

# ANIMAÇÃO E DESIGN DE PERSONAGENS

---

## INTRODUÇÃO

Existem diversas ferramentas para animação 2D ou 3D; porém, o animador não é um mero usuário de softwares. Embora seja importante conhecer as ferramentas e seus recursos, é essencial compreender que elas não realizam o trabalho por conta própria, nem esse é seu propósito.

Independentemente do nível de sofisticação dos recursos disponíveis, essas ferramentas servem para auxiliar o animador ou modelador a expressar suas ideias de forma mais eficiente. A criatividade e a intencionalidade são sempre atribuições do profissional, e não da ferramenta.

## TEMA 1 – CRIAÇÃO DE PERSONAGENS DIGITAIS

Existem diversas ferramentas tanto para ilustração 2D quanto para modelagem 3D na criação de personagens. No entanto, mais importante do que as ferramentas em si são os conceitos fundamentais que permeiam essas áreas. Por exemplo, na animação 2D, compreender os princípios de sprites e quadros-chave para interpolação é mais relevante do que dominar um software específico. De modo similar, na modelagem 3D, não ter noções de topologia pode ser um obstáculo muito maior do que dificuldades para usar certo software.

### 1.1 Técnicas na criação de personagens

Ao longo do tempo, diferentes técnicas foram utilizadas na criação de personagens para jogos e animação. Um exemplo clássico é o Mortal Kombat, que, em suas primeiras versões, utilizava fotos digitalmente tratadas de atores para criar um efeito realista. Já para criaturas mais fantasiosas do jogo, foram utilizados bonecos fotografados quadro a quadro, em uma abordagem semelhante ao stop motion. Atualmente, a criação de personagens envolve uma série de etapas estruturadas, sendo uma das mais importantes a construção do model sheet.

O **model sheet** é um guia visual que apresenta as proporções e as poses de um personagem, auxiliando em sua consistência visual ao longo do processo de animação e modelagem. Existem diferentes formatos de model sheets, a saber:

- model sheets com quatro vistas (frente, costas, perfil e topo): para modelagem 3D;
- model sheets com oito vistas: mais detalhados, usados para refinar a

---

silhueta do personagem;

- model sheets com poses de ação: focados na expressividade e no movimento do personagem.

Embora não haja uma regra para o formato do model sheet, sua criação é essencial para garantir a consistência visual do personagem.

No 3D, o model sheet é frequentemente utilizado como blueprint (não confundir com o sistema de código visual do Unreal Engine), servindo como referência direta para a modelagem. Após a modelagem, o personagem passa pelas fases de texturização e rigging, ficando pronto para ser animado.

No 2D, o processo pode variar conforme o tipo de animação utilizada:

- **Animação baseada em esqueleto (rigging):** Exige que os membros do personagem estejam separados para aplicação do sistema de ossos.
- **Animação quadro a quadro:** A consistência entre os quadros é fundamental, tornando o model sheet uma referência para evitar discrepâncias visuais, pois sua ausência ou seu uso inadequado pode resultar em inconsistências, fazendo o personagem finalizado diferir do conceito original.

## 1.2 Modularidade na criação de personagens

A modularidade na criação de personagens oferece maior personalização e flexibilidade dentro do jogo.

Na modelagem 3D, o modelador deve considerar a topologia do personagem para facilitar a animação e otimizar a performance do sistema. Além disso, personagens podem ser criados com componentes intercambiáveis, como roupas e acessórios, possibilitando personalizações diretas dentro da game engine.

Na arte 2D, personagens criados com modularidade podem ser desenhados em camadas separadas, o que facilita troca de partes e aplicações de rigging. Quando se utiliza rigging, é preciso separar os membros do personagem corretamente para ter mais controle do esqueleto digital.

## TEMA 2 – FERRAMENTAS DE ILUSTRAÇÃO E ANIMAÇÃO 2D

Há diversas ferramentas disponíveis para a criação de arte 2D de

---

personagens para jogos, e a escolha pode variar de acordo com o estilo da arte desejado. Algumas ferramentas são mais adequadas para pixel art (arte pixelada que remete aos jogos antigos), ao passo que outras trabalham melhor com vetores (objetos cujas formas e linhas são baseadas em cálculos matemáticos, garantindo que não fiquem pixelados ao serem escalados).

## 2.1 Softwares de ilustração e pintura digital

Para a criação de sprites e ilustrações 2D, há várias opções populares:

- Adobe Photoshop: ampla utilização para ilustração digital e criação de sprites;
- Adobe Illustrator: especializado em arte vetorial, útil para assets escaláveis;
- Krita: alternativa gratuita para pintura digital e animação quadro a quadro;
- Clip Studio Paint: muito usado para ilustração e animação estilo mangá;
- Affinity Designer: alternativa ao Illustrator, eficiente para design vetorial;
- Aseprite: especializado em pixel art, utilizado em jogos retrô;
- Piskel: ferramenta on-line gratuita para pixel art e sprites animados.

Além dos softwares tradicionais, existem ferramentas que se integram a engines e facilitam o fluxo de trabalho com arte modular e animação.

## 2.2 Softwares de animação 2D

Para animação 2D baseada em rigging (esqueleto digital), algumas das principais ferramentas são:

- Adobe Animate: voltado para animações vetoriais, muito usado na criação de jogos e interações web;
- Toon Boom Harmony: um dos mais completos para animação 2D, usado por grandes estúdios;
- Spine 2D: muito popular para animação em jogos, com integração nativa ao Unity e ao Unreal;
- DragonBones: alternativa gratuita ao Spine, compatível com engines como Godot e Unity;
- Live2D Cubism: excelente para criar animações semi-3D, como em visual novels e VTubers;

- Open Toonz: software gratuito e open-source, adotado em animação tradicional.

Caso não se utilize rigging na animação, a abordagem pode ser quadro a quadro, sendo possível recorrer a softwares como:

- TVPaint Animation: um dos melhores para animação tradicional digital;
- Toon Boom Harmony: Além de rigging, tem suporte robusto para animação quadro a quadro;
- Krita: é gratuito e oferece suporte para animação quadro a quadro;
- RoughAnimator: alternativa acessível para animação desenhada à mão;
- FlipBook: simples e eficiente para animação tradicional.

A escolha do software depende do estilo de animação desejado e da engine utilizada no jogo.

## 2.3 Integração com engines e modularidade

O ilustrador ou animador precisa ter clareza sobre como o personagem será utilizado dentro do jogo, pois uma escolha errada pode gerar dificuldades na programação. Por exemplo, se um personagem precisa de diferentes armaduras ao longo do jogo, sua arte deve ser modular, de modo que elementos sejam aplicados dinamicamente no código. Em jogos que utilizam rigging, o personagem precisa ser criado separado em camadas, para que os ossos possam ser ajustados sem restrições.

Entre as soluções para animação modular em jogos, destacam-se:

- Spine 2D: uma das melhores opções para animação modular em jogos, sendo altamente otimizado para Unity, Unreal e Godot;
- DragonBones: alternativa gratuita ao Spine, também voltada para jogos;
- Anima2D: plug-in para Unity que permite animações baseadas em rigging 2D;
- Paper2D: sistema nativo do Unreal Engine para animação 2D e sprites;
- Unity 2D Animation Package: ferramenta oficial do Unity para rigging e animação de personagens 2D.

A escolha da engine e do sistema de animação impacta diretamente no desempenho do jogo. O rigging pode ser uma opção mais leve, pois carregar

---

muitos sprites pode consumir muita memória. No entanto, dependendo da complexidade da animação, pode ser necessário equilibrar qualidade e desempenho.

## **2.4 Uso de inteligência artificial na ilustração**

Uma opção que tem ganhado espaço no fluxo de trabalho é o uso de inteligência artificial generativa para a criação de assets e conceitos visuais. Alguns softwares e plataformas que têm sido empregados nessa tarefa são:

- Stable Diffusion (SDXL): utilizado para gerar imagens detalhadas;
- Leonardo.AI: possibilita a criação de arte conceitual e estilos variados;
- Runway ML: IA voltada para edição e aprimoramento de animações;
- ControlNet para Stable Diffusion: possibilita modificar imagens mantendo coerência estrutural.

Embora essas ferramentas acelerem a produção, frisamos que a IA não substitui o trabalho criativo do artista; seu papel é auxiliar no processo, seja para gerar referências visuais, seja para aprimorar elementos dentro de softwares de edição de imagem.

O principal desafio no uso de IA é obter coerência e consistência visual, especialmente para jogos que exigem uniformidade no estilo artístico. Para isso, modelos como LoRA (Low-Rank Adaptation) podem ser treinados para manter padrões visuais consistentes.

Escolher a ferramenta mais adequada não deve levar em consideração apenas critérios técnicos; deve ser uma decisão alinhada com as necessidades do projeto e a engine utilizada. Independentemente do software escolhido, o conhecimento sobre modularidade, animação e integração com engines é fundamental para criar personagens e animações eficientes no jogo.

Além disso, o avanço das ferramentas de IA abre novas possibilidades para o desenvolvimento de ilustrações e animações, mas não substitui o artista na criação de personagens únicos e bem planejados.

## **TEMA 3 – FERRAMENTAS DE MODELAGEM E ANIMAÇÃO 3D**

A direção de arte é crucial na escolha das ferramentas mais adequadas para o trabalho em 3D. Dependendo do nível de detalhamento desejado, a

---

modelagem pode ser abordada com técnicas distintas, exigindo softwares específicos para low poly ou high poly.

Por exemplo, para personagens low poly, geralmente não é necessário um software de escultura digital. Técnicas como box modeling são mais eficientes e podem ser aplicadas em programas como 3ds Max, Blender ou Maya. Já para personagens mais detalhados, a escultura digital é essencial. Ferramentas como ZBrush e Mudbox são indicadas para criar detalhes minuciosos antes de aplicar um retopology para otimização da malha.

No entanto, mais importante do que saber usar o software em si, o modelador 3D precisa ter domínio dos conceitos fundamentais de modelagem, texturização e animação para obter bons resultados.

### 3.1 Conceitos fundamentais de modelagem 3D

O conhecimento não está vinculado a uma ferramenta específica, mas a conceitos que permeiam toda a modelagem 3D. Alguns dos mais importantes são:

- **Poligonagem:** Representa a quantidade de faces de um modelo 3D, impactando diretamente no desempenho do jogo.
- **Topologia:** Refere-se à organização das faces e arestas do modelo. Uma topologia mal estruturada pode inviabilizar rigging e animação.

Os tipos de polígonos são:

- Quads (quatro vértices): mais comuns em modelagem para animação;
- Tris (três vértices): utilizados em engines e modelagem otimizada;
- Ngons (mais de quatro vértices): podem gerar problemas na animação e precisam ser evitados.

Além disso, um objeto 3D é composto de vértices, arestas e faces, e sua estrutura deve equilibrar fidelidade visual e otimização de desempenho.

### 3.2 Técnicas de otimização na modelagem 3D

Otimização é um fator elementar no desenvolvimento de jogos 3D. Algumas das principais técnicas utilizadas são:

- **Level of Detail (LOD):** consiste em utilizar modelos de diferentes resoluções dependendo da distância do personagem em relação ao

---

espectador. O Unreal Engine dispõe de suporte nativo para LODs em sistema próprio, o Nanite, que facilita lidar com modelos de alta poligonagem de maneira eficiente.

- **Normal Maps:** conferem sensação de detalhes extras sem aumentar a poligonagem do modelo. São empregados para personagens esculpidos no ZBrush, possibilitando projetar detalhes de alta resolução em modelos otimizados.

As principais ferramentas para otimização são:

- Simplygon: usado para geração automática de LODs;
- Decimation Master (ZBrush): reduz a quantidade de polígonos sem perder detalhes;
- Quad Remesher: plug-in para retopologia automática;
- Instant Meshes: opção gratuita para retopologia procedural.

### 3.3 Inteligência artificial na modelagem 3D

A inteligência artificial também vem sendo utilizada para criação e refinamento de modelos 3D. Algumas ferramentas que se destacam são:

- Rodin: oferece a geração de modelos 3D a partir de imagens ou descrições textuais;
- DALL-E & Leonardo.AI: podem ser usadas para criar conceitos visuais antes de modelar;
- Kaedim: IA que gera modelos 3D a partir de imagens 2D.

Alertamos que, embora esses recursos possam ser úteis para protótipos, muitos modelos gerados por IA apresentam problemas de topologia e texturização, exigindo refinamento posterior.

### 3.4 Ferramentas de modelagem 3D

Dependendo do fluxo de trabalho, algumas ferramentas são mais indicadas para cada etapa do processo de modelagem. Eis algumas opções:

- Blender: gratuito e open source, é usado para modelagem e animação;
- Autodesk Maya: um dos mais populares na indústria, é excelente para animação;



- 
- Autodesk 3ds Max: muito utilizado para modelagem de ambientes e objetos;
  - ZBrush: especializado em escultura digital, ideal para personagens detalhados;
  - Mudbox: alternativa ao ZBrush para escultura digital;

Entre os softwares para retopologia e otimização, destacamos:

- TopoGun: específico para retopologia manual;
- Blender (Remesh Modifier): tem ferramentas nativas para redução de polígonos.

### 3.5 Ferramentas de rigging e animação 3D

Após a modelagem, um personagem precisa de rigging e animação para ser utilizado em um jogo.

São softwares para rigging automático:

- Mixamo: plataforma da Adobe que oferece rigging automático e biblioteca de animações;
- AccuRig: alternativa gratuita para rigging automatizado;
- Rokoko Studio: possibilita captura de movimento para animação mais realista.

São softwares para animação 3D:

- Cascadeur: oferece um sistema baseado em IA para criar animações físicas realistas;
- Autodesk MotionBuilder: focado em captura e edição de movimento;
- Blender: tem um poderoso sistema de animação com suporte para rigging.

### 3.6 Modularidade na modelagem 3D

A modularidade na modelagem 3D não se aplica apenas à personalização de personagens, mas também à integração com game engines:

- Unreal Engine: seu esqueleto-padrão pode ser utilizado para agilizar o processo de animação. Caso o modelo tenha um esqueleto diferente, o Unreal oferece sistema de retargeting para compatibilizar animações.

- Unity: trata personagens humanoides com o sistema Humanoid Rig, possibilitando que animações sejam reutilizadas facilmente.

Utilizar estruturas compatíveis com a engine pode economizar tempo e otimizar a produção, garantindo que as animações funcionem de forma eficiente.

## TEMA 4 – MOTORES DE JOGOS

Atualmente, existem diversas game engines disponíveis para o desenvolvimento de jogos, cada uma com suas características e especializações. Alguns exemplos populares são: Unreal Engine, Unity, Godot, GameMaker Studio, e RPG Maker.

Com tantas opções, surge a pergunta: Qual engine escolher e como cada uma lida com animação e design de personagens? A resposta varia conforme diversos fatores, como linguagem de programação, estilo visual desejado e necessidades do projeto.

### 4.1 Escolha da engine mais adequada

A escolha da game engine deve ser consciente e alinhada às necessidades do projeto. Alguns fatores a considerar são:

- **Linguagem de programação:** cada engine tem uma linguagem principal.
  - a) Unreal Engine: C++, com suporte ao sistema Blueprints (programação visual);
  - b) Unity: C#;
  - c) Godot: Compatível com C#, GDScript (inspirado em Python) e C++;
  - d) RPG Maker: tem configurações totalmente visuais, com suporte ao Ruby on Rails para scripts avançados.
- **Foco da engine:** algumas engines são mais otimizadas para determinados estilos de jogos:
  - a) Unreal Engine: melhor para gráficos realistas e projetos 3D detalhados;
  - b) Unity e Godot: equilibram bem 2D e 3D, oferecendo grande flexibilidade;
  - c) GameMaker Studio: focado em jogos 2D, utilizado para plataformas e RPGs;

---

d) RPG Maker: ideal para jogos de RPG tradicionais, com sistema pré-configurado para facilitar a produção.<sup>1</sup>

- **Nível da linguagem de programação utilizada:** Linguagens de baixo nível, como C++, são mais rápidas e eficientes, mas exigem maior conhecimento técnico. Já linguagens de alto nível, como C# e GDScript, são mais acessíveis, mas podem ter uma leve perda de performance. No entanto, independentemente da linguagem escolhida, boas práticas de otimização são essenciais. Um código mal estruturado pode tornar qualquer engine lenta, independentemente da linguagem utilizada.

## 4.2 O papel das engines na animação e no design de personagens

Cada game engine lida com personagens e animações de forma específica, oferecendo ferramentas para implementação e manipulação de animações.

- **RPG Maker:** utiliza um sistema baseado em sprites para animação de personagens. As animações podem ser:
  - a) criadas por meio de troca de sprites e programação de movimentação;
  - b) inseridas como cenas pré-renderizadas em vídeo, que são reproduzidas em momentos específicos;
  - c) animadas dentro da limitação da engine, utilizando scripts para controle de movimentos e interações.
- **Unreal Engine, Unity e Godot:** engines mais robustas em que o tratamento da animação e dos personagens é mais flexível e avançado. As animações podem ser trabalhadas de forma similar a softwares profissionais, utilizando:
  - a) quadros-chave (keyframes), que definem poses principais e interpolam movimentos entre elas;
  - b) interpolação de movimento, que oferece a possibilidade de suavizar

---

<sup>1</sup> Alguns argumentam que Unity, Godot e Unreal podem fazer tudo o que o RPG Maker faz e mais. No entanto, embora pareça mais limitado em comparação com outras engines, essa abordagem pode ser mais eficiente para o estilo de jogo que o RPG Maker foi projetado para criar. O RPG Maker conta com um sistema otimizado para criação rápida de RPGs, o que economiza tempo de desenvolvimento. Criar essa mesma estrutura do zero em outra engine pode demandar um esforço significativo, a menos que se utilizem assets prontos das lojas da Unity ou Unreal. De qualquer modo, se o desenvolvedor tem experiência em Unity ou Godot, pode ser mais interessante utilizá-las para aproveitar o conhecimento adquirido e evitar a curva de aprendizado de um novo software.

---

transições entre animações;

- c) sistemas de câmeras e enquadramento, os quais facilitam a criação de cenas cinematográficas no próprio jogo;
- d) blend trees, que misturam diferentes animações para tornar os movimentos mais fluidos.

Engines como Unreal Engine são tão sofisticadas que ferramentas de animação como Sequencer são usadas até na indústria cinematográfica. Enquanto o RPG Maker depende de sprites e scripts simples, Unreal, Unity e Godot possibilitam criar animações diretamente na engine, trabalhar com cinemáticas em tempo real, e utilizar captura de movimento (motion capture) para animações realistas. A flexibilidade maior, no entanto, exige mais trabalho para estruturar e programar tudo do zero.

### 4.3 Modularidade e organização dos personagens

Independentemente da game engine, um fator crucial no desenvolvimento de personagens é a modularidade e a organização do código.

O RPG Maker conta com uma estrutura pré-definida, separando personagens, inimigos e eventos no próprio editor. Já em engines como Unity, Godot e Unreal, o desenvolvedor precisa construir essa estrutura do zero. Uma das práticas recomendadas nesses motores é criar uma classe-pai (base) para representar todos os personagens, e criar subclasses (herança) para personagens específicos, adicionando particularidades a cada um.

Essa abordagem garante que qualquer mudança na estrutura do personagem seja aplicada em todos os personagens do jogo, facilitando o gerenciamento e dispensando a necessidade de duplicar código.

Assinalamos que não existe engine perfeita, mas sim a ferramenta mais adequada para o projeto. É essencial que o desenvolvedor avalie as necessidades do jogo e sua familiaridade com a engine para obter maior produtividade. No que diz respeito à animação, engines como RPG Maker lidam com sprites e scripts simples, ao passo que Unreal, Unity e Godot oferecem abordagens mais avançadas, como interpolação de movimento, blend trees e animação cinematográfica.

Por fim, conceitos como abstração, encapsulamento e modularidade continuam sendo indispensáveis para estruturar personagens em qualquer

---

engine. Quanto mais bem planejada for a estrutura, mais fácil será escalar o projeto e manter a organização do código e das animações.

## TEMA 5 – DIREÇÃO ARTÍSTICA PARA JOGOS

Ao longo deste material, citamos diversos princípios de animação, ferramentas de criação e game engines. No entanto, nenhuma dessas ferramentas será eficaz se a direção artística do jogo não for bem planejada.

Independentemente da qualidade técnica das animações ou do realismo dos modelos 3D, um personagem precisa estar em harmonia com o projeto como um todo. A intencionalidade na direção artística é primordial para que todas as escolhas visuais e estilísticas contribuam para a imersão e a coerência da experiência do jogador.

O uso das melhores tecnologias, sem um direcionamento artístico claro, pode resultar em inconsistências visuais que comprometem a experiência. Em um jogo 2D, 2.5D ou 3D, a unidade gráfica é fundamental para que os personagens e o ambiente se comuniquem visualmente de forma coesa.

### 5.1 O perigo da inconsistência visual e o vale da estranheza

Um dos problemas mais comuns em projetos mal planejados é a desconexão entre a animação e a direção de arte. Por exemplo, um jogo altamente realista, mas com animações robóticas ou exageradamente caricatas, pode gerar um efeito não previsto de estranhamento; igualmente, um jogo estilizado, mas com animações excessivamente realistas, pode perder expressividade e impacto visual.

Esse fenômeno pode levar ao que chamamos de *vale da estranheza* (uncanny valley), quando um personagem ou animação parece realista o suficiente para gerar familiaridade, mas contém elementos sutis que causam desconforto ou estranhamento no espectador.

A quebra da imersão gerada pelo vale da estranheza é similar ao conceito de suspensão de descrença, descrito por McKee (1997). Segundo o autor, toda obra de ficção estabelece um contrato invisível com o espectador, no qual as regras do universo narrativo são definidas pelo criador. Se essas regras forem quebradas de maneira abrupta ou inconsistente, a imersão se perde.

---

Isso também ocorre nos jogos: se um jogador perceber algo incoerente no mundo do jogo, sua imersão será comprometida.

Em jogos realistas, por exemplo, motores gráficos frequentemente utilizam sistemas de física realista, como Ragdoll Physics, para simular quedas e colisões. No entanto, se a física for aplicada de forma exagerada ou mal ajustada, o resultado pode ser cômico, transformando uma cena dramática em algo engraçado.

Já em jogos estilizados, um erro oposto pode acontecer: animações muito realistas podem tornar a movimentação monótona e pouco expressiva. Em cenas de ataques especiais ou momentos cinematográficos, o animador pode precisar exagerar deformações e poses dramáticas para tornar a ação mais impactante.

Portanto, a consistência da direção artística e da animação deve ser cuidadosamente planejada para evitar esses problemas.

## 5.2 Pós-processamento e efeitos visuais

Outro fator importante na direção artística é o uso de efeitos visuais e pós-processamento oferecidos pelas game engines. Os efeitos gráficos podem transformar completamente a estética de um jogo e contribuir para a narrativa visual. Alguns exemplos são:

- **neblina densa:** estabelece um clima de mistério e restringe a visibilidade do jogador;
- **cel shading:** confere aos modelos 3D um aspecto de desenho animado;
- **filtros retrô:** simulam estéticas VHS, pixels borrados ou cores desbotadas, criando um efeito nostálgico.

Embora pareçam apenas detalhes estéticos, esses efeitos têm impacto direto na performance do jogo. Shaders complexos, iluminação dinâmica e efeitos em tempo real podem exigir alto processamento, reduzindo o desempenho do jogo em hardware menos potente. Por isso, é fundamental avaliar cuidadosamente o impacto desses efeitos na jogabilidade e se certificar de que eles se alinham à proposta do jogo.

## 5.3 O papel da direção artística na coesão do jogo

Para garantir uma direção artística bem-sucedida, algumas perguntas

---

devem ser respondidas durante o desenvolvimento:

- a) Qual é o estilo visual do jogo? (Realista? Estilizado? Low poly? Pixel art? Cel shading?)
- b) As animações são compatíveis com esse estilo? (Um jogo realista deve ter animações fluidas e naturais, ao passo que um jogo estilizado pode se beneficiar de exageros e deformações.)
- c) Os efeitos visuais reforçam ou enfraquecem a imersão? (Neblina, iluminação e pós-processamento devem contribuir para a atmosfera.)
- d) A engine escolhida permite atingir esse resultado de forma eficiente? (Se um motor gráfico exige um esforço excessivo para atingir o estilo desejado, talvez outra ferramenta seja mais adequada.)

A direção artística não deve ser um processo isolado, mas um planejamento estratégico que envolve todas as áreas do desenvolvimento. Desde a modelagem dos personagens e a animação até a escolha da engine e os efeitos visuais, tudo deve estar alinhado para criar uma experiência coesa e imersiva.

## DECLARAÇÃO DE USO DE INTELIGÊNCIA ARTIFICIAL GENERATIVA

Este material foi integralmente elaborado pelo autor, contando, no entanto, com o auxílio de uma ferramenta de inteligência artificial generativa – especificamente o ChatGPT, versão 4.0 – para sugestões pontuais de organização do conteúdo. Todas as decisões finais sobre a redação, estrutura e argumentação foram tomadas pelo autor, garantindo a originalidade e a autoria do texto.

---

## REFERÊNCIAS

BLENDER FOUNDATION. **Animation & Rigging**. Blender 4.4 Manual. Disponível em: <<https://docs.blender.org/manual/en/latest/animation/index.html>>. Acesso em: 21 fev. 2025.

EPIC GAMES. Unreal Engine 5.0 Documentation. Disponível em: <<https://docs.unrealengine.com/5.0/en-US>>. Acesso em: 21 fev. 2025.

GODOT ENGINE. **Introduction to 2D**. Disponível em: <[https://docs.godotengine.org/pt-br/4.x/tutorials/2d/introduction\\_to\\_2d.html](https://docs.godotengine.org/pt-br/4.x/tutorials/2d/introduction_to_2d.html)>. Acesso em: 1º abr. 2025.

MCKEE, R. **Story**: Substance, Structure, Style and the Principles of Screenwriting. New York: ReganBooks, 1997.

UNITY TECHNOLOGIES. **Introduction to 2D Animation**. Disponível em: <<https://docs.unity3d.com/Packages/com.unity.2d.animation@10.1/manual/index.html>>. Acesso em: 1º abr. 2025.

WILLIAMS, R. **The Animator's Survival Kit**: A Manual of Methods, Principles, and Formulas for classical, computer, games, stop motion and internet animators. London: Faber & Faber, 2001.