

CURSO DE CIÊNCIA DA COMPUTAÇÃO – TCC		
(X) PRÉ-PROJETO	() PROJETO	ANO/SEMESTRE: 2020/1

GERAÇÃO PROCEDURAL DE TERRENOS VIRTUAIS COM APARÊNCIA NATURAL UTILIZANDO GPU

Alex Seródio Gonçalves

Prof. Dalton Solano dos Reis – Orientador

1 INTRODUÇÃO

A forma como construímos nossas cidades e sociedades ao longo da história está diretamente relacionada com o relevo das paisagens ao nosso redor. Essas paisagens estão em constante mudança, sendo influenciadas frequentemente não apenas por forças naturais, mas também por forças econômicas e culturais (PLIENINGER et al., 2015).

Algumas dessas mudanças podem se mostrar complexas de serem analisadas e estudadas no mundo real. Nesses casos, simulações computacionais possibilitam a criação e validação de modelos que representam tais cenários, permitindo assim a execução de experimentos, análise dos resultados e validação de teorias em um ambiente controlado (HEERMANN, 1990, p. 8). Tais simulações podem ser utilizadas para analisar os processos naturais e sociais causadores de mudanças na paisagem, sendo necessário, porém, a utilização de um cenário adequado à simulação.

Para garantir a utilização de terrenos naturais e possibilitar uma variedade de cenários diferentes para as simulações, é possível utilizar terrenos gerados de forma procedural através de modelos físicos. Estes modelos realizam a execução de algoritmos de erosão repetidamente até que o nível de erosão do terreno virtual se equipare ao nível de erosão de um terreno real, exposto a tais processos constantemente durante sua formação (OLSEN, 2004). Esses algoritmos exigem um poder de processamento considerável para atingir resultados realistas, tornando em muitos casos sua execução em tempo real impraticável (MEI; DECAUDIN; HU, 2007, p. 47).

Com os constantes avanços no desenvolvimento de hardware, a utilização da *Graphic Processing Unit* (GPU) para a resolução de problemas computacionalmente complexos vem se tornando cada vez mais viável. Dessa forma é possível utilizar o poder de paralelismo das GPUs modernas para realizar o processamento dos algoritmos de erosão em tempo real e ainda preservar o nível de realismo necessário para simulações (MEI; DECAUDIN; HU, 2007, p. 48).

Diante do apresentado, este trabalho propõe o desenvolvimento de uma ferramenta para geração procedural de terrenos virtuais com aparência natural utilizando modelos físicos. Para tal, propõe também uma versão otimizada em GPU dos algoritmos de erosão de terreno

comumente utilizados pelos modelos físicos, para que seja possível executá-los repetidamente durante a geração do terreno, a fim de alcançar uma aparência natural no menor tempo possível.

1.1 OBJETIVOS

O objetivo deste trabalho é desenvolver uma ferramenta para geração de terrenos virtuais com aparência natural que sejam adequados para uso em simulações.

Os objetivos específicos são:

- a) desenvolver um algoritmo para geração de terrenos com aparência natural;
- b) realizar o processamento do algoritmo em GPU;
- c) disponibilizar uma interface para visualização do terreno gerado.

2 TRABALHOS CORRELATOS

A seguir são apresentados três trabalhos que possuem características semelhantes aos principais objetivos do estudo proposto. O primeiro apresenta um método para sintetizar terrenos fractais de aparência natural utilizando algoritmos procedurais e de erosão (OLSEN, 2004), o segundo é um simulador de dinâmica de relevos que utiliza algoritmos de erosão térmica e erosão hídrica (DIEGOLI NETO, 2017) e o terceiro é um framework para modelagem de terrenos complexos com saliências, arcos e cavernas (PEYTAVIE et al., 2009).

2.1 REALTIME PROCEDURAL TERRAIN GENERATION

Olsen (2004) apresenta em seu trabalho técnicas para geração procedural e erosão de superfícies, que são utilizadas em conjunto para geração de terrenos fractais com uma aparência natural, com o objetivo de serem adequados para utilização em jogos. A implementação foi realizada na linguagem de programação Java e o mapa de altura do terreno gerado é representado utilizando uma matriz bidimensional quadrada de dimensões 512 x 512, onde a altura de cada célula é definida por um número decimal entre 0 e 1 (OLSEN, 2004, p. 1).

O processo apresentado por Olsen (2004) pode ser dividido em duas etapas. A primeira consiste na geração do terreno base através de duas técnicas de geração procedural, sendo elas o *diamond-square* e diagrama de Voronoy, ambos algoritmos para geração de ruídos (OLSEN, 2004, p. 2-4). A segunda etapa consiste em tornar o terreno base recém criado mais natural, submetendo-o à algoritmos de erosão térmica, que simula os materiais dos pontos mais altos se soltando e deslizando pelas encostas até o fundo; e erosão hidráulica, que simula os materiais do terreno sendo dissolvidos, carregados por água corrente e depositados em outro local

(OLSEN, 2004, p. 5-11). A Figura 1 apresenta um terreno gerado a partir da execução dos processos descritos acima.

Figura 1 - Terreno gerado através das técnicas de geração procedural e erosão



Fonte: Olsen (2004, p. 1).

Além disso, são apresentadas análises de performance dos algoritmos de erosão térmica e hidráulica utilizados, as quais são utilizadas para propor uma nova versão otimizada dos mesmos, onde a qualidade da simulação (do ponto de vista físico) é prejudicada em troca de um ganho considerável na velocidade de processamento (OLSEN, 2004, p. 5-11).

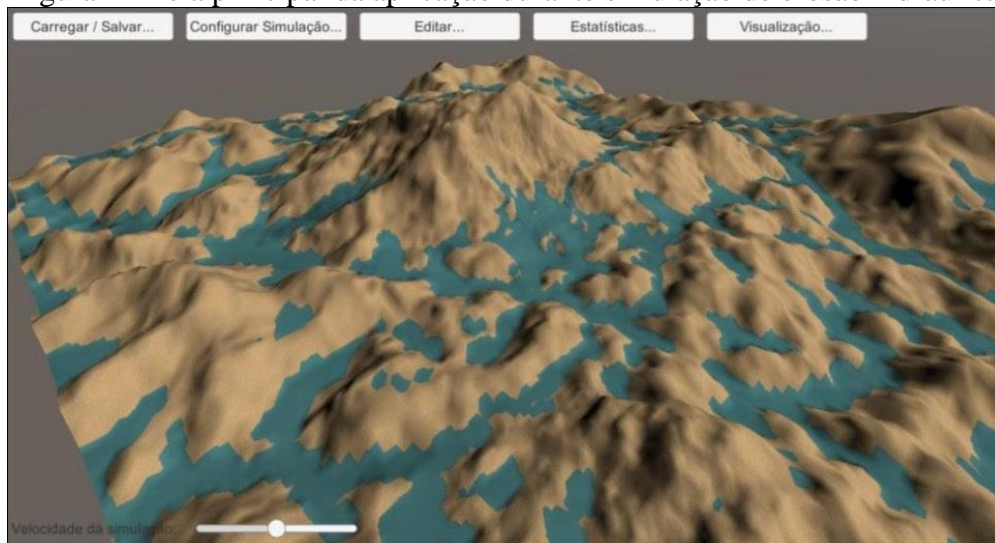
Por fim, Olsen (2004) conclui que as otimizações realizadas no algoritmo de erosão térmica fizeram com que ele se tornasse rápido o suficiente para ser utilizado em um processamento em tempo real, porém resultando em uma redução na qualidade do terreno. Já no caso da versão otimizada do algoritmo de erosão hidráulica, a qualidade do terreno gerado se mostrou maior, porém o tempo de execução continuou sendo muito lento para utilização em tempo real. Ainda assim, os terrenos gerados mostraram-se adequados ao objetivo principal de serem utilizados em jogos (OLSEN, 2004, p. 19).

2.2 SIMULADOR DE DINÂMICA DO RELEVO

Diegoli Neto (2017) apresenta o desenvolvimento de um simulador de deslizamentos de terra. A ferramenta foi desenvolvida utilizando o motor gráfico Unity para visualização da simulação e a linguagem de programação C# para criação dos algoritmos (DIEGOLI NETO, 2017, p. 14). A ferramenta também utiliza algoritmos de erosão térmica e hidráulica para transformar o relevo e assim como Olsen (2004), utiliza uma matriz bidimensional quadrada para representar o terreno, na qual o valor de cada célula representa a altura do terreno naquela

posição. A Figura 2 apresenta a tela principal da aplicação durante a simulação de erosão hidráulica.

Figura 2 - Tela principal da aplicação durante simulação de erosão hidráulica



Fonte: Diegoli Neto (2017, p. 40).

Antes da simulação começar o usuário deve carregar um relevo a partir de um mapa de altura já existente de um terreno (DIEGOLI NETO, 2017, p. 48), podendo escolher diferentes tipos de superfície para diferentes áreas do terreno. As superfícies disponíveis são solo exposto, grama, floresta e concreto, sendo que o tipo da superfície impacta na execução da simulação (DIEGOLI NETO, 2017, p. 42).

Antes de iniciar a simulação é possível selecionar quais algoritmos de erosão serão utilizados, podendo escolher entre erosão térmica e erosão hidráulica (ou ambas). Além disso, é possível configurar algumas variáveis para cada simulação, sendo elas *inclinação máxima* e *fator de alteração* para a erosão térmica e *intensidade de chuva*, *intervalo entre chuvas*, *fator de evaporação* e *fator de solubilidade* para a erosão hidráulica (DIEGOLI NETO, 2017, p. 49-50).

Uma vez que a simulação foi iniciada, é possível acompanhar visualmente as alterações no terreno em tempo real. Para informações mais detalhadas sobre o estado atual do relevo é possível acessar a tela de estatísticas, onde é fornecido várias informações sobre o solo, umidade, altitude, entre outras (DIEGOLI NETO, 2017, p. 45). Também é possível selecionar uma visão por mapas de calor, podendo escolher entre mapas de profundidade do solo, profundidade da água, umidade do solo e inclinação (DIEGOLI NETO, 2017, p. 46).

Diegoli Neto (2017) conclui que a simulação desenvolvida apresenta uma visualização condizente com a realidade, mesmo não sendo capaz de oferecer uma representação precisa dos processos realizados. Além disso, relata que mesmo com o motor gráfico Unity apresentando

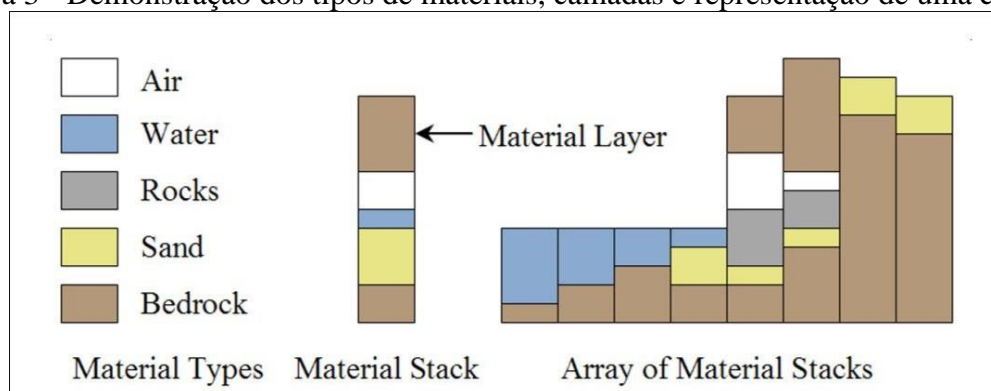
facilidades no desenvolvimento da ferramenta, a plataforma não se mostrou otimizada para trabalhar com atualizações constantes no relevo, o que resultou em uma perda de performance durante as simulações (DIEGOLI NETO, 2017, p. 57-58).

2.3 ARCHES

Enquanto que os trabalhos de Olsen (2004) e Diegoli Neto (2017) apresentados até então se concentram apenas nas características mais gerais do terreno, como montanhas e vales, Peytavie et al. (2009) apresenta um framework focado principalmente na modelagem de formações rochosas mais complexas e específicas, como cavernas, arcos, saliências e penhascos.

Para conseguir modelar essas formações rochosas, Peytavie et al. (2009) utiliza uma forma diferente de representar o terreno gerado. Ao invés de utilizar uma matriz bidimensional onde a altura de cada ponto é representada por um valor escalar (como nos dois trabalhos anteriores), mantém-se a matriz, porém utilizando uma estrutura de camadas em cada posição para representar os vários materiais existentes naquele ponto, sendo que cada material é caracterizado por sua espessura e tipo, podendo ser dos tipos: ar, água, areia, rocha e pedra. Dessa forma, torna-se possível representar uma caverna, por exemplo, simplesmente por inserir uma camada de ar entre duas camadas de rocha (PEYTAVIE et al., 2009, p. 3), como mostrado na Figura 3, possibilitando assim a criação de relevos que não seriam possíveis utilizando apenas um mapa de altura comum.

Figura 3 - Demonstração dos tipos de materiais, camadas e representação de uma caverna



Fonte: Peytavie et al. (2009, p. 3).

Ao invés de trabalhar com geração de terrenos do zero, o framework proposto fornece ferramentas de edição de terreno que permitem que o usuário modele as características principais do terreno através de dois modelos diferentes: um modelo volumétrico discreto, contendo várias camadas de diferentes materiais e um modelo implícito, que representa a superfície do relevo (PEYTAVIE et al., 2009, p. 2).

Após a modelagem inicial, o terreno é submetido a um algoritmo de simulação de estabilização de materiais, no qual materiais granulares como areia, pedra e água interagem uns com os outros e se misturam, a fim de criar um cenário mais próximo do real de forma automática, poupando o usuário deste trabalho (PEYTAVIE et al., 2009, p. 4). Além disso, a etapa de edição do terreno também engloba ferramentas que possuem algoritmos para criar rachaduras ou redes de túneis, realizar a erosão de superfícies e criar pilhas de rochas de forma procedural (PEYTAVIE et al., 2009, p. 5-8).

Peytavie et al. (2009) conclui afirmando ter apresentado uma abordagem original para a representação de estruturas complexas em terrenos, disponibilizando ferramentas que possibilitam a modelagem de arcos, cavernas, penhascos, pilhas de rochas e outras estruturas. Afirma também que a ferramenta se mostrou eficaz, sendo utilizada na construção de uma variedade de cenas com características geológicas únicas (PEYTAVIE et al., 2009, p. 10).

3 PROPOSTA DA FERRAMENTA

Nessa seção será apresentada a justificativa para a realização da pesquisa proposta, assim como os requisitos principais da ferramenta a ser desenvolvida e a metodologia utilizada durante a realização da pesquisa.

3.1 JUSTIFICATIVA

O Quadro 1 abaixo apresenta uma comparação entre os trabalhos correlatos apresentados anteriormente na seção 2.

Quadro 1 - Comparativo entre os trabalhos correlatos

Características \ Correlatos	Olsen (2004)	Diegoli Neto (2017)	Peytavie et al. (2009)
tipo de ferramenta	geração	simulação	modelagem
utiliza algoritmos de erosão	Sim	Sim	Sim
utiliza algoritmos de geração procedural	Sim	Não	Sim
utiliza processamento em GPU	Não	Não	Não
permite visualização do terreno	Sim	Sim	Sim

Fonte: elaborado pelo autor.

Conforme é possível observar no Quadro 1, os três trabalhos apresentados possuem objetivos diferentes. Olsen (2004) é o único cuja finalidade é a mesma que o trabalho proposto, ou seja, a geração de terrenos virtuais, enquanto Diegoli Neto (2017) apresenta um simulador de processos de dinâmica de relevo e Peytavie et al. (2009) apresenta uma ferramenta para modelagem de terrenos.

Apesar disso, todos realizam a modificação do terreno base através de algoritmos de erosão, porém em contextos diferentes. Olsen (2004) os utiliza para melhorar a qualidade do terreno gerado, tornando-o mais natural, Diegoli Neto (2017) os utiliza como componente principal nas simulações de deslizamento e Peytavie et al. (2009) os utiliza como uma ferramenta de edição que pode ser aplicada em uma área específica do terreno. Os algoritmos de geração procedural, por outro lado, são utilizados apenas por Olsen (2004) e Peytavie et al. (2009), sendo que o primeiro os utiliza para a geração do terreno base, enquanto o segundo não os utiliza diretamente no terreno, mas sim na criação de pilhas de rochas.

Em relação a utilização de GPU como forma de otimização dos algoritmos, nenhum deles possui tal implementação. Apesar de Peytavie et al. (2009) mencionar que a etapa de renderização do terreno produzido por sua ferramenta poderia ser facilmente implementada como um *shader* em GPU (PEYTAVIE et al., 2009, p. 2), de Diegoli Neto (2017) sugerir que a renderização de seu terreno fosse feita através de outra plataforma que não a Unity a fim de conseguir uma simulação com melhor desempenho (DIEGOLI NETO, 2017, p. 58), e de Olsen (2004) ter criado uma versão otimizada de alguns algoritmos de erosão, mas que não se mostraram totalmente eficazes (OLSEN, 2004, p. 19).

Por fim, também é possível observar que todos possuem uma forma de visualização dos terrenos dentro da própria ferramenta, sem a necessidade de utilizar ferramentas externas para tal.

A partir do apresentado conclui-se que todos os correlatos possuem uma certa preocupação com performance, apesar de nenhum deles tentar resolver este problema com a utilização de programação em GPU. Todos apresentam técnicas semelhantes (porém adaptadas a suas necessidades) de algoritmos de erosão e dois deles, de geração procedural.

Dessa forma, o trabalho proposto mostra-se relevante no que diz respeito a apresentar uma forma de otimização em GPU dos algoritmos de geração procedural e erosão já amplamente utilizados, sem necessitar de uma perda de qualidade no terreno resultante para tal. A pesquisa realizada também estudará a viabilidade de utilizar uma ferramenta de desenvolvimento de jogos em alto nível como Unity em conjunto com programação em GPU. Além disso, por se tratar de uma área ainda não muito estudada em nossa universidade, este trabalho tem potencial de possibilitar a abertura de novas pesquisas referentes a programação em GPU, como simulação de fluídos e partículas, que tendem a serem direcionadas a GPU e portanto utilizar os mesmos princípios que serão estudados aqui.

3.2 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

A ferramenta proposta neste trabalho deverá:

- a) permitir a geração de terrenos virtuais com aparência natural (RF - Requisito Funcional);
- b) permitir a parametrização de algumas propriedades do terreno gerado (RF);
- c) permitir a visualização do terreno em 3D (RF);
- d) permitir a navegação pelo terreno através da movimentação da câmera (RF);
- e) permitir a exportação do terreno gerado no formato de mapa de altura (RF);
- f) permitir a importação de um terreno previamente gerado (RF);
- g) permitir a visualização de características do terreno através de mapas de calor como mapas de altura e inclinação (RF);
- h) ser desenvolvido para computadores com placa de vídeo (RNF - Requisito Não Funcional);
- i) ser desenvolvido utilizando a plataforma Unity para a visualização do terreno e a linguagem C# para implementação dos algoritmos gerais (RNF);
- j) utilizar as linguagens ShaderLab e HLSL para otimização dos algoritmos de erosão em GPU (RNF).

3.3 METODOLOGIA

O trabalho será desenvolvido observando as seguintes etapas:

- a) levantamento bibliográfico: realizar levantamento bibliográfico sobre formação natural de relevos, técnicas para geração de terrenos virtuais com aparência natural e processamento paralelo em GPU;
- b) elicitação de requisitos: detalhar e reavaliar os requisitos de acordo com o levantamento bibliográfico;
- c) especificação da solução: formalizar as funcionalidades da ferramenta através dos diagramas da *Unified Modeling Language* (UML), utilizando a ferramenta Draw.io;
- d) implementação da ferramenta: implementar a ferramenta proposta utilizando as linguagens C#, ShaderLab, HLSL e o motor gráfico Unity;
- e) validação: validação dos resultados alcançados através de testes de performance para avaliar a velocidade dos algoritmos implementados e análise da naturalidade dos terrenos gerados por um profissional da área.

As etapas serão realizadas nos períodos relacionados no Quadro 2.

Quadro 2 - Cronograma

etapas / quinzenas	2020									
	ago.		set.		out.		nov.		dez.	
	1	2	1	2	1	2	1	2	1	2
levantamento bibliográfico										
elicitação de requisitos										
especificação da solução										
implementação da ferramenta										
validação										

Fonte: elaborado pelo autor.

4 REVISÃO BIBLIOGRÁFICA

Este capítulo apresenta os assuntos que fundamentarão a pesquisa a ser realizada, sendo eles: criação de terrenos procedurais e programação em GPU.

Para a criação de terrenos de forma procedural pretende-se utilizar a combinação das técnicas de Fournier, Fussell e Carpenter (1982) e Ebert et al. (2003), apresentadas por Olsen (2004) para a geração do terreno base, seguido da aplicação repetitiva dos algoritmos de erosão térmica e hidráulica implementados por Diegoli Neto (2017).

A programação em GPU será utilizada para adaptar os algoritmos implementados por Diegoli Neto (2017) a fim de melhorar sua performance. Para que tal adaptação seja possível, serão utilizados como base os guias de programação em GPU de Pharr e Fernando (2005) e Nguyen (2007), assim como o Unity Manual (2020) que apresenta explicações sobre criação de *shaders* e outros componentes gráficos na plataforma Unity.

REFERÊNCIAS

DIEGOLI NETO, Guilherme. **Simulação de dinâmica do relevo através da transformação de mapas de altura**. 2017. 64 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

EBERT, David. S. et al. **Texturing and Modeling: A Procedural Approach**. 3. ed. São Francisco: Morgan Kaufmann, 2003.

FOURNIER, Alain; FUSSELL, Don; CARPENTER, Loren. Computer rendering of stochastic models. **Communications of the ACM**, New York, v. 25, n. 6, p. 371-384, jun. 1982.

HEERMANN, Dieter W. **Computer Simulation Methods in Theoretical Physics**. 2. ed. Heidelberg: Springer, 1990.

MEI, Xing; DECAUDIN, Philippe; HU, Bao-Gang. Fast hydraulic erosion simulation and visualization on GPU. In: PACIFIC CONFERENCE ON COMPUTER GRAPHICS AND APPLICATIONS, 15., 2007, Maui. **Proceedings...** Piscataway: IEEE, 2007. p. 47-56.

NGUYEN, Hubert. **GPU Gems 3: Programming Techniques for High-Performance Graphics and General-Purpose Computation**. Boston: Addison-Wesley, 2007.

OLSEN, Jacob. **Realtime Procedural Terrain Generation**: Realtime Synthesis of Eroded Fractal Terrain for Use in Computer Games. 2004. 20 f. Department of Mathematics and Computer Science, University of Southern Denmark, Odense.

PEYTAVIE, Adrien et al. Arches: a framework for modeling complex terrains. **Eurographics**, Blackwell, v. 28, n. 2, p. 457-467, abr. 2009.

PHARR, Matt; FERNANDO, Randima. **GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation**. Boston: Addison-Wesley, 2005.

PLIENINGER, Tobias et al. Exploring ecosystem-change and society through a landscape lens: recent progress in European landscape research. **Ecology and Society**, [S.l.], v. 20, n. 2, art. 5, jun. 2015.

UNITY TECHNOLOGIES. **Unity manual**. [S.l.], 2020. Disponível em: <<https://docs.unity3d.com/Manual/index.html>>. Acesso em: 10 abr. 2020.

ASSINATURAS

(Atenção: todas as folhas devem estar rubricadas)

Assinatura do(a) Aluno(a): _____

Assinatura do(a) Orientador(a): _____

Assinatura do(a) Coorientador(a) (se houver): _____

Observações do orientador em relação a itens não atendidos do pré-projeto (se houver):

FORMULÁRIO DE AVALIAÇÃO – PROFESSOR TCC I

Acadêmico(a): _____

Avaliador(a): _____

ASPECTOS AVALIADOS ¹		atende	atende parcialmente	não atende
ASPECTOS TÉCNICOS	1. INTRODUÇÃO O tema de pesquisa está devidamente contextualizado/delimitado?			
	O problema está claramente formulado?			
	2. OBJETIVOS O objetivo principal está claramente definido e é passível de ser alcançado?			
	Os objetivos específicos são coerentes com o objetivo principal?			
	3. TRABALHOS CORRELATOS São apresentados trabalhos correlatos, bem como descritas as principais funcionalidades e os pontos fortes e fracos?			
	4. JUSTIFICATIVA Foi apresentado e discutido um quadro relacionando os trabalhos correlatos e suas principais funcionalidades com a proposta apresentada?			
	São apresentados argumentos científicos, técnicos ou metodológicos que justificam a proposta?			
	São apresentadas as contribuições teóricas, práticas ou sociais que justificam a proposta?			
	5. REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO Os requisitos funcionais e não funcionais foram claramente descritos?			
	6. METODOLOGIA Foram relacionadas todas as etapas necessárias para o desenvolvimento do TCC?			
	Os métodos, recursos e o cronograma estão devidamente apresentados e são compatíveis com a metodologia proposta?			
ASPECTOS METODOLÓGICOS	7. REVISÃO BIBLIOGRÁFICA (atenção para a diferença de conteúdo entre projeto e pré-projeto) Os assuntos apresentados são suficientes e têm relação com o tema do TCC?			
	As referências contemplam adequadamente os assuntos abordados (são indicadas obras atualizadas e as mais importantes da área)?			
	8. LINGUAGEM USADA (redação) O texto completo é coerente e redigido corretamente em língua portuguesa, usando linguagem formal/científica?			
	A exposição do assunto é ordenada (as ideias estão bem encadeadas e a linguagem utilizada é clara)?			
	9. ORGANIZAÇÃO E APRESENTAÇÃO GRÁFICA DO TEXTO A organização e apresentação dos capítulos, seções, subseções e parágrafos estão de acordo com o modelo estabelecido?			
	10. ILUSTRAÇÕES (figuras, quadros, tabelas) As ilustrações são legíveis e obedecem às normas da ABNT?			
	11. REFERÊNCIAS E CITAÇÕES As referências obedecem às normas da ABNT?			
	As citações obedecem às normas da ABNT?			
	Todos os documentos citados foram referenciados e vice-versa, isto é, as citações e referências são consistentes?			

PARECER – PROFESSOR DE TCC I OU COORDENADOR DE TCC (PREENCHER APENAS NO PROJETO):

O projeto de TCC será reprovado se:

- qualquer um dos itens tiver resposta NÃO ATENDE;
- pelo menos **4 (quatro)** itens dos **ASPECTOS TÉCNICOS** tiverem resposta ATENDE PARCIALMENTE; ou
- pelo menos **4 (quatro)** itens dos **ASPECTOS METODOLÓGICOS** tiverem resposta ATENDE PARCIALMENTE.

PARECER: () APROVADO () REPROVADO

Assinatura: _____ Data: _____

¹ Quando o avaliador marcar algum item como atende parcialmente ou não atende, deve obrigatoriamente indicar os motivos no texto, para que o aluno saiba o porquê da avaliação.

FORMULÁRIO DE AVALIAÇÃO – PROFESSOR AVALIADOR

Acadêmico(a): _____

Avaliador(a): _____

ASPECTOS AVALIADOS ¹		atende	atende parcialmente	não atende	
ASPECTOS TÉCNICOS	1. INTRODUÇÃO O tema de pesquisa está devidamente contextualizado/delimitado? O problema está claramente formulado?				
	2. OBJETIVOS O objetivo principal está claramente definido e é passível de ser alcançado? Os objetivos específicos são coerentes com o objetivo principal?				
	3. TRABALHOS CORRELATOS São apresentados trabalhos correlatos, bem como descritas as principais funcionalidades e os pontos fortes e fracos?				
	4. JUSTIFICATIVA Foi apresentado e discutido um quadro relacionando os trabalhos correlatos e suas principais funcionalidades com a proposta apresentada? São apresentados argumentos científicos, técnicos ou metodológicos que justificam a proposta? São apresentadas as contribuições teóricas, práticas ou sociais que justificam a proposta?				
	5. REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO Os requisitos funcionais e não funcionais foram claramente descritos?				
	6. METODOLOGIA Foram relacionadas todas as etapas necessárias para o desenvolvimento do TCC? Os métodos, recursos e o cronograma estão devidamente apresentados e são compatíveis com a metodologia proposta?				
	7. REVISÃO BIBLIOGRÁFICA (atenção para a diferença de conteúdo entre projeto e pré-projeto) Os assuntos apresentados são suficientes e têm relação com o tema do TCC? As referências contemplam adequadamente os assuntos abordados (são indicadas obras atualizadas e as mais importantes da área)?				
	ASPECTOS METODOLÓGICOS	8. LINGUAGEM USADA (redação) O texto completo é coerente e redigido corretamente em língua portuguesa, usando linguagem formal/científica? A exposição do assunto é ordenada (as ideias estão bem encadeadas e a linguagem utilizada é clara)?			

**PARECER – PROFESSOR AVALIADOR:
(PREENCHER APENAS NO PROJETO)**

O projeto de TCC deverá ser revisado, isto é, necessita de complementação, se:

- qualquer um dos itens tiver resposta NÃO ATENDE;
- pelo menos **5 (cinco)** tiverem resposta ATENDE PARCIALMENTE.

PARECER: () APROVADO () REPROVADO

Assinatura: _____ Data: _____

¹ Quando o avaliador marcar algum item como atende parcialmente ou não atende, deve obrigatoriamente indicar os motivos no texto, para que o aluno saiba o porquê da avaliação.