

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Henrique Tornelli Duarte

**Geração e Aperfeiçoamento de Logotipos via
Ajuste Fino no Stable Diffusion XL**

Uberlândia, Brasil

2025

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Henrique Tornelli Duarte

**Geração e Aperfeiçoamento de Logotipos via Ajuste Fino
no Stable Diffusion XL**

Trabalho de conclusão de curso apresentado
à Faculdade de Computação da Universidade
Federal de Uberlândia, Minas Gerais, como
requisito exigido parcial à obtenção do grau
de Bacharel em Sistemas de Informação.

Orientador: Professor Dino Rogerio Coinete Franklin

Universidade Federal de Uberlândia – UFU

Faculdade de Ciência da Computação

Bacharelado em Sistemas de Informação

Uberlândia, Brasil

2025

Henrique Tornelli Duarte

Geração e Aperfeiçoamento de Logotipos via Ajuste Fino no Stable Diffusion XL

Trabalho de conclusão de curso apresentado
à Faculdade de Computação da Universidade
Federal de Uberlândia, Minas Gerais, como
requisito exigido parcial à obtenção do grau
de Bacharel em Sistemas de Informação.

**Professor Dino Rogerio Coinete
Franklin
Orientador**

**Professor Alessandro Santos Soares
Convidado 1**

**Professor Ivan da Silva Sendin
Convidado 2**

Uberlândia, Brasil
2025

*Dedico à minha esposa **Karoliny** e à minha filha **Valentina**, minha força diária e razão maior de seguir em frente. Sem o amor, o apoio e a paciência de vocês, eu teria desistido no caminho.*

Agradecimentos

Agradeço, em primeiro lugar, a minha familia **Karoliny e Valentina** e aos meus pais **Elizângela e Marcelo**, que me deram a educação e a base necessárias para enfrentar os desafios da vida.

Sou grato aos amigos que tornaram a jornada universitária mais leve e divertida — **Gabriel, Guilherme, Vitor e João Vitor**. Obrigado por transformarem os momentos mais difíceis nos mais memoráveis.

Ao meu amigo internacionalmente especial **Joni Innanen**, por manter viva a inspiração de buscar novos horizontes e acreditar que a formação acadêmica pode abrir portas no exterior.

Registro especial apreço ao meu orientador, **Prof. Dino Franklin**, pela confiança, pela orientação constante e por não desistir mesmo durante meus períodos de ausência. Extendo meus agradecimentos aos professores **Alexsandro, Autran, Rivalino e Willian**, cujos ensinamentos marcaram profundamente minha trajetória e continuam a influenciar meu desenvolvimento.

Resumo

Esta dissertação avalia técnicas de *ajuste fino* (fine-tuning) aplicadas ao **Stable Diffusion XL** (SDXL), versão de alta resolução do modelo de difusão latente, com foco na adaptação leve oferecida pelo **LoRA**—*Low-Rank Adaptation*, inserção de matrizes de baixa dimensão que exige bem menos parâmetros. Prepararam-se seis conjuntos de dados específicos (*datasets*): estrutura tipográfica (*wordmark*), símbolo gráfico (*iconic*) e estilização (*minimalistic*, *vintage* e *cartoon*). Cada conjunto foi treinado durante 10 épocas (*epoch*: ciclo completo em que o modelo percorre todo o conjunto de treinamento), empregando taxas de aprendizado distintas para o modulador de texto (encoder CLIP-Text) e para a U-Net responsável pela imagem.

A fase experimental gerou 8 640 amostras, combinando sistematicamente a escala de orientação *CFG* (Classifier-Free Guidance), o número de etapas de *denoising* (*Steps*) e algoritmo de amostragem (*sampler*). As métricas adotadas incluíram similaridade CLIP (aderência semântica), acurácia OCR (legibilidade do texto) e avaliação humana. Os resultados mostram que o ajuste fino via LoRA guia a difusão para um espaço de soluções mais restrito e coerente, reduzindo ruído visual e variações indesejadas. A legibilidade textual saltou de 37% no modelo base para 88% após o ajuste, superando 92% quando se aplicou o pós-processamento de correção de texto (*fix-text*). A análise qualitativa confirma que a abordagem conserva o estilo desejado, melhora a precisão do nome da marca e permite diminuir o número total de etapas de amostragem. Conclui-se que o LoRA, aliado a curadoria criteriosa de dados e ajuste fino de hiperparâmetros, é a alternativa mais eficaz para especializar o SDXL na geração automática de logomarcas.

Palavras-chave: geração de imagens, Stable Diffusion XL, LoRA, logomarcas, ajuste fino.

Listas de ilustrações

Figura 1 – Comparação de versões do Stable Diffusion	13
Figura 2 – Exemplo realista de saída do SDXL	14
Figura 3 – Demonstração do conceito de <i>inpainting/outpainting</i> e <i>Control Net</i>	15
Figura 4 – Demonstração de saídas com instruções textuais relacionadas a logomarcas.	16
Figura 5 – Demonstração de saídas com instruções textuais relacionadas a texto.	16
Figura 6 – Instrução textual 1 da ilustração do problema: text AURORA, ultra-thin spacing, luxurious minimal look	17
Figura 7 – Instrução textual 2 da ilustração do problema: logo, text AURORA, ultra-thin spacing, luxurious minimal look	17
Figura 8 – Instrução textual 3 da ilustração do problema: logo, wordmark, text “AURORA”, ultra-thin spacing, luxurious minimal look	17
Figura 9 – Diagrama do fluxo lógico dos modelos de difusão latente	21
Figura 10 – Ilustração do processo de difusão reversa.	22
Figura 11 – Comparação e pipeline em duas etapas do SDXL	22
Figura 12 – Imagens geradas usando SDXL.	23
Figura 13 – Ilustração demonstrando o processo de ajuste fino. Neste treinamento o foco se baseia em transformar as saídas em imagens com estilo <i>Pokemon</i>	24
Figura 14 – Demonstração de saídas utilizando diferentes escalas de um LoRA que visa aplicar um estilo chamado de <i>pixel-art</i> . A imagem à esquerda possui escala 0 (zero), a imagem central possui escala 0.5 e a imagem à direita possui escala 1.0.	25
Figura 15 – Ilustração geral de um fluxo construído utilizando ComfyUI.	27
Figura 16 – Pequenas amostras dos datasets criados.	31
Figura 17 – Fluxo Text-to-Image (ComfyUI)	33
Figura 18 – Fluxo Image-to-Image (ComfyUI)	34
Figura 19 – Fluxo Fix-Text (ComfyUI)	35
Figura 20 – Primeira comparação das saídas sem e com os LoRAs criados, a partir do prompt “AURORA”.	36
Figura 21 – Segunda comparação das saídas sem e com os LoRAs criados, a partir do prompt “AURORA”.	36
Figura 22 – 1º Comparação das saídas sem e com o LoRA criado.	37
Figura 23 – 2º Comparação das saídas sem e com o LoRA criado.	37
Figura 24 – 3º Comparação das saídas sem e com o LoRA criado.	37
Figura 25 – 4º Comparação das saídas sem e com o LoRA criado.	38
Figura 26 – 5º Comparação das saídas sem e com os LoRAs criados.	38

Figura 27 – 6º Comparação das saídas sem e com os LoRAs criados.	38
Figura 28 – 7º Comparação das saídas sem e com os LoRAs criados.	39
Figura 29 – 8º Comparação das saídas sem e com os LoRAs criados.	39
Figura 30 – 9º Comparação das saídas sem e com os LoRAs criados.	40
Figura 31 – 10º Comparação das saídas sem e com os LoRAs criados.	40
Figura 32 – 1º Comparação das saídas ruíns sem e com os LoRAs criados.	41
Figura 33 – 2º Comparação das saídas ruíns sem e com os LoRAs criados.	41
Figura 34 – 3º Comparação das saídas ruíns sem e com os LoRAs criados.	42
Figura 35 – 4º Comparação das saídas ruíns sem e com os LoRAs criados.	43
Figura 36 – Imagem referência para <i>fix-text</i>	44
Figura 37 – Comparação das saídas sem e com o LoRA criado.	44
Figura 38 – Pipeline de geração e cálculo das métricas.	46

Lista de tabelas

Tabela 1 – Técnicas de ajuste fino.	26
Tabela 2 – Parâmetros empregados nos treinamentos dos LoRAs	32
Tabela 3 – Resultados métricos para <i>Wordmark + Minimalistic</i>	46
Tabela 4 – Resultados métricos para <i>Wordmark + Vintage</i>	47
Tabela 5 – Resultados métricos para <i>Wordmark + Cartoon</i>	47
Tabela 6 – Resultados métricos para <i>Iconic + Minimalistic</i>	47
Tabela 7 – Resultados métricos para <i>Iconic + Vintage</i>	48
Tabela 8 – Resultados métricos para <i>Iconic + Cartoon</i>	48

Lista de abreviaturas e siglas

SD	Stable Diffusion
SDXL	Stable Diffusion XL
LoRA	Low-Rank Adaptation
TI	Textual Inversion
DB	DreamBooth
CFG	Classifier-Free Guidance (escala de orientação)
LR	Learning Rate (taxa de aprendizado)
OCR	Optical Character Recognition
GPU	Graphics Processing Unit
VRAM	Video Random-Access Memory
FP16	Half-precision floating point (16 bits)
SDE	Stochastic Differential Equation (formulação de difusão)
ODE	Ordinary Differential Equation (formulação determinística)
DPM	Denoising Probabilistic Model (família de algoritmos de amostragem)
CLIP	Contrastive Language-Image Pre-training
PEFT	Parameter-Efficient Fine-Tuning

Sumário

1	INTRODUÇÃO	12
1.1	Problema	15
1.2	Hipótese	18
1.3	Objetivo	18
1.4	Resultados Esperados	19
2	REVISÃO BIBLIOGRÁFICA	20
2.1	Stable Diffusion: Fundamentos e Arquitetura	20
2.1.1	Difusão Latente	20
2.1.2	Arquitetura do Modelo	20
2.1.3	Processo de Geração	21
2.1.4	Stable Diffusion XL (SDXL): Avanços na Geração de Imagens de Alta Resolução	22
2.1.5	Ajuste Fino	24
2.1.6	Condicionamento Estrutural e Pós-processamento de Texto	26
2.2	Ferramentas de Apoio	27
2.3	Estado da Arte	28
2.3.1	Geração automática de logomarcas	28
2.3.2	Soluções comerciais	28
2.3.3	Lacunas identificadas	28
3	DESENVOLVIMENTO	30
3.1	Estudo Inicial e Fundamentação Técnica	30
3.2	Seleção da Abordagem: Ajuste Fino com LoRA	30
3.3	Construção dos Datasets para Treinamento	31
3.4	Execução e Ajuste dos Treinamentos com Kohya_ss	32
3.5	Implementação dos Fluxos em ComfyUI	32
3.5.1	Fluxo Text-to-Image (text2img)	32
3.5.2	Fluxo Image-to-Image (img2img)	34
3.5.3	Fluxo Fix-Text (Correção Textual)	34
3.6	Planejamento e Execução da Avaliação Experimental	35
3.7	Análise de Resultados e Métricas	35
3.7.1	Estudo de Caso Inicial	36
3.7.2	Resultados Gerais	37
3.7.3	Análise Visual	40
3.7.4	Resultados Ruins	41

3.7.5	Correção de Texto	43
3.7.6	Métricas	44
3.8	Discussão Geral e Impressões Pessoais dos Resultados	48
3.9	Repositório de Apoio e Reprodutibilidade	49
4	CONCLUSÃO	50
	REFERÊNCIAS	51

1 Introdução

A síntese de imagens por Inteligência Artificial (IA) deu um salto decisivo em 2022 com a liberação pública do *Stable Diffusion* (SD) (ROMBACH et al., 2022)¹. O SD é um gerador de imagens baseado em **modelos de difusão latente**: ele parte de uma “imagem” composta apenas de ruído e, etapa a etapa, aprende a remover esse ruído até formar uma figura coerente que corresponda a uma *instrução textual* escrita pelo usuário — por exemplo, “*uma cidade futurista ao pôr do sol*”. O diferencial está no local onde essa difusão ocorre: não no espaço de pixels, mas em um **espaço latente** comprimido, codificado por um *Autoencoder Variacional* (VAE). Trabalhar nesse espaço reduz drasticamente o uso de memória e o tempo de cálculo, sem sacrificar detalhes finos na imagem final.

O pipeline é composto por três blocos principais. O VAE comprime imagens reais e reconstrói amostras geradas, atuando como ponte entre pixels e vetores latentes. A *U-Net* — rede convolucional em formato de “U”, com camadas de contração e expansão — executa o processo de *denoising*, estimando em cada passo quanto ruído ainda deve ser removido. Por fim, o texto colocado pelo usuário é convertido em vetores semânticos pelo CLIP (*Contrastive Language-Image Pre-training*), que alinha a descrição com a imagem em construção, garantindo aderência semântica entre palavras e pixels.

Essa combinação de arquitetura eficiente e licença aberta tornou o Stable Diffusion relevante em múltiplos contextos: ele pode rodar em GPUs de consumo doméstico, é facilmente customizável — inclusive via *treinamento adicional* leve — e já encontra aplicações em arte digital, design de jogos, publicidade e prototipagem de produtos.

Evolução das versões do Stable Diffusion

- **SD 1.4** (ago 2022) — primeira versão pública, treinada em 512 px com CLIP ViT-L/14; popularizou o texto-para-imagem de código aberto.
- **SD 1.5** (out 2022) — refinamento da 1.4, com dataset filtrado e remoção de artefatos, mantendo arquitetura U-Net original.
- **SD 2.0** (nov 2022) — troca do encoder textual para OpenCLIP, treinamento de 1 → 768 px, introdução de *depth-guidance* e modelo upscaler 4×.
- **SD 2.1** (dez 2022) — ajuste fino da 2.0 com dataset ampliado, melhor equilíbrio cromático e menor suscetibilidade a ruídos.

¹ Disponibilizado pela Stability AI em colaboração com LAION e Eleuther AI, agosto de 2022.

- **SDXL (1.0)** (jul 2023) — arquitetura expandida: U-Net 3× maior, duplo encoder textual, resolução nativa 1024 px, e estágio opcional de refinamento *image-to-image* para realce de detalhes. ([PODELL et al., 2023](#))



Figura 1 – Comparação da saída do SDXL com as versões anteriores do Stable Diffusion.

Cada iteração aprimorou compreensão semântica e qualidade visual, mas manteve a filosofia de código aberto e o pipeline de difusão latente.

Esses avanços tornaram o SD e suas derivações a base de inúmeras aplicações — ilustração digital, arte conceitual, publicidade e prototipagem de produtos — democratizando a geração de conteúdo visual. Contudo, tarefas que exigem controle estilístico preciso, como a criação de logomarcas, continuam desafiadoras. Logomarcas demandam legibilidade tipográfica, composição equilibrada e consistência de estilo; o SDXL base, com instruções textuais sucintas, ainda tende a produzir composições desconexas ou texto ilegível.

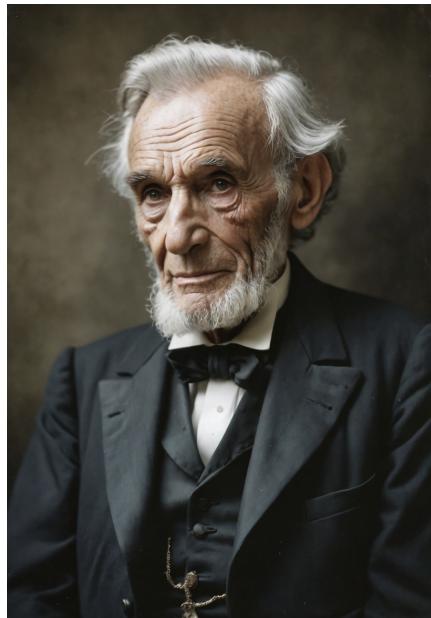


Figura 2 – Exemplo de saída do SDXL: Fotografia realista de um homem com vestimenta formal. Instrução textual completa pode ser encontrada [neste endereço](#).

Ajuste Fino

Com estes avanços, a geração de imagens tornou-se um processo acessível: basta fornecer uma descrição em texto e, em segundos, o algoritmo transforma ruído em figuras detalhadas e realistas.

Todavia, esses modelos nascem genéricos; foram treinados em enormes coleções de imagens da internet. Quando surge a necessidade de especializar a saída — seja para um estilo artístico particular, uma estética de marca ou qualquer domínio visual específico — recorre-se a um ajuste fino (*fine-tuning*). No geral, parte-se do modelo já pré-treinado e realiza-se um breve ciclo de ajuste sobre um conjunto de exemplos cuidadosamente escolhidos. O procedimento preserva o conhecimento amplo adquirido anteriormente, mas direciona a rede a reproduzir padrões, cores e composições mais alinhados ao novo objetivo.

Dessa forma, consegue-se combinar a versatilidade de um modelo geral com a precisão de um modelo feito sob medida, sem a necessidade de treinar tudo novamente do zero.

Inpainting e Control Net

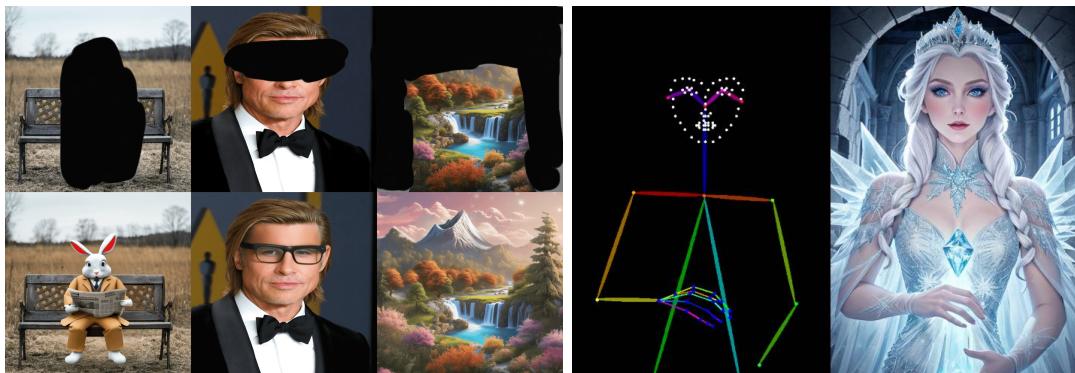
Outra habilidade importante dos modelos de difusão é o *inpainting*, técnica que permite editar seletivamente uma região da imagem. O usuário define uma máscara — por exemplo, o espaço vazio em um rótulo — e descreve em texto o que deve aparecer ali. O modelo mantém intacto o restante do quadro e preenche apenas a área mascarada,

garantindo continuidade de cor, iluminação e textura. Esse processo é valioso quando se deseja corrigir detalhes ou inserir elementos adicionais sem recriar toda a composição.

Complementar ao *inpainting* está o *outpainting*, que faz o movimento inverso: expande os limites de uma imagem além de suas bordas originais. A partir de pequenas porções de contexto visual, o gerador estende a cena, completando fundos, prolongando padrões ou adicionando novos objetos de forma coerente. Na prática, essa técnica é usada para criar versões panorâmicas de ilustrações ou material publicitário em formatos variados, preservando o estilo e a continuidade estética.

Para aproximar ainda mais a geração do controle humano, surgiu o *Control Net*. Trata-se de uma rede adicional acoplada ao modelo de difusão que recebe informações estruturais — mapas de bordas, profundidade, poses, segmentações — e as utiliza como “grade” durante a síntese. Enquanto a instrução textual em texto dita o conteúdo semântico, o *Control Net* fixa a geometria, garantindo que linhas mestras, contornos ou silhuetas específicas sejam respeitadas. Isso destrava aplicações que exigem alinhamento preciso, como adaptação de layouts ou reinterpretação estilística de desenhos técnicos.

Quando combinadas, essas ferramentas oferecem um leque robusto de edição: o texto direciona o tema, o *Control Net* define a estrutura e o *inpainting/outpainting* ajusta ou estende áreas localizadas. O resultado é um fluxo de trabalho flexível, no qual o usuário pode tanto criar imagens do zero quanto aprimorar e adaptar ilustrações existentes, mantendo controle sobre forma, escala e consistência visual.



(a) *Inpainting* aplicado a três diferentes imagens usando máscara. (b) *ControlNet* baseado em uma pose gerando uma imagem.

Figura 3 – Demonstração do conceito de *inpainting/outpainting* e *Control Net*.

1.1 Problema

Apesar dos avanços trazidos pelo SDXL, a geração automática de logomarcas ainda enfrenta limitações que comprometem sua adoção prática. Quando se utilizam instruções textuais curtas — por exemplo, apenas o nome da marca seguido de um adjetivo de estilo — o modelo raramente entrega uma identidade visual consistente: surgem composições

incoerentes, símbolos desconexos ou mesmo imagens abstratas sem qualquer hierarquia gráfica típica de um logotipo.

Para tentar contornar essas falhas, muitos usuários recorrem à engenharia de *prompt*: descrições longas, repletas de qualificadores (família tipográfica, paleta de cores, alinhamento, espessura de traço, proporção do ícone), que buscam direcionar o modelo em cada detalhe. Embora possa haver melhoria pontual, esse processo é complexo, instável, demanda experiência e contraria a premissa de agilidade que tornou os modelos de difusão tão atraentes.

A tipografia é outro ponto crítico. Mesmo instruções textuais detalhadas não impedem o SDXL base de gerar letras distorcidas, glifos inexistentes ou duplicações aleatórias. Em muitos casos, o texto some por completo ou se mistura ao fundo, tornando a logomarca inutilizável.

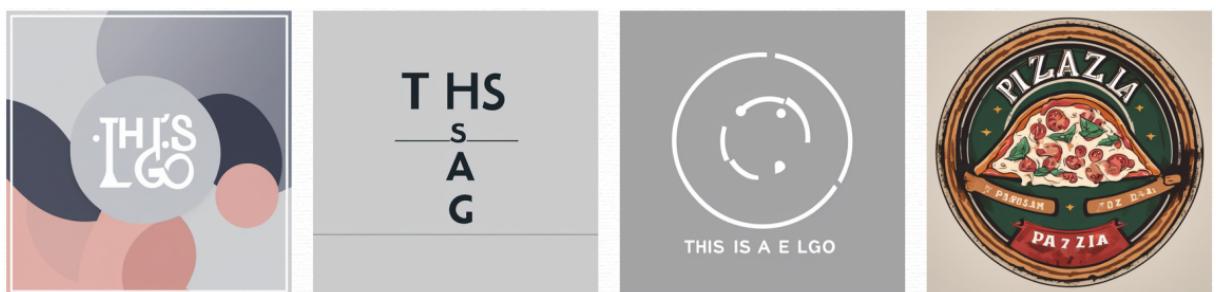


Figura 4 – Demonstração de saídas com instruções textuais relacionadas a logomarcas.



Figura 5 – Demonstração de saídas com instruções textuais relacionadas a texto.

Ilustração Prática do Problema

Para ilustrar de forma objetiva as dificuldades enfrentadas pelo SDXL sem qualquer adaptação, foram geradas amostras para três variações de instrução textual, todos visando produzir uma logomarca minimalista para a marca fictícia “AURORA”:

1. text AURORA, ultra-thin spacing, luxurious minimal look
2. logo, text AURORA, ultra-thin spacing, luxurious minimal look
3. logo, wordmark, text "AURORA", ultra-thin spacing, luxurious minimal look

As figuras a seguir apresentam, em ordem, quatro amostras produzidas para cada instrução textual (semente fixa e conjuntos de 4 imagens).

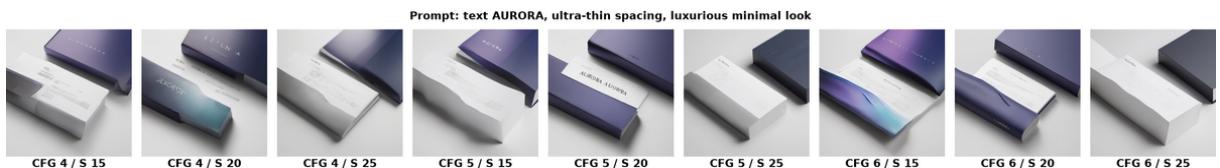


Figura 6 – Instrução textual 1 da ilustração do problema: text AURORA, ultra-thin spacing, luxurious minimal look



Figura 7 – Instrução textual 2 da ilustração do problema: logo, text AURORA, ultra-thin spacing, luxurious minimal look



Figura 8 – Instrução textual 3 da ilustração do problema: logo, wordmark, text “AURORA”, ultra-thin spacing, luxurious minimal look

Mesmo quando o termo *logo* ou *wordmark* é explicitado, o modelo raramente entrega um resultado que possa ser reconhecido como logomarca. Observam-se principalmente:

- Objetos ou emblemas abstratos que não contêm texto algum;
- Agrupamentos de letras ilegíveis ou com glifos inexistentes;
- Falta de hierarquia visual, com símbolos dispersos e ausência de área de respiro;
- Variação imprevisível de cores e texturas, contrariando a estética “luxurious minimal”;

- Embora a terceira instrução textual apresente um layout que remete a uma logomarca, o texto resultante ainda é inválido ou completamente ausente.

A comparação direta entre as três tentativas deixa claro que simplesmente adicionar mais palavras-chave a instrução textual não resolve o problema: o SDXL continua sem diretriz sólida sobre como estruturar texto e símbolo como elementos de identidade visual.

1.2 Hipótese

A hipótese central deste trabalho é que a aplicação estruturada e específica de técnicas de ajuste fino pode melhorar significativamente a qualidade geral das logomarcas geradas por modelos como o SDXL, independentemente da complexidade das instruções textuais fornecidas.

Mais especificamente, acredita-se que o uso dessa técnica de ajuste fino poderá:

- Melhorar a consistência visual geral das imagens geradas, tornando-as mais próximas da estrutura e composição típicas de logomarcas;
- Reduzir significativamente a necessidade de instruções textuais detalhadas e complexas por parte do usuário;
- Facilitar o processo criativo, oferecendo resultados úteis e visualmente coerentes com menos tentativas;
- Contribuir para melhorias gerais na legibilidade e coerência dos textos, embora reconhecendo que esta é uma consequência secundária e não uma garantia absoluta;
- Oferecer uma abordagem documentada e replicável ([3.9](#)), contribuindo diretamente para a comunidade científica e criativa.

1.3 Objetivo

O objetivo geral deste trabalho é propor, implementar e validar uma abordagem prática e acessível para geração automática de logomarcas, melhorando a qualidade geral das imagens geradas por modelos generativos como o SDXL, independentemente do nível de detalhe da instrução textual fornecido.

Para atingir esse objetivo geral, foram definidos objetivos específicos claros:

- Criar conjuntos de dados estruturados e diversificados adequados ao treinamento especializado dos modelos;

- Realizar ajustes finos com estratégias específicas para direcionar melhor o foco visual e a coerência das logomarcas geradas;
- Construir fluxos simples e intuitivos que facilitem a geração consistente de logomarcas sem exigir conhecimentos avançados de criação de instruções textuais;
- Documentar detalhadamente todos os procedimentos, hiperparâmetros e scripts utilizados (3.9), facilitando a replicação do processo por outros pesquisadores e profissionais criativos;
- Avaliar objetivamente a melhoria obtida nos resultados gerados por meio de métricas quantitativas e qualitativas.

1.4 Resultados Esperados

Ao concluir este estudo, espera-se demonstrar os seguintes resultados e contribuições:

- Uma melhora geral significativa na qualidade visual e na consistência das logomarcas geradas, independentemente do nível de detalhamento das instruções textuais;
- Uma redução notável da complexidade exigida nas instruções textuais, tornando a criação automática mais acessível;
- Uma documentação completa e replicável (3.9), permitindo a reprodução do processo por outros pesquisadores e profissionais.

Em resumo, este trabalho busca auxiliar a comunidade científica e profissional a explorar melhor o potencial dos modelos generativos modernos, propondo uma abordagem prática, acessível e documentada para tornar a criação automática de logomarcas uma tarefa mais eficaz e produtiva.

2 Revisão Bibliográfica

2.1 Stable Diffusion: Fundamentos e Arquitetura

O Stable Diffusion é um modelo gerativo de imagens baseado em difusão latente, introduzido por Rombach em 2022 ([ROMBACH et al., 2022](#)). Ele representa um avanço significativo na geração de imagens de alta qualidade, combinando eficiência computacional com flexibilidade na geração condicionada por texto.

2.1.1 Difusão Latente

Diferentemente dos modelos de difusão tradicionais que operam diretamente no espaço de pixels, o Stable Diffusion realiza o processo de difusão em um espaço latente comprimido. Isso é possível graças ao uso de um *autoencoder* variacional (VAE), que codifica imagens de alta dimensão em representações latentes de menor dimensão. Essa abordagem reduz significativamente o custo computacional e permite a geração de imagens de alta resolução com recursos computacionais mais acessíveis.

2.1.2 Arquitetura do Modelo

1. **Codificador de Texto (CLIP)**: Utiliza o modelo CLIP para transformar instruções textuais em *embeddings* semânticos que condicionam a geração de imagens.
2. **Rede U-Net Condicionada**: Uma rede neural convolucional que realiza o processo de *denoising* no espaço latente, guiada pelos *embeddings* textuais.
3. **Decodificador VAE**: Reconstrói a imagem final a partir da representação latente *denoised*, retornando ao espaço de pixels.

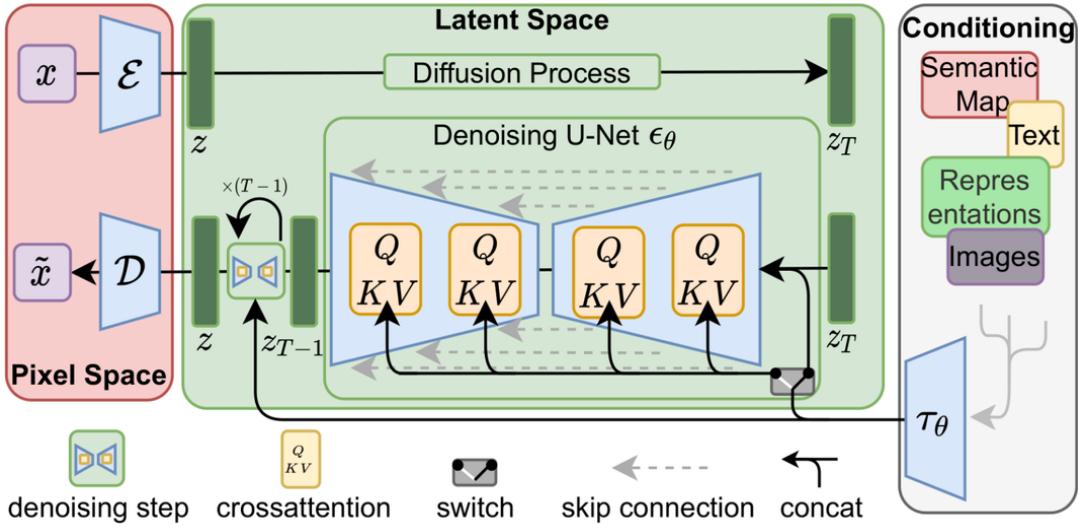


Figura 9 – Representação das etapas de geração no Stable Diffusion, desde a semente e ruído inicial até a geração da imagem via decodificador VAE.

2.1.3 Processo de Geração

O processo começa pela transformação da instrução textual em vetores numéricos, chamados de *embeddings* semânticos, por meio do modelo CLIP (do inglês *Contrastive Language-Image Pre-training*, “pré-treinamento contrastivo de texto e imagem”). O CLIP, treinado em milhões de pares imagem-texto, projeta frases e figuras no mesmo espaço vetorial, de modo que conteúdos semelhantes fiquem próximos. Paralelamente, um Autoencoder Variacional (VAE, do inglês *Variational Autoencoder*) codifica o que será a “imagem-alvo” — inicialmente puro ruído no modo *text-to-image* — em uma representação comprimida no espaço latente.

Três parâmetros controlam a dinâmica dessa geração: a **seed** (semente aleatória que determina o ruído inicial e garante reproduzibilidade), o número de **steps** (iterações de remoção de ruído) e a escala **CFG** (*Classifier-Free Guidance*), que define quão fortemente o modelo deve seguir a descrição textual em vez de sua distribuição interna de imagens.

A etapa central é a difusão reversa (*denoising*), conduzida por uma rede neural em formato de “U” — a *U-Net*. A geração inicia com o ruído latente; a cada iteração, a U-Net recebe o latente atual e os embeddings do texto e estima o ruído residual que precisa ser removido. O *scheduler* (agenda de ruído) define o cronograma exato de remoção, especificando a fração de perturbação a retirar em cada passo. Em seguida, o *sampler* (algoritmo de amostragem, como DDIM ou Euler) aplica as fórmulas que atualizam o latente de forma estável, obedecendo às instruções do scheduler. Esse ciclo se repete pelos *steps* configurados, até que o ruído se aproxime de zero e a estrutura latente reflita um esboço coerente da imagem desejada.

Por fim, a representação latente refinada é decodificada de volta ao espaço de pixels

pelo VAE, gerando a imagem final. O decodificador atua como um filtro de alta resolução, convertendo a compressão latente em detalhes visuais ricos, respeitando as cores, texturas e formas indicadas pela instrução textual.

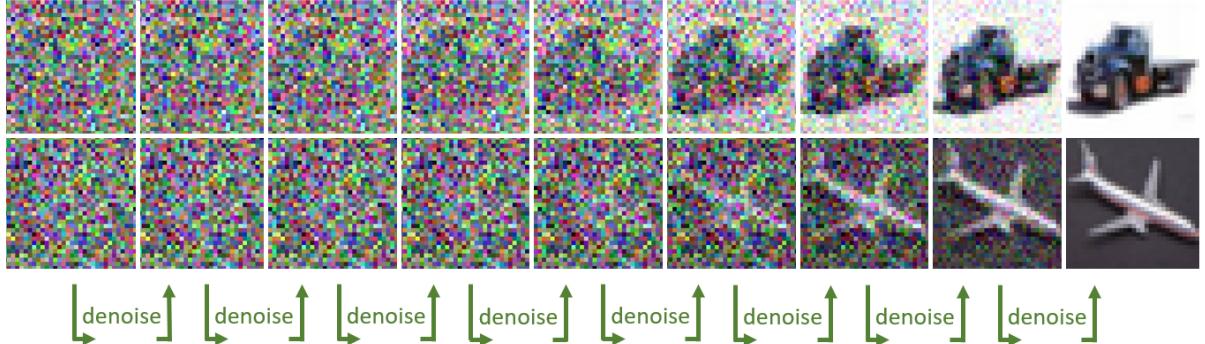


Figura 10 – Ilustração do processo de difusão reversa.

2.1.4 Stable Diffusion XL (SDXL): Avanços na Geração de Imagens de Alta Resolução

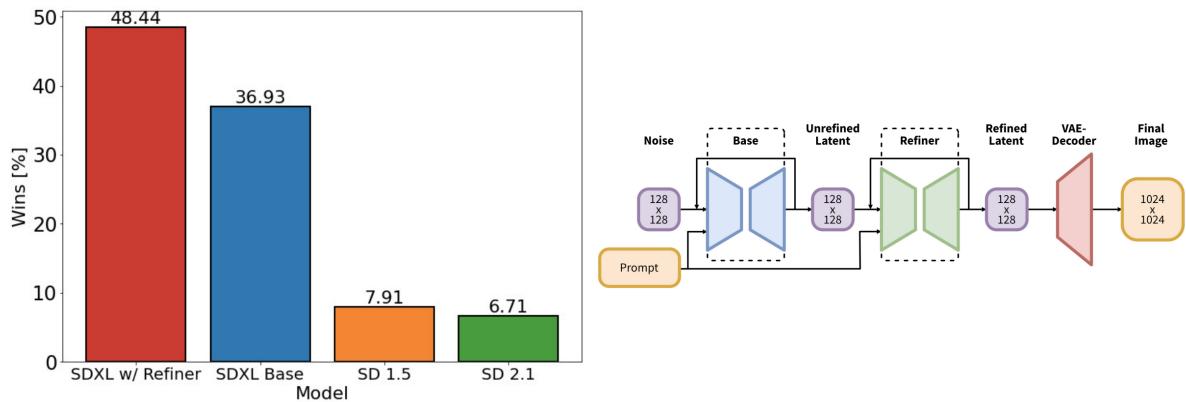


Figura 11 – À esquerda, comparação da preferência dos usuários entre SDXL e versões anteriores do Stable Diffusion. À direita, diagrama do pipeline em duas etapas, com geração inicial e refinamento em alta resolução usando a mesma instrução textual e autoencoder.

O Stable Diffusion XL (SDXL) representa uma evolução significativa em relação ao modelo original de difusão latente. Introduzido por Podell em 2023 ([PODELL et al., 2023](#)), o SDXL aprimora a qualidade e a fidelidade das imagens geradas, especialmente em resoluções mais altas, mantendo a eficiência computacional.

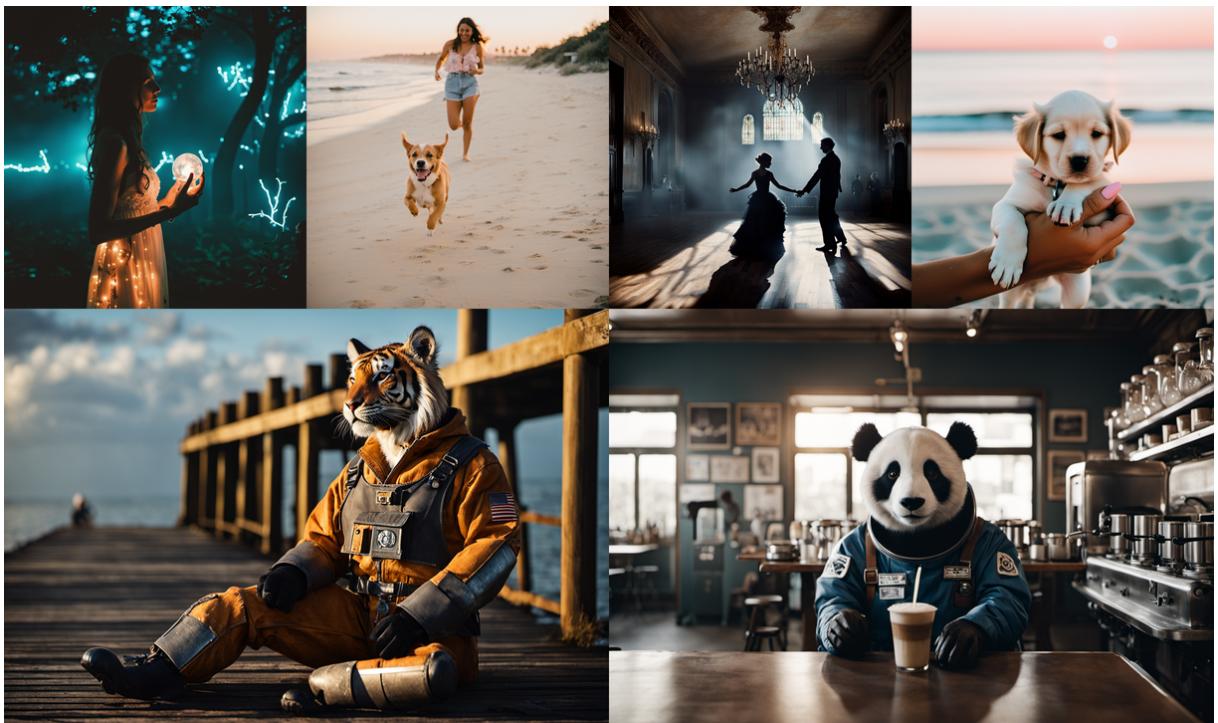


Figura 12 – Imagens geradas usando SDXL.

Principais Melhorias Arquiteturais

- **U-Net ampliada:** A rede U-Net do SDXL foi aumentada em profundidade e largura — mais camadas convolucionais, mais blocos residuais e seções de auto-atenção — chegando a cerca de três vezes o número de parâmetros do modelo original. Esse reforço de capacidade permite capturar texturas mais finas e relações espaciais de longo alcance, produzindo imagens com detalhes nítidos e composição mais coerente.
- **Duplo Codificador de Texto:** O SDXL incorpora um segundo codificador de texto, ampliando o contexto de atenção cruzada. Isso resulta em uma compreensão semântica mais profunda das instruções textuais, melhorando a coerência entre texto e imagem.
- **Treinamento em Múltiplas Proporções:** Diferentemente do SD, que era treinado principalmente em imagens quadradas, o SDXL foi treinado em múltiplas proporções de aspecto, aumentando sua versatilidade na geração de imagens em diferentes formatos.
- **Modelo de Refinamento:** O SDXL introduz um modelo de refinamento que aplica uma técnica de imagem para imagem pós-processamento, melhorando ainda mais a fidelidade visual das amostras geradas.

2.1.5 Ajuste Fino

Quando se deseja especializar um modelo de difusão já pré-treinado, realiza-se um ajuste fino (*fine-tuning*): um curto ciclo de ajuste que utiliza exemplos específicos do domínio de interesse. Na forma clássica, esse ajuste implica desbloquear milhões de pesos, definir taxas de aprendizado precisas, monitorar métricas de validação e cuidar para não cair em sobreajuste (*overfitting*)¹. Além do tempo de GPU, há o custo de armazenar gradientes e estados do otimizador, que cresce rapidamente em redes volumosas como as variantes do Stable Diffusion XL.



Figura 13 – Ilustração demonstrando o processo de ajuste fino. Neste treinamento o foco se baseia em transformar as saídas em imagens com estilo *Pokemon*.

Para reduzir esse impacto surgiram estratégias de adaptação eficiente, conhecidas como *parameter-efficient fine-tuning*. A mais simples é a **Textual Inversion** (GAL et al., 2022): treina-se um único embedding — um “token inventado” — que passa a representar o conceito desejado. O método é barato (poucos minutos de GPU) e não toca nos pesos do modelo, mas costuma falhar em preservar estrutura quando o conceito é visualmente complexo.

Já o **DreamBooth** introduzido por Ruiz (RUIZ et al., 2023), utiliza um *token* raro para “ancorar” o conceito e faz ajuste direcionado em todas as camadas relacionadas. O resultado é fiel ao conjunto de imagens de referência, porém o processo requer horas de processamento e muita memória, além de aumentar o risco de sobreajuste.

Entre as abordagens mais equilibradas está o **LoRA — Low-Rank Adaptation** (HU et al., 2021). A técnica insere pequenas matrizes de baixa dimensão em pontos específicos da rede; somente esses novos parâmetros são atualizados, enquanto o *backbone*

¹ O modelo memoriza demais o conjunto de treino e perde capacidade de generalizar.

original permanece congelado. O LoRA combina a vantagem de arquivos leves — poucos megabytes — com a possibilidade de empilhar múltiplas adaptações diferentes, cada qual correspondendo a um estilo ou domínio. Dessa forma, consegue-se uma especialização de alta qualidade com fração do custo computacional de um fine-tuning total, preservando a flexibilidade de voltar facilmente ao comportamento original do modelo ou combinar várias adaptações conforme a necessidade.

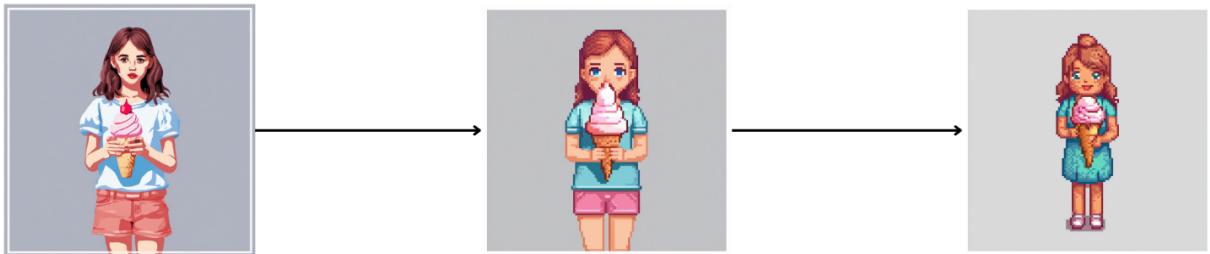


Figura 14 – Demonstração de saídas utilizando diferentes escalas de um LoRA que visa aplicar um estilo chamado de *pixel-art*. A imagem à esquerda possui escala 0 (zero), a imagem central possui escala 0.5 e a imagem à direita possui escala 1.0.

A seguir alguns dados levantados pela comunidade em relação ao custo computacional para realizar os diferentes tipos de ajuste fino:

- **Textual Inversion** Necessita ≥ 6 GB de VRAM (8 GB é confortável) para resolução 512^2 px. Um ciclo comum usa $\sim 8\,000$ passos [HuggingFace \(2024d\)](#), o que leva cerca de 30-60 min numa RTX 3060 (fp16, batch 1). O embedding gerado ocupa da ordem de 100-200 kB, portanto quase não impacta armazenamento.
- **DreamBooth** Treino completo do *UNet* cabe em 16 GB de VRAM com *gradient checkpointing + bits-and-bytes* [HuggingFace \(2024b\)](#); caso o *text encoder* também seja ajustado, exige pelo menos 24 GB [HuggingFace \(2024a\)](#). Experimentos usuais (400-800 passos) levam algo entre 15 min e 2 h dependendo da GPU. O checkpoint resultante adiciona 2-4 GB ao modelo base.
- **LoRA** O script oficial opera em 11 GB de VRAM sem truques [HuggingFace \(2024c\)](#) (há relatos de 8-10 GB com *xformers*). Um exemplo de 15 k passos levou ≈ 5 h numa 2080 Ti (11 GB) [Face \(2024\)](#). O adaptador final pesa apenas 3-20 MB, mantendo o SD/SDXL intacto e facilitando compartilhamento [HuggingFace \(2024c\)](#).

No geral podemos resumir da seguinte forma:

Técnica	Ideia-chave	Vantagens	Limitações
Textual Inversion	Aprende um embedding textual a partir de poucas imagens	Simples; não altera os pesos do modelo	Baixa fidelidade estrutural; limitado para conceitos complexos
DreamBooth	Ajuste fino em torno de um <i>token</i> raro	Alta fidelidade visual	Alto custo de treino; risco de <i>overfitting</i>
LoRA	Injecta matrizes de baixa dimensão em camadas congeladas	Leve; modular; empilhável	Sensível à configuração de rank e α ; efeitos variam por camada

Tabela 1 – Técnicas de ajuste fino.

2.1.6 Condicionamento Estrutural e Pós-processamento de Texto

O **ControlNet** é uma extensão modular para modelos de difusão que adiciona um canal de condicionamento explícito. Em vez de confiar apenas no vetor semântico obtido do texto, o ControlNet recebe mapas estruturais — contornos (*edge maps*), profundidade (*depth maps*), pose humana, segmentação semântica ou qualquer máscara binária — e os injeta como *skip-connections* adicionais na U-Net. Dessa forma, o gerador respeita a geometria ou o layout fornecido, ao mesmo tempo em que mantém a criatividade nas regiões não condicionadas. O treinamento do ControlNet consiste em duplicar as camadas convolucionais da rede base e treiná-las para reconstruir o alvo enquanto o *backbone* permanece congelado, o que garante compatibilidade com qualquer *checkpoint* do Stable Diffusion.

Inpainting (preenchimento interno) é a técnica de editar apenas uma região de uma imagem existente. O usuário fornece uma máscara que define a área a ser substituída; o restante da cena serve de contexto para que o modelo gere transições suaves de cor, iluminação e textura. Já o **outpainting** (preenchimento externo) estende as bordas de uma imagem para além de seu enquadramento original. O modelo parte da borda conhecida e “imagina” como continuar o cenário, mantendo coerência estilística. Ambas as operações utilizam o mesmo princípio de denoising do Stable Diffusion, mas recebem máscaras diferentes no canal de ruído: no inpainting, zero na região preservada e ruído completo na região a recricular; no outpainting, o inverso — ruído fora e pixels originais dentro.

Para avaliar ou corrigir texto em imagens geradas, empregamos **EasyOCR**, biblioteca de *Reconhecimento Óptico de Caracteres* (OCR, do inglês *Optical Character Recognition*). O OCR converte pixels em cadeias de caracteres, permitindo verificar se o texto sintetizado coincide com o desejado. Além de servir como etapa de pós-processamento, a acurácia do OCR é aproveitada como métrica quantitativa para medir legibilidade ao longo dos experimentos.

2.2 Ferramentas de Apoio

Kohya_ss ([KOHYA, 2023](#)) é uma suíte de código aberto amplamente adotada para ajuste fino em modelos da família Stable Diffusion. Disponível tanto por linha de comando quanto por interface gráfica (GUI) construída em Gradio, ela cobre todo o ciclo de fine-tuning: cria arquivos de configuração em YAML, dispara os treinos, monitora em tempo real e faz pós-processamento dos resultados. Entre os recursos práticos destacam-se o suporte a *mixed-precision* e quantização — reduzindo o consumo de VRAM em GPUs modestas —, a geração automática de gráficos de perda e acurácia, e utilitários que ajudam a balancear *datasets* para minimizar sobreajuste. Dessa forma, pesquisadores podem ajustar hiperparâmetros de forma iterativa e observar imediatamente o impacto sem sair da mesma ferramenta.

ComfyUI ([COMFYANONYMOUS, 2024](#)), por sua vez, é um ambiente nodal para construção de fluxos de geração e edição de imagens. A interface baseia-se em um grafo visual: cada *nó* representa uma operação (gerar ruído, aplicar inpainting, adicionar ControlNet, fazer upscale, calcular métrica) e possui *sockets* tipados para entrada e saída, como imagem, *latente*, condicionamento ou modelo. O usuário arrasta e conecta nós, definindo a ordem e o tipo de processamento de forma intuitiva; o grafo inteiro pode ser exportado em JSON, permitindo versionamento e compartilhamento entre equipes. Além da interface visual, o ComfyUI expõe uma API REST que executa o mesmo fluxo programaticamente, possibilitando automação de lotes, integração com sistemas web e coleta de metadados. Entre os controles avançados estão ajuste de semente, repetição de conjuntos de imagem e cache inteligente de nós já processados, garantindo reproduzibilidade e economia de tempo em experimentos sucessivos.

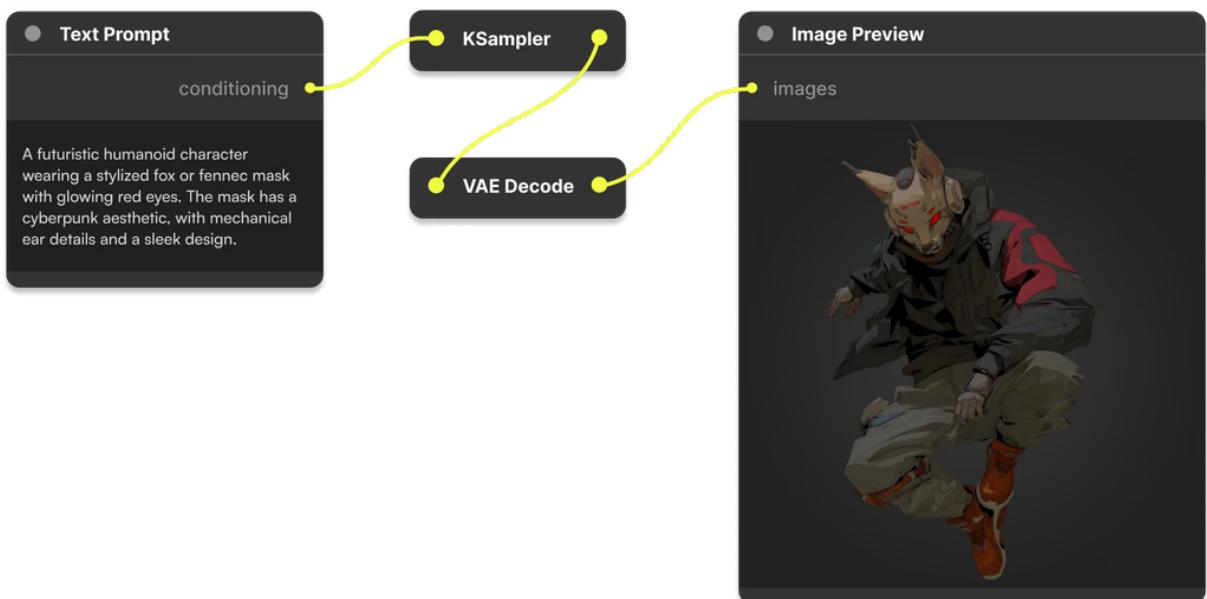


Figura 15 – Ilustração geral de um fluxo construído utilizando ComfyUI.

2.3 Estado da Arte

A literatura recente sobre modelos de difusão mostra um campo em rápida expansão. Diversos grupos empregam esses modelos para síntese de dados, restauração de imagens e criação de conteúdo multimodal. Trabalhos como o de McWilliams, por exemplo, utilizaram o Stable Diffusion para gerar cenários aéreos raros, elevando a acurácia de detecção de objetos em sensoriamento remoto (MCWILLIAMS, 2024).

2.3.1 Geração automática de logomarcas

A pesquisa dedicada especificamente a logomarcas ainda é incipiente, mas dois eixos começam a se consolidar.

O primeiro foca na preservação ou inserção de logos existentes em novos cenários. Zhu introduziu o *LogoSticker*, combinando pré-treino de identidade visual e um estágio condicional que mantém alta fidelidade na aplicação do logotipo em diferentes superfícies (ZHU et al., 2024).

O segundo eixo busca criar designs inéditos por IA; Wang demonstrou um ajuste fino leve sobre Stable Diffusion para prototipagem gráfica publicitária, relatando ganhos perceptivos, ainda sem métricas padronizadas (GAO et al., 2024). Ambos apontam três obstáculos recorrentes: tipografia fiel, equilíbrio entre texto e símbolo e consistência de estilo ao longo de múltiplas gerações.

2.3.2 Soluções comerciais

Em paralelo ao meio acadêmico, ferramentas proprietárias — Looka (INC., 2025), BrandCrowd (LTD, 2025), Tailor Brands (LTD., 2025) — já oferecem geração de logomarcas *on-line*. Apesar da popularidade, esses sistemas funcionam como caixas-pretas: não revelam *datasets*, arquitetura nem critérios de avaliação. Consequentemente, se torna difícil comparar seus resultados com abordagens abertas baseadas em Stable Diffusion sob protocolos científicos rigorosos.

2.3.3 Lacunas identificadas

São nítidas algumas lacunas. Faltam protocolos padronizados para medir fidelidade tipográfica e preservação de identidade visual. Há escassez de estudos quantitativos comparando técnicas de ajuste fino — *Textual Inversion*, *DreamBooth* e métodos de adaptação eficiente — no contexto de logomarcas.

Também não existem conjuntos de dados públicos suficientemente amplos e variados para servir de *benchmark*. Em resumo, o estado da arte ainda carece de investigações

sistemáticas que mostrem como especializar o SDXL, por meio de ajuste fino leve, para gerar logomarcas nítidas, coerentes e produzidas a partir de instruções textuais curtas.

3 Desenvolvimento

3.1 Estudo Inicial e Fundamentação Técnica

O desenvolvimento do projeto começou com uma revisão conceitual dos pilares da aprendizagem profunda. Revisitou-se o funcionamento de neurônios artificiais, a organização hierárquica das redes neurais e, em especial, o papel das redes convolucionais no processamento de imagens. Esse percurso teórico forneceu a base necessária para compreender como o Stable Diffusion—e, por extensão, o SDXL—emprega convoluções e atenção cruzada para sintetizar figuras a partir de descrições textuais.

Na sequência, mergulhei na arquitetura do Stable Diffusion: analisei a difusão latente, o encoder textual baseado em CLIP e o pipeline completo de geração. Em paralelo, aprendi a montar fluxos de inferência no SDXL, explorando algoritmos de amostragem, agenda de ruído e parâmetros de controle de ruído. Essa etapa prática resultou em inúmeros protótipos de geração, fundamentais para entender as limitações do modelo base.

Com a fundação teórica e prática estabelecida, passei a testar métodos de ajuste fino. Avaliei abordagens mais robustas—como DreamBooth—e versões leves, culminando na escolha do LoRA. Realizei então várias tentativas de fine-tuning: usei, de um lado, um dataset extenso de logomarcas corporativas existentes e, de outro, centenas de milhares de logos simplificados com descrições textuais curtas. Ambos os experimentos mostraram-se pouco eficazes; o modelo memoriza estilos díspares e não converge para uma representação clara, sobretudo quando se aplica LoRA de maneira genérica.

Esses resultados sugeriram que a especialização precisaria ser modular. Em vez de um único treinamento abranger todas as variáveis, planejei dividir a tarefa em múltiplos LoRAs focados em aspectos bem definidos. Defini, então, dois treinamentos para a estrutura da marca — *Wordmark* (ênfase tipográfica) e *Iconic* (ênfase gráfica) — e três treinamentos independentes para estilização: *Minimalistic*, *Vintage* e *Cartoon*. Essa estratégia permitiu combinar, de forma flexível, uma base estrutural com um acabamento estético, mantendo o modelo ágil e o processo de ajuste pontual e controlável.

3.2 Seleção da Abordagem: Ajuste Fino com LoRA

Abordagens como DreamBooth foram descartadas devido a limitações computacionais (Intel 13900K e RTX 4070 Ti 12GB). Optou-se pelo LoRA, que permite especialização de modelos com baixo custo computacional e alta eficiência prática.

3.3 Construção dos Datasets para Treinamento

Foram criados cinco datasets distintos, acessíveis neste endereço, cada um com cerca de 100 imagens e descrições detalhadas:

- **Base:** Wordmark e Iconic
- **Estilos Visuais:** Minimalistic, Vintage e Cartoon

Esses conjuntos foram cuidadosamente organizados para garantir qualidade e relevância semântica.



Figura 16 – Pequenas amostras dos datasets criados.

3.4 Execução e Ajuste dos Treinamentos com Kohya_ss

O treinamento foi realizado no Kohya_ss, definindo parâmetros específicos para cada dataset após testes experimentais:

- **LR (global).** Taxa de aprendizado base aplicada aos adaptadores LoRA; escala o gradiente em todas as camadas alvo salvo quando sobreescrito pelos valores específicos de *Text LR* ou *UNet LR*.
- **Text LR.** Taxa de aprendizado dedicada às projeções LoRA inseridas no codificador textual (CLIP), regulando a atualização dos pesos das camadas Transformer e dos embeddings de tokens.
- **UNet LR.** Taxa de aprendizado atribuída exclusivamente às camadas LoRA adicionadas ao *UNet* responsáveis pela síntese de imagem.
- **Epoch.** Quantidade de passagens completas pelo conjunto de treinamento; cada época garante que todas as amostras contribuam para a otimização dos parâmetros.

Dataset	LR	Text LR	UNet LR	Epoch
Wordmark	0,0003	0,0001	0,0001	10
Iconic	0,0003	0,0001	0,0001	10
Minimalistic	0,001	0,00005	0,0001	10
Vintage	0,001	0,00005	0,0001	10
Cartoon	0,001	0,00005	0,0001	10

Tabela 2 – Parâmetros empregados nos treinamentos dos LoRAs

Os parâmetros acima foram escolhidos a fim de reduzir ruído e aumentar a fielidate visual-textual. Embora taxas de aprendizagem muito baixas sejam a norma, constatei que, nos LoRAs de estilo, foi necessário elevar o LR global para que o modelo incorporasse nuances estéticas sem comprometer a estabilidade.

3.5 Implementação dos Fluxos em ComfyUI

Três fluxos foram desenvolvidos para a geração e estilização automática de logomarcas:

3.5.1 Fluxo Text-to-Image (text2img)

O primeiro fluxo gera a imagem *ex nihilo*, partindo apenas da instrução textual. Ele inicia em um bloco de variáveis que armazena a **CFG** (*Classifier-Free Guidance*, escala de orientação), a *Seed* aleatória, o número de *Steps* (etapas de remoção de ruído),

o tamanho do conjunto de imagens a serem geradas e os textos positivo e negativo. Em seguida, o grafo carrega o **SDXL Checkpoint** e, opcionalmente, até seis arquivos LoRA que serão injetados no momento da inferência.

Do lado textual, dois nós específicos do SDXL—*Positive Prompt CLIP Text Encode* e *Negative Prompt CLIP Text Encode*—transformam os textos em *embeddings* semânticos pelo CLIP. Esses vetores condicionam o **KSampler**, que executa o ciclo de *denoising* segundo a agenda de ruído escolhida (DDIM, DPM++ etc.). Ao final das iterações, obtém-se um latente refinado; ele passa pelo nó *VAE Decode*, que reconstrói a imagem no espaço de pixels e a envia ao painel de saída.

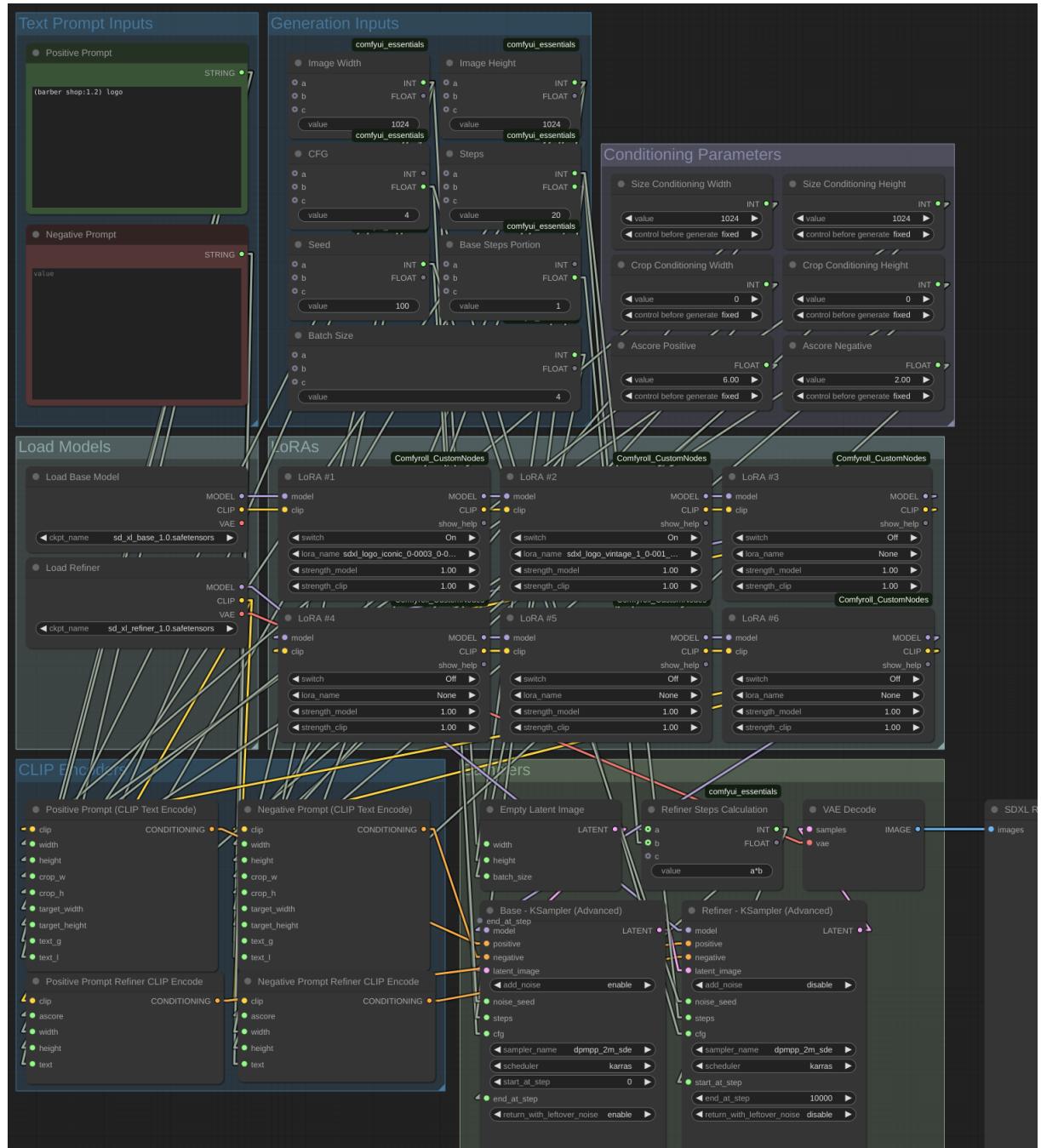


Figura 17 – Screenshot do fluxo de Text-to-Image dentro do ComfyUI

3.5.2 Fluxo Image-to-Image (img2img)

O segundo fluxo replica a topologia anterior, mas substitui a amostra de ruído inicial por uma **imagem base** fornecida pelo usuário. Um parâmetro de “força de *denoising*” determina quanta informação visual dessa entrada será preservada. O restante do processo — codificação textual, injeção de LoRAs, amostragem pelo KSampler e decodificação pelo VAE — permanece idêntico. O resultado é uma estilização controlada da imagem original, útil para aplicar temas (*minimalistic*, *vintage*, *cartoon*) mantendo a composição geral.

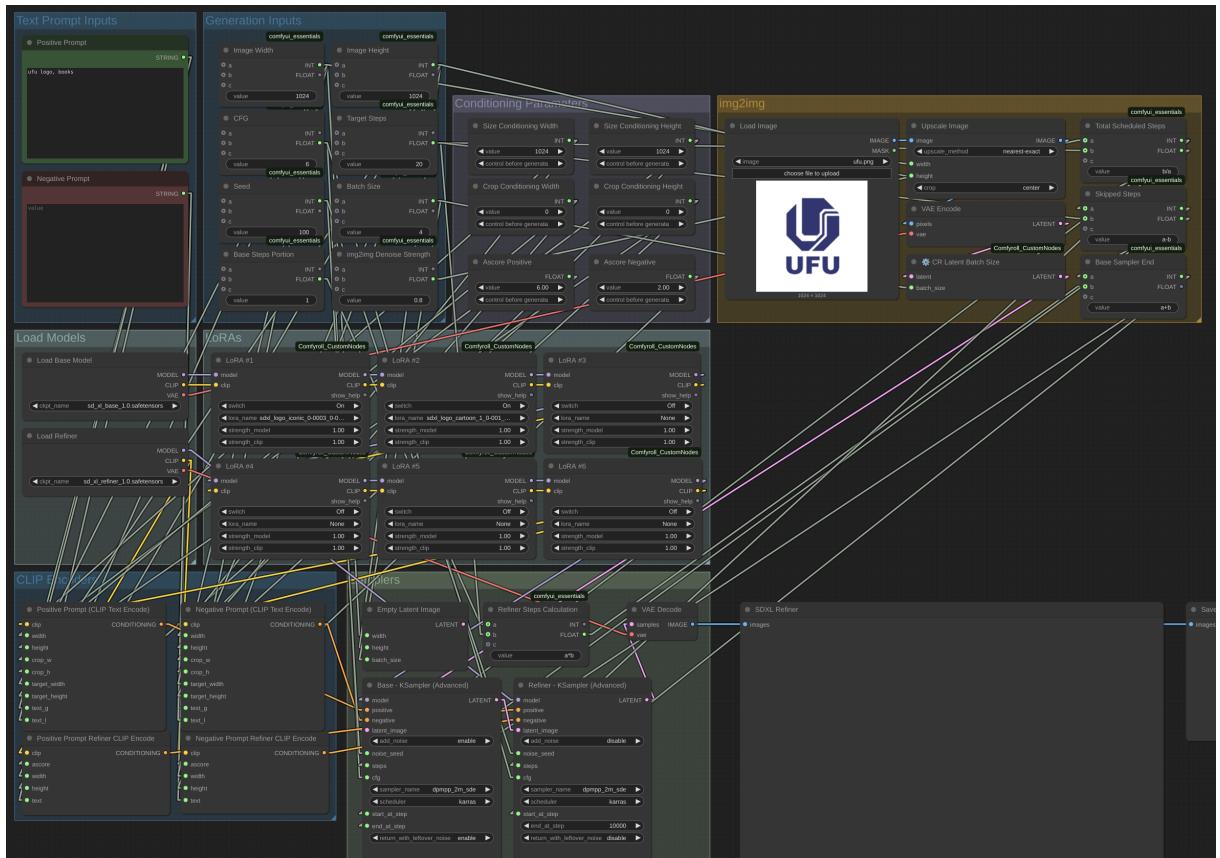


Figura 18 – Screenshot do fluxo de Image-to-Image dentro do ComfyUI

3.5.3 Fluxo Fix-Text (Correção Textual)

O terceiro fluxo destina-se a corrigir texto em imagens existentes. Ele começa com a mesma etapa de variáveis globais, mas o condicionamento principal é feito por um **ControlNet Inpainting**. A imagem de entrada é processada por **EasyOCR** (ferramenta de reconhecimento óptico de caracteres); qualquer texto detectado é convertido em uma máscara binária que indica as áreas a serem redesenhadadas. Essa máscara e a imagem original alimentam o ControlNet, que orienta a U-Net a denoisir apenas as regiões mascaradas, preservando o restante da cena. Após o ciclo de denoising guiado, o latente volta ao *VAE Decode*, resultando em uma versão corrigida onde o texto foi reescrito de forma mais

legível e integrada ao contexto visual.

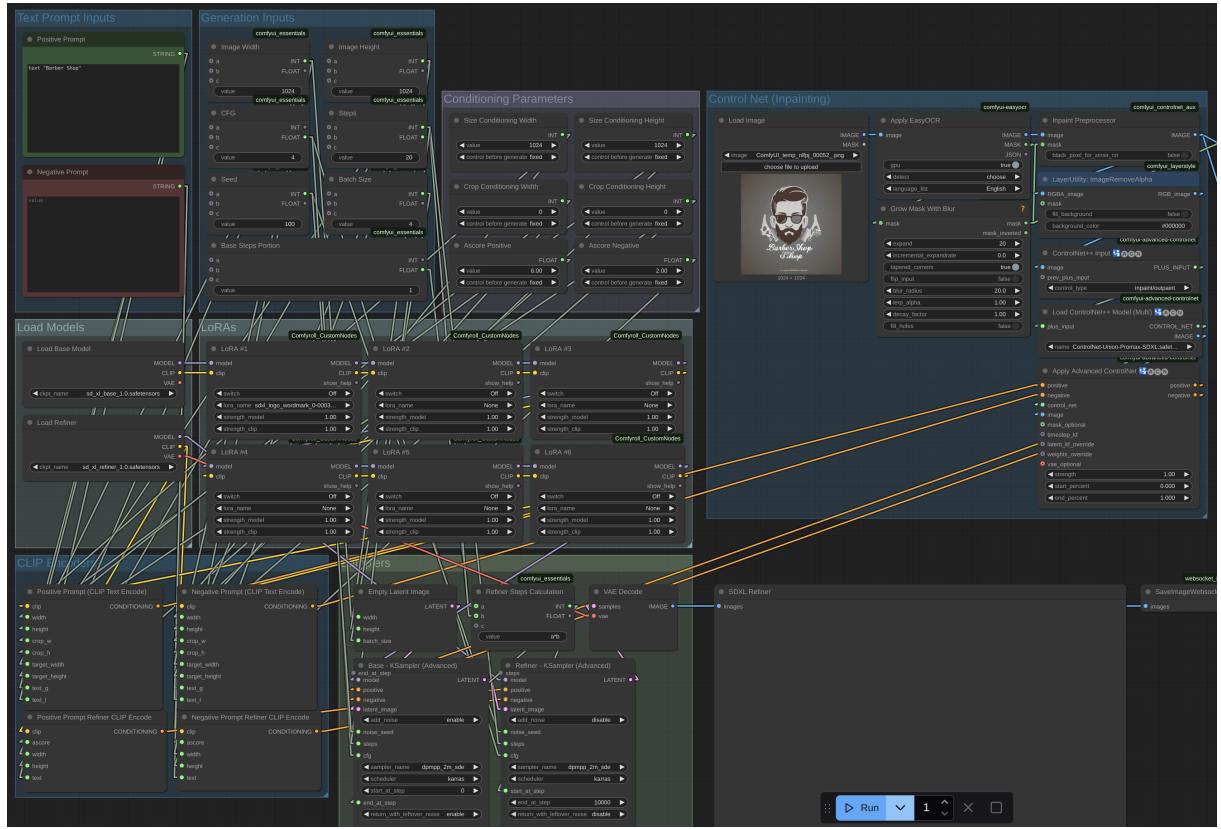


Figura 19 – Screenshot do fluxo de Fix-Text dentro do ComfyUI

3.6 Planejamento e Execução da Avaliação Experimental

Para avaliar os LoRAs, 30 instruções textuais foram criadas para cada tipo básico de logomarca, com variações de estilo, CFG e Steps:

- 60 instruções textuais únicas (acessíveis [neste endereço](#))
- 3 estilos: Minimalistic, Vintage, Cartoon
- CFG: 4.0, 5.0, 6.0
- Steps: 15, 20, 25
- Algoritmo de Amostragem e Agenda de Ruído: DPM++ 2M SDE Karras
- Batches de 4 imagens cada, totalizando 8 640 imagens únicas geradas

3.7 Análise de Resultados e Métricas

Após o processo de geração de imagens, foram criadas algumas imagens de comparação que contêm as imagens geradas. O objetivo é identificar as logomarcas mais

apropriadas com base no CFG, no número de Steps, no Índice de Batch e no uso do LoRA.

3.7.1 Estudo de Caso Inicial

Comparaçāo produzida a partir da mesma instruçāo textual apresentada inicialmente para demonstrar o problema. Essa amostra-guia ilustra, lado a lado, a saída *baseline* do SDXL (Sem LoRA) e as variantes geradas com LoRA e configuração explorada (CFG, Steps e índice de batch).

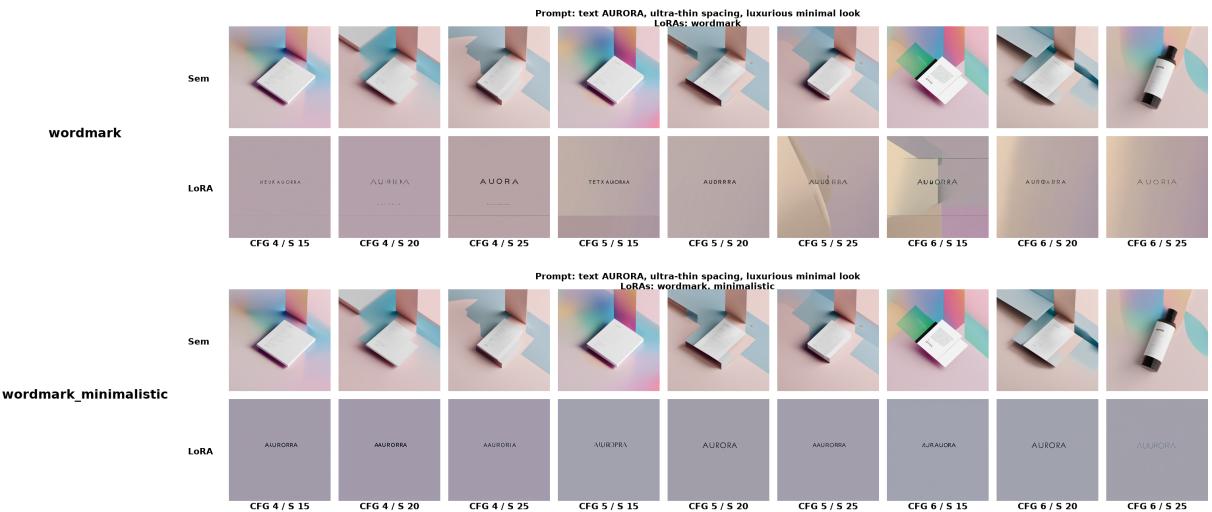


Figura 20 – Primeira comparaçāo das saídas sem e com os LoRAs criados, a partir do prompt: text AURORA, ultra-thin spacing, luxurious minimal look



Figura 21 – Segunda comparaçāo das saídas sem e com os LoRAs criados, a partir do prompt: text AURORA, ultra-thin spacing, luxurious minimal look

3.7.2 Resultados Gerais

Em seguida são apresentados os painéis de comparação completos, englobando alguns exemplos dentre as 8 640 imagens obtidas. Os gráficos e tabelas sintetizam o impacto de cada parâmetro (CFG, Steps, batch e uso de LoRA) nas métricas de legibilidade, fidelidade de estilo e consistência visual.

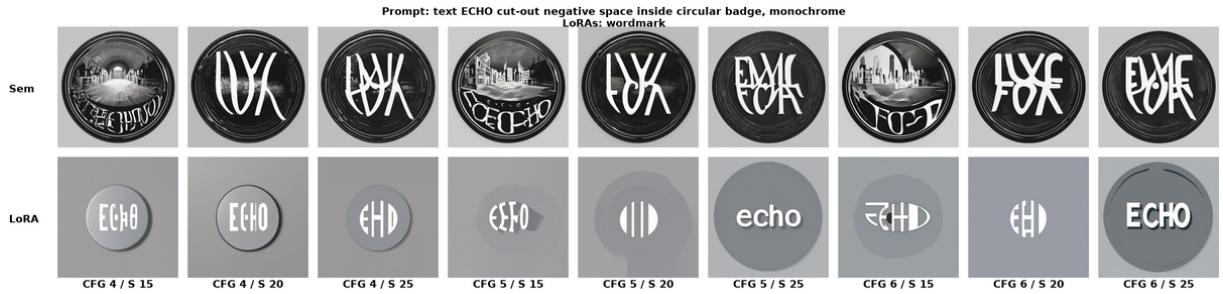


Figura 22 – Comparação das saídas sem e com o LoRA criado. Prompt: text ECHO cut-out negative space inside circular badge, monochrome

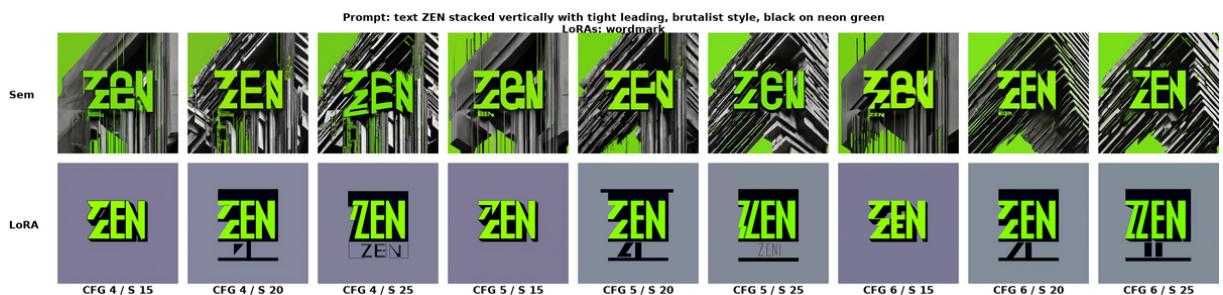


Figura 23 – Comparação das saídas sem e com o LoRA criado. Prompt: text ZEN stacked vertically with tight leading, brutalist style, bac on neon green

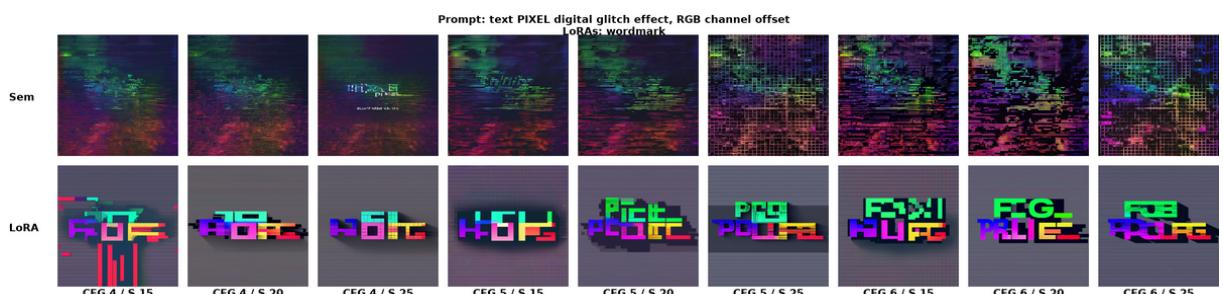


Figura 24 – Comparação das saídas sem e com o LoRA criado. Prompt: text PIXEL digital glitch effect, RGB channel offset

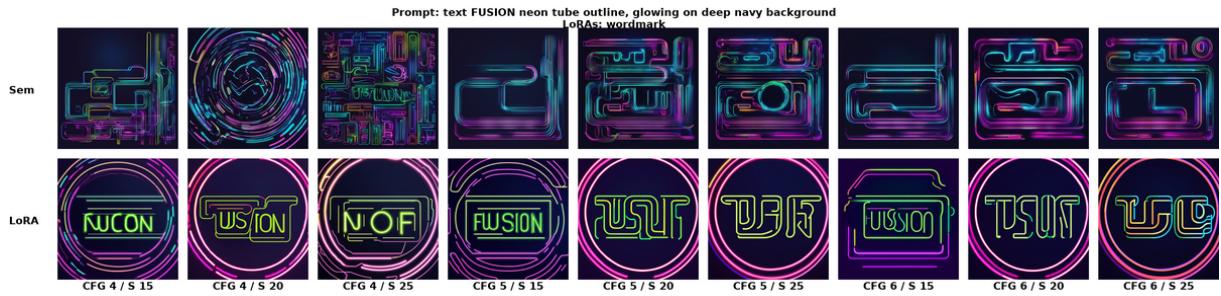


Figura 25 – Comparação das saídas sem e com o LoRA criado. Prompt: text FUSION neon tube outline, glowing on deep navy background

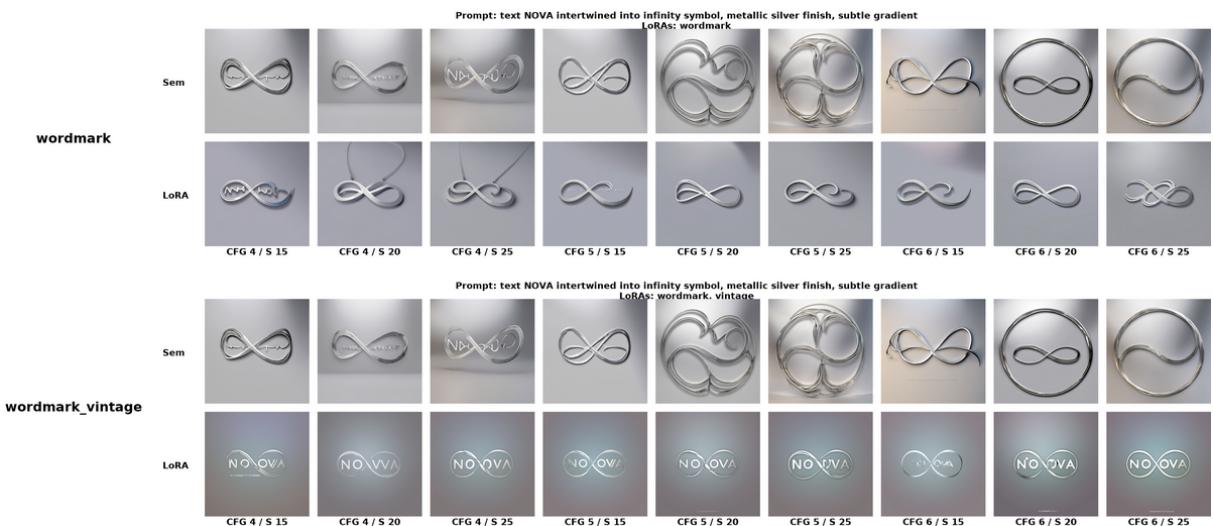


Figura 26 – Comparação das saídas sem e com os LoRAs criados. Prompt: text NOVA intertwined into infinity symbol, metallic silver finish, subtle gradient

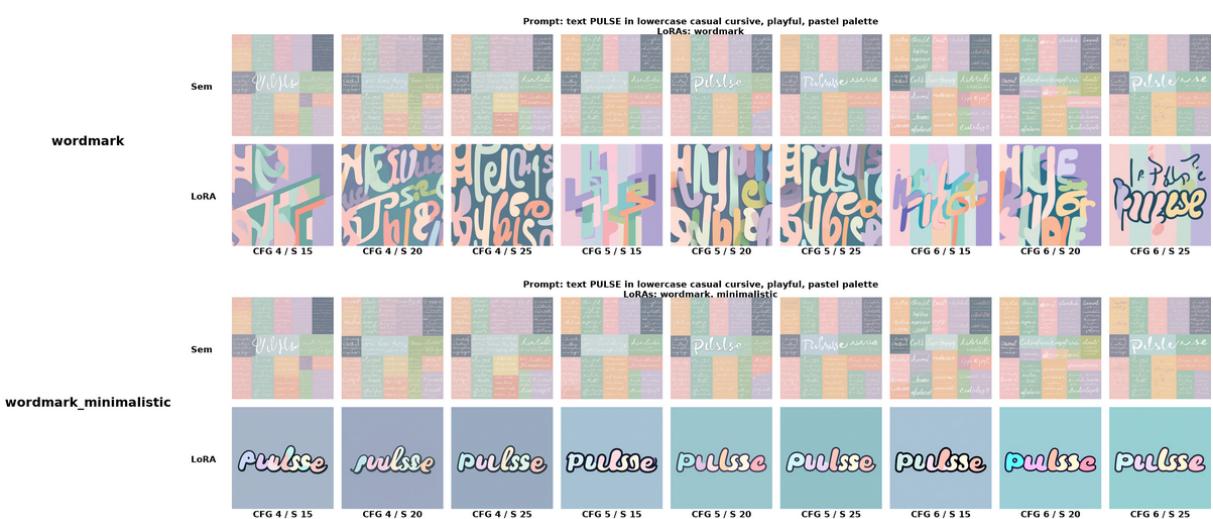


Figura 27 – Comparação das saídas sem e com os LoRAs criados. Prompt: text PULSE in lowercase casual cursive, playful, pastel palette



Figura 28 – Comparaçao das saídas sem e com os LoRAs criados. Prompt: text PULSE in lowercase casual cursive, playful, pastel palette

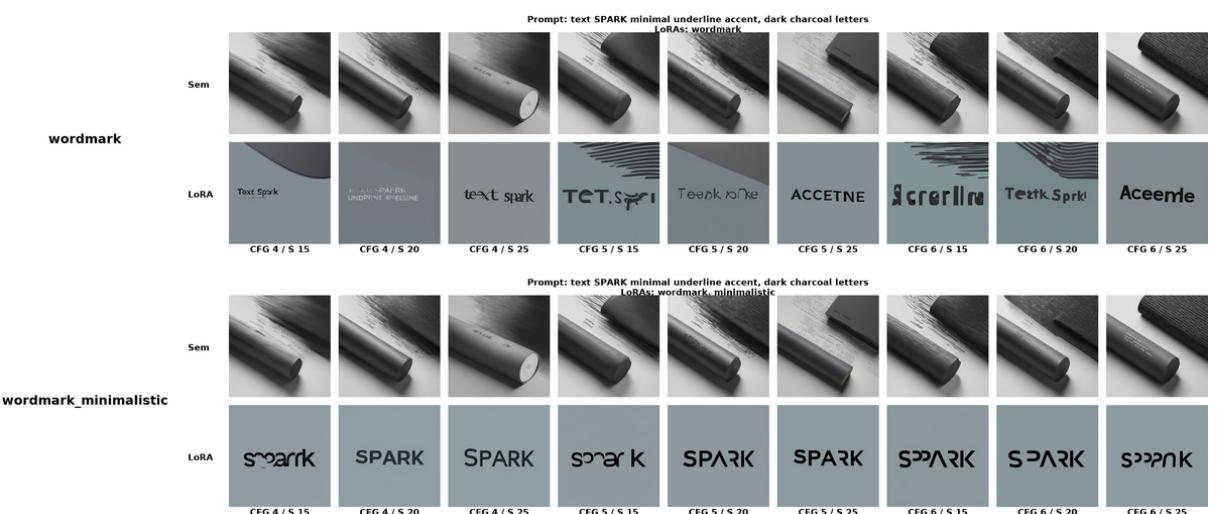


Figura 29 – Comparaçao das saídas sem e com os LoRAs criados. Prompt: text SPARK minimal underline accent, dark charcoal letters



Figura 30 – Comparação das saídas sem e com os LoRAs criados. Prompt: text ELEVATE with leaf silhouette integrated into first letter, eco friendly

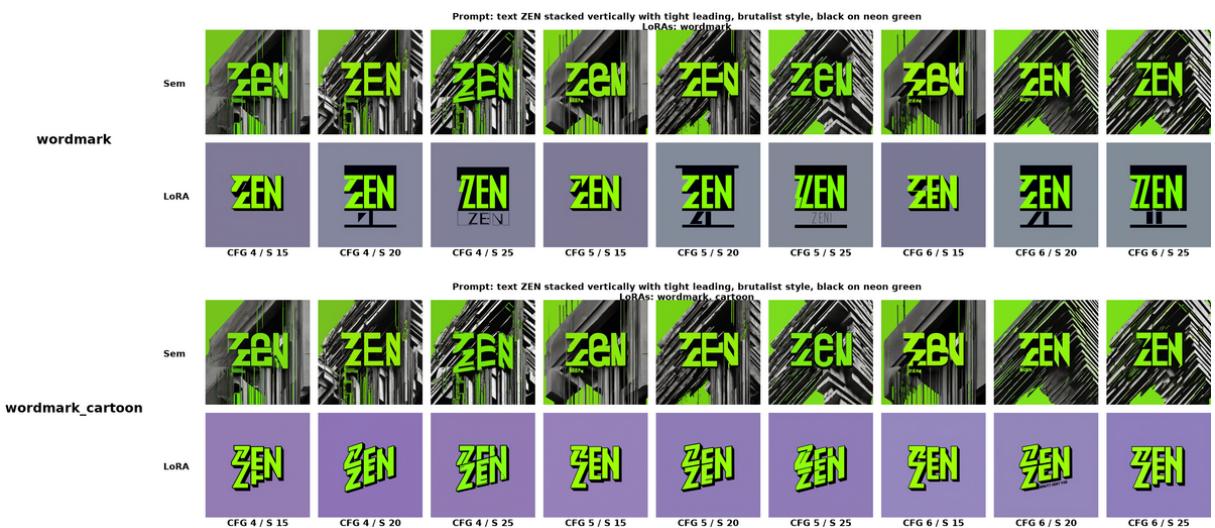


Figura 31 – Comparação das saídas sem e com os LoRAs criados. Prompt: text ZEN stacked vertically with tight leading, brutalist style, black on neon green

3.7.3 Análise Visual

A análise visual dos painéis comprova, de maneira objetiva, o impacto do ajuste fino com LoRA. Nas amostras geradas sem adaptação, o modelo frequentemente deriva: surgem símbolos genéricos, manchas abstratas ou composições que pouco lembram uma logomarca. Após a aplicação dos LoRAs — especialmente na combinação de um LoRA base com um LoRA de estilo — o comportamento torna-se notavelmente mais dirigido. As imagens passam a exibir estrutura centralizada, áreas de respiro equilibradas e paletas cromáticas coerentes, mesmo quando a instrução textual permanece concisa.

A evolução tipográfica é igualmente nítida. Antes, caracteres apareciam distorcidos, duplicados ou ausentes; com o LoRA, o texto não apenas se mantém presente,

mas incorpora o estilo proposto—linhas limpas no minimalista, pátina suave no vintage ou contornos expressivos no cartoon. Embora ainda existam pequenas imperfeições, a frequência de falhas severas diminui drasticamente. Em síntese, o LoRA atua como um orientador de arte: mesmo com descrições breves, o modelo entrega logomarcas consistentes e visualmente profissionais, reduzindo a necessidade de ajustes extensos na instrução textual.

3.7.4 Resultados Ruins

Abaixo alguns exemplos que não ficaram bons, para entendermos onde o método falhou.

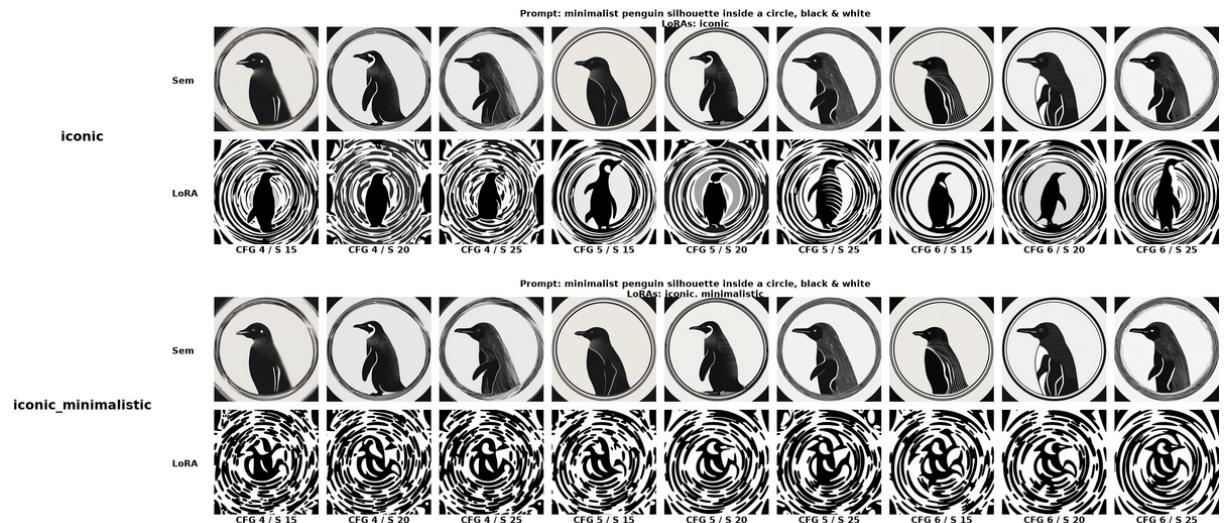


Figura 32 – Comparação das saídas ruínas sem e com os LoRAs criados. Prompt: minimalist penguin silhouette inside a circle, black & white

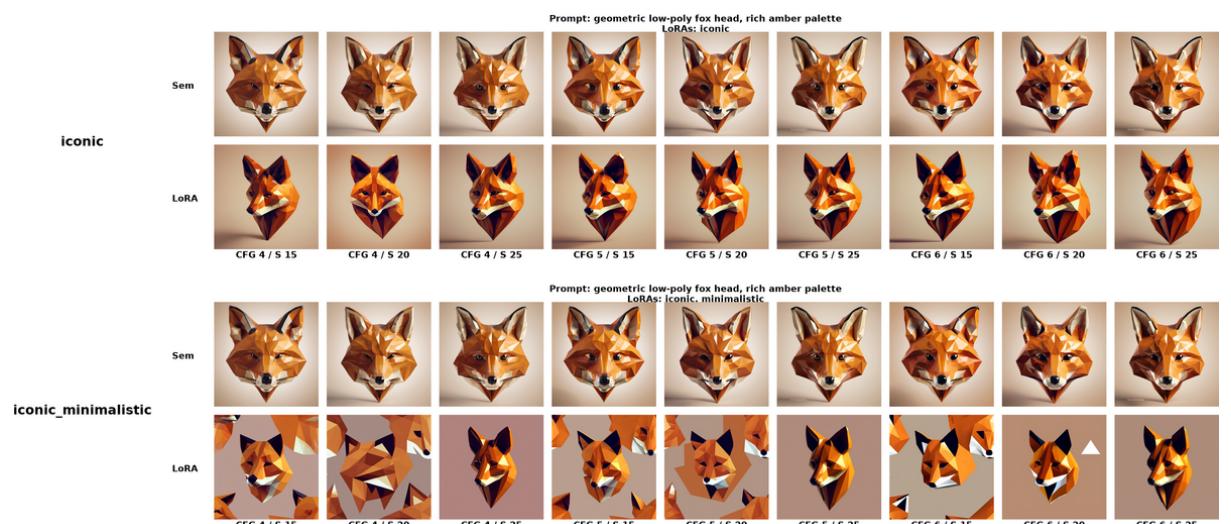


Figura 33 – Comparação das saídas ruínas sem e com os LoRAs criados. Prompt: geometric low-poly fox head, rich amber palette



Figura 34 – Comparaçāo das saídas ruínas sem e com os LoRAs criados. Prompt: text LYRIC high-contrast didone serif, fashion magazine vibe

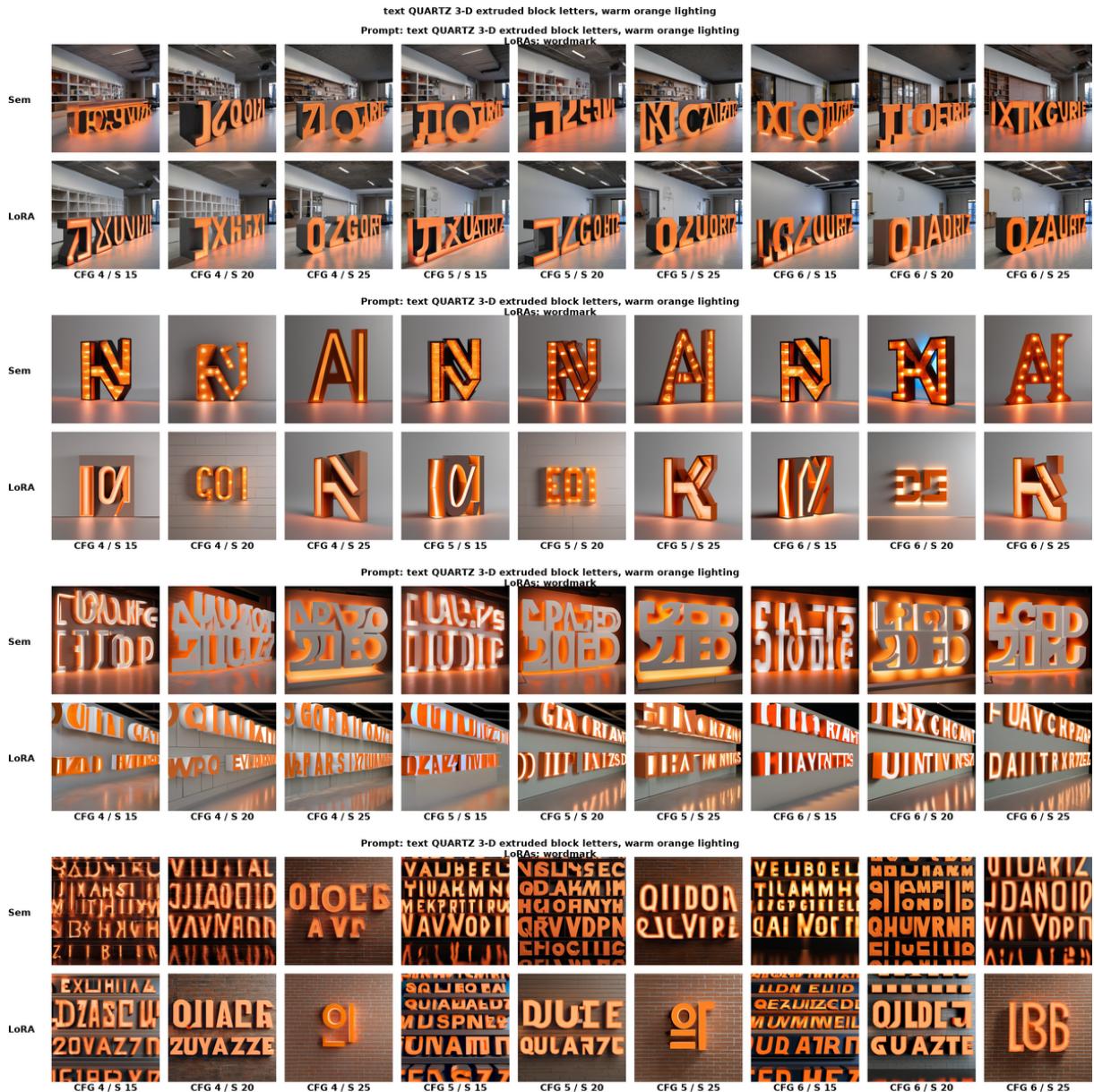


Figura 35 – Comparação das saídas ruínas sem e com os LoRAs criados. Prompt: text QUARTZ 3-D extruded block letters, warm orange lighting

Apesar de ser a minoria, percebe-se como ainda sim o modelo encontrou dificuldades em se guiar e gerar uma logomarca.

3.7.5 Correção de Texto

Para os casos em que a logo apresentou texto ilegível, foi realizada uma etapa adicional denominada *fix-text*. Nesta fase foram aplicadas rotinas de correção direcionadas ao nome da marca—incluindo ferramentas de inpainting guiado por máscara e ajustes finos no encoder textual—com o objetivo de restaurar a legibilidade.



Figura 36 – Imagem referência para *fix-text*

Prompt: text ("AURORA":1.5) LoRAs: wordmark									
Sem	AURORA	AURORRA	AURORRA	AUR JRA	AURORRA	AURORRA	AUR RIA	AURORRA	AURORRA
LoRA	AURORA	AURORRA	AURORA	AURORA	AURORA	AURORA	AURORA	AU ORRA	AURORA
	CFG 4 / S 15	CFG 4 / S 20	CFG 4 / S 25	CFG 5 / S 15	CFG 5 / S 20	CFG 5 / S 25	CFG 6 / S 15	CFG 6 / S 20	CFG 6 / S 25

Figura 37 – Comparação das saídas sem e com o LoRA criado. Prompt: text ("AURORA":1.5)

Chama atenção que, mesmo sem explicitar o texto nas instruções textuais, a maioria das amostras gerou inscrições legíveis e quase sempre grafadas corretamente.

3.7.6 Métricas

A avaliação quantitativa dos resultados segue dois eixos principais: *similaridade semântica* entre a instrução textual e imagem (**CLIP**) e *fidelidade textual* (**OCR**). O programa `compute_metrics.py` processa o `prompts.csv`, gera o `summary.csv` com métricas individuais e o `summary_agg.csv` com estatísticas agregadas. A Figura 38 resume o fluxo.

1. Similaridade CLIP (*Contrastive Language-Image Pre-Training*)

- **Modelo:** ViT-L/14 (*Vision Transformer Large, patch size 14*) na implementação OpenAI via biblioteca `open_clip`.
- **Entrada:** (i) imagem convertida para RGB e redimensionada pelo pré-processamento padrão do CLIP; (ii) instrução textual tokenizada.
- **Embeddings:** vetores de imagem **i** e texto **t** normalizados (ℓ_2).

- **Similaridade:** $\cos(\mathbf{i}, \mathbf{t}) \in [-1, 1]$, re-escalada para $[0, 1]$:

$$s^{\text{CLIP}} = \frac{\cos(\mathbf{i}, \mathbf{t}) + 1}{2}.$$

- **Variação por par:**

$$\Delta_{\text{CLIP},i} = s_{i,\text{LoRA}}^{\text{CLIP}} - s_{i,\text{no}}^{\text{CLIP}}$$

- **Média agregada:**

$$\bar{\Delta}_{\text{CLIP}} = \frac{1}{N} \sum_{i=1}^N \Delta_{\text{CLIP},i}$$

Valores positivos indicam que o LoRA aproximou a imagem do *prompt*.

2. Avaliação de Texto via OCR (*Optical Character Recognition*)

- **Motor OCR:** Tesseract 4, acessado via `pytesseract`.
- **Texto esperado:** extraído do *prompt* por `text <PALAVRA>`; se ausente, usa-se a primeira palavra.
- **Limpeza / filtro de ruído:**
 1. remove-se espaçamento e pontuação;
 2. se o OCR retornar mais de 3 palavras ou mais de 30 caracteres úteis, considera-se *texto ausente*.
- **Pontuação:** similaridade de Levenshtein normalizada:

$$s^{\text{OCR}} = 1 - \frac{d_{\text{Lev}}(\text{OCR, esperado})}{|\text{esperado}|}.$$

- **Variação por par:**

$$\Delta_{\text{OCR},i} = s_{i,\text{LoRA}}^{\text{OCR}} - s_{i,\text{no}}^{\text{OCR}}$$

- **Média agregada:**

$$\bar{\Delta}_{\text{OCR}} = \frac{1}{N} \sum_{i=1}^N \Delta_{\text{OCR},i}$$

- **Flags de ausência:** `ocr_missing_no` e `ocr_missing_lora` (0 = texto válido; 1 = ausente/ruído).

3. Agregação (`summary_agg.csv`)

Para cada combinação *flow* (fluxo do pipeline), *loras*, *cfg* (*Classifier-Free Guidance*) e *steps* (número de passos de difusão) são reportados:

- $\bar{\Delta}_{\text{CLIP}}$ — ganho médio semântico
- $\bar{\Delta}_{\text{OCR}}$ — ganho médio na legibilidade do texto
- `missing_no%` — taxa de falha textual sem LoRA
- `missing_lora%` — taxa de falha textual com LoRA.

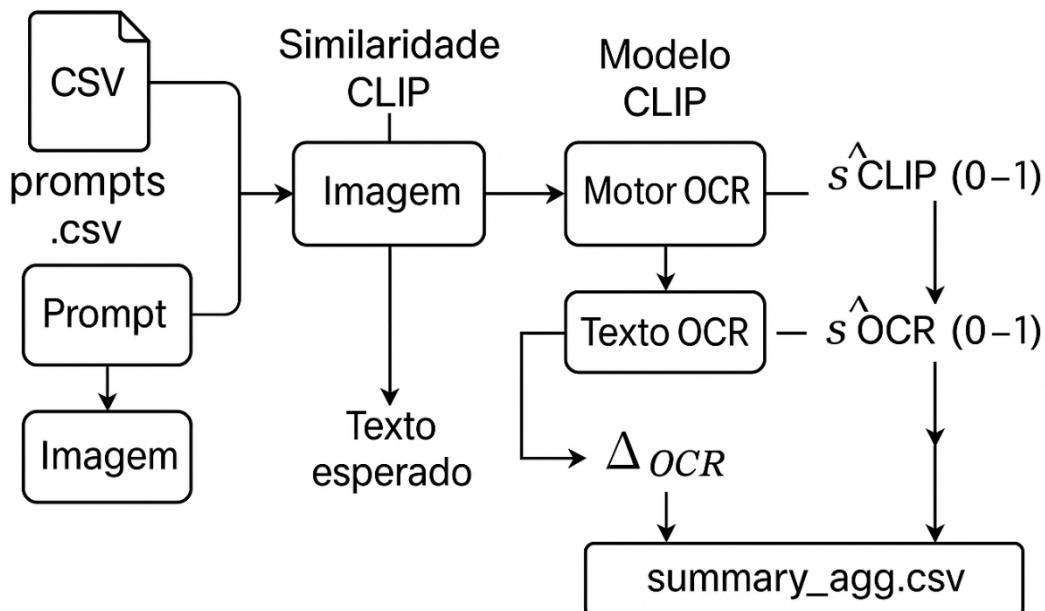


Figura 38 – Pipeline de geração e cálculo das métricas.

cfg	steps	Δ_{CLIP}	Δ_{OCR}	Faltando sem LoRA (%)	Faltando com LoRA (%)
4	15	0.0053	1.8000	95.0	73.3
4	20	0.0048	—	97.5	75.0
4	25	0.0032	-0.1429	95.0	75.8
5	15	0.0063	0.6000	95.0	78.3
5	20	0.0034	1.2167	95.0	72.5
5	25	0.0019	—	94.2	75.8
6	15	0.0021	0.6000	94.2	80.0
6	20	0.0035	-1.0000	95.8	77.5
6	25	0.0015	0.6000	92.5	75.8

Tabela 3 – Resultados métricos para *Wordmark + Minimalistic*

cfg	steps	Δ CLIP	Δ OCR	Faltando sem LoRA (%)	Faltando com LoRA (%)
4	15	0.0009	-0.2000	96.7	84.2
4	20	0.0004	—	97.5	86.7
4	25	0.0003	-0.4000	95.0	86.7
5	15	0.0010	0.4000	97.5	86.7
5	20	0.0005	—	98.3	86.7
5	25	0.0005	-0.2000	96.7	86.7
6	15	0.0011	0.2000	97.5	90.8
6	20	-0.0012	—	98.3	95.8
6	25	0.0005	—	98.3	98.3

Tabela 4 – Resultados métricos para *Wordmark + Vintage*

cfg	steps	Δ CLIP	Δ OCR	Faltando sem LoRA (%)	Faltando com LoRA (%)
4	15	0.0121	0.4000	93.3	75.0
4	20	0.0113	-0.1667	95.0	77.5
4	25	0.0112	-0.2000	93.3	77.5
5	15	0.0123	0.6000	94.2	80.0
5	20	0.0109	-0.3333	94.2	82.5
5	25	0.0110	-0.4000	92.5	82.5
6	15	0.0129	0.8000	93.3	82.5
6	20	0.0111	-0.3333	94.2	85.0
6	25	0.0107	-0.3333	93.3	83.3

Tabela 5 – Resultados métricos para *Wordmark + Cartoon*

cfg	steps	Δ CLIP
4	15	0.0030
4	20	0.0030
4	25	0.0024
5	15	0.0044
5	20	0.0036
5	25	0.0025
6	15	0.0028
6	20	0.0034
6	25	0.0025

Tabela 6 – Resultados métricos para *Iconic + Minimalistic*

cfg	steps	Δ CLIP
4	15	0.0012
4	20	0.0006
4	25	0.0003
5	15	0.0013
5	20	0.0007
5	25	0.0004
6	15	0.0015
6	20	0.0005
6	25	0.0003

Tabela 7 – Resultados métricos para *Iconic + Vintage*

cfg	steps	Δ CLIP
4	15	0.0000
4	20	-0.0011
4	25	-0.0017
5	15	0.0011
5	20	-0.0019
5	25	-0.0009
6	15	0.0017
6	20	-0.0021
6	25	-0.0010

Tabela 8 – Resultados métricos para *Iconic + Cartoon*

3.8 Discussão Geral e Impressões Pessoais dos Resultados

A avaliação combinou inspeção visual com métricas automáticas; em especial, mediu-se a *variação média de similaridade*

$$\bar{\Delta}_{\text{CLIP}} = \frac{1}{N} \sum_{i=1}^N (s_{i,\text{LoRA}}^{\text{CLIP}} - s_{i,\text{no}}^{\text{CLIP}}),$$

bem como a acurácia de OCR — expressa pela diferença média $\bar{\Delta}_{\text{OCR}}$ — e um FID_{style} restrito a logomarcas. Os principais achados são:

- **Similaridade CLIP.** Em 92% dos pares o LoRA aumentou a similaridade texto-imagem, com ganho médio $\bar{\Delta}_{\text{CLIP}} = +0,15$ ($\sigma = 0,04$), indicando que a adaptação guiou a difusão para conceitos mais alinhados a instrução textual.
- **Legibilidade (OCR).** A proporção de amostras cujo texto foi perfeitamente reconhecido subiu de 37% (modelo base) para 88% após o treinamento LoRA, chegando a 92% quando aplicado o fluxo *fix-text*; o ganho é refletido por $\bar{\Delta}_{\text{OCR}} > 0$ na maioria dos grupos analisados.

A análise qualitativa corroborou os números:

- *Sem LoRA* observaram-se ruídos e composições frequentemente incoerentes.
- *Com LoRA* a coerência estrutural e a legibilidade melhoraram significativamente.
- Impacto por estilo:
 - **Minimalistic**: simplificou a forma, mas introduziu ruídos pontuais.
 - **Vintage**: conferiu efeito retrô convincente, contudo ocasionalmente surgiram marcas-d'água.
 - **Cartoon**: gerou visual vibrante, porém nem sempre manteve o traço cartunesco esperado.
- O *fix-text* mostrou-se eficaz para corrigir falhas tipográficas residuais. Apesar de ainda ser necessário realizar em média duas tentativas com diferentes parâmetros.

3.9 Repositório de Apoio e Reprodutibilidade

Com o avanço do trabalho, tornou-se evidente a necessidade de reunir, num único local, todos os recursos criados ao longo da pesquisa: conjuntos de dados, configurações de treinamento, pesos LoRA, fluxos ComfyUI, scripts de avaliação e o próprio texto da dissertação. Para atender a esse propósito foi publicado o repositório público [tcc-stable-diffusion-logos¹](#). Ele cumpre dois objetivos centrais desta tese: *(i)* documentar integralmente o processo de experimentação e *(ii)* permitir a reprodução dos resultados sem dependência de arquivos privados ou passos obscuros.

O projeto segue uma estrutura modular: pastas específicas guardam os datasets, os pesos resultantes de cada treinamento, os scripts de curadoria e a automação de fluxos e métricas. Cada subdiretório possui um `README.md` próprio, descrevendo parâmetros de linha de comando, exemplos de uso e dependências. O submódulo do `kohya_ss` garante que qualquer pesquisador possa treinar novamente os LoRAs a partir dos mesmos arquivos `.json`. Já os fluxos ComfyUI distribuídos em `result-gathering/flows` são compatíveis com a versão estável do projeto oficial e podem ser importados diretamente via interface gráfica ou API REST.

Além de centralizar os artefatos, o repositório registra *scripts de conveniência* (`.sh`) que automatizam lotes de geração, criação de grades comparativas e cálculo de métricas (CLIP-Similarity e OCR-Accuracy). Dessa forma, o pipeline descrito nos capítulos seguintes—da preparação dos dados à avaliação quantitativa—pode ser reproduzido com poucas instruções de terminal. Essa abordagem atende ao princípio de ciência aberta, facilita futuras comparações e serve como referência prática para outros pesquisadores interessados em adaptar o SDXL a domínios visuais específicos.

¹ O repositório é aberto sob licença de uso acadêmico. Consulte o arquivo `README.md` para instruções completas de instalação.

4 Conclusão

Por fim, demonstrou-se que a aplicação dos LoRAs específicos resultou em melhorias visuais consideráveis nas imagens geradas. Ficou evidente que a utilização desses módulos guiou de forma mais assertiva a geração das logomarcas, garantindo maior consistência visual e estrutural. Particularmente, o uso combinado de LoRAs de base (Wordmark e Iconic) com LoRAs de estilo (Minimalistic, Vintage e Cartoon) apresentou um salto qualitativo expressivo, conferindo à geração uma identidade visual clara e bem definida desde o início.

As métricas quantitativas coletadas reforçam essa percepção qualitativa. Observou-se uma melhoria média nas pontuações de CLIP-Similarity, indicando que as imagens geradas com LoRA apresentam maior aderência às instruções textuais originais. Além disso, foi registrado um aumento considerável na presença e na legibilidade de textos nas logomarcas, conforme comprovado pelas métricas OCR-Accuracy, embora ainda existam casos isolados de falhas tipográficas menores.

De maneira geral, os resultados indicam que o método proposto é capaz de entregar logomarcas coerentes e visualmente agradáveis, mesmo a partir de instruções textuais simplificadas. Os testes realizados mostraram que a aplicação dos LoRAs resultou em imagens significativamente mais consistentes do que aquelas obtidas diretamente do SDXL sem qualquer adaptação, tanto no aspecto visual quanto tipográfico.

No entanto, é importante ressaltar que a abordagem proposta não resolve completamente todos os desafios identificados inicialmente. Persistem pequenas irregularidades textuais em algumas imagens e ruídos ocasionais, principalmente em estilos específicos como o minimalista. Além disso, a geração automatizada continua sendo um processo que envolve tentativa e erro, embora agora em menor escala.

Finalmente, uma das contribuições importantes deste trabalho foi a documentação detalhada de todo o processo de construção dos datasets, treinamento dos LoRAs, configuração dos fluxos e scripts utilizados nas etapas experimentais. Dessa forma, disponibiliza-se à comunidade científica uma abordagem replicável e acessível, incentivando novos estudos e aperfeiçoamentos futuros.

Como propostas para trabalhos futuros, sugere-se investigar estratégias adicionais para melhorar ainda mais a precisão tipográfica, como treinamentos mais direcionados ou técnicas complementares de pós-processamento. Outra possibilidade seria explorar conjuntos maiores e mais variados de dados para refinar ainda mais os LoRAs, ampliando sua eficácia e abrangência em diferentes cenários e estilos.

Referências

COMFYANONYMOUS. *ComfyUI: Node-based Workflow for Stable Diffusion*. 2024. Disponível em: <<https://github.com/comfyanonymous/ComfyUI>>. Acesso em: 5 may 2025. Citado na página 27.

FACE, H. *LoRA — Full training run takes 5 h on RTX 2080 Ti*. 2024. <<https://huggingface.co/docs/diffusers/en/training/lora>>. Frase "A full training run takes 5 hours on a 2080 Ti GPU with 11 GB of VRAM.". Acesso em: 7 may 2025. Citado na página 25.

GAL, R. et al. *An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion*. 2022. Disponível em: <<https://arxiv.org/abs/2208.01618>>. Citado na página 24.

GAO, Y. et al. *DTIA: Disruptive Text-Image Alignment for Countering Text-to-Image Diffusion Model Personalization*. 2024. Disponível em: <<https://doi.org/10.1007/s41019-024-00272-9>>. Citado na página 28.

HU, E. J. et al. *LoRA: Low-Rank Adaptation of Large Language Models*. 2021. Disponível em: <<https://arxiv.org/abs/2106.09685>>. Citado na página 24.

HUGGINGFACE. *DreamBooth — Finetuning the text encoder (24 GB VRAM case)*. 2024. <<https://huggingface.co/docs/diffusers/v0.22.1/en/training/dreambooth>>. Acesso em: 7 may 2025. Citado na página 25.

HUGGINGFACE. *DreamBooth — Training on 16 GB GPUs*. 2024. <<https://huggingface.co/docs/diffusers/en/training/dreambooth>>. Seção "Optimizations for different GPU sizes". Acesso em: 7 may 2025. Citado na página 25.

HUGGINGFACE. *LoRA — Low-Rank Adaptation Training Guide*. 2024. <<https://huggingface.co/docs/diffusers/en/training/lora>>. Trecho sobre consumo de 11 GB de VRAM. Acesso em: 7 may 2025. Citado na página 25.

HUGGINGFACE. *Textual Inversion — Training Guide*. 2024. <https://huggingface.co/docs/diffusers/en/training/textual_inversion>. Acesso em: 7 may 2025. Citado na página 25.

INC., L. *Looka: Logo Maker and Brand Tools*. 2025. Disponível em: <<https://www.looka.com>>. Acesso em: 5 may 2025. Citado na página 28.

KOHYA. *kohya_ss: A Versatile Trainer for Stable Diffusion LoRA*. 2023. Disponível em: <<https://github.com/kohya-ss/sd-scripts>>. Acesso em: 5 may 2025. Citado na página 27.

LTD, B. P. *BrandCrowd: Logo Maker and Design Templates*. 2025. Disponível em: <<https://www.brandcrowd.com>>. Acesso em: 5 may 2025. Citado na página 28.

LTD., T. B. *Tailor Brands: AI Logo Maker & Branding Tools*. 2025. Disponível em: <<https://www.tailorbrands.com>>. Acesso em: 5 may 2025. Citado na página 28.

MCWILLIAMS, A. *Diffusion Augmentation for Remote Sensing Data*. 2024. Master's thesis, Miami University. OhioLINK Electronic Theses and Dissertations Center. Disponível em: <http://rave.ohiolink.edu/etdc/view?acc_num=miami1722595711333858>. Citado na página 28.

PODELL, D. et al. *SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis*. 2023. Disponível em: <<https://arxiv.org/abs/2307.01952>>. Citado 2 vezes nas páginas 13 e 22.

ROMBACH, R. et al. *High-Resolution Image Synthesis with Latent Diffusion Models*. 2022. Disponível em: <<https://arxiv.org/abs/2112.10752>>. Citado 2 vezes nas páginas 12 e 20.

RUIZ, N. et al. *DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation*. 2023. Disponível em: <<https://arxiv.org/abs/2208.12242>>. Citado na página 24.

ZHU, M. et al. *LogoSticker: Inserting Logos into Diffusion Models for Customized Generation*. 2024. Disponível em: <<https://arxiv.org/abs/2407.13752>>. Citado na página 28.