



Container para Experimentos main.py

Script para executar experimentos usando **main.py** em container Docker, aproveitando o container Ollama que já está rodando.

🎯 Pré-requisitos

Você já deve ter o Ollama rodando:

```
docker ps | grep ollama # Deve mostrar container ollama rodando
```

🚀 Uso Rápido

```
# 1. Testar se tudo está funcionando  
./test-main-container.sh  
  
# 2. Ver modelos disponíveis  
./run-main-experiment.sh --list-models  
  
# 3. Simular experimento (dry-run)  
./run-main-experiment.sh --dry-run  
  
# 4. Executar experimento específico  
./run-main-experiment.sh --single ollama_mistral_7b progressive_hint  
  
# 5. Executar experimento completo (108 classificações)  
./run-main-experiment.sh --full
```

—
PROF

⚡ Comandos Disponíveis

Experimentos

```
# Experimento completo (todos os 27 modelos × 4 técnicas)  
./run-main-experiment.sh --full  
  
# Experimento específico  
./run-main-experiment.sh --single <modelo> <técnica>  
  
# Exemplos de experimentos específicos:  
./run-main-experiment.sh --single ollama_mistral_7b progressive_hint  
./run-main-experiment.sh --single ollama_falcon3_10b self_hint  
./run-main-experiment.sh --single ollama_qwen2_7b hypothesis_testing
```

Utilitários

```
# Simular execução (ver comandos)
./run-main-experiment.sh --dry-run

# Listar modelos disponíveis
./run-main-experiment.sh --list-models

# Apenas construir a imagem
./run-main-experiment.sh --build

# Ajuda
./run-main-experiment.sh --help
```

Como Funciona

1. **Container Ollama:** Usa seu container Ollama existente via `--network host`
2. **Container Framework:** Cria um container temporário para cada execução
3. **Volumes:** Monta diretórios `data/`, `results/`, `logs/`, `config/`
4. **Execução:** Roda `python main.py` com os parâmetros corretos

III Estrutura do Experimento

Comando Executado

```
python main.py data/ --columns target --model <modelo> --technique
<técnica> --output xlsx
```

Modelos Testados (27 total)

PROF

- DeepSeek: `ollama_deepseek_15b`, `ollama_deepseek_r1_14b`, etc.
- Falcon: `ollama_falcon3_10b`, `ollama_falcon3_7b`, etc.
- Gemma: `ollama_gemma2_27b`, `ollama_gemma2_9b`, etc.
- Llama: `ollama_llama32_3b`, `ollama_llama33_70b`, etc.
- Mistral: `ollama_mistral_7b`, `ollama_mistral_large`, etc.
- Phi: `ollama_phi3_14b`, `ollama_phi3_mini`, etc.
- Qwen: `ollama_qwen2_7b`, `ollama_qwen2_5_32b`, etc.

Técnicas de Prompt (4 total)

- `progressive_hint` - Dicas progressivas
- `progressive_rectification` - Correção progressiva
- `self_hint` - Auto-sugestão
- `hypothesis_testing` - Teste de hipóteses

📁 Resultados

```
results/
├── classification_results_YYYYMMDD_HHMMSS.xlsx
├── progressive_hint/
│   ├── ollama_mistral_7b_results.xlsx
│   └── ...
└── progressive_rectification/
├── self_hint/
└── hypothesis_testing/

logs/
├── framework_YYYYMMDD.log
└── classification_errors.log
```

🐳 Detalhes Técnicos

Dockerfile

- **Base:** `python:3.11-slim`
- **Dependências:** Apenas essenciais (`curl`, `jq`, Python packages)
- **Otimizado:** Para execução rápida, sem ferramentas de compilação

Container

- **Network:** `--network host` (acessa Ollama localhost:11434)
- **Volumes:** Monta diretórios locais para persistir resultados
- **Temporário:** `--rm` remove container após execução
- **Variáveis:** `OLLAMA_BASE_URL=http://localhost:11434`

⚡ Exemplos Práticos

PROF

Teste Rápido

```
# Testar um modelo específico
./run-main-experiment.sh --single ollama_mistral_7b progressive_hint
```

Experimento Parcial

```
# Executar alguns modelos manualmente
./run-main-experiment.sh --single ollama_mistral_7b progressive_hint
./run-main-experiment.sh --single ollama_mistral_7b self_hint
./run-main-experiment.sh --single ollama_falcon3_10b progressive_hint
```

Experimento Completo

```
# Executar todos os 108 experimentos (4-8 horas)
./run-main-experiment.sh --full
```

🔍 Monitoramento

```
# Ver resultados em tempo real
ls -la results/

# Ver logs
tail -f logs/framework_*.log

# Verificar recursos
docker stats
```

🛠 Solução de Problemas

Problema: Ollama não encontrado

```
# Verificar se Ollama está rodando
docker ps | grep ollama

# Se não estiver, iniciar:
docker run -d -p 11434:11434 --name ollama ollama/ollama
```

Problema: Erro de rede

PROF

```
# Testar conectividade
curl http://localhost:11434/api/tags

# Verificar se porta está ocupada
netstat -tlnp | grep 11434
```

Problema: Falta de dados

```
# Verificar arquivos de dados
ls -la data/

# Deve ter pelo menos um arquivo .xlsx
```

Vantagens

- ✓ **Reutiliza Ollama:** Usa container Ollama existente
- ✓ **Container Simples:** Sem complexidade de Docker Compose
- ✓ **Execução Rápida:** Container temporário para cada experimento
- ✓ **Flexível:** Pode executar experimentos específicos ou completos
- ✓ **Logs Detalhados:** Acompanha progresso e erros
- ✓ **Fácil Debug:** Cada execução é independente

Execute `./test-main-container.sh` para verificar se tudo está funcionando! 