# Plano de Ensino-Aprendizagem Integral

**Fluxo**

| Situação | Data | Executor | Descrição |
|---|---|---|---|
| Disponível para elaboração | 15-12-2025 15:45:11 | Alan Rafael Moser | |
| Em elaboração | 16-12-2025 15:08:16 | Dalton Solano dos Reis | |

**Informações FURB**

## Plano de Desenvolvimento Institucional - PDI

**Missão:** promover o ensino, a pesquisa e a extensão, fomentando o desenvolvimento socioeconômico sustentável e o bem-estar social.

**Visão:** ser uma Universidade pública, reconhecida pela qualidade da sua contribuição na vida regional, nacional e global.

**Valores:** transparência; participação; valorização dos discentes e dos servidores; formação integral do ser humano; democracia; ética; pluralidade; desenvolvimento social e sustentável; manutenção da sua identidade e tradição; respeito à natureza e a todas as formas de vida.

## Projeto Pedagógico Institucional - PPI

**Princípios do Ensino:** Democracia e Direitos Humanos; ética e Cidadania ambiental; relações étnico-sociais; formação Crítica.

**Diretrizes para o Ensino:** aprendizagem como foco do processo; educação geral; flexibilização; tecnologias digitais, internacionalização.

**Identificação**

| | | |
|---|---|---|
| **Ano/Semestre:** | 2026/1 | **Turma:** CMP.0186.00.002 |
| **Nome da Disciplina:** | Computação Gráfica | |
| **Centro:** | Centro de Ciências Exatas e Naturais | |
| **Departamento:** | Departamento de Sistemas e Computação | |

**Carga Horária**

| Créditos | | | Carga Horária semestral | | |
|---|---|---|---|---|---|
| **Teóricos:** 4 | **Práticos:** 0 | **Total:** 4 | **Teórica:** 72 | **Prática:** 0 | **Total:** 72 |

**Cursos**

**28 - Ciência da Computação (Matutino)** — **Currículo: 2019/1 Fase(s): 7/A**

**Objetivo do curso**

O curso de Ciência da Computação da Universidade Regional de Blumenau tem como objetivo formar um profissional com conhecimento científico e base sólida em computação, atendendo de forma proativa e ética às demandas da comunidade regional.

**Objetivo geral da disciplina**

Compreender e implementar técnicas básicas que permitam a visualização e edição interativa de modelos gráficos vetoriais.

**Ementa**

Conceitos básicos de Computação Gráfica: primitivas geométricas, modelagem de curvas, sistemas de cores, componentes de sistemas gráficos e programação utilizando API gráfica. Computação

Gráfica 2D: representação e modelagem de objetos, transformações geométricas e processo clássico de visualização. Computação Gráfica 3D: representação e modelagem de objetos, transformações geométricas, câmera sintética, aplicação de textura e iluminação.

**Pré-Requisitos**

| Nome da Disciplina | Código da disciplina | Tipo |
|---|---|---|

**Professor(es)**

**Dalton Solano dos Reis (Cursando Doutorado em Ciências da Computação)**

Dados Complementares do(a) Professor(a):

E-mail professor: dalton@furb.br
Material da disciplina (AVA3 e no Repositório GIT): https://github.com/dalton-reis/disciplina_CG_2026_1
Cronograma detalhado da disciplina: https://github.com/dalton-reis/disciplina_CG_2026_1/blob/main/cronograma.md

Home: https://github.com/dalton-reis/dalton-reis

| Unidades e Subunidades | Objetivos Específicos | Procedimentos Metodológicos | Instrumentos e Critérios de Avaliação |
|---|---|---|---|
| 1.Introdução a sistemas gráficos:<br>1.1. Histórico e aplicações,<br>1.2. Conceitos gerais,<br>1.3. Principais áreas da Computação Gráfica,<br>1.4. Dispositivos de entrada e saída gráficos,<br>1.5 Introdução a biblioteca gráfica (OpenGL). | Identificar os conceitos gerais da Computação Gráfica e as principais áreas de atuação. | Aula expositiva dialogada, material programado e atividades em grupo (laboratório). | Instrumento:<br>Seminário com avaliação individual (N1).<br><br>Critério:<br>Compreensão conceitos básicos das principais áreas gráficas, dos dispositivos de entrada/saída e introdução a uma biblioteca gráfica. |
| 2. Conceitos básicos de Computação Gráfica:<br>2.1 Estruturas de dados para geometria,<br>2.2 Sistemas de coordenadas na biblioteca gráfica (OpenGL),<br>2.3 Primitivas básicas (vértices, linhas, polígonos, círculos e curvas cúbicas - splines). | Aplicar os conceitos básicos de sistemas de referências e modelagem geométrica em Computação Gráfica. | Aula expositiva dialogada, material programado e atividades em grupo (laboratório). | Instrumento:<br>Exercícios em grupo com avaliação individual (N2).<br><br>Critério:<br>Capacidade de entender dados geométricos, coordenadas e primitivas básicas. |
| 3. Conceitos básicos de 2D:<br>3.1 Programação orientada a eventos,<br>3.2 Funções callback (teclado e mouse),<br>3.3 Algoritmos de seleção e Boundaring Box,<br>3.4 Modelos de Cores,<br>3.5 Iluminação e aplicação de texturas. | Demonstrar conhecimento no desenvolvimento de sistemas com interface gráfica com o usuário. Interpretar, especificar e desenvolver aplicações simples que envolvam conceitos básicos 2D. | Aula expositiva dialogada, material programado e atividades em grupo (laboratório). | Instrumento:<br>Trabalho em grupo com avaliação individual (N3).<br><br>Critério:<br>Compreensão sobre algoritmos de seleção e modelos de cores e iluminação. Implementação correta dos problemas propostos. |

| | | | |
|---|---|---|---|
| 4. Conceitos básicos de 3D:<br>4.1 Pipeline de visualização: loop, display e render,<br>4.2 Sistemas de referência e Câmera sintética (profundidade de campo),<br>4.3 Projeções (ortogonal e perspectiva) e viewport,<br>4.4 Coordenadas homogêneas, Transformações geométricas 2D/3D e Composição de transformações geométricas. | Demonstrar conhecimentos teóricos e práticos nos algoritmos básicos de computação gráfica 3D. | Aula expositiva dialogada, material programado e atividades em grupo (laboratório). | Instrumento:<br>Trabalho em grupo com avaliação individual (N4).<br><br>Critério:<br>Compreensão sobre sistemas de referência, câmera sintética, projeção e transformações de coordenadas.<br>Implementação de um pequeno cenário 3D escolhido. |

**Procedimentos de Avaliação**

São 4 (quatro) avaliações compostas por exercícios e trabalhos de implementação computacional. Todos os trabalhos envolvem uma contribuição teórica e/ou implementação. Nas apresentações dos trabalhos, TODOS os integrantes que desenvolveram o trabalho devem participar. A nota da avaliação é individual, mesmo que seja resultado de um trabalho em equipe. Mesmo que no cronograma da disciplina conste mais de um dia para apresentação das avaliações, sempre será considerado como data final de entrega (postagem no AVA3 ou no GitHub, conforme enunciado) um dia antes do primeiro dia das apresentações, e a equipe DEVE utilizar somente do material postado no AVA3 até o seu limite de entrega na apresentação. O aluno deve demonstrar conhecimento do código implementado respondendo principalmente questões relacionadas ao conteúdo apresentado, e não somente saber "ler" o código desenvolvido.

Os critérios de avaliação dos trabalhos são: apresentação oral do trabalho; corretude das soluções apresentadas; uso de metodologia compatível com o conteúdo teórico apresentado pelo professor; atendimento aos requisitos do trabalho; organização, robustez e qualidade geral do código fonte criado; conformidade com os conceitos apresentados em sala de aula.
As atividades práticas podem ser desenvolvidas no sistema operacional Windows, Linux ou MacOS. Aconselhasse manter os mesmos integrantes da equipe em todas as atividades em grupo.

A média final será calculada através de média ponderada das avaliações:
MF = (0.10*N1) + (0.15*N2) + (0.30*N3) + (0.45*N4)
N1: Teórico - áreas gráficas (seminário individual)
N2: Prática em 2D (OpenGL/OpenTK/C#) - lista de exercícios (atividade em equipe)
N3: Prática em 2D (OpenGL/OpenTK/C#) - aplicação, conceitos gráficos básicos (atividade em equipe)
N4: Prática em 3D (OpenGL/OpenTK/C#) - aplicação (atividade em equipe)
obs.: a apresentação oral dos trabalhos N1 e N4 são obrigatórias, pois caso não sejam apresentados a nota do respectivo trabalho será igual a zero.

IMPORTANTE: os trabalhos desenvolvidos nas avaliações N2, N3 e N4 devem usar como base (extendendo e/ou alterando as classes fornecidas SOMENTE quando realmente for necessário - autorizado pelo professor) os exemplos de códigos disponibilizados no repositório da disciplina. O material produzido em TODAS as atividades deve ser postado no GitHub criado pelo professor para cada equipe.

De acordo com o regimento geral da FURB, artigo 66, o aluno que faltar a alguma atividade de avaliação poderá requerer ao professor nova oportunidade em até 5 (cinco) dias úteis, mediante expressa justificativa fundamentada. No caso de provas, esta nova oportunidade será concedida ao final do semestre.

O Regimento Geral da FURB estabelece, no seu Art. 62, que a frequência mínima exigida, para fins de

aprovação, é de 75% da carga horária total da disciplina e média final igual ou superior a 6.0. Abono de faltas e atividades compensatórias irão seguir as determinações legais, que podem ser consultadas neste link:
https://www.furb.br/web/1615/servicos/portal-academico/guia-academico/avaliacao

**Observações**

Ferramentas básicas para a utilização da disciplina (ver https://github.com/dalton-reis/dalton-reis#ferramentas):
- GIT: ferramenta de versionamento de código para uso no GitHub e VSCode.
- IDE: Visual Studio Code (VSCode).
- Linguagem de Programação: C#.
- Biblioteca Gráfica: OpenTK (OpenGL).

Ferramentas opcionais para a utilização da disciplina:
- Blender: ver https://www.blender.org
- Extensões do VSCode: ver https://github.com/dalton-reis/dalton-reis/blob/main/_._/VSCode/VsCodeExtensoes.md

Não é admitida, sob hipótese alguma, cópia de trabalhos ou "compartilhamento de código" com colegas. Todos os trabalhos nos quais o professor concluir que houve cópia (mesmo que parcial) receberão nota zero, não havendo possibilidade de reavaliação dos trabalhos. Os alunos devem tomar os devidos cuidados para proteger seu código contra cópias para reuso em outros trabalhos. Caso venha a usar um repositório (GIT), use-o em modo privado.
Na modalidade de aulas presenciais, o professor pode pedir para desligar o computador (pessoal e/ou do laboratório) durante as aulas (teóricas e/ou práticas), caso julgue necessário.

Mais referências bibliográficas serão disponibilizadas pelo professor durante o desenvolvimento da disciplina (ver http://www.zotero.org/groups/daltonreis_cg), caso seja necessário.
Ressalta-se a importância dos estudantes acompanharem as comunicações via e-mail institucional (@furb.br) e as postagens feitas no Ambiente Virtual de Aprendizagem (AVA3), pois toda comunicação digital será feita por esse e-mail institucional.

As atividades desta disciplina, seguindo a Resolução FURB nº 61/2021, e aprovada no Colegiado de Curso, serão desenvolvidas no modelo "PRESENCIAL", com Professor, Aluno, Aulas e Avaliações todos de forma presencial.

O conteúdo dessa disciplina tem Interdisciplinaridade com:
- Introdução à programação (1a fase): N2, N3 e N4
- Fundamentos Matemáticos (1a fase): N2, N3 e N4
- Programação Orientada a Objetos (2a fase): N2, N3 e N4
- Algoritmos e Estrutura de Dados (3a fase): N3 e N4
- Teoria dos Grafos (4a fase): N3 e N4
- Álgebra Linear (5a fase): N2, N3 e N4
- Geometria Analítica (6a fase): N2, N3 e N4

**Documentos Recomendados**

**Básico**

- AZEVEDO, Eduardo; CONCI, Aura. **Computação gráfica: teoria e prática.** Rio de Janeiro : Elsevier, 2003. xv, 353 p, il. , 1CD-ROM. Acompanha CD-ROM.

- CONCI, Aura; AZEVEDO, Eduardo; LETA, Fabiana R. **Computação gráfica**: [teoria e prática], 2. Rio de Janeiro : Elsevier : Campus, 2008. xi, 407 p, il. , 1 CD-ROM.

- FIGUEIREDO, Luiz Henrique de; CÉSAR, Paulo; INSTITUTO DE MATEMÁTICA PURA E APLICADA (BRASIL). **Introdução a geometria computacional.** Rio de Janeiro : IMPA, 1991. 111p, il. Trabalho apresentado no 18. Colóquio Brasileiro de Matemática.

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

- SILVA, Isabel Cristina Siqueira da. **Aprendendo computação gráfica com OpenGL e Blender**. Porto Alegre : Ed. UniRitter, 2007. 192 p, il. (Experiência acadêmica, 7).

- WOO, Mason; OPENGL ARCHITECTURE REVIEW BOARD. **OpenGL programming guide: the official guide to learning OpenGL, version 1.2.** 3rd ed. Boston : Addison Wesley, 1999. xi, 730p, il.

**Complementar**

- ANGEL, Edward. **OpenGL: a primer.** Boston : Addison-Wesley, 2002. xiii, 235p, il.

- ARVO, James. **Graphics gems II.** Boston : Academic Press : HBJ, c1991. xxxii, 643p, il. (The graphics gems series).

- FOLEY, James D. **Computer graphics: principles and practice.** 2nd ed. Reading : Addison-Wesley, c1990. xxiii, 1175 p, il. , 1 disquete. (Addison-Wesley systems programming series). Acompanha disquete.

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

- HILL, Francis S. **Computer graphics using OpenGL.** 2nd ed. Upper Saddle River, N.J : Prentice Hall, 2001. xxxi, 922p, il.

- MANTYLA, Martti. **An introduction to solid modeling.** Rockville : Computer Sci, c1988. 401p, il. (Principles of computer science series, 13).

- MORTENSON, Michael E. **Mathematics for computer graphics applications.** 2nd ed. New York : Industrial Press, 1999. 354p, il.

- REINICKE, José Fernando. **Modelando personagens com o Blender 3D**. São Paulo: Novatec, 2008. 266 p, il.

- SCHNEIDER, Philip J; EBERLY, David H. **Geometric tools for computer graphics.** Amsterdam : Boston : Morgan Kaufmann Publishers, 2003. xlv, 1009p, il. (The Morgan Kaufmann series in computer graphics and geometric modeling).

- VELHO, Luiz; GOMES, Jonas. **Sistemas gráficos 3D**. Rio de Janeiro : IMPA, 2001. iii, 330 p, il. (Série de computação e matemática).

- WRIGHT, Richard S; LIPCHAK, Benjamin. **OpenGL SuperBible**.3rd ed. Indianopolis : SAMS, 2005. xvii, 1173 p, il. , 1 CD-ROM.

**Eletrônico**

- Biblioteca gráfica OpenGL Acesso em: 04 Fev. 2026.

- Blender is the free open source 3D content creation suite (model, shade, animate, render, composite and interactive 3d), available for all major operating systems under the GNU General Public License. Acesso em: 04 Fev. 2026.

- Game Development Acesso em: 04 Fev. 2026.

- Guia de C# O guia do C# fornece muitos recursos sobre a linguagem C#. Dependendo de sua experiência com programação ou com a linguagem C# e .NET, convém explorar as diferentes seções deste guia. Acesso em: 04 Fev. 2026.

- OpenTK The Open Toolkit is set of fast, low-level C# bindings for OpenGL, OpenGL ES and OpenAL. It runs on all major platforms and powers hundreds of apps, games and scientific research. OpenTK provides several utility libraries, including a math/linear algebra package, a windowing system, and input handling. Acesso em: 04 Fev. 2026.

- The Khronos Group Acesso em: 04 Fev. 2026.

- Welcome to OpenGL Welcome to the online book for learning OpenGL! Whether you are trying to learn OpenGL for academic purposes, to pursue a career or simply looking for a hobby, this book will teach you the basics, the intermediate, and all the advanced knowledge using modern (core-profile) OpenGL. The aim of LearnOpenGL is to show you all there is to modern OpenGL in an easy-to-understand fashion with clear examples, while also providing a useful reference for later studies. Acesso em: 04 Fev. 2026.

- Zotero - Grupo de CG Algumas referências bibliográficas utilizadas na disciplina de Computação Gráfica.

- github.com/dalton-reis/disciplina_CG_2026_1 REIS, Dalton S. dos. Computação Gráfica: notas de aula (GitHub). Blumenau, 2021. Disponível em: https://github.com/dalton-reis/CG_2026_1. Acesso em: 04 Fev. 2026.


DTI - Seção de Desenvolvimento de Sistemas [04-Fev-2026 10:56:32]

**Início** **Meus Planos de Ensino na Graduação** **Sair**