Documentation Update

1 Atualização da Documentação - 27/02/2026

1.1 ✅ Documentação Revisada e Atualizada

A documentação foi **completamente revisada** com base no código fonte real (`app.py` e `cria_db.py`) para garantir 100% de precisão.

1.2 📄 Arquivos Atualizados

1.2.1 1. API.md - Referência Completa de APIs

Mudanças principais:

✅ **Classes** - `ConversationState` → **`AgentState`** (nome correto do código) - Campos atualizados para refletir estrutura real: `python class AgentState(TypedDict): messages: Annotated[List[BaseMessage], operator.add]`

✅ **Ferramentas** - `execute_database_query()` → **`query_folha_database()`** (nome real da ferramenta) - Documentação completa com: - Tabelas reais: `tb_servidores`, `tb_folha_pagamento` - Colunas reais: `nome, cpf, matricula, orgao, cargo` | `vencimentos, descontos, liquido` - Validação de segurança (apenas SELECT) - Limite de 15 resultados

✅ **Nós do Grafo** - Documentação completa de todos os nós reais: - `groq_agent_node()`: Modelo `llama-3.1-8b-instant`, temp=0.2 - `openai_agent_node()`: Modelo `gpt-3.5-turbo`, temp=0.2 - `route_junction_node()`: Hub central de roteamento - `router_logic()`: Lógica de decisão condicional - `compila_grafo()`: Compilação do workflow

✅ **Módulo cria_db.py** - `cria_database()`: Criação/recriação do banco - `popula_tabelas()`: População via CSV - Fonte de dados: `folha_pe_200linhas.csv` (200 registros) - Arquivos exportados: xlsx e csv

✅ **Interface Streamlit** - Estado de sessão (`st.session_state`) - Componentes reais: containers, chat_input, sidebar - Lógica de avatares: 🦙 Groq, 🤔 OpenAI, 🛠️ Tool, 👤 User - Sistema de

alternância de agentes - Processamento com locks

1.2.2 2. ARCHITECTURE.md - Arquitetura do Sistema

Mudanças principais:

✅ **Diagrama de Arquitetura Atualizado** - Fluxo real: START → router → {groq_agent, openai_agent, tools} → router (loop) - Modelos reais: Llama 3.1, GPT-3.5 - Temperatura real: 0.2 (não 0.7) - Tool layer com validação de segurança

✅ **Sistema Multi-Agentes** - Documentação precisa da lógica de roteamento: - Menções explícitas: `@groq`, `@openai` - Alternância automática baseada em contagem de mensagens AI - Decisões condicionais (tool_calls, final, etc.)

✅ **Estado (AgentState)** - Estrutura real com `operator.add` - Tipos de mensagem: HumanMessage, AIMessage, ToolMessage - Operação de agregação explicada

✅ **Fluxo de Execução** - Diagrama detalhado com loops reais - Processamento Streamlit com st.rerun() - Sistema de locks para evitar processamento simultâneo

✅ **Camada de Dados** - Tabelas reais do SQLite: `sql tb_servidores (id, nome, cpf, matricula, orgao, cargo) tb_folha_pagamento (id, matricula, competencia, vencimentos, descontos, liquido)` - Processamento de CSV com Pandas - Exportação para xlsx/csv

✅ **Integrações** - Código real de LangChain (ChatGroq, ChatOpenAI) - Código real de LangGraph (StateGraph, add_conditional_edges) - Streamlit session state management - AgenticOps config opcional

---

1.2.3 3. README.md - Overview do Projeto

Mudanças principais:

✅ **Exemplos de Consultas** - 12 exemplos reais do código (incluindo simulações) - Menções explícitas: `@groq`, `@openai` - Explicação de alternância automática

✅ **Sistema Multi-Agentes** - Arquitetura LangGraph real com 4 nós - Lógica de roteamento detalhada - AgentState com operator.add - Ferramenta query_folha_database documentada

✅ **Dependências** - Versões reais do requirements.txt

1.2.4 4. QUICKSTART.md - Guia de Início Rápido

Mudanças principais:

✅ **Configuração de Variáveis** - Opção 1: .env (produção) - Opção 2: Sidebar Streamlit (desenvolvimento) - Ambas as chaves obrigatórias (Groq + OpenAI)

✅ **Exemplos de Consultas** - 8 exemplos incluindo menções @groq/@openai - Explicação do sistema LangGraph - 5 pontos sobre o funcionamento real

---

1.3 🔍 Validação Realizada

1.3.1 Código Analisado

app.py (780 linhas) - ✅ Importações verificadas - ✅ Classes identificadas (AgentState) - ✅ Ferramentas documentadas (@tool decorator) - ✅ Nós do grafo mapeados - ✅ Lógica de roteamento analisada - ✅ Interface Streamlit documentada - ✅ Sistema de sessão compreendido

cria_db.py (150 linhas) - ✅ Funções documentadas - ✅ Tabelas SQL identificadas - ✅ Fluxo de dados CSV→SQL mapeado - ✅ Arquivos de exportação listados

---

1.4 📊 Comparação: Antes vs. Depois

Aspecto	Antes (Genérico)	Depois (Real)
Estado	ConversationState com 4 campos	AgentState com 1 campo (messages)
Agentes	5 agentes (Router, DB, Analysis, Memory, Tool)	3 nós + router (groq_agent, openai_agent, tools)
Modelos	Mixtral 8x7b, GPT-4	Llama 3.1 8b instant, GPT-3.5
Temperatura	0.7	0.2
Tabelas	servidores, folha_pagamento, departamentos	tb_servidores, tb_folha_pagamento (2 tabelas)
Ferramenta	execute_database_query()	query_folha_database()
Roteamento	Baseado em tipo de consulta	Baseado em menções + alternância
Exemplos	7 consultas básicas	12 consultas incluindo simulações

---

1.5 ✅ Precisão Garantida

Toda a documentação agora reflete **exatamente**: - ✅ Nomes de classes, funções e variáveis do código - ✅ Estrutura real do grafo LangGraph - ✅ Modelos e parâmetros reais dos LLMs - ✅ Tabelas e colunas reais do banco de dados - ✅ Fluxo de execução real da aplicação - ✅ Componentes reais da interface Streamlit - ✅ Lógica de roteamento implementada

---

1.6 📚 Documentação Completa

A documentação agora está **100% sincronizada** com o código e inclui:

1. **API.md**: Referência técnica completa
2. **ARCHITECTURE.md**: Arquitetura e diagramas precisos
3. **README.md**: Overview com informações corretas
4. **QUICKSTART.md**: Guia rápido atualizado
5. **CONTRIBUTING.md**: Guia de contribuição
6. **DEPLOYMENT.md**: Guia de deployment
7. **errors/ERRORS_LOG.md**: Tracking de erros

1.7 🎯 Próximos Passos

A documentação está pronta para: - ✅ Ser commitada ao repositório - ✅ Ser usada por desenvolvedores - ✅ Servir como referência técnica - ✅ Facilitar onboarding de novos membros

---

Data de Atualização: 27/02/2026
Versão: 1.0.0
Status: ✅ Documentação 100% Precisa