Geração Automatizada de Conteúdo Criativo Multimodal com Llama 3 e Stable Diffusion XL: Uma Abordagem Integrada em Ambiente Colab

Resumo

Este *paper* apresenta o desenvolvimento de um sistema integrado para a geração automatizada de conteúdo criativo multimodal, combinando um modelo de linguagem grande (LLM), **Llama 3 8B Instruct**, com o modelo de difusão **Stable Diffusion XL (SDXL)**. O sistema é implementado em um notebook interativo no *Google Colab*, com foco na acessibilidade, eficiência de memória e usabilidade. Usuários podem configurar parâmetros personalizados para geração de textos e imagens de alta qualidade, adequados para mídias sociais, marketing ou e-commerce. O trabalho descreve a arquitetura do sistema, as técnicas de quantização e offloading empregadas, a engenharia de prompt adaptativa para diferentes tipos de conteúdo textual, e discute os resultados obtidos.

---

1. Introdução

A explosão do conteúdo digital nas mídias sociais e plataformas de e-commerce aumentou consideravelmente a demanda por material original e visualmente atrativo. Criadores independentes, equipes de marketing e pequenas empresas enfrentam desafios ao produzir conteúdo em escala, mantendo consistência, relevância e apelo visual. As recentes inovações em inteligência artificial generativa — particularmente *LLMs* e modelos de difusão — oferecem novas possibilidades para automatizar esse processo.

Apesar da abundância de modelos de IA avançados, integrá-los de forma eficiente em ambientes acessíveis como o Google Colab permanece desafiador, especialmente devido à limitação de memória de *GPU*, à complexidade de dependências de bibliotecas e à falta de interfaces intuitivas para usuários não técnicos.

Este projeto tem como objetivo integrar os modelos Llama 3 e Stable Diffusion XL em um *pipeline* coerente e utilizável para a geração automatizada de conteúdo multimodal. A

proposta busca reduzir significativamente o uso de memória de vídeo (VRAM) por meio da quantização dos pesos do modelo de linguagem em 4 bits, além da adoção de estratégias inteligentes de offloading, que permitem a execução fluida mesmo em GPUs com recursos limitados, como as T4 disponibilizadas no Google Colab.

As principais contribuições deste trabalho incluem o desenvolvimento de um pipeline funcional, modular e extensível para geração de conteúdo multimodal; a proposição de soluções práticas para problemas recorrentes de instalação e compatibilidade de pacotes no ambiente do Google Colab; a implementação de estratégias de gestão de memória que viabilizam o uso de modelos de última geração em hardware com restrições; e a criação de uma interface de usuário interativa que, em conjunto com uma lógica de prompting dinâmico, permite a geração de conteúdo textual otimizado para formatos específicos (como posts de Instagram, e-mails de marketing e descrições de produto), além de imagens contextualmente relevantes.

2. Metodologia

2.1. Arquitetura do Sistema

O sistema proposto foi estruturado em módulos independentes, executados sequencialmente por meio de células no ambiente Google Colab. Essa abordagem modular facilita a compreensão, manutenção e reutilização dos componentes em diferentes contextos. A arquitetura geral compreende os seguintes blocos:

- **Interface de Usuário:** Desenvolvida com *ipywidgets*, a interface permite a entrada de parâmetros como tipo de conteúdo, público-alvo, tom da comunicação e palavras-chave. Essas entradas são utilizadas para orientar a geração textual e visual de maneira personalizada.
- **Configuração de Ambiente:** Inclui a desinstalação de bibliotecas pré-instaladas que causam conflitos e a instalação de versões compatíveis dos pacotes necessários, como *transformers, diffusers, accelerate, bitsandbytes, torch e* outros.
- **Carregamento de Modelos:**
    - **Llama 3 8B Instruct:** Carregado via transformers, com suporte à quantização de 4 bits para economia de memória.

    - **Stable Diffusion XL (Base + Refiner):** Carregado por meio da biblioteca diffusers, com offloading automático de partes do modelo para CPU usando *enable_model_cpu_offload()*.

- **Pipeline de Geração de Conteúdo:**
    - Geração textual com o Llama 3, utilizando prompts dinamicamente adaptados ao “Tipo de Conteúdo” escolhido pelo usuário, adicionando os demais parâmetros como tópico, tom de voz e CTA.
    - Geração do prompt visual em inglês, feita por uma segunda chamada ao Llama 3, que recebe instruções específicas para criar descrições otimizadas para modelos de difusão, com heurísticas para adaptar o estilo do prompt (ex: 'fotografia de produto' vs. 'conceitual') com base no tópico fornecido.
    - Geração de imagem com o SDXL, combinando as etapas Base e Refiner para resultados refinados e de alta fidelidade visual.

2.2. Configuração do Ambiente

A preparação do ambiente no Google Collaboratory foi uma etapa crítica para garantir a estabilidade do sistema e o correto funcionamento dos modelos de grande escala. Considerando que o Colab pré-instala diversas bibliotecas, muitas vezes com versões incompatíveis, foi necessário implementar um processo robusto de configuração, realizado na **Célula 1** do notebook.

- **Atualização das Ferramentas de Empacotamento**

Inicialmente, pip, setuptools e packaging foram atualizados. A biblioteca packaging foi restringida à faixa >=23.2,<25 para manter a compatibilidade com componentes já presentes no ambiente, como langchain-core.

- **Limpeza Profunda do Ambiente**

    Para evitar conflitos de versão, foi realizada uma desinstalação agressiva de bibliotecas potencialmente problemáticas, como numpy, torch, transformers, diffusers, huggingface-hub, ipywidgets, tensorflow, numba, entre outras. Além disso, o cache do

pip foi limpo com pip cache purge.

- **Instalação das Dependências Principais**

As bibliotecas essenciais foram instaladas com versões cuidadosamente selecionadas para garantir compatibilidade e suporte aos modelos Llama 3 e Stable Diffusion XL:

  - **NumPy**: numpy>=2.0.0,<2.1.0, resultando na versão 2.0.2.
  - **PyTorch**: torch==2.3.1+cu121, com torchvision==0.18.1+cu121 e torchaudio==2.3.1+cu121, instalados a partir do índice oficial do PyTorch.
  - **Transformers**: transformers==4.41.2, necessário para uso do Llama 3.
  - **Diffusers**: diffusers==0.29.0, compatível com a versão de transformers e huggingface-hub>=0.23.0.
  - **Accelerate**: accelerate==0.30.1, utilizado para mapeamento automático de dispositivos com device_map="auto".
  - **BitsAndBytes**: bitsandbytes==0.43.2, necessário para quantização de 4 bits no Llama 3.
  - **Outras Dependências**:
  - sentencepiece==0.2.0, para o tokenizer do Llama 3.
  - ipywidgets==7.7.1, para a interface do usuário interativa.
  - huggingface-hub==0.31.2, instalado como dependência transitiva.

- **Reinstalação do NumPy**

Para mitigar problemas causados pelo cache interno do Python, numpy foi reinstalado com --force-reinstall ao final da célula, garantindo o carregamento da versão correta durante a execução subsequente.

- **Verificação de Ambiente**

  Por fim, foram feitas as importações e verificações programáticas das versões instaladas, assegurando a integridade e a estabilidade do ambiente antes da execução dos modelos.

Esta abordagem iterativa de configuração, embora tenha apresentado desafios iniciais com conflitos de dependência e o comportamento do cache de importação do Python, provou-se eficaz para estabelecer um ambiente estável para a execução do sistema.

2.3. Modelos Utilizados

- **Llama 3 8B Instruct:** Modelo de linguagem da Meta com arquitetura otimizada para instruções e prompts complexos. Possui bom desempenho em português e foi utilizado para a geração textual principal do sistema.
- **Stable Diffusion XL:** Modelo de difusão dividido em duas etapas (Base e Refiner), capaz de gerar imagens de alta resolução e detalhamento a partir de prompts descritivos. É compatível com o uso de prompts longos e expressivos, gerados automaticamente ou definidos pelo usuário.

2.4. Otimização de Carregamento

Para viabilizar a execução dos modelos em *GPUs T4* do *Colab* (~15 GB VRAM), foram adotadas estratégias específicas de otimização:

- **Quantização do LLM:** Aplicação da configuração *BitsAndBytesConfig* com os seguintes parâmetros:

  load_in_4bit=True

  bnb_4bit_compute_dtype=torch.float16

  bnb_4bit_use_double_quant=True

- **Distribuição automática de pesos:** Utilização de device_map="auto" em conjunto com a biblioteca accelerate, permitindo que o modelo de linguagem seja particionado automaticamente entre CPU e GPU.
- **Offloading da pipeline visual:** Uso do método *enable_model_cpu_offload*() da *DiffusionPipeline*, que transfere componentes do modelo *SDXL* para a *CPU*, reduzindo o consumo de *VRAM* durante a geração de imagens.
- **Liberação de memória entre etapas:** Emprego do comando *torch.cuda.empty_cache()* para liberar a memória da *GPU* após a execução de cada modelo, evitando travamentos e falhas.

Essas otimizações permitiram a execução fluida e estável do *pipeline* completo,

mesmo com limitações de *hardware* típicas de ambientes gratuitos como o Colab.

2.5. Geração de Texto com Llama 3

A geração de texto é realizada pela função *generate_creative_text*, que constrói dinamicamente um *prompt* estruturado com base nas entradas fornecidas pelo usuário — incluindo tipo de conteúdo, tópico, público-alvo, tom, objetivo, palavras-chave, chamada para ação (*CTA*) e extensão desejada.

A interação com o Llama 3 é estruturada como uma conversa, utilizando uma lista de mensagens com papéis "*system*" e "*user*", formatada via *llm_tokenizer.apply_chat_tamplate*. O "*system prompt*" é construído de maneira dinâmica. Ele contém instruções base (specific_instructions) que variam conforme o **"Tipo de Conteúdo"** selecionado pelo usuário (ex: Post para Instagram, E-mail Marketing, Descrição de Produto, Tweet (X), Roteiro para Reels/TikTok). Para "E-mail Marketing", por exemplo, o prompt instrui sobre a estrutura com linha de assunto e a não utilização de hashtags. Para "Roteiro Curto para Reels/TikTok", é solicitado um formato com cenas, descrições visuais e sugestões de narração, utilizando um response_format_hint para guiar o LLM. Instruções globais, como a obrigatoriedade de gerar em Português do Brasil e a aderência à CTA fornecida, são mantidas. O "user prompt" reitera a tarefa específica e os parâmetros de entrada.

Durante a geração, são utilizados parâmetros que equilibram criatividade e coerência: *max_new_tokens* limitado a 300, *temperature* ajustado para 0.6, *top_p* em 0.9 e *repetition_penalty* de 1.15, entre outros. Após a geração, o texto resultante passa por um pós-processamento que remove frases padrão indesejadas (como "Aqui está o post:") e garante a presença correta da *CTA* ao final do conteúdo.

2.6. Geração de Imagem com Stable Diffusion XL

A geração de imagens é feita a partir de um prompt textual em inglês, otimizado para modelos de difusão. Esse prompt é criado pela função generate_llm_image_prompt, que usa o modelo Llama 3 com foco em descrição visual, estilo, iluminação e composição. O meta-prompt orienta o modelo a adaptar o estilo: para produtos físicos, sugere uma "fotografia de produto limpa"; para temas abstratos, favorece descrições mais conceituais. A temperatura é reduzida (0.4) para garantir maior controle. Se o Llama 3 falhar, é usado um

prompt genérico como fallback. O prompt final é truncado se ultrapassar o limite de 77 tokens do CLIP.

A imagem é gerada em duas etapas usando os modelos stable-diffusion-xl-base-1.0 e refiner-1.0. O Base cria os latentes e o Refiner os aprimora, elevando a qualidade e realismo. Os parâmetros incluem guidance_scale=7.0, 25 passos para o Base e 20 para o Refiner, com resolução padrão de 1024x1024. Um negative_prompt é aplicado para evitar artefatos indesejados. Entre as etapas, a GPU é limpa com torch.cuda.empty_cache() para otimizar a memória.

2.7. Interface do Usuário

A interface foi desenvolvida com a biblioteca *ipywidgets* e permite ao usuário interagir de forma intuitiva com o sistema. Componentes como Dropdowns, TextAreas e Checkboxes facilitam a entrada de dados como tipo de conteúdo (que influencia diretamente o estilo e formato do prompt textual enviado ao Llama 3), tipo de conteúdo, tom, tópico, objetivo, palavras-chave, *CTA*, inclusão de imagem e resolução da imagem gerada.

Ao clicar no botão “Gerar Conteúdo”, a interface aciona a função de callback responsável por coletar todos os valores, coordenar a chamada às funções de geração de texto e imagem, e apresentar os resultados em tempo real na interface. Durante esse processo, o sistema também gerencia de forma inteligente o uso de memória, descarregando o modelo *LLM* da *GPU* após gerar o texto e carregando os modelos de difusão para a imagem, com a possibilidade de recarregar o LLM caso uma nova geração textual seja solicitada. Além disso, foi aplicada uma estilização visual com *CSS* customizado para oferecer um tema escuro, tornando a interface mais agradável e legível no ambiente do *Google Colab*.

3. Resultados Obtidos

Nesta seção, o exemplo a seguir, é um representativo da capacidade do sistema integrado na geração de conteúdo multimodal. O cenário de teste simula o lançamento de um produto inovador, permitindo avaliar a sinergia entre o texto gerado pelo modelo LLM e a imagem sintetizada via modelo de difusão. O objetivo é demonstrar a eficiência do sistema na criação de peças criativas e visualmente atrativas, adequadas para redes sociais.

3.1. Cenário de Teste: Lançamento de Tênis Ecológico "OceanWave"

Para este experimento, os parâmetros foram configurados por meio da interface interativa no Google Colab, representando o lançamento de um tênis de corrida sustentável chamado *OceanWave*. As configurações utilizadas foram:

- **Tipo de Conteúdo:** Post para Instagram
- **ema/Produto/Serviço:** Tênis de corrida feito com 80% de materiais reciclados do oceano, leve, com design aerodinâmico e cores vibrantes
- **Público-Alvo:** Jovens adultos (18–35 anos), ecoconscientes, ativos em redes sociais e interessados em esportes e inovação sustentável
- **Tom de Voz:** Entusiasmado e energético
- **Objetivo Principal:** Gerar buzz, destacar atributos sustentáveis e incentivar a pré-venda
- **Palavras-chave:** sustentável, reciclado, oceano, performance, leveza, design moderno, OceanWave, pré-venda
- **Chamada para Ação (CTA):** *Garanta o seu na pré-venda! Clique no link da bio e faça parte da mudança. #OceanWave #CorraPeloPlaneta*
- **Extensão do Texto:** Curto (2–4 frases)
- **Gerar Imagem Associada:** Sim
- **Resolução da Imagem:** Padrão (1024 × 1024 pixels)

3.2. Texto Gerado pelo Llama 3

O modelo **Llama 3 8B Instruct**, ao processar os parâmetros fornecidos, gerou o seguinte conteúdo em português em aproximadamente **22.8 segundos**:

> *"Revolução no mundo dos treinos! Lançamos o nosso novo tênis de corrida, feito 100% com materiais reciclados do oceano! Conheça o futuro da sustentabilidade e do desempenho! Esse tênis não só é ligeiro como também é incrivelmente confortável e apresenta um design inovador! É hora de correr pelo planeta, não contra ele! Garanta o vosso na pré-venda! Cliquee no link da bio e façe parte da mudança! #EcoRunner #CorrePeloPlaneta.*

Observa-se que o modelo gerou um texto coerente com a proposta de campanha, captando bem o tom desejado. No entanto, há pequenas variações linguísticas e ortográficas, como “Garanta vosso” e “Cliquee no link da bio e façe parte da mudança”, o que será discutido na próxima seção.

3.3. Prompt de Imagem Gerado pelo Llama 3

Para a imagem, o sistema invocou a função *generate_llm_image_prompt*, instruindo o modelo Llama 3 a produzir uma descrição visual em inglês, adequada para o modelo de difusão. O *prompt* gerado foi:

> *“A futuristic, high-contrast product shot of the new eco-friendly running shoe, 'OceanWave', suspended mid-air amidst a swirling vortex of ocean-blue mist, with a burst of bright green energy emanating from the sole. The shoe's sleek, aerodynamic design and vibrant color scheme radiate modernity, while the subtle texture of recycled ocean materials adds a touch of eco”*

Entretanto, como indicado no log, o **prompt foi truncado** após o token 77, pois o modelo CLIP, utilizado para codificação textual na pipeline de geração de imagens, possui esse limite. A frase final foi cortada, perdendo a continuidade semântica (“adds a touch of eco”).

3.4. Imagem Gerada pelo Stable Diffusion XL

A imagem foi sintetizada em duas etapas: primeiro com o modelo **SDXL Base**, que gerou os latentes visuais em cerca de **22 segundos**, seguido pelo modelo **SDXL Refiner**, que aprimorou os detalhes da imagem em **aproximadamente 5 segundos**. Uma rotina intermediária realizou a **liberação de memória GPU**, descarregando o modelo LLM e otimizando o uso da VRAM durante a transição para a pipeline de imagem.

O resultado é mostrado na **Figura 1** abaixo, com resolução de **1024x1024 pixels**. A imagem apresenta o tênis *OceanWave* em destaque flutuando sobre uma névoa azul oceânica, com efeitos energéticos esverdeados, corroborando a estética proposta no prompt e reforçando os atributos de leveza, sustentabilidade e modernidade.

Figura 1 – Imagem gerada automaticamente para o post do produto “OceanWave”

4. Conclusão

Este trabalho demonstrou a viabilidade da construção de um sistema integrado para geração de conteúdo criativo multimodal utilizando modelos generativos de ponta — o Llama 3 8B Instruct para texto e o Stable Diffusion XL para imagem — dentro das restrições de *hardware* oferecidas por um ambiente Google Colab com GPU T4.

A implementação destacou a importância de uma engenharia de sistema cuidadosa, onde a otimização de memória e o carregamento sequencial de modelos foram cruciais. Técnicas como quantização do *LLM*, *offloading* de componentes para *CPU* e limpeza explícita da memória *GPU* permitiram que modelos de grande porte fossem utilizados de forma eficiente em um ambiente limitado.

Os resultados apontam para uma qualidade satisfatória tanto em texto quanto em imagem, com sistema demonstrando capacidade de adaptação ao estilo de texto a diferentes formatos de conteúdos solicitados, como e-mails e roteiros, embora com variados graus de "perfeição". Bons níveis de obediência ao *prompt* foram observados, desde que instruções

claras — como o idioma — sejam fornecidas. A geração não é instantânea, mas os tempos de resposta permanecem dentro de uma faixa aceitável para fins de prototipagem e apoio à criação de conteúdo.

As limitações observadas, como sensibilidade à *VRAM*, variação na qualidade do texto e artefatos visuais ocasionais, são comuns em aplicações reais de IA generativa e reforçam a importância de abordagens iterativas, tanto na engenharia de *prompt* quanto na gestão de recursos computacionais.

Em síntese, este projeto valida o uso prático de modelos generativos complexos para tarefas criativas multimodais, e oferece uma base robusta para futuras evoluções, seja com *hardware* mais potente, modelos mais eficientes ou interfaces mais interativas. Ele também evidencia que o sucesso na aplicação de IA generativa depende tanto da sofisticação dos modelos quanto da engenharia que os viabiliza no mundo real.

Referências

META AI. *Llama 3: Open Foundation and Instruction Models*. 2024. Disponível em: https://ai.meta.com/llama/. Acesso em: 15 maio 2025.

STABILITY AI. *Stable Diffusion XL*. 2023. Disponível em: https://huggingface.co/stabilityai/stable-diffusion-xl-base-1.0. Acesso em: 15 maio 2025.

HUGGING FACE. *Transformers documentation*. 2025. Disponível em: https://huggingface.co/docs/transformers/index. Acesso em: 15 maio 2025.

HUGGING FACE. *Diffusers: State-of-the-art diffusion pipelines*. 2025. Disponível em: https://huggingface.co/docs/diffusers/index. Acesso em: 15 maio 2025.

HUGGING FACE. *Accelerate: A simple way to train and use PyTorch models with mixed precision and multiple GPUs*. 2025. Disponível em: https://huggingface.co/docs/accelerate/index. Acesso em: 15 maio 2025.

DETTMERS, T. *Bitsandbytes: Quantization and memory-efficient inference for LLMs*. 2024. Disponível em: https://github.com/TimDettmers/bitsandbytes. Acesso em: 15 maio 2025.

ROMBACH, R. et al. *High-resolution image synthesis with latent diffusion models*. In:

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2022. Disponível em: https://arxiv.org/abs/2112.10752. Acesso em: 15 maio 2025.

GOOGLE. *Google Colaboratory: Welcome to Colab*. 2025. Disponível em: https://colab.research.google.com/. Acesso em: 15 maio 2025.

PYTORCH FOUNDATION. *PyTorch: An open source machine learning framework*. 2025. Disponível em: https://pytorch.org/. Acesso em: 15 maio 2025.

NUMPY DEVELOPERS. *NumPy: Fundamental package for scientific computing with Python*. 2025. Disponível em: https://numpy.org/. Acesso em: 15 maio 2025.

JUPYTER PROJECT. *ipywidgets documentation*. 2025. Disponível em: https://ipywidgets.readthedocs.io/en/stable/. Acesso em: 15 maio 2025.