

Documento Simplificado — Projeto “Pontos Históricos do Recife”

1) Definição do projeto

- **O que é**: Aplicativo educacional gamificado que usa IA para reconhecer pontos históricos do Recife a partir de imagens e avaliar descrições textuais.
- **Como o usuário interage**:
 - Modo Foto: envia uma imagem para o modelo CNN identificar/comparar.
 - Modo Descrição: escreve um texto; o sistema avalia com NLP.
 - Modo Mistério: desafio de adivinhar o local.
- **Stack**: Python, Flask (backend), PyTorch (CNN), sklearn + NLTK (NLP), HTML/JS (frontend).

2) Problemática escolhida

- **Problema**: Dificuldade de engajar estudantes no aprendizado de história local com métodos tradicionais.
- **Solução**: Chatbot/jogo com:
 - Reconhecimento de imagem para identificar pontos históricos.
 - Avaliação automática de texto (semântica + keywords).
 - Gamificação (XP, níveis, streak) para motivar.

3) Datasets escolhidos

- **Imagens (estado atual no filesystem)**: 6 fotos presentes em ``data/recife_historic/casa_da_cultura/``. As demais pastas de classes existem mas estão vazias neste momento (12 classes definidas por pasta).
- **Descrições**: ``data/photo_descriptions.json`` (23 registros) com:
 - ``id``, ``image_path``, ``name``, ``official_description``, ``keywords``, ``difficulty``, ``points``.

4) Pipeline/ETL (simplificado)

- **Entrada (Foto)**:
 - Base64 → PIL → Resize/Normalize → CNN ``ImprovedCNN`` → classe + confiança.
- **Entrada (Texto)**:
 - Limpeza (lowercase, remoção de acentos/stopwords, lematização) → TF-IDF → similaridade cosseno.
 - Score final = 60% similaridade + 40% presença de keywords.

- ****Gamificação****:
- Ação → calcula pontos/XP → atualiza tentativas, acertos, streak e nível.
- Não há ETL externo; os dados são locais (imagens e JSON).

5) Modelos escolhidos

- ****Visão (Deep Learning)****:
- `ImprovedCNN` (PyTorch) treinada do zero.
- Estrutura: Conv (64,128,256,512) + AdaptiveAvgPool → FC (1024, 512, 256) → 12 classes.
- Acurácia reportada: ~96% (no conjunto do projeto).
- ****Texto (NLP)****:
- `TfidfVectorizer(max_features=1000)` + `cosine_similarity`.
- Pré-processamento com NLTK (stopwords PT) e lematização.
- Observação: O código também suporta comparar similaridade visual entre fotos do usuário e a foto do desafio.

6) Estratégias de avaliação (as usadas no projeto)

- ****CNN (modo foto)****:
- Similaridade alta quando as classes batem (mesmo local).
- Métrica prática: thresholds no score/semelhança para pontuação (≥ 0.8 excelente, ≥ 0.6 bom, ≥ 0.4 aceitável).
- ****NLP (modo descrição)****:
- Score por similaridade de TF-IDF + keywords.
- Regras de pontuação simples por faixas: $\geq 80\%$ excelente; $\geq 60\%$ muito bom; $\geq 40\%$ bom; $< 40\%$ tentativa.
- ****Gamificação****:
- XP, nível, streak, acurácia acumulada do jogador.

7) Arquitetura do projeto (figura simples)

flowchart LR

```
U[Frontend HTML/JS] -->|HTTP| F[Flask Backend]
```

subgraph Dados Locais

```
D1[data/recife_historic/ (imagens)]
```

```

D2[data/photo_descriptions.json]

end

F -->|POST /api/process_image| V[CNN ImprovedCNN]

F -->|POST /api/photo_game/submit_description| T[TF-IDF + Cosine]

F -->|POST /api/compare_visual_similarity| V

V --> F

T --> F

F --> G[Gamificação (XP, nível, streak)]

D1 --> V

D2 --> T

```

8) POC/Protótipo da aplicação

- ****Como rodar**:**
- `pip install -r requirements.txt`
- `python main.py`
- Acesse `http://localhost:5000` (interface) e `http://localhost:5000/chatbot` (chatbot).
- ****Endpoints principais**:**
- `POST /api/process_image`: processa imagem (modo foto) e retorna classe, confiança e texto explicativo simples.
- `POST /api/photo_game/submit_description`: avalia descrição (modo texto) e retorna score e `points_earned`.
- `POST /api/compare_visual_similarity`: compara foto do usuário com a foto do desafio e atribui XP por similaridade.
- `GET /api/player_stats/:player_id`: retorna estatísticas (XP, level, streak).
- ****Telas**:**
- `templates/index.html`: dashboard.
- `templates/chatbot.html`: chatbot e modos de jogo.
- ****O que demonstra**:**
- Identificação de 12 locais históricos via imagem.
- Avaliação de descrições em português com score.

- Evolução de XP e nível do jogador.

Dica para apresentação

- Mostre rapidamente: uma imagem → classe + confiança. Em seguida, uma descrição boa vs. ruim → diferença de score. Finalize com a barra de XP subindo.