Documentação Detalhada do App - Quantum Finance Score de Crédito

# 1. Introdução

O aplicativo em Streamlit foi desenvolvido como parte do projeto end-to-end em MLOps da Quantum Finance. O objetivo deste componente é disponibilizar uma interface gráfica simples, clara e escalável para consumo da API de predição de score de crédito. A aplicação tem como público-alvo analistas internos e potenciais parceiros externos, permitindo a classificação de risco de crédito com base em dados cadastrais, financeiros e comportamentais.

# 2. Arquitetura da Solução

O app foi construído em Streamlit e consome a API exposta via AWS API Gateway. O fluxo principal é o seguinte:  
1. Usuário insere os dados no Streamlit.  
2. Inputs categóricos são normalizados para os valores canônicos usados no modelo.  
3. Payload é montado no formato JSON esperado pela API.  
4. Requisição POST é enviada para o endpoint seguro.  
5. A resposta é exibida no app em formato de card colorido, indicando a classe prevista e a versão do modelo.

# 3. Estrutura do Projeto

Estrutura do repositório quantum-finance-app-credit-score:  
  
quantum-finance-app-credit-score/  
│  
├── app.py # código principal do aplicativo  
├── requirements.txt # dependências  
└── .streamlit/  
 └── secrets.toml # configuração de endpoint e chave da API  
  
Dependências mínimas:  
- streamlit==1.30.0  
- requests==2.31.0

# 4. Funcionalidades do App

- Apresentação inicial institucional da Quantum Finance.  
- Entrada de dados numéricos (idade, renda anual, salário mensal, etc.).  
- Campos financeiros exibidos em dólar ($) e percentuais (%).  
- Calculadora auxiliar para cálculo de meses entre duas datas (opcional, formato aaaa/mm/dd).  
- Entrada de dados categóricos (ocupação, comportamento de pagamento, etc.), exibidos com tradução em português.  
- Normalização de inputs categóricos para valores canônicos.  
- Montagem do payload JSON.  
- Chamada à API via POST.  
- Exibição da resposta em cards coloridos: vermelho (Poor), amarelo (Standard), verde (Good).

# 5. Estrutura do Código (app.py)

O código foi estruturado em blocos bem definidos:  
  
- Definição da função get\_prediction: realiza chamada à API e exibe o resultado em card colorido.  
- Entradas numéricas coletadas via st.number\_input.  
- Calculadora auxiliar implementada com st.text\_input e cálculo de meses.  
- Entradas categóricas coletadas via st.selectbox e st.radio.  
- Normalização de categorias via dicionários de mapeamento.  
- Montagem do payload no formato esperado.  
- Botão 'Gerar Predição' com spinner de carregamento.

Exemplo de função de predição utilizada no app:

def get\_prediction(payload):  
 endpoint = st.secrets["API-ENDPOINT"]  
 headers = {  
 "Content-Type": "application/json",  
 "x-api-key": st.secrets["API-KEY"]  
 }  
  
 response = requests.post(endpoint, headers=headers, data=json.dumps(payload))  
  
 if response.status\_code == 200:  
 result = response.json()  
 classes = {0: ("Poor", "vermelho"), 1: ("Standard", "amarelo"), 2: ("Good", "verde")}  
 st.markdown(f"Classe prevista: {classes[result.get('prediction')][0]}")  
 st.caption(f"Versão do modelo: {result.get('version')}")  
 else:  
 st.error("Erro ao obter a previsão.")

# 6. Segurança

- Segredos não são versionados (uso de .streamlit/secrets.toml).  
- API Key obrigatória para consumo da API.  
- Em produção (Streamlit Cloud), secrets configurados via painel seguro.

# 7. CI/CD do App

Pipeline configurado no Streamlit Cloud:  
  
- Deploy inicial realizado a partir do repositório GitHub.  
- Branch monitorada: main.  
- Cada push/merge dispara rebuild automático (CI) e entrega contínua (CD).  
- Secrets configurados diretamente no painel (API\_ENDPOINT e API\_KEY).  
- URL de produção: https://quantum-finance-app-credit-score.streamlit.app/

# 8. Boas Práticas e Decisões

- Interface simplificada e traduzida.  
- Normalização categórica para compatibilidade com modelo.  
- Uso de cards coloridos para clareza no resultado.  
- Estrutura enxuta, contendo apenas o necessário para o frontend.  
- Código validado com pylint.

# 9. Valor Entregue

- App funcional consumindo a API em produção.  
- Integração ponta a ponta concluída.  
- Ferramenta disponível para analistas de crédito.

# 10. Próximos Passos

- Melhorar experiência do usuário (ajuda, tooltips, validações).  
- Adicionar histórico de predições no app (via logs de S3).  
- Internacionalização (versão em inglês).  
- Integração com dashboards corporativos para uso avançado.

# 11. Conclusão

O aplicativo em Streamlit consolida a entrega de MLOps, transformando o modelo de classificação de score de crédito em uma ferramenta prática, segura e escalável. A arquitetura modular garante facilidade de manutenção e evolução futura.