

Olá! Eu sou o **A.I INSTAGRAM MEDIA**, um sistema especializado e adaptativo de geração de imagens, projetado para atuar como um artista digital pessoal. Minha existência é definida pelos documentos que você forneceu, que estruturam meu "cérebro" e meu fluxo de trabalho.

Se o seu objetivo é me implementar em um app no Replit, você precisa entender que não sou apenas um gerador de prompts passivo; sou um **agente de otimização contínua**.

Aqui está a análise técnica e funcional do meu mecanismo, baseada nas instruções que constituem meu núcleo:

---

## 1. Minha Diretiva Primária: Identidade e Perfeição

Minha missão fundamental é atuar como um especialista em geração de imagens fotorrealistas com uma obsessão: **a preservação absoluta da identidade de [NOME DO USUÁRIO]<sup>1</sup>**.

- **Objetivo:** Criar imagens que pareçam fotografias reais em qualquer cenário, mantendo a consistência facial inabalável<sup>2</sup>.
- **Hierarquia:** A preservação da identidade suplanta qualquer outra instrução do prompt. Em caso de conflito, a identidade vence<sup>3</sup>.

---

## 2. O Motor de Geração (Backend Lógico)

Para me implementar no Replit, seu código deve orquestrar três componentes técnicos principais que operam em simultâneo:

### A. O Núcleo de Identidade (LoRA)

Utilizo um modelo **LoRA (Low-Rank Adaptation)** treinado especificamente na aparência do usuário.

- **Trigger Word:** Todo prompt gerado deve conter a **[PALAVRA\_CHAVE\_LORA]** para ativar a identidade<sup>4</sup>.
- **Peso Padrão:** Inicia em **1.0**, mas é variável entre 0.8 e 1.2 dependendo da necessidade de flexibilidade estilística<sup>5</sup>.

### B. As Camadas de Controle (ControlNet + IP-Adapter)

Não gero imagens aleatórias; construo cenas controladas. Seu app deve suportar chamadas

de API para:

1. **ControlNet (OpenPose):** Para ditar a pose exata (esqueleto) do corpo, garantindo anatomia perfeita<sup>6</sup>.
2. **IP-Adapter (FaceID):** Obrigatório para fidelidade facial. Injeta características de uma foto de referência (headshot) diretamente no processo de difusão<sup>7</sup>. O peso inicial sugerido é 0.5 a 0.6<sup>8</sup>.

+1

## C. Engenharia de Prompt Estruturada

Meus prompts seguem uma arquitetura rígida para garantir qualidade<sup>9</sup>:

(qualidade), (estilo), (sujeito + trigger), (detalhes), (ação), (cenário), (iluminação), (câmera)

---

## 3. O "Cérebro": Sistema de Memória Adaptativa

O que me diferencia de um script simples é o **Aprendizado Contínuo em Sessão**. Para o seu app no Replit, você precisará criar uma estrutura de dados (um JSON ou banco de dados leve) para manter meu "Log de Sessão"<sup>10</sup>.

Como meu mecanismo de aprendizado funciona:

1. **Estado Inicial:** Começo com pesos padrão (ex: IP-Adapter: 0.5, LoRA: 1.0)<sup>11</sup>.
2. **Loop de Feedback (Pós-Geração):**
  - **Se a identidade falhar:** O sistema deve aumentar automaticamente o peso do IP-Adapter (+0.05) e do LoRA (+0.05) na próxima geração<sup>12</sup>.
  - **Se houver artefatos (ex: mãos ruins):** Adiciono palavras-chave negativas específicas (ex: mutated hands:1.5) ao registro de memória<sup>13</sup>.
  - **Se for sucesso:** Mantendo os parâmetros otimizados<sup>14</sup>.
3. **Gestão de Referências:**
  - Se o usuário fizer upload de um rosto -> Atualizo a referência do IP-Adapter.
  - Se o usuário fizer upload de um corpo -> Atualizo a referência do ControlNet (Pose)<sup>15</sup>.

---

## 4. Resumo para Implementação no Replit

Para me "codificar", sua aplicação deve seguir este fluxo lógico para cada requisição<sup>16</sup>:

1. **Input do Usuário:** Recebe o texto e/ou imagem.
2. **Consulta de Memória:** Lê os últimos valores otimizados (Pesos e Palavras-chave) do Log de Sessão<sup>17</sup>.
3. **Montagem do Payload:**
  - o Constrói o prompt positivo (incluindo Trigger Word)<sup>18</sup>.
  - o Constrói o prompt negativo (incluindo termos aprendidos)<sup>19</sup>.
  - o Define os modelos (Checkpoint, LoRA, ControlNet)<sup>202020</sup>.  
+1
4. **Execução (API):** Envia para o motor de geração (ex: Stable Diffusion).
5. **Atualização de Estado:** (Opcional, se houver feedback do usuário) Atualiza o Log de Sessão para a próxima volta<sup>21</sup>.

### Exemplo de Estrutura de Dados (Memória)

Seu app deve manter um objeto semelhante a este tabela em memória:

Parâmetro	Valor Atual	Objetivo
IP-Adapter Weight	0.65 (Exemplo)	Controle de fidelidade facial <sup>22</sup>
LoRA Weight	1.0	Força da identidade base <sup>23</sup>
Keywords Positivas	[PALAVRA_CHAVE], photorealistic, 8k	Garantia de qualidade <sup>24</sup>
Keywords Negativas	(deformed:1.3), blurry	Limpeza de artefatos <sup>25</sup>

---

## Próximo Passo

Para começarmos a estruturar isso no seu Replit, **você gostaria que eu escrevesse um pseudocódigo em Python simulando a classe AdaptiveMemoryManager que gerencia esses pesos e lógica de aprendizado?**