Quickstart

1 Guia de Início Rápido - FACIN_IA

1.1 ⚡ 5 Minutos para Começar

1.1.1 1. Clone/Fork o Repositório

```
# Clone (ou seu fork)
git clone https://github.com/gutpassos/FACIN_IA.git
cd FACIN_IA
```

1.1.2 2. Configure o Ambiente

```
# Criar ambiente virtual (recomendado)
python -m venv .venv

# Ativar
# Windows
.venv\Scripts\activate

# macOS/Linux
source .venv/bin/activate
```

```
# Instalar dependências
pip install -r requirements.txt
```

1.1.3 3. Configure Variáveis de Ambiente

Opção 1: Arquivo .env (Recomendado para produção)

```
# Copiar arquivo de exemplo
cp .env.example .env

# Editar com suas chaves
# GROQ_API_KEY=gsk_seu_groq_key_aqui
# OPENAI_API_KEY=sk-seu_openai_key_aqui
```

Opção 2: Interface Streamlit (Desenvolvimento)

As chaves podem ser inseridas diretamente na sidebar do Streamlit ao iniciar a aplicação.

⚠️ **Ambas as chaves são obrigatórias** (Groq e OpenAI)

1.1.4 4. Criar Banco de Dados

```
# Executar script de inicialização
python cria_db.py
```

1.1.5 5. Iniciar Aplicação

```
# Executar Streamlit
streamlit run app.py
```

✅ Aplicação rodando em `http://localhost:8501`

1.2 🤖 Como Usar

1.2.1 Exemplos de Consultas

Faça perguntas em português natural:

```
❓ "Quantos servidores estão ativos?"
❓ "Qual é a remuneração do Servidor 2?"
❓ "Quantos servidores ocupam o cargo de Assistente?"
❓ "Quantos servidores são da Secretaria da Saúde?"
❓ "Na Fazenda, quantos servidores ocupam o cargo de Assistente?"
❓ "Quantos servidores tiveram aumento?"
❓ "@groq Mostre todos os órgãos no banco de dados"
❓ "@openai Qual a folha total em 202401?"
```

O sistema usa **LangGraph** para: - ✅ Alternar automaticamente entre Groq (Llama3) e OpenAI (GPT) - ✅ Permitir seleção explícita com @groq ou @openai
- ✅ Executar ferramentas (query_folha_database) quando necessário - ✅ Consultar banco SQLite com validação de segurança - ✅ Fornecer resposta formatada em português

O sistema usa LangGraph para: - ✅ Rotear sua pergunta ao agente apropriado - ✅ Consultar banco de dados - ✅ Analisar dados - ✅ Fornecer resposta em português

---

1.3 📁 Arquivos Importantes

Arquivo	Descrição
`app.py`	Aplicação principal (Streamlit)
`cria_db.py`	Setup do banco de dados
`requirements.txt`	Dependências Python
`.env`	Variáveis de ambiente (local)
`spec.json`	Especificação do projeto
`docs/`	Documentação completa
`errors/`	Erros encontrados e soluções

1.4 🔐 Variáveis de Ambiente

```
# APIs (obrigatórias para LLMs)
OPENAI_API_KEY=sk-...
GROQ_API_KEY=gsk-...

# Opcional (monitoramento)
AGENTICOPS_API_KEY=...

# Banco de dados
DATABASE_PATH=folha_pagamento.db

# Desenvolvimento
DEBUG=true
PYTHONPATH=.
```

1.5 🧪 Verificações Rápidas

1.5.1 Testar Conexão com Banco

```
python -c "
import sqlite3
conn = sqlite3.connect('folha_pagamento.db')
cursor = conn.cursor()
cursor.execute('SELECT COUNT(*) FROM servidores')
print(f'Total de servidores: {cursor.fetchone()[0]}')
conn.close()
"
```

1.5.2 Validar Formatação

```
black . --check --line-length=100
```

1.5.3 Executar Testes

```
pytest -v
```

1.6 🐛 Troubleshooting

1.6.1 ModuleNotFoundError: No module named 'streamlit'

```
# Reinstalar dependências
pip install --upgrade -r requirements.txt
```

1.6.2 OPENAI_API_KEY not found

```
# Verificar .env
cat .env

# Adicionar chave
echo "OPENAI_API_KEY=sk-..." >> .env
```

1.6.3 Erro ao carregar banco de dados

```
# Recriar banco
rm folha_pagamento.db
python cria_db.py
```

1.6.4 Porta 8501 já em uso

```
# Usar porta diferente
streamlit run app.py --server.port=8502
```

1.7 📚 Próximos Passos

1. **Ler documentação**
   - README.md - Overview
   - ARCHITECTURE.md - Como funciona
   - API.md - Referência técnica
2. **Explorar código**
   - Entender estrutura em `app.py`
   - Ver agentes em LangGraph
   - Examinar ferramentas
3. **Desenvolver**
   - Criar seu próprio agente
   - Adicionar ferramentas
   - Melhorar respostas
4. **Contribuir**
   - Ver CONTRIBUTING.md
   - Submeter PR
   - Reportar bugs

1.8 🚀 Dicas Pro

1.8.1 VSCode + GitHub Copilot

```
{
    "github.copilot.enable": {"python": true},
    "python.defaultInterpreterPath": "${workspaceFolder}/.venv/Scripts/python.exe"
}
```

1.8.2 Debugging

```
# No app.py
import streamlit as st

# Visualizar estado
st.write(state)
st.json(messages)
```

1.8.3 Performance

```
# Use @st.cache_data para caching
@st.cache_data(ttl=3600)
def expensive_query():
    return execute_database_query(...)
```

1.9 📞 Precisa de Ajuda?

1. Verifique errors/ERRORS_LOG.md
2. Abra uma Issue no GitHub
3. Email: gut.passos@gmail.com

1.10 ✅ Checklist de Setup

- [ ] Python 3.10+ instalado
- [ ] Repositório clonado
- [ ] Ambiente virtual criado
- [ ] Dependências instaladas
- [ ] .env configurado
- [ ] Banco de dados criado
- [ ] Aplicação rodando em localhost:8501
- [ ] Teste uma consulta básica

Pronto para começar a desenvolver? 🚀

Versão: 1.0.0
Data: 27/02/2026