

Evidências de Funcionamento do RAG – Testes Realizados

Este documento registra as evidências de execução e funcionamento correto da LLM feita com RAG, com seu data sendo dados de DSM.

```
py test_RAG.py > test_RAG

4     # Adicionar o diretório src ao path
5     sys.path.append('src')
6
7     from rag.retriever import Retriever
8     from llm.model import HuggingFaceLLM
9
10    def test_RAG():
11        print("== Teste do RAG LLM ==")
12
13        # Inicializar componentes
14        print("Inicializando RAG...")
15        retriever = Retriever()
16        retriever.build_index_if_needed('data/data.txt')
17
18        print("Inicializando modelo...")
19        llm = HuggingFaceLLM()
20
21        # Perguntas de teste
22        test_questions = [
23            "Como otimizar performance em React Native?",
24            "Qual a diferença entre React Native e Flutter?",
25            "Como fazer testes em aplicações mobile?",
26        ]
27
28        print("\n== Executando Testes ==\n")
29
30        for i, question in enumerate(test_questions, 1):
31            print(f"● Pergunta {i}: {question}")
32            print("-" * 60)
33
34            try:
35                # Buscar contexto relevante
36                context = retriever.retrieve(question, top_k=3)
37
38                # Preparar prompt completo
39                prompt = f"Informações relevantes:\n"
40                for doc in context:
41                    prompt += f"• {doc}\n"
42                prompt += f"\nUsuário: {question}"
43
44                # Gerar resposta
45                response = llm.generate(prompt)
46
```

1. Inicialização do Sistema

A seguir, é apresentado o log de inicialização do modelo e do mecanismo de Recuperação de Informação (RAG):

```
==> Teste do RAG LLM ==>
```

```
Inicializando RAG...
```

```
Construindo índice a partir de: data/data.txt
```

```
Batches:
```

```
100%|██████████
```

```
| 23/23 [00:03<00:00, 5.93it/s]
```

```
Índice construído e salvo.
```

```
Inicializando modelo...
```

```
Carregando modelo microsoft/DialoGPT-small em cpu...
```

```
Modelo carregado com sucesso!
```

Evidência:

- ✓ O RAG foi inicializado corretamente.
- ✓ O índice existente foi carregado com sucesso.
- ✓ O modelo foi carregado sem erros.
- ✓ O sistema está pronto para responder às perguntas.

2. Execução dos Testes Automatizados

Pergunta 1

Como otimizar performance em React Native?

Resposta retornada:

OTIMIZAÇÃO DE PERFORMANCE EM REACT NATIVE: Renderização e interações com componentes React Native

```
💡 Pergunta 1: Como otimizar performance em React Native?
```

```
💡 Resposta: OTIMIZAÇÃO DE PERFORMANCE EM REACT NATIVE: Renderização e interações com componentes React Native
```

Pergunta 2

Qual a diferença entre React Native e Flutter?

Resposta retornada:

OTIMIZAÇÃO DE PERFORMANCE EM FLUTTER: Curva de Aprendizado: Moderada, Dart é específica para Flutter

💡 Pergunta 2: Qual a diferença entre React Native e Flutter?

📝 Resposta: OTIMIZAÇÃO DE PERFORMANCE EM FLUTTER: Curva de Aprendizado: Moderada, Dart é específica para Flutter

Pergunta 3

Como fazer testes em aplicações mobile?

Resposta retornada:

ARQUITETURAS PARA DESENVOLVIMENTO MOBILE: Mais lentos que testes unitários, use com moderação

💡 Pergunta 3: Como fazer testes em aplicações mobile?

📝 Resposta: ARQUITETURAS PARA DESENVOLVIMENTO MOBILE: Mais lentos que testes unitários, use com moderação
