteljau(float t, Vector3[] p)
202 {
203     //Rotina que retorna o valor dos pontos que criam a spline.
204     Vector3[] p1 = new Vector3[p.Length];
205     for (int i = 0; i < p.Length; i++)
206     {
207         p1[i] = p[i];
208     }
209     float nt = 1.0f - t;
210     int lnght = p1.Length;
211     while (lnght > 1)
212     {
213         for (int i = 0, e = lnght - 1; i < e; i++)
214         {
215             p1[i] = nt * p1[i] + t * p1[i + 1];
216         }
217         lnght = lnght - 1;
218     }
219     return p1[0];
220 }

```

Fonte: elaborado pelo autor.

Cada um desses pontos é desenhado como uma spline pela rotina `pDrawNCurve`, que apresenta um loop que depende da quantidade de pontos da spline, valor que é armazenado na variável `wQtdPoints`, conforme pode ser observado na figura 8. Foi preciso criar uma classe `uSphere` que é atribuída a cada um desses pontos da spline, para possibilitar demonstrar o valor de cada ponto em execução. Nesta classe é chamada a classe de `tooltip` no momento que o mouse em um ponto da spline, e em seguida é demonstrado o valor do ponto em formato de `hint`.

Figura 8 - Método que desenha a spline

```

188 private void pDrawNCurve()
189 {
190     //Rotina que desenha a linha da spline.
191     for (int i = 0; i <= wQtdPoints; i++)
192     {
193         float t = i / (float)wQtdPoints;
194         wPositions[i] = CalculateNCasteljau(t, wPoints2);
195         wCList[i].transform.position = wPositions[i];
196         wCList[i].GetComponent<uSphere>().setT(t);
197         wCList[i].transform.position = wPositions[i];
198     }
199     wLineRenderer.SetPositions(wPositions);
200 }

```

Fonte: elaborado pelo autor.

Para ambas as cenas, foi necessário criar uma classe `cameraScript` que inicializa a câmera nas cenas e apresenta as rotinas de movimentação da câmera.

### 3.3 OPERACIONALIDADE

A ferramenta apresenta uma tela inicial com as opções `matriz`, `splines` e `fechar`, como mostra na figura 9. A opção `fechar` fecha a ferramenta, a opção de `splines` abre a tela de splines e a opção de `matriz` abre a tela de matrizes.

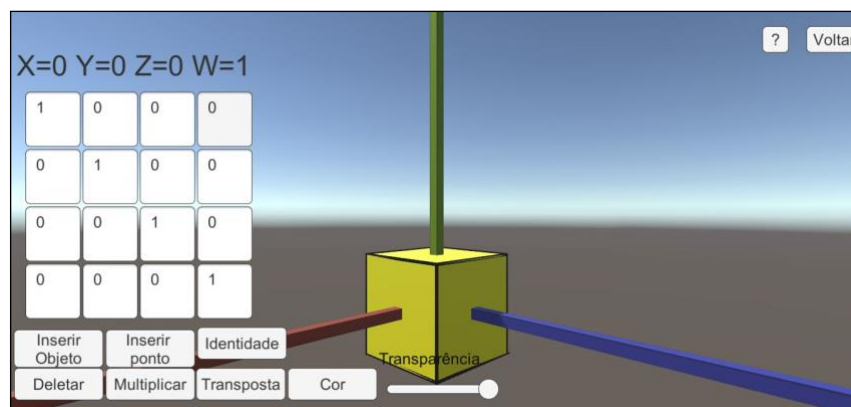
Figura 9 - Tela inicial



Fonte: elaborado pelo autor.

Esta tela apresenta campos para inserir os valores da matriz do objeto atualmente selecionado, um botão **inserir objeto** que insere mais um objeto gráfico na cena, um botão **inserir ponto** que insere um objeto gráfico que representa um ponto na cena, um botão **deletar** que deleta o objeto gráfico selecionado, uma opção que converte a matriz do objeto atual para a matriz identidade, uma opção que converte a matriz do objeto atual para a matriz transposta, um botão que altera a cor do objeto gráfico atualmente selecionado, um slide que permite ajustar a transparência do objeto gráfico atualmente selecionado, um botão **multiplicar** que abre uma subtela que permite realizar diversos cálculos básicos na matriz do objeto atualmente selecionado, um botão de instruções que apresenta as teclas que realizam alguma função na ferramenta, um botão de voltar que retorna para a tela inicial previamente descrita, além da própria cena gráfica com os objetos gráficos gerados, como pode ser observado na figura 10.

Figura 10 - Tela de matrizes

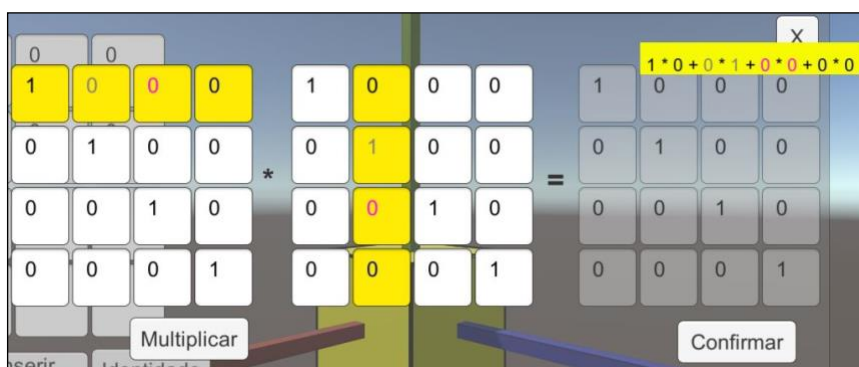


Fonte: elaborado pelo autor.

A subtela de **cálculos visível** na figura 11 apresenta a matriz anterior, campos para inserir dados da matriz com que será feito o **cálculo** e apresenta também a **matriz** resultante do cálculo. Nessa **subtela**, **existe** um botão que realiza a multiplicação, um botão para confirmar as alterações feitas na matriz inicial e um botão voltar que fecha a subtela.



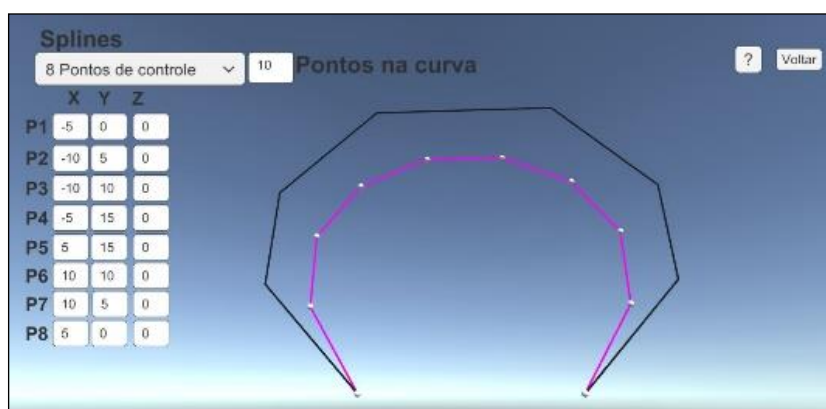
Figura 11 - Subtela de multiplicação



Fonte: elaborado pelo autor.

Ao pressionar o botão de **splines** na tela inicial, é aberta a tela de splines, que apresenta um campo que permite selecionar a quantidade de pontos de apoio que formarão a curva de spline, um campo que pode ser inserida a quantidade de pontos na spline, diversos campos para cada ponto de apoio existente no qual é possível alterar seus valores, existe um campo voltar que volta para a tela inicial além da própria cena em si, onde são visíveis os pontos de apoio, a spline gerada e os seus pontos. As splines apresentam como padrão, o formato de uma forma geométrica dependendo da quantidade de pontos de controle, como por exemplo, um octógono aparece ao selecionar 8 pontos de controle, como pode ser visto na figura 12.

Figura 12 - Tela de splines



Fonte: elaborado pelo autor.

## 4 RESULTADOS

De modo a ampliar o seu caráter científico, todos os TCCs devem apresentar e discutir resultados não limitados à comparação com os trabalhos correlatos. Devem ser apresentados os casos de testes do software, destacando objetivo do teste, como foi realizada a coleta de dados e a apresentação dos resultados obtidos, preferencialmente em forma de gráficos ou tabelas, fazendo comentários sobre os mesmos. Também é sugerida a comparação com os trabalhos correlatos apresentados na fundamentação teórica.

## 5 CONCLUSÕES

As conclusões devem refletir os principais resultados alcançados, realizando uma avaliação em relação aos objetivos previamente formulados. Deve-se deixar claro se os objetivos foram atendidos, se as ferramentas utilizadas foram adequadas e quais as principais contribuições do trabalho sociais ou práticas para o seu grupo de usuários bem como para o desenvolvimento científico e ou tecnológico da área.

Deve-se incluir também as limitações e as possíveis extensões do TCC.

## REFERÊNCIAS

FERNANDES, Daniela. **Brasil avança em conhecimento básico de matemática, mas continua atrás em ranking**. 2016. Disponível em: <[https://www.bbc.com/portuguese/noticias/2016/02/160209\\_ocde\\_alunos\\_baixa\\_performance\\_pai\\_df](https://www.bbc.com/portuguese/noticias/2016/02/160209_ocde_alunos_baixa_performance_pai_df)>. Acesso em: 19 set. 2019.

GOMES, Jonas; VELHO, Luiz. **Fundamentos da computação gráfica**. Rio de Janeiro : IMPA, 2003. 603 p, il. (Série de computação e matemática). Acesso em: 3 out. 2019.

HOHENWARTER, M. *et al.* (Austria). **Geogebra**. 2019. Disponível em: <<https://www.geogebra.org/>>. Acesso em: 25 out. 2019.

IDOETA, Paula Adamo. **Enem**: o que as questões de matemática 'mais difíceis' dizem sobre a educação no Brasil. 2018. Disponível em: <<https://www.bbc.com/portuguese/brasil-44888935>>. Acesso em: 18 set. 2019.

MACHADO, Oswaldo Bay. **VISEDU-MAT 2.0**: Visualizador de material educacional. 2014. Disponível em: <<http://dsc.inf.furb.br/tcc/index.php?cd=6&tcc=1643>>. Acesso em: 11 set. 2019.

MORAIS, Felipe de; JAQUES, Patrícia A.. **PAT2Math+Handwriting**: Evoluindo Sistemas Tutores de Matemática com reconhecimento da escrita à mão. 2017. Disponível em: <<https://www.br-ie.org/pub/index.php/sbie/article/view/7652/5448>>. Acesso em: 11 set. 2019.

PERSIANO, Ronaldo Cesar Marinho; OLIVEIRA, Antonio A. F. de (Antonio Alberto F. de); INSTITUTO BRASILEIRO DE PESQUISA EM INFORMATICA. **Introducao a computacao grafica**. Rio de Janeiro : Livros Tecnicos e Cientificos : IBPI, 1988. xiv, 225p, il, 23cm. (Ciencia de computacao). Acesso em: 3 out. 2019.

SILVA, Cláudio Xavier da; BARRETO FILHO, Benigno. **Toda matemática**: livro do professor, [2. grau]. São Paulo : Atica, 1990. 374, [2], 24 p, il. (algumas col.). Acesso em: 3 out. 2019.

TOLEDO, Bruno de Souza. **O USO DE SOFTWARES COMO FERRAMENTA DE ENSINO/APRENDIZAGEM NA EDUCAÇÃO DO ENSINO MÉDIO/TÉCNICO NO INSTITUTO FEDERAL DE MINAS GERAIS**. 2015. Disponível em: <<http://www.fumec.br/revistas/sigc/article/view/3163/1891>>. Acesso em: 2 nov. 2019.

**VARGAS, Douglas N. Editor dirigido por sintaxe**. 1992. Relatório de pesquisa n. 240 arquivado na Pró-Reitoria de Pesquisa, Universidade Regional de Blumenau, Blumenau.

## **APÊNDICE A – DIAGRAMAS DE ESPECIFICAÇÃO**

É fundamental que todo projeto apresente alguma forma de especificação do que foi desenvolvido. A descrição é opcional. Assim, **este apêndice deve conter os diagramas de especificação que não couberam ao longo do texto.** Os diagramas devem conter legendas numeradas na sequência do artigo.

Cada apêndice deve iniciar em uma nova página.

## **APÊNDICE B – XXX**

Podem ser inseridos outros apêndices no artigo tais como códigos de implementação, telas de interface, instrumentos de coleta de dados, entre outros. **Apêndices são textos elaborados pelo autor** a fim de complementar sua argumentação. Os apêndices são identificados por letras maiúsculas consecutivas, seguidas de um travessão e pelos respectivos títulos. Deve haver no mínimo uma referência no texto anterior para cada apêndice. Colocar sempre um preâmbulo no apêndice. Caso existam tabelas ou ilustrações, identifique-as através da legenda, seguindo a numeração normal das legendas do artigo.

## **ANEXO A – DESCRIÇÃO**

Elemento opcional, **anexos são documentos não elaborados pelo autor**, que servem de fundamentação, comprovação ou ilustração, como mapas, leis, estatutos, entre outros. Os anexos são identificados por letras maiúsculas consecutivas, seguidas de um travessão e pelos respectivos títulos. Deve haver no mínimo uma referência no texto anterior para cada anexo. Colocar sempre um preâmbulo no anexo. Caso existam tabelas ou ilustrações, identifique-as através da legenda, seguindo a numeração normal das legendas do artigo.

## 6 DESCRIÇÃO DA FORMATAÇÃO

A seguir são apresentadas observações gerais sobre o texto do artigo do Trabalho de Conclusão de Curso (TCC).  
**Observa-se que esta descrição deve ser retirada do texto final.**

Na confecção do texto deve-se:

- a) usar frases curtas. Segundo Teodorowitsch (2003, p. 3), “Frases com mais de duas linhas aumentam o risco de o leitor não compreender a ideia ou de entendê-la de forma equivocada.”;
- b) usar linguagem impessoal (usar a terceira pessoa do singular) e verbo na voz ativa (a ação é praticada pelo sujeito), com conexão entre os parágrafos;
- c) não usar palavras coloquiais;
- d) não usar palavras repetidas em demasia;
- e) usar verbos no presente quando for referir-se a partes do trabalho que já se encontram disponíveis no texto;
- f) destacar palavras em língua estrangeira em *itálico*, conforme descrito abaixo:
  - nome de software, ferramenta, aplicativo, linguagem de programação, plataforma, empresa: não deve ser escrito em *itálico* (exemplos: Delphi 7, Pascal, Object Pascal, Java, JavaScript, Java 2 Micro Edition, Basic, Microsoft Visual C++, C, Windows, Linux, MySQL, Oracle, Eclipse 3.0, Enterprise Architect, Rational Rose, Microsoft, Sun Microsystems),
  - citações: o sobrenome do autor ou o nome da instituição responsável pela autoria do documento citado não deve ser escrito em *itálico* (exemplo: Segundo Sun Microsystems (2004), ...),
  - palavras em língua estrangeira encontradas nos dicionários nacionais: não devem ser grafadas em *itálico* (exemplos: software, hardware, web, Internet),
  - demais palavras em língua estrangeira: devem ser escritas em *itálico* (exemplos: *palmtop*, *classpath*, *play*, etc.). No entanto, Teodorowitsch (2003, p. 7), sugere que alguns termos em língua inglesa devem ser substituídos por termos em português (exemplos: núcleo em vez de *kernel*, aprendizagem de máquina em vez de *machine learning*, etc.);
- g) observar as seguintes regras quanto ao uso de siglas:
  - colocar as siglas entre parênteses precedidas pela forma completa do nome, quando aparecem pela primeira vez no texto (exemplos: Associação Brasileira de Normas Técnicas (ABNT), Trabalho de Conclusão de Curso (TCC)). Caso exista uma lista de siglas na parte pré-textual do volume final, pode-se usar somente a sigla, quando aparecer pela primeira vez no texto,
  - usar apenas a sigla nas demais ocorrências no texto,
  - escrever as siglas em letras maiúsculas e não usar *itálico*,
  - escrever o plural das siglas sem apóstrofo (exemplos: PCs, APIs, PDAs) e determinar o gênero da sigla conforme o gênero do primeiro substantivo do seu nome (exemplo: o TCC – o Trabalho de Conclusão de Curso).

### 6.1 FORMATAÇÃO

A formatação geral para apresentação do documento, descrita na NBR 14724 (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2011), é a seguinte:

- a) o texto divide-se em capítulos, seções e subseções (até cinco divisões);
- b) a apresentação de citações em documentos deve seguir a NBR 10520 (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2002b);
- c) a descrição das referências bibliográficas deve estar de acordo com a NBR 6023 (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2002a).

Observa-se ainda que todo capítulo, seção ou subseção deve ter no mínimo um texto relacionado.

O artigo deve ser digitado usando as fontes e formatação de parágrafos deste modelo, indicadas no Quadro 2.

Quadro 2– Estilos do modelo

USO	FORMATO
título de capítulo ou seção primária (1)	TF-TÍTULO 1 (Times New Roman, 10pt, negrito, maiúsculas)
título de seção secundária (1.1)	TF-TÍTULO 2 (Times New Roman, 10pt, maiúsculas)
título de seção terciária (1.1.1)	TF-Título 3 (Times New Roman, 10pt, minúsculas, exceto a 1ª letra da 1ª palavra do título e de nomes próprios)
título de seção quaternária (1.1.1.1)	TF-Título 4 (mesma formatação seção terciária)
título de seção quinária (1.1.1.1.1)	TF-Título 5 (mesma formatação seção terciária)
texto	TF-TEXTO (Times New Roman, 10pt)



citação direta com mais de três linhas	TF-CITAÇÃO (Times New Roman, 9pt, com recuo de 4 cm)
ítems (alíneas)	ver descrição abaixo (Times New Roman, 10pt)
referência bibliográfica	TF-REFERÊNCIA ITEM (Times New Roman, 10pt, alinhada à margem esquerda)
fonte, legenda, texto de quadro/tabela e figura	TF-FONTE (Times New Roman, 9pt, centralizada) TF-LEGENDA, (Times New Roman, 10pt, centralizada) TF-TEXTO- QUADRO (Times New Roman, 10pt) TF-FIGURA (Times New Roman, 10pt, centralizada)

Fonte: elaborado pelo autor.

O espaçamento, também definido no modelo, deve ser conforme indicado no Quadro 3.

Quadro 3 - Espaçamento

USO	ESPAÇAMENTO
título de capítulo ou seção primária (1) título de seção secundária (1.1) título de seção terciária (1.1.1) título de seção quaternária (1.1.1.1) título da seção quinária (1.1.1.1.1)	espaço simples, com 12pt antes do parágrafo
texto	espaço simples, com 6 pt antes do parágrafo
citação direta com mais de três linhas	espaço simples com 6pt antes e depois do parágrafo
ítems (alíneas)	espaço simples, com 6 pt antes do parágrafo
referência bibliográfica	espaço simples, com 6 pt antes do parágrafo
legenda e texto de ilustração/tabela	espaço simples, com 6 pt antes do parágrafo
fonte	espaço simples, com 0pt antes do parágrafo

Fonte: elaborado pelo autor.

Na disposição gráfica de ítems (alíneas) devem ser observados os seguintes quesitos:

- o texto que antecede os ítems termina com dois pontos;
- cada item deve iniciar com uma letra minúscula seguida de fecho parênteses e terminar com um ponto e vírgula, sendo que o último item termina com ponto (FORMATO: TF-ALÍNEA);
- o texto de cada item inicia com letra minúscula, exceto nomes próprios;
- quando contiver subítems, os mesmos devem iniciar com hífen colocado sob a primeira letra do texto do item correspondente (FORMATO: TF-SUBALÍNEA nível 1 ou TF-SUBALÍNEA nível 2, conforme o caso). Nesse caso, cada subitem deve terminar com uma vírgula, exceto o último que termina com ponto ou com ponto e vírgula.

Segue um exemplo:

- cada item inicia com letra minúscula, cada item inicia com letra minúscula, cada item inicia com letra minúscula, cada item inicia com letra minúscula (FORMATO: TF-ALÍNEA);
- cada item inicia com letra minúscula, cada item inicia com letra minúscula, cada item inicia com letra minúscula, cada item inicia com letra minúscula (FORMATO: TF-ALÍNEA):
  - cada subitem (nível 1) inicia com letra minúscula, cada subitem (nível 1) inicia com letra minúscula (FORMATO: TF-SUBALÍNEA nível 1):
    - cada subitem (nível 2) inicia com letra minúscula, cada subitem (nível 2) inicia com letra minúscula (FORMATO: TF-SUBALÍNEA nível 2),
    - cada subitem (nível 2) inicia com letra minúscula, cada subitem (nível 2) inicia com letra minúscula (FORMATO: TF-SUBALÍNEA nível 2),
  - cada subitem (nível 1) inicia com letra minúscula, cada subitem (nível 1) inicia com letra minúscula (FORMATO: TF-SUBALÍNEA nível 1);
- cada item inicia com letra minúscula, cada item inicia com letra minúscula, cada item inicia com letra minúscula, cada item inicia com letra minúscula (FORMATO: TF-ALÍNEA).

6.1.1.1 Exemplo de título de seção quaternária [FORMATO: TF-TÍTULO 4]

Formato: TF-TEXTO.

6.1.1.1.1 Exemplo de título de seção quinária [FORMATO: TF-TÍTULO 5]

Formato: TF-TEXTO.

### 6.1.2 Formatação de quadros, figuras e tabelas

Um quadro contém apenas informações textuais, que podem ser agrupadas em colunas. Uma figura contém, além das informações textuais, pelo menos um elemento gráfico. Uma tabela é uma apresentação tabular de informações **numéricas** relacionadas.

Os quadros, figuras e tabelas são identificados na parte superior por uma legenda (a qual deve estar centralizada) composta pela palavra designativa (Figura, Quadro ou Tabela, conforme o caso), seguida de seu número em algarismo arábico (usar numeração progressiva, uma sequência para os quadros, outra para as figuras e outra para as tabelas), de hífen e do título. As ilustrações devem:

- aparecer centralizadas no texto;
- estar delimitadas por uma moldura simples (com exceção das tabelas não quais não devem ser usadas bordas (linhas) verticais em suas extremidades);
- aparecer numa única página (quando o tamanho não exceder o da página), inclusive a legenda;
- serem inseridas o mais próximo possível do trecho a que se referem pela primeira vez.

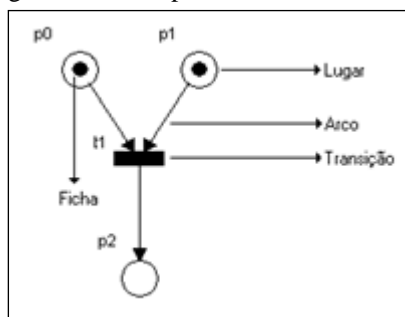
Toda ilustração deve ter fonte, centralizada. Quando foi o próprio autor que fez a ilustração, deve inserir o texto: “Fonte: elaborado pelo autor”.

Observa-se que quando um código fonte for descrito dentro de um quadro, deve-se utilizar letra do tipo `courier new 9pt`. (TF-CÓDIGO-FONTE)

Exemplos de como se deve referenciar uma figura, um quadro e uma tabela bem como descrevê-los são mostrados a seguir.

Um exemplo de uma rede de Petri pode ser visto na Figura 13.

Figura 13– Exemplo de uma rede de Petri



Fonte: Schubert (2003, p. 18).

Um exemplo de código fonte gerado a partir de uma especificação pode ser visto no Quadro 4.

Quadro 4 – Funções que verificam se as transições estão sensibilizadas

```
Function TEstruturaMalha.T1Sensibilizada: boolean;
begin
    result := (Fp2 and Fp4);
end;

function TEstruturaMalha.T2Sensibilizada: boolean;
begin
    result := (Fp1 and Fp3);
end;

function TEstruturaMalha.T3Sensibilizada: boolean;
begin
    result := (Fp2 and Fp4);
end;
```

Fonte: Schubert (2003, p. 63).

A quantidade de trabalhos finais realizados no Curso de Ciência da Computação (de 2010 até 2014) é apresentada na Tabela 1.

Tabela 1 – Trabalhos finais realizados no Curso de Ciência da Computação

Ano	Estágios	TCC's	Totais
2010/1	0	16	16
2010/2	0	21	21
2011/1	0	25	25
2011/2	0	23	23
2012/1	0	23	23
2012/2	0	22	22
2013/1	0	25	25
2013/2	0	16	16
2014/1	0	18	18
2014/2	0	13	13
	<b>0</b>	<b>202</b>	<b>202</b>

Fonte: elaborado pelo autor.

### 6.1.3 Exemplos de citações retiradas de documentos ou de nomes constituintes de uma entidade

A apresentação de citações em documentos deve seguir a NBR 10520 (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2002b). O sistema a ser usado é o alfabético. Exemplos de citações são: “Numa publicação recente (SEBESTA, 2000) é exposto ...” e “Segundo Silva *et al.* (1987), execução controlada de programas é ...”.

Quando a citação referir-se a uma parte específica do documento consultado, especificar no texto do artigo a(s) página(s). Esta(s) deverá(ão) seguir a data, separada(s) por vírgula(s) e precedida(s) pelo designativo que a(s) caracteriza(m). Como exemplo, mostra-se: “(SCHIMT, 1999, p. 50)” ou “... visto que Schimt (1999, p. 50) implementou ...”.

As citações diretas (transcrição textual de parte da obra do autor consultado), no texto, com mais de três linhas, devem ser destacadas com recuo de 4 cm da margem esquerda, com letra menor que a do texto utilizado e sem aspas (FORMATO: TF-CITAÇÃO), conforme o exemplo a seguir.

A Associação Brasileira de Normas Técnicas (ABNT) é o Fórum Nacional de Normalização. As Normas Brasileiras, cujo conteúdo é de responsabilidade dos Comitês Brasileiros (ABNT/CB) e dos Organismos de Normalização Setorial (ABNT/ONS), são elaboradas por Comissões de Estudo (CE), formadas por representantes dos setores envolvidos, delas fazendo parte: produtores, consumidores e neutros (universidades, laboratórios e outros). (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2002b, p. 1).

Quando da citação de um nome (identificador) constituinte de uma entidade ou de um elemento de interface em um texto, deve-se utilizar o tipo de letra *courier new*, com tamanho nove (9). Para facilitar a formatação, existe o estilo de palavra denominado TF-COURIER9. Como exemplo cita-se nome de classe, atributo ou método. A seguir são apresentados exemplos.

As classes `TTabelaTransicao` e `TExpressaoRegular` são classes de interface, porém estão sendo consideradas como classes de domínio da aplicação.

Ao clicar no botão `Confirmar`, o software abre uma nova tela.