# 📑 Relatório de Justificativas Técnicas

Projeto: Chatbot Gamificado sobre Pontos Históricos do Recife

Foco: Justificar escolhas de modelos, métodos e arquitetura

---

## 1) Por que treinar uma CNN do zero (ImprovedCNN) em vez de Transfer Learning?

* \*\*Contexto do domínio\*\*: As classes são locais históricos específicos do Recife (estilo visual, textura e composição próprios). Modelos pré-treinados (ImageNet) aprendem objetos genéricos (gatos, carros, etc.). Treinar do zero favorece features mais ajustadas ao domínio.
* \*\*Simplicidade e controle\*\*: Arquitetura enxuta, transparente e fácil de depurar. Útil para projeto educacional e para explicar cada estágio (Conv → Pool → Dense).
* \*\*Dataset pequeno mas estável\*\*: Embora o Transfer Learning fosse benéfico com poucos dados, a CNN simples alcançou boa performance (96%). O ganho adicional do TL não justificaria a complexidade neste estágio.
* \*\*Custo computacional\*\*: Arquitetura leve, treina rápido (~3 min), viável em máquinas comuns.

Conclusão: A CNN customizada equilibra aprendizagem específica do domínio, rapidez e clareza pedagógica.

---

## 2) Por que TF‑IDF + Similaridade do Cosseno para o Modo Descrição?

* \*\*Linguagem em português\*\*: Solução robusta e leve para PT-BR sem depender de modelos grandes.
* \*\*Transparência\*\*: Fácil explicar ao usuário o que influencia o score (palavras-chave e termos relevantes).
* \*\*Eficiência\*\*: Baixo custo computacional; responde em tempo real.
* \*\*Controle de critérios\*\*: Combinação 60% similaridade semântica + 40% keywords conecta fluência textual com precisão factual.

Conclusão: TF‑IDF + Cosseno fornece avaliação rápida, explicável e adequada ao contexto educacional.

---

## 3) Por que a métrica de similaridade visual baseada em “mesma classe + confiança”?

* \*\*Objetivo do produto\*\*: Não buscamos “similaridade estética” arbitrária, e sim se a foto é do mesmo local histórico.
* \*\*Robustez prática\*\*: Comparar classes previstas (e suas confianças) é estável com datasets pequenos, evitando ruído de métricas puramente pixel/feature space.
* \*\*Sinal pedagógico\*\*: A mensagem ao usuário fica clara (“mesmo local vs. diferente”), alinhando expectativa e pontuação.

Conclusão: A decisão centrada em classe+confiança alinha métrica técnica ao objetivo do jogo e ao tamanho do dataset.

---

## 4) Por que um sistema unificado de XP (sem moedas)?

* \*\*Redução de complexidade cognitiva\*\*: Um único indicador de progresso evita confusão e melhora UX.
* \*\*Traço educacional\*\*: O foco vira aprendizado e consistência, não economia de itens.
* \*\*Coerência\*\*: Todas as ações (foto, descrição) convergem em XP e nível, simplificando feedback e metas.

Conclusão: XP único torna a gamificação mais clara, motivadora e fácil de manter.

---

## 5) Por que a fórmula de nível `level = int((XP/100) \*\* 0.5) + 1`?

* \*\*Progressão suavemente desacelerada\*\*: Cresce rápido no início (engajamento) e estabiliza depois (retenção), sem exigir curvas complexas.
* \*\*Previsível e simples\*\*: Fácil de explicar e de ajustar (constantes e expoente).

Conclusão: Fórmula simples, estável e com curva de dificuldade intuitiva.

---

## 6) Por que a arquitetura web com Flask + Templates estáticos?

* \*\*Rapidez de prototipação\*\*: Flask é minimalista e perfeito para POCs e demos educacionais.
* \*\*Integração direta com PyTorch\*\*: Chamadas síncronas simples, sem overhead de serviços separados.
* \*\*Baixa barreira de entrada\*\*: HTML/CSS/JS vanilla facilitam manutenção e contribuições iniciais.

Conclusão: Stack leve, coerente com os objetivos de ensino e prototipação.

---

## 7) Por que o pipeline de NLP simples (regex, stopwords, lemmatization)?

* \*\*Resiliência em PT-BR\*\*: Pipeline robusto sem dependência de modelos pesados.
* \*\*Explicabilidade\*\*: Fácil mostrar como o texto é limpo e analisado.
* \*\*Custo/benefício\*\*: Entrega valor rápido com custo mínimo de infraestrutura.

Conclusão: O pipeline cobre bem a necessidade de avaliar descrições curtas e objetivas.

---

## 8) Alternativas consideradas e por que não agora

* \*\*Transfer Learning (ResNet/EfficientNet)\*\*: Melhor base com poucos dados, porém aumenta dependência e complexidade; pode ser um próximo passo quando ampliarmos o dataset.
* \*\*Embeddings semânticos (SBERT) para descrição\*\*: Melhor semântica, mas custo maior (modelo pesado) e explicabilidade menor para usuários iniciantes.
* \*\*Métricas visuais sofisticadas (LPIPS/CLIP-Similarity)\*\*: Úteis para estética/conteúdo amplo; nosso objetivo é “mesmo local histórico”, a abordagem por classe atende direto.
* \*\*Stack SPA (React/Vue)\*\*: Melhora UX sofisticada, porém aumenta complexidade do projeto sem necessidade imediata.

---

## 9) Critérios de qualidade que guiaram as escolhas

* Clareza e explicabilidade para o usuário final
* Baixa latência e baixo custo computacional
* Aderência ao objetivo pedagógico
* Simplicidade de manutenção e evolução
* Escalonamento gradual (rota para upgrades futuros)

---

## 10) Próximos passos recomendados

* Adotar Transfer Learning quando o dataset crescer
* Incluir embeddings semânticos leves para PT-BR (e.g., sentence-transformers-mini)
* Ampliar dataset com variação de ângulo/iluminação
* Criar testes automatizados de regressão de métricas

---

Em suma, as escolhas privilegiam simplicidade, explicabilidade e aderência ao objetivo educacional, garantindo um produto funcional, leve e com caminho claro para evoluções técnicas futuras.