REFATORANDO E ADICIONANDO NOVAS FUNCIONALIDADES NO VISEDU-CG COM MOTOR DE JOGOS UNITY

Douglas Eduardo Bauler, 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

dbauler@furb.br, dalton@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**: Ensino aprendizagem. Unity. Computação Gráfica. Transformações geométricas.

# Introdução

A tecnologia está em constante evolução de maneiras muito significativas, melhorando o dia a dia, aumentando a produtividade e o entendimento de vários assuntos. Existem ferramentas interativas de ensino que dão interlúdio ao assunto a ser abordado, tornando-o de uma maneira mais lúdica na qual muitas vezes pode dificultar o aprendizado.

A contribuição didática para uma pedagogia voltada para o sujeito requer assumir, entre outras coisas, o uso das mídias e das tecnologias da educação. O professor deve ser capaz de utilizar aparatos tecnológicos não apenas para seu uso próprio, mas trabalhar com esses recursos em sala de aula, em favor da aprendizagem dos alunos (SILVA, 2011, p.6).

Por meio dessas metodologias a ferramenta VisEdu-CG tem como objetivo trazer essas melhorias no aprendizado aos acadêmicos da matéria de Computação Gráfica. Conforme Reis (2011, apud BUTTENBERG, 2020, p. 1), “o VisEdu-CG é um projeto para desenvolver uma plataforma Web que permita os alunos da disciplina de Computação Gráfica do curso de Ciências da Computação praticarem os conceitos ministrados nesta disciplina”, estando atualmente na versão 5.0 do projeto.

Essa aplicação contou com o desenvolvimento de vários módulos específicos, dentre eles pode-se citar o motor de jogos, matemática, estatística, processamento de imagens, realidade aumentada e simulação. Para que a ferramenta tenha uma evolução constante foi realizado um processo de migração de linguagem e refatoração do código. Uma das motivações para fazer a refatoração foi em relação à algumas funcionalidades do WebGL, o que torna o VisEdu-CG um sistema igualmente limitado (BUTTENBERG, 2020).

Um processo de migração de uma ferramenta já consolidada numa linguagem, não é um processo simples de realizar, devido à complexidade com diversas funcionalidades da ferramenta, tendo também o curto espaço de tempo para seu desenvolvimento e a falta de estrutura do código para melhor entendimento e manutenção, fazendo com que a migração não seja realizada completamente. Em razão dessas dificuldades, este trabalho propõe continuar o processo de migração das funcionalidades, assim como a refatoração do código já migrado para melhor compreensão, manutenção e adição de novas funções, utilizando a motor de jogos Unity. Os objetivos específicos são:

1. permitir criar atividades em forma de exercícios práticos;
2. disponibilizar novas peças do tipo Iteração, Polígonos e Spline;
3. ampliar o tutorial existente com as novas funcionalidades implementadas.

# FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são destacados os principais assuntos do desenvolvimento da ferramenta. Primeiramente são apresentados os conceitos utilizados para o desenvolvimento como: Refatoração, tutoriais interativos. Respectivamente é apresentada a versão anterior do sistema, o VisEdu-CG 5.0. Por fim, são apresentados os trabalhos correlatos.

## Refatoração de código

A refatoração é o processo de modificar um sistema de software de modo que não altere o comportamento externo do código, embora melhore a sua estrutura interna. É uma maneira disciplinada de reorganizar o código, minimizando as chances de introduzir *bug*. Em sua essência, ao refatorar, aperfeiçoará o design do código depois que ele foi escrito (FOWLER, 2020).

*Refactorings* são transformações de código que melhoram a manutenibilidade de um sistema, mas sem afetar o seu funcionamento. A definição de transformações de código, está referindo a modificações no código, como dividir uma função em duas, renomear uma variável, mover uma função para outra classe, extrair uma interface de uma classe etc. Segundo melhorar a manutenibilidade do sistema, isto é, melhorar sua modularidade, melhorar seu projeto ou arquitetura, melhorar sua estabilidade, tornar o código mais legível, mais fácil de entender e modificar (VALENTE, 2020).

É arriscado ser feito uma refatoração, se exige mudanças que podem introduzir a *bugs* sutis em um código que está funcionando. A implementação se não for feita de forma adequada, pode-se fazer atrasar em dias ou até semanas. Ao começa a explorar o código, logo se descobre novas oportunidades para alterá-lo, à medida que o analisa com mais detalhes. Quanto mais se avalia, surgem novos detalhes da necessidade de mudanças a serem feitas (FOWLER, 2020).

Quando os coautores e Flower escreveram o livro *Padrões de Projeto,* mencionaram que os padrões de projeto oferecem alvos para as refatorações. No entanto, identificar o alvo é somente uma parte do problema e transformar o código para atingir seus objetivos é outro desafio (FOWLER, 2020).

Padrões projetos e refatoração estão amplamente conectados, visando garantir uma melhor compreensão e fácil manutenção de código de um projeto específico. Conforme afirma Rapeli (2006), os padrões de projetos favorecessem a implementações mais eficientes tendo clareza e fácil entendimento do código, em casos de sistemas não projetados em seu uso, é possível aplicá-los sem alterar suas funcionalidades existentes.

## tutoriais interativos

A acelerada expansão e disseminação das tecnologias digitais permitiu inovar, a nível conceitual e pedagógico, os espaços formais de ensino e aprendizagem e as práticas que dentro deles se organizam. Ao analisar esses espaços formais, nomeadamente as escolas, contêm os requisitos e infraestruturas necessárias para reagir às mudanças científicas e tecnológicas vigentes, de modo a se adaptarem às exigências educativas do século XXI, onde a mudança nas práticas de ensino e aprendizagem em sala de aula se torna cada vez mais necessárias (PEDRO; BAETA, 2017).

As ferramentas tecnológicas são instrumentos que podem auxiliar no acompanhamento e nas orientações dos alunos, oferecendo soluções como: facilitar o acompanhamento das atividades em execução por cada aluno de maneira síncrona e/ou assíncrona; facilitar a difusão de esclarecimento de dúvidas coletivas e individuais; integrar espaços coletivos e individuais de produção; e oferecer auxílios a cada aluno conforme o estágio de aprendizagem (SANTOS, 2014).

Tutoriais interativos podem ser implementados de forma de assistentes visuais, na qual contribuem para um melhor aproveitamento dos recursos e entendimento a respeito da ferramenta a ser utilizada. Essas novas tecnologias são de grande ajuda no processo de ensino aprendizagem, fazendo com que os usuários se sintam mais próximos e motivados para aproveitar todos os recursos disponíveis.

A instrução através do tutorial visa potencializar o processo de construção do conhecimento. Para tanto, o aprendiz terá que ler e exercitar o conhecimento adquirido. A conexão com outros *sites* disponíveis na internet, pode apresentar informações auxiliares que nortearão no processo de ensino-aprendizagem (SANTOS; BEZERRA, 2018).

Conforme afirma Santos e Bezerra (2018), na internet há uma predominância de tutoriais informais, uma vez que se pode encontrar em *sites* como o Youtube, com diversos vídeos de tutoriais ensinando sobre os mais variados assuntos, feitos por quaisquer pessoas independentemente de ter um conhecimento científico. Todavia, a qualidade vai depender da sua capacidade de expressão e conhecimento das características dessa ferramenta. Portanto tutoriais podem ser considerados mecanismos informais de auxílio aprendizagem, destacando seu caráter didático descrevendo por uma sequência de passos a utilização de uma ferramenta.

## Versão anterior da ferramenta

O objetivo principal da versão 5.0 da ferramenta VisEdu-CG, foi realizar a migração para utilizar o motor gráfico Unity. Apesar de apresentar problemas de visualização em alguns objetos, como por exemplo, a iluminação *spot*, a plataforma teve resultado satisfatório na construção de cenários contendo conceitos básicos de computação gráfica, como as transformações geométricas e em conceitos com maior complexidade como é o caso das iluminações (BUTTENBERG, 2020).

O segundo objetivo foi criar uma proposta de tutorial informativo de maneira simples, em que a criação de uma cena destacasse os conceitos essenciais. Buttenberg (2020) salienta que o tutorial pode ser melhorado, como por exemplo a forma de exibição dos passos no tutorial. Além disso, algumas peças ainda não foram migradas como: o Polígono, o Spline e as melhorias na implementação da peça iluminação.

O terceiro objetivo, de utilizar representações visuais a partir de peças de encaixe para gerar uma cena gráfica, foi atingido. As peças importadas de uma ferramenta de criação de modelos 3D se comportaram adequadamente no Unity e os encaixes das peças nos slots foram bem-sucedidos. Quase todas as peças tiveram suas representações gráficas efetuadas, com exceção das peças spline e polígono (Figura 6). Ele destaca essas peças a serem adicionadas nas próximas versões. Além das funções de look at, near e far da câmera (BUTTENBERG, 2020).

Figura 6 - Tela principal do VisEdu-CG 5.0

Interface gráfica do usuário

Descrição gerada automaticamente

Fonte: elaborado pelo autor.

Com relação a conclusão a respeito do resultado da ferramenta, Buttenberg (2018), destaca um nível no consumo de memória maior na maioria dos navegadores, com exceção do Google Chrome sendo estável. E a melhor plataforma em desempenho foi na versão desktop Windows. Por este motivo ele recomenda ser gerado executáveis não apenas para a versão web, e sim para as demais plataformas, visando a melhor resolução de desempenho da ferramenta e tornando-a multiplataforma. Sendo assim, também disponibilizou um tutorial com base nas funcionalidades disponíveis da versão, atendendo seus principais requisitos do projeto, como a migração, podendo manipular peças como: a câmera, o objeto gráfico, o cubo e iluminação, com exceção do Polígono, Spline e outras funcionalidades, como exportação/importação de projetos e a guia de ajuda. Além de que o tutorial inicialmente implementado não trata de todas as funções já desenvolvidas.

## TRABALHOS CORRELATOS

São apresentados três trabalhos correlatos com características semelhantes aos objetivos do trabalho proposto. O primeiro trabalho é uma ferramenta chamada Duolingo, aplicativo para auxílio de aprendizado de múltiplas linguagens e multiplataforma, sendo utilizado em forma de um jogo com desafios diários e metas e recompensas com objetivo de estimular o estudo de outras línguas. O segundo trabalho é o QuestMeter, conforme descreve Vieira (2019), é uma ferramenta de quiz construída com elementos de gamificação juntamente com o conceito de Clickers. O terceiro trabalho é o Toweljs, conforme descreve Zanluca (2018), é um motor gráfico que utiliza JavaScript e WebGL, com objetivo de facilitar a implementação e abstrair o uso dessas duas ferramentas.

Quadro 1 – Duolingo

|  |  |
| --- | --- |
| Referência | Duolingo (2021). |
| Objetivos | Aplicativo com objetivo na aprendizagem de idiomas em formato de jogo. |
| Principais funcionalidades | Dispondo de regras, pontuações, moedas e punições, estimulando atividades diárias de curta duração (em torno de vinte minutos podendo-se ser definido pelo usuário) premiando-o regularmente. |
| Ferramentas de desenvolvimento |  |
| Resultados e conclusões | Segundo MELO (2021), a experiência com o Duolingo no ensino formal de alemão durante um semestre letivo em um curso de Letras/Alemão com estudantes em níveis iniciais de aprendizagem mostrou-se relevante para ampliar o vocabulário do idioma. |

Fonte: elaborado pelo autor.

Nas versões mais atuais, o Duolingo oferece um ambiente interativo tendo uma progressão de atividades realizadas. Cada exercício a ser relizado possui um tutorial explicando pronúncias e como serão abordadas as questões sobre o assunto em questão. A versão gratuita possui vidas onde geralmente são três, ou seja, errando três atividades será obrigatório ter que aguardar elas serem recarregadas, caso seja pago, suas vidas e tentativas serão ilimitadas. Ao longo de todos os testes é sempre possível realizar uma tentativa de aptidão da linguagem, para verificar seu nível de desempenho no conhecimento de novas palavras, por exemplo. Na Figura 1 é possível visualizar a tela principal do Duolingo na versão web.

Figura 1 - Tela principal do aplicativo Duolingo na versão Web

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamente

Fonte: Duolingo (2021).

Quadro 2 – QuestMeter

|  |  |
| --- | --- |
| Referência | QuestMeter (2019). |
| Objetivos | Auxiliar os professores na realização de atividades diversificadas para motivar e engajar os alunos em sala de aula. |
| Principais funcionalidades | O aplicativo possui dois papeis, cada um possuindo funcionalidades diferentes. O papel de professor na ferramenta é manter a funcionalidades como: atividades, questões dentro das atividades, respostas dentro das questões, gerar turmas dentro de atividades, apresentar a atividade criada para os alunos e controlar o andamento da apresentação. Enquanto o papel do aluno é ingressar em atividades disponibilizadas, escolher as opções oferecidas em cada questão da atividade, visualizar as respostas escolhidas, podendo ver as respostas corretas e verificar seu progresso na ferramenta na tela de perfil. |
| Ferramentas de desenvolvimento | Framework Ionic e a plataforma Firebase. |
| Resultados e conclusões | Segundo o autor foi concluido que, foram cumpridos os objetivos definidos, embora os resultados de usabilidade, engajamento e motivação, obtidos tenham sido razoáveis. Os *feedbacks* recebidos dos alunos e dos professores durante os testes e comentários disponíveis nos formulários, foram positivos em sua maioria. |

Fonte: elaborado pelo autor.

É uma ferramenta de quiz construída com elementos de gamificação com o conceito de *Clickers,* utilizando o framework Ionic e a plataforma Firebase. Com o objetivo auxiliar os professores na realização de atividades diversificadas para motivar e engajar os alunos em sala de aula, além disso, outro propósito da ferramenta é testar a interação dos alunos com ferramentas diferenciadas em sala (VIEIRA, 2019).

|  |  |
| --- | --- |
| Referência | Toweljs (2018). |
| Objetivos | Aumentar o nível de abstração na implementação entre o motor de jogos utilizando JavaScript e WebGL. |
| Principais funcionalidades | Disponibiliza a criação de objetos gráficos (cubos e esferas) e luzes permitido juntar tudo numa cena, permite também a criação de dois tipos diferentes de câmera sintética (perspectiva e ortogonal). |
| Ferramentas de desenvolvimento | JavaScript e WebGL. |
| Resultados e conclusões | Segundo o autor, descreve que dentre as dificuldades durante o desenvolvimento, se destaca a forma como foram implementadas as transformações geométricas e as luzes. Para evitar o processamento desnecessário do recalculo da matriz de transformação do objeto a cada vez que o desenhasse, optou-se por recalculá-la somente quando realmente houvesse alguma alteração nos seus valores. O autor também afirma a respeito da arquitetura utilizada, na qual é baseada em componentes, em que se mostrou eficiente para ser utilizada num motor de jogos, que poderá alcançar um baixo nível de acoplamento e um grande grau de reutilização dos componentes. |

Segundo Zanluca (2018), Toweljs é uma implementação de um motor de jogos utilizando JavaScript e WebGL, tendo como principal objetivo facilitar a implementação e aumentar o nível de abstração para aplicações desenvolvidas utilizando essas duas ferramentas. O motor por sua vez disponibiliza a criação de objetos gráficos (cubos e esferas) e luzes permitido juntar tudo numa cena, permite também a criação de dois tipos diferentes de câmera sintética (perspectiva e ortogonal). Tudo isso feito utilizando uma arquitetura baseada em componentes, que ajudou na organização e facilitará futuras expansões do código.

# DESCRIÇÃO

Nesta seção é apresentado o que foi desenvolvido na visão geral da ferramenta. Na primeira seção é realçado uma visão geral da ferramenta atentando as novas funcionalidades, peças e formas de uso. Em seguida a segunda seção são apresentadas as principais técnicas utilizadas na implementação.

## visão geral da ferramenta

A ferramenta implementada tem a tela dividida em quatro painéis, sendo o primeiro painel contendo os seguintes menus: Arquivo, Fábrica de peças, Propriedades das peças e Ajuda. O segundo painel é denominado como Renderer, é utilizado para manter os encaixes das peças e exclusão. Enquanto o terceiro painel tem como função representar as peças graficamente tendo como base os encaixes definidos no segundo painel. E por fim, o painel Visualizador, responsável apenas em demonstrar apenas a forma geométrica das peças dispostas no terceiro painel conforme suas configurações.

// figura

Conforme descreve Buttenberg (2020), ao abrir a ferramenta é exibida a seguinte mensagem: “Deseja realizar o tutorial para aprender os conceitos básicos da ferramenta?”, podendo ser respondido com os botões Sim e Pular tutorial, sendo o segundo botão mostrado a cada passo executado. Se o botão Sim for selecionado, em cada passo em um total de dez passos será demonstrados algumas formas de maneiras de encaixes e definições de algumas propriedades das peças disponíveis a serem encaixadas.

A Fabrica de peças dispõe das peças que podem ser utilizadas no Renderer. São dez peças no total que podem ser encaixadas: a primeira é a Camera, a segunda Objeto Gráfico, a terceira Cubo, a quarta Poligono, a quinta Spline, a sexta, sétima e oitava são as peças de transformações geométricas, são elas: Transladar, Rotacionar, Escalar respectivamente, a nona peça Iteração responsável pelas iterações das transformações com as peças e a última peça Iluminação.

Cada peça possui suas únicas propriedades, como pode ser observado na figura . A peça Polígono possui o nome (propriedade em comum com todas as peças), pontos, a primitiva (tendo opções como: Vértices, Aberto, Fechado, Cheio), cor, a posição (x, y, z) e ativo (funcionalidade para ativar ou não a peça, além da sua exibição no ambiente gráfico). Ao passo que a peça Spline possui nome, pontos de controle (enumerados como: P1, P2, P3, P4, P5), cor e ativo.

// Figura

A Iteração compreende um conjunto de propriedades diferentes dependendo do tipo de transformação geométrica encaixado na peça, podendo ser do tipo Transladar, Rotacionar, Escalar. Suas propriedades são: nome, posições X, Y, Z (Intervalo, Min, Max) e ativo. Onde o intervalo é definido como a quantidade de valor a ser incrementado na posição na qual pertence, e Min é o valor mínimo a ser inicializado quando o valor de incremento chegue ao valor máximo e o Max é o valor máximo que pode ser incrementado.

// Figura iteração

O painel Arquivo tem como principais funcionalidades: importação e exportação de um projeto. Caso seja selecionado a opção de exportação será aberto uma tela de seleção de diretório com um nome sugestivo do arquivo definido como padrão (ProjectVisEdu.json). Selecionado o diretório, consistirá a ser realizado o processo de exportação na estrutura em JSON conforme as peças estejam dispostas nos encaixes do painel Renderer. Na hipótese de ser selecionado a opção de importação, será aberto a tela de seleção de diretório na qual se deve selecionar o projeto a ser importado. Onde ao selecionar e clicar em abrir será realizado o processo de importação conforme a estrutura do projeto, em situação de um projeto já estar definido nos encaixes será descartado para importação completa do projeto selecionado.

// Figura painel Arquivos

Na exportação será verificado uma propriedade chamada de bloqueio de campos, na qual tem como principal objetivo de bloquear as propriedades das peças embaralhamento seus valores, impedindo com que seja alterado ou visualizado pelo usuário. Em situação de importação do projeto, o valor dos campos caso estejam bloqueados é realizado o processo de desembaralhamento. A figura abaixo demonstra o comportamento do bloqueio dos campos, representado por um ícone de cadeado.

// Figura demonstrando o cadeado para bloqueio de campos

O painel de Ajuda contém as informações com o objetivo de documentação das principais funcionalidades da ferramenta. Sendo separado pelos menus: Dicas, Sobre, Fábrica, Espaço Gráfico, Visão da Câmera, Propriedade de Peças, Arquivo, Versões. Cada um deles documentando e auxiliando as respectivas funcionalidades. Na figura abaixo é demonstrado a documentação dos menus Visão de Câmera e Propriedade de Peças.

// Figura do painel Ajuda

## implementação

Os requisitos Funcionais (RF) e Não Funcionais (RNF) são apresentados nos Quadros 5 e 6 respectivamente e foram utilizados como suporte a implementação da ferramenta.

// StartCorroutine, Refatoração, Base64 para bloqueio de campos.

# 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 eles. Também é sugerida a comparação com os trabalhos correlatos apresentados na fundamentação teórica.

# 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

As referências devem ser apresentadas em ordem alfabética. Só podem ser inseridas nas referências os documentos citados ao longo do artigo. Todos os documentos citados obrigatoriamente têm que estar inseridos nas referências. A seguir são apresentados alguns exemplos de referências bibliográficas. Destaca-se que deve ser seguida a norma da ABNT.

[parte de um documento:]

AMADO, Gilles. Coesão organizacional e ilusão coletiva. In: MOTTA, Fernando C. P.; FREITAS, Maria E. (Org.). **Vida psíquica e organização**. Rio de Janeiro: FGV, 2000. p. 103-115.

[trabalho acadêmico ou monografia (TCC/Estágio, especialização, dissertação, tese):]

AMBONI, Narcisa F. **Estratégias organizacionais**: um estudo de multicasos em sistemas universitários federais das capitais da região sul do país. 1995. 143 f. Dissertação (Mestrado em Administração) - Curso de Pós-Graduação em Administração, Universidade Federal de Santa Catarina, Florianópolis.

[norma técnica:]

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6023**: informação e documentação: referências - elaboração. Rio de Janeiro, 2002a. 24 p.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 10520**: informação e documentação: citações em documentos: apresentação. Rio de Janeiro, 2002b. 7 p.

[livro:]

BASTOS, Lília R.; PAIXÃO, Lyra; FERNANDES, Lúcia M. **Manual para a elaboração de projetos e relatórios de pesquisa, teses e dissertações**. Rio de Janeiro: Zahar, 1979.

[trabalho acadêmico ou monografia (TCC/Estágio, especialização, dissertação, tese):]

BRUXEL, Jorge L. **Definição de um interpretador para a linguagem Portugol, utilizando gramática de atributos**. 1996. 77 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

[verbete de enciclopédia em meio eletrônico:]

EDITORES gráficos. In: WIKIPEDIA, a enciclopédia livre. [S.l.]: Wikimedia Foundation, 2006. Disponível em: http://pt.wikipedia.org/wiki/Editores\_graficos. Acesso em: 13 maio 2006.

[artigo em evento:]

FRALEIGH, Arnold. The Algerian of independence. In: ANNUAL MEETING OF THE AMERICAN SOCIETY OF INTERNATIONAL LAW, 61, 1967, Washington. **Proceedings…** Washington: Society of International Law, 1967. p. 6-12.

[norma técnica:]

IBGE. **Normas para apresentação tabular**. 3. ed. Rio de Janeiro, 1993. 61 p. Disponível em: http://biblioteca.ibge.gov.br/visualizacao/monografias/GEBIS%20-%20RJ/normastabular.pdf. Acesso em: 27 ago. 2013.

[artigo em periódico:]

KNUTH, Donald E. Semantic of context-free languages. **Mathematical Systems Theory**, New York, v. 2, n. 2, p. 33-50, jan./mar. 1968.

[trabalho acadêmico ou monografia (TCC/Estágio, especialização, dissertação, tese):]

SCHUBERT, Lucas A. **Aplicativo para controle de ferrovia utilizando processamento em tempo real e redes de Petri**. 2003. 76 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

[página da internet com autor]

SCHULER, João P. S. **Tutorial de Delphi**. Porto Alegre, [2002]. Disponível em: http://www.schulers.com/jpss/pascal/dtut/. Acesso em: 27 ago. 2013.

[página da internet sem autor]

SCHRATCH. **Program, imagine, share**. [S.l.], [2013?]. Disponível em: <https://scratch.mit.edu/>. Acesso em: 27 maio 2013.

[relatório de pesquisa:]

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.

# 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:

1. 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.”;
2. 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;
3. não usar palavras coloquiais;
4. não usar palavras repetidas em demasia;
5. usar verbos no presente quando for referir-se a partes do trabalho que já se encontram disponíveis no texto;
6. destacar palavras em língua estrangeira em itálico, conforme descrito abaixo:
   1. 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),
   2. 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), ...),
   3. palavras em língua estrangeira encontradas nos dicionários nacionais: não devem ser grafadas em itálico (exemplos: software, hardware, web, Internet),
   4. 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.);
7. observar as seguintes regras quanto ao uso de siglas:
   1. 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,
   2. usar apenas a sigla nas demais ocorrências no texto,
   3. escrever as siglas em letras maiúsculas e não usar itálico,
   4. 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).

## formatação

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

1. o texto divide-se em capítulos, seções e subseções (até cinco divisões);
2. a apresentação de citações em documentos deve seguir a NBR 10520 (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2002b);
3. 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 ternária) |
| título de seção quinária (1.1.1.1.1) | TF-Título 5 (mesma formatação seção ternária) |
| 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) |
| itens (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 |
| itens (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 itens (alíneas) devem ser observados os seguintes quesitos:

1. o texto que antecede os itens termina com dois pontos;
2. cada item deve iniciar com uma letra minúscula seguida de fecha parênteses e terminar com um ponto e vírgula, sendo que o último item termina com ponto (FORMATO: TF-ALÍNEA);
3. o texto de cada item inicia com letra minúscula, exceto nomes próprios;
4. quando contiver subitens, 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:

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);
2. 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):
   1. cada subitem (nível 1) inicia com letra minúscula, cada subitem (nível 1) inicia com letra minúscula (FORMATO: TF-SUBALÍNEA nível 1):
      1. cada subitem (nível 2) inicia com letra minúscula, cada subitem (nível 2) inicia com letra minúscula (FORMATO: TF-SUBALÍNEA nível 2),
      2. cada subitem (nível 2) inicia com letra minúscula, cada subitem (nível 2) inicia com letra minúscula (FORMATO: TF-SUBALÍNEA nível 2),
   2. cada subitem (nível 1) inicia com letra minúscula, cada subitem (nível 1) inicia com letra minúscula (FORMATO: TF-SUBALÍNEA nível 1);
3. 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).

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

Formato: TF-TEXTO.

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

Formato: TF-TEXTO.

### 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:

1. aparecer centralizadas no texto;
2. estar delimitadas por uma moldura simples (com exceção das tabelas não quais não devem ser usadas bordas (linhas) verticais em suas extremidades);
3. aparecer numa única página (quando o tamanho não exceder o da página), inclusive a legenda;
4. 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 1.

Figura 1– 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 |
|  | **0** | **202** | **202** |

Fonte: elaborado pelo autor.

### 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 as 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.