Table of Contents

# 🛠️ CÓRTEX BI - Guia Completo de Instalação, Configuração e Administração

**Versão:** 2.0  
**Data:** Outubro 2025  
**Desenvolvido em parceria com:** Manus AI

## 📋 Índice

1. [Pré-requisitos](#pré-requisitos)
2. [Instalação](#instalação)
3. [Configuração Inicial](#configuração-inicial)
4. [Inicialização do Sistema](#inicialização-do-sistema)
5. [Configuração Avançada](#configuração-avançada)
6. [Administração do Sistema](#administração-do-sistema)
7. [Integração Microsoft 365](#integração-microsoft-365)
8. [Monitoramento e Manutenção](#monitoramento-e-manutenção)
9. [Troubleshooting](#troubleshooting)
10. [Segurança e Backup](#segurança-e-backup)

## 🔧 Pré-requisitos

### Requisitos de Sistema Operacional

O CÓRTEX BI pode ser instalado nos seguintes sistemas operacionais:

**Linux:** - Ubuntu 20.04 LTS ou superior - CentOS 8 ou superior - Red Hat Enterprise Linux (RHEL) 8 ou superior - Debian 10 ou superior

**Windows:** - Windows 10 (versão 1909 ou superior) - Windows 11 - Windows Server 2019 ou superior

**macOS:** - macOS 10.15 (Catalina) ou superior - macOS 11 (Big Sur) ou superior - macOS 12 (Monterey) ou superior

### Requisitos de Hardware

**Mínimos:** - Processador: Dual-core 2.0 GHz - Memória RAM: 4 GB - Espaço em disco: 10 GB livres - Conexão de rede: 10 Mbps

**Recomendados:** - Processador: Quad-core 3.0 GHz ou superior - Memória RAM: 8 GB ou superior - Espaço em disco: 20 GB livres (SSD recomendado) - Conexão de rede: 100 Mbps ou superior

### Software Necessário

**Obrigatório:** - **Python 3.8 ou superior** (recomendado Python 3.10) - **pip** (gerenciador de pacotes Python) - **Git** (para clonar o repositório)

**Opcional (para funcionalidades específicas):** - **SQL Server** - Para conexão com banco de dados corporativo - **Power BI Desktop** - Para integração com datasets do Power BI - **Microsoft 365** - Para integração com Copilot Studio, Teams e SharePoint

### Verificação de Pré-requisitos

Execute os seguintes comandos para verificar se os pré-requisitos estão instalados:

# Verificar versão do Python  
python3 --version  
# Deve retornar: Python 3.8.x ou superior  
  
# Verificar pip  
pip3 --version  
  
# Verificar Git  
git --version  
  
# Verificar espaço em disco (Linux/macOS)  
df -h  
  
# Verificar espaço em disco (Windows)  
wmic logicaldisk get size,freespace,caption

## 📥 Instalação

### Método 1: Instalação Automática (Recomendado)

#### Linux / macOS

# 1. Clonar o repositório  
git clone https://github.com/Rimkus85/cortex-bi.git  
cd cortex-bi  
  
# 2. Dar permissão de execução ao script  
chmod +x scripts/install.sh  
  
# 3. Executar instalação automática  
./scripts/install.sh

O script automático irá: - ✅ Verificar versão do Python - ✅ Criar ambiente virtual Python - ✅ Instalar todas as dependências - ✅ Criar diretórios necessários - ✅ Configurar permissões - ✅ Validar a instalação

#### Windows

# 1. Abrir PowerShell como Administrador  
  
# 2. Clonar o repositório  
git clone https://github.com/Rimkus85/cortex-bi.git  
cd cortex-bi  
  
# 3. Executar instalação automática  
.\scripts\install.bat

O script irá: - ✅ Verificar instalação do Python - ✅ Criar ambiente virtual - ✅ Instalar dependências - ✅ Configurar variáveis de ambiente - ✅ Criar atalhos para inicialização

### Método 2: Instalação Manual

Se preferir controle total sobre o processo:

# 1. Clonar o repositório  
git clone https://github.com/Rimkus85/cortex-bi.git  
cd cortex-bi  
  
# 2. Criar ambiente virtual Python  
python3 -m venv venv  
  
# 3. Ativar ambiente virtual  
# Linux/macOS:  
source venv/bin/activate  
# Windows:  
venv\Scripts\activate  
  
# 4. Atualizar pip  
pip install --upgrade pip  
  
# 5. Instalar dependências  
pip install -r requirements.txt  
  
# 6. Criar diretórios necessários  
mkdir -p data output logs database templates  
  
# 7. Verificar instalação  
python src/main\_ai.py --help

### Verificação da Instalação

Após a instalação, execute o script de verificação:

# Linux/macOS  
python3 scripts/verificar\_integracao\_copilot.py --url http://localhost:5000  
  
# Windows  
python scripts\verificar\_integracao\_copilot.py --url http://localhost:5000

## ⚙️ Configuração Inicial

### Arquivo de Configuração (.env)

O CÓRTEX BI utiliza variáveis de ambiente para configuração. Crie um arquivo .env na raiz do projeto:

# Copiar template de exemplo  
cp .env.example .env  
  
# Editar configurações  
nano .env # Linux/macOS  
notepad .env # Windows

### Variáveis de Ambiente Principais

# ===== SERVIDOR =====  
HOST=0.0.0.0  
PORT=5000  
DEBUG=False  
WORKERS=4  
  
# ===== SEGURANÇA =====  
API\_KEY=cHKALRHOHMpDnoFGGuHimNigg3HugUrq  
SECRET\_KEY=sua-chave-secreta-aqui  
CORS\_ORIGINS=https://copilotstudio.microsoft.com,https://teams.microsoft.com  
  
# ===== BANCO DE DADOS =====  
DB\_TYPE=sqlite # ou 'sqlserver', 'postgresql'  
DB\_HOST=localhost  
DB\_PORT=1433  
DB\_NAME=cortexbi  
DB\_USER=seu\_usuario  
DB\_PASSWORD=sua\_senha  
  
# ===== SQL SERVER (Opcional) =====  
SQL\_SERVER\_HOST=seu-servidor.database.windows.net  
SQL\_SERVER\_DATABASE=seu\_banco  
SQL\_SERVER\_USERNAME=seu\_usuario  
SQL\_SERVER\_PASSWORD=sua\_senha  
SQL\_SERVER\_DRIVER=ODBC Driver 17 for SQL Server  
  
# ===== POWER BI (Opcional) =====  
POWERBI\_CLIENT\_ID=seu-client-id  
POWERBI\_CLIENT\_SECRET=seu-client-secret  
POWERBI\_TENANT\_ID=seu-tenant-id  
  
# ===== SHAREPOINT (Opcional) =====  
SHAREPOINT\_SITE\_URL=https://suaempresa.sharepoint.com/sites/seu-site  
SHAREPOINT\_CLIENT\_ID=seu-client-id  
SHAREPOINT\_CLIENT\_SECRET=seu-client-secret  
SHAREPOINT\_TENANT\_ID=seu-tenant-id  
SHAREPOINT\_UPLOAD\_PATH=/Documentos/CORTEX\_BI  
  
# ===== EMAIL (Opcional) =====  
SMTP\_HOST=smtp.office365.com  
SMTP\_PORT=587  
SMTP\_USER=seu-email@empresa.com  
SMTP\_PASSWORD=sua-senha  
SMTP\_FROM=cortexbi@empresa.com  
  
# ===== LOGS =====  
LOG\_LEVEL=INFO # DEBUG, INFO, WARNING, ERROR, CRITICAL  
LOG\_FILE=logs/cortexbi.log  
LOG\_MAX\_SIZE=10485760 # 10MB  
LOG\_BACKUP\_COUNT=5  
  
# ===== CACHE =====  
CACHE\_ENABLED=True  
CACHE\_TTL=1800 # 30 minutos em segundos  
CACHE\_MAX\_SIZE=1000  
  
# ===== IA E ML =====  
ML\_MODEL\_PATH=models/  
NLP\_MODEL=pt\_core\_news\_lg  
FEEDBACK\_THRESHOLD=0.7  
RECOMMENDATION\_MIN\_CONFIDENCE=0.6

### Configuração de Diretórios

Crie a estrutura de diretórios necessária:

# Estrutura de diretórios  
mkdir -p data/{raw,processed,archive}  
mkdir -p output/{reports,presentations,exports}  
mkdir -p logs  
mkdir -p database  
mkdir -p templates/{pptx,excel,pdf}  
mkdir -p models  
mkdir -p cache

### Permissões (Linux/macOS)

Configure as permissões adequadas:

# Dar permissão de execução aos scripts  
chmod +x scripts/\*.sh  
  
# Configurar permissões dos diretórios  
chmod 755 data output logs database templates models cache  
chmod 644 .env

## 🚀 Inicialização do Sistema

### Método 1: Scripts Automáticos (Recomendado)

#### Linux / macOS

# Iniciar o servidor  
./scripts/start\_ai.sh  
  
# O script irá:  
# 1. Ativar ambiente virtual  
# 2. Verificar dependências  
# 3. Iniciar servidor FastAPI  
# 4. Abrir documentação no navegador

#### Windows

# Iniciar o servidor  
.\scripts\start\_ai.bat

### Método 2: Inicialização Manual

# 1. Ativar ambiente virtual  
source venv/bin/activate # Linux/macOS  
venv\Scripts\activate # Windows  
  
# 2. Iniciar servidor  
python src/main\_ai.py  
  
# 3. Servidor estará disponível em:  
# http://localhost:5000

### Verificação do Sistema

Após iniciar, verifique se o sistema está funcionando:

# 1. Health Check  
curl http://localhost:5000/health  
  
# Resposta esperada:  
{  
 "status": "healthy",  
 "timestamp": "2025-10-14T18:30:00.000000",  
 "services": {  
 "data\_loader": "active",  
 "analytics\_engine": "active",  
 "pptx\_generator": "active",  
 "nlp\_engine": "active",  
 "ml\_engine": "active",  
 "recommendation\_engine": "active",  
 "feedback\_system": "active",  
 "admin\_system": "active"  
 }  
}

### Acessar Interfaces

**Documentação Interativa (Swagger UI):**

http://localhost:5000/docs

**Dashboard Administrativo:**

http://localhost:5000/admin/admin\_dashboard.html

**API Health Check:**

http://localhost:5000/health

## 🔧 Configuração Avançada

### Configuração de Banco de Dados SQL Server

Para conectar o CÓRTEX BI a um banco de dados SQL Server corporativo:

**1. Instalar driver ODBC:**

# Ubuntu/Debian  
curl https://packages.microsoft.com/keys/microsoft.asc | sudo apt-key add -  
curl https://packages.microsoft.com/config/ubuntu/20.04/prod.list | sudo tee /etc/apt/sources.list.d/mssql-release.list  
sudo apt-get update  
sudo ACCEPT\_EULA=Y apt-get install -y msodbcsql17  
  
# Windows  
# Baixar e instalar de: https://docs.microsoft.com/en-us/sql/connect/odbc/download-odbc-driver-for-sql-server

**2. Configurar conexão no .env:**

DB\_TYPE=sqlserver  
SQL\_SERVER\_HOST=seu-servidor.database.windows.net  
SQL\_SERVER\_DATABASE=seu\_banco  
SQL\_SERVER\_USERNAME=seu\_usuario  
SQL\_SERVER\_PASSWORD=sua\_senha  
SQL\_SERVER\_DRIVER=ODBC Driver 17 for SQL Server

**3. Testar conexão:**

python3 -c "  
from src.agents.data\_loader import DataLoader  
loader = DataLoader()  
loader.test\_sql\_connection()  
"

### Configuração de Templates PPTX

Personalize os templates de apresentação:

**1. Criar template personalizado:**

# Copiar template base  
cp templates/template\_relatorio.pptx templates/meu\_template.pptx  
  
# Editar no PowerPoint adicionando placeholders:  
# {{titulo\_principal}}  
# {{data\_geracao}}  
# {{total\_vendas}}  
# {{crescimento\_percentual}}  
# {{insights\_principais}}

**2. Registrar template no sistema:**

# Via API  
curl -X POST "http://localhost:5000/admin/template/register" \  
 -H "Content-Type: application/json" \  
 -H "X-API-Key: sua-api-key" \  
 -d '{  
 "template\_name": "meu\_template.pptx",  
 "description": "Template personalizado para relatórios executivos",  
 "placeholders": [  
 "titulo\_principal",  
 "data\_geracao",  
 "total\_vendas",  
 "crescimento\_percentual",  
 "insights\_principais"  
 ]  
 }'

### Configuração de Cache

Otimize o desempenho configurando o cache:

# No arquivo .env  
CACHE\_ENABLED=True  
CACHE\_TTL=1800 # 30 minutos  
CACHE\_MAX\_SIZE=1000 # Máximo de itens no cache

### Configuração de Logs

Personalize o sistema de logs:

# No arquivo .env  
LOG\_LEVEL=INFO  
LOG\_FILE=logs/cortexbi.log  
LOG\_MAX\_SIZE=10485760 # 10MB  
LOG\_BACKUP\_COUNT=5  
  
# Logs separados por tipo  
LOG\_AI\_INTERACTIONS=logs/ai\_interactions.log  
LOG\_ERRORS=logs/errors.log  
LOG\_AUDIT=logs/audit.log

## 👨‍💼 Administração do Sistema

### Dashboard Administrativo

Acesse o dashboard em: http://localhost:5000/admin/admin\_dashboard.html

**Funcionalidades disponíveis:**

**1. Métricas em Tempo Real:** - Número de análises executadas - Usuários ativos - Tempo médio de resposta - Taxa de sucesso das operações - Uso de recursos (CPU, memória)

**2. Gerenciamento de Templates:** - Visualizar templates disponíveis - Upload de novos templates - Editar placeholders - Ativar/desativar templates - Visualizar preview

**3. Gerenciamento de Usuários:** - Criar/editar usuários - Definir permissões - Gerar API Keys - Visualizar histórico de acesso - Revogar acessos

**4. Configurações do Sistema:** - Ajustar parâmetros de performance - Configurar integrações - Gerenciar cache - Configurar logs - Backup e restore

### Gerenciamento de API Keys

**Criar nova API Key:**

curl -X POST "http://localhost:5000/admin/api-key/create" \  
 -H "Content-Type: application/json" \  
 -H "X-API-Key: admin-api-key" \  
 -d '{  
 "user\_id": "usuario@empresa.com",  
 "permissions": ["read", "write", "admin"],  
 "expiration\_days": 90  
 }'

**Listar API Keys:**

curl -X GET "http://localhost:5000/admin/api-keys" \  
 -H "X-API-Key: admin-api-key"

**Revogar API Key:**

curl -X DELETE "http://localhost:5000/admin/api-key/{key\_id}" \  
 -H "X-API-Key: admin-api-key"

### Gerenciamento de Templates

**Via Dashboard Administrativo:**

1. Acesse http://localhost:5000/admin/admin\_dashboard.html
2. Navegue até “Gerenciar Templates”
3. Clique em “Upload Novo Template”
4. Selecione arquivo PPTX
5. Configure placeholders
6. Salve e ative

**Via API:**

# Listar templates  
curl -X GET "http://localhost:5000/admin/templates/usuario" \  
 -H "X-API-Key: sua-api-key"  
  
# Atualizar placeholders  
curl -X POST "http://localhost:5000/admin/template/update" \  
 -H "Content-Type: application/json" \  
 -H "X-API-Key: sua-api-key" \  
 -d '{  
 "template\_name": "template\_relatorio.pptx",  
 "new\_placeholders": {  
 "total\_vendas": "receita\_total",  
 "crescimento": "variacao\_percentual"  
 }  
 }'

### Configuração de Integrações

**SharePoint:**

curl -X POST "http://localhost:5000/admin/sharepoint/config" \  
 -H "Content-Type: application/json" \  
 -H "X-API-Key: sua-api-key" \  
 -d '{  
 "base\_url": "https://suaempresa.sharepoint.com",  
 "site\_path": "/sites/seu-site",  
 "upload\_path": "/Documentos/CORTEX\_BI",  
 "client\_id": "seu-client-id",  
 "client\_secret": "seu-client-secret",  
 "tenant\_id": "seu-tenant-id"  
 }'

**Power BI:**

curl -X POST "http://localhost:5000/admin/powerbi/config" \  
 -H "Content-Type: application/json" \  
 -H "X-API-Key: sua-api-key" \  
 -d '{  
 "client\_id": "seu-client-id",  
 "client\_secret": "seu-client-secret",  
 "tenant\_id": "seu-tenant-id",  
 "workspace\_id": "seu-workspace-id"  
 }'

## 🔗 Integração Microsoft 365

### Pré-requisitos para Integração

* Microsoft 365 Business ou superior
* Microsoft Copilot for Microsoft 365 licenciado
* Permissões de administrador no tenant
* CÓRTEX BI rodando em servidor acessível

### Integração com Copilot Studio

**Passo 1: Registrar o CÓRTEX BI como Plugin**

1. Acesse o Microsoft Copilot Studio
2. Navegue até “Plugins” → “Add Plugin”
3. Selecione “API Plugin”
4. Configure:

{  
 "schema\_version": "v1",  
 "name\_for\_human": "CÓRTEX BI",  
 "name\_for\_model": "cortex\_bi",  
 "description\_for\_human": "Agente de análise de dados e business intelligence",  
 "description\_for\_model": "Analisa dados, gera relatórios e apresentações PPTX automaticamente. Responde perguntas em português sobre métricas, vendas, performance e KPIs.",  
 "auth": {  
 "type": "service\_http",  
 "authorization\_type": "bearer",  
 "verification\_tokens": {  
 "api\_key": "sua-api-key"  
 }  
 },  
 "api": {  
 "type": "openapi",  
 "url": "http://seu-servidor:5000/openapi.json"  
 },  
 "logo\_url": "http://seu-servidor:5000/static/logo.png",  
 "contact\_email": "admin@empresa.com",  
 "legal\_info\_url": "http://seu-servidor:5000/legal"  
}

**Passo 2: Criar Flows no Power Automate**

Para cada funcionalidade do CÓRTEX BI, crie um Flow:

**Flow 1: CÓRTEX Health Check**

Trigger: When Copilot calls action  
Action: HTTP Request  
 Method: GET  
 URL: http://seu-servidor:5000/health  
 Headers: X-API-Key: sua-api-key  
Response: Return to Copilot

**Flow 2: CÓRTEX Analyze**

Trigger: When Copilot calls action with parameters  
Action: HTTP Request  
 Method: POST  
 URL: http://seu-servidor:5000/analyze  
 Headers: X-API-Key: sua-api-key  
 Body: {  
 "file\_path": @{triggerBody()['file\_path']},  
 "analysis\_type": @{triggerBody()['analysis\_type']}  
 }  
Response: Format and return to Copilot

**Flow 3: CÓRTEX NLP Query**

Trigger: When Copilot calls action with query  
Action: HTTP Request  
 Method: POST  
 URL: http://seu-servidor:5000/nlp/query  
 Headers: X-API-Key: sua-api-key  
 Body: {  
 "query": @{triggerBody()['query']},  
 "user\_id": @{triggerBody()['user\_id']}  
 }  
Response: Return formatted results

**Passo 3: Configurar Tópicos no Copilot Studio**

Crie tópicos para ativar o CÓRTEX BI:

**Tópico 1: Análise de Dados** - Gatilho: “analisar dados”, “análise”, “métricas” - Ação: Chamar Flow “CÓRTEX Analyze”

**Tópico 2: Perguntas Naturais** - Gatilho: “como foram”, “mostre”, “qual foi” - Ação: Chamar Flow “CÓRTEX NLP Query”

**Tópico 3: Gerar Relatório** - Gatilho: “gerar relatório”, “criar apresentação” - Ação: Chamar Flow “CÓRTEX Generate PPTX”

**Passo 4: Testar Integração**

No Microsoft Teams:

@Copilot, use CÓRTEX BI para analisar vendas do último trimestre

### Script de Configuração Automática

Execute o script automatizado:

python scripts/configurar\_integracao\_copilot.py \  
 --url http://seu-servidor:5000 \  
 --api-key sua-api-key \  
 --copilot-url https://copilotstudio.microsoft.com

## 📊 Monitoramento e Manutenção

### Monitoramento Contínuo

**Script de Monitoramento:**

# Executar monitoramento contínuo  
python scripts/monitorar\_integracao\_copilot.py \  
 --url http://localhost:5000 \  
 --interval 60 # Verificar a cada 60 segundos

**Métricas Monitoradas:** - Status de cada agente - Tempo de resposta - Taxa de sucesso - Uso de memória e CPU - Número de requisições - Erros e exceções

### Logs do Sistema

**Visualizar logs em tempo real:**

# Log principal  
tail -f logs/cortexbi.log  
  
# Logs de IA  
tail -f logs/ai\_interactions.log  
  
# Logs de erro  
tail -f logs/errors.log  
  
# Logs de auditoria  
tail -f logs/audit.log

**Analisar logs:**

# Buscar erros  
grep ERROR logs/cortexbi.log  
  
# Contar requisições por hora  
grep "POST /analyze" logs/cortexbi.log | cut -d' ' -f1 | cut -d'T' -f2 | cut -d':' -f1 | sort | uniq -c  
  
# Usuários mais ativos  
grep "user\_id" logs/audit.log | cut -d'"' -f4 | sort | uniq -c | sort -nr | head -10

### Backup e Restore

**Backup Automático:**

#!/bin/bash  
# Script de backup diário  
  
BACKUP\_DIR="/backup/cortexbi"  
DATE=$(date +%Y%m%d)  
  
# Criar diretório de backup  
mkdir -p $BACKUP\_DIR/$DATE  
  
# Backup do banco de dados  
cp -r database/ $BACKUP\_DIR/$DATE/  
  
# Backup de templates  
cp -r templates/ $BACKUP\_DIR/$DATE/  
  
# Backup de configurações  
cp .env $BACKUP\_DIR/$DATE/  
  
# Backup de logs  
cp -r logs/ $BACKUP\_DIR/$DATE/  
  
# Compactar  
tar -czf $BACKUP\_DIR/cortexbi\_$DATE.tar.gz $BACKUP\_DIR/$DATE/  
  
# Remover backups antigos (manter últimos 30 dias)  
find $BACKUP\_DIR -name "cortexbi\_\*.tar.gz" -mtime +30 -delete

**Restore:**

#!/bin/bash  
# Restaurar backup  
  
BACKUP\_FILE=$1  
  
# Extrair backup  
tar -xzf $BACKUP\_FILE -C /tmp/  
  
# Parar serviço  
./scripts/stop\_ai.sh  
  
# Restaurar arquivos  
cp -r /tmp/cortexbi\_\*/database/ .  
cp -r /tmp/cortexbi\_\*/templates/ .  
cp /tmp/cortexbi\_\*/.env .  
  
# Reiniciar serviço  
./scripts/start\_ai.sh

### Atualização do Sistema

# 1. Fazer backup  
./scripts/backup.sh  
  
# 2. Parar serviço  
./scripts/stop\_ai.sh  
  
# 3. Atualizar código  
git pull origin master  
  
# 4. Atualizar dependências  
pip install -r requirements.txt --upgrade  
  
# 5. Executar migrações (se houver)  
python scripts/migrate.py  
  
# 6. Reiniciar serviço  
./scripts/start\_ai.sh  
  
# 7. Verificar saúde  
curl http://localhost:5000/health

## 🔍 Troubleshooting

### Problema: Servidor não inicia

**Sintomas:** - Erro ao executar python src/main\_ai.py - Porta já em uso

**Soluções:**

# Verificar se porta está ocupada  
netstat -tlnp | grep :5000 # Linux  
netstat -ano | findstr :5000 # Windows  
  
# Matar processo na porta  
# Linux:  
lsof -ti:5000 | xargs kill -9  
# Windows:  
netstat -ano | findstr :5000  
taskkill /PID <PID> /F  
  
# Tentar porta alternativa  
PORT=8000 python src/main\_ai.py

### Problema: Erro de dependências

**Sintomas:** - ModuleNotFoundError - Erros de importação

**Soluções:**

# Reinstalar dependências  
pip install -r requirements.txt --force-reinstall  
  
# Verificar versão do Python  
python3 --version # Deve ser 3.8+  
  
# Limpar cache do pip  
pip cache purge  
  
# Criar novo ambiente virtual  
python3 -m venv venv\_new  
source venv\_new/bin/activate  
pip install -r requirements.txt

### Problema: Erro de conexão SQL Server

**Sintomas:** - Erro ao conectar com banco de dados - Timeout de conexão

**Soluções:**

# Verificar drivers ODBC  
odbcinst -q -d # Linux  
# Deve listar: ODBC Driver 17 for SQL Server  
  
# Testar conectividade  
telnet seu-servidor.database.windows.net 1433  
  
# Verificar credenciais no .env  
cat .env | grep SQL\_SERVER  
  
# Testar conexão manualmente  
python3 -c "  
import pyodbc  
conn = pyodbc.connect(  
 'DRIVER={ODBC Driver 17 for SQL Server};'  
 'SERVER=seu-servidor;'  
 'DATABASE=seu-banco;'  
 'UID=usuario;'  
 'PWD=senha'  
)  
print('Conexão bem-sucedida!')  
"

### Problema: Template PPTX não encontrado

**Sintomas:** - Erro ao gerar apresentação - Arquivo não encontrado

**Soluções:**

# Verificar templates disponíveis  
ls -la templates/  
  
# Criar template básico se necessário  
python3 -c "  
from pptx import Presentation  
prs = Presentation()  
prs.save('templates/template\_basico.pptx')  
print('Template criado!')  
"  
  
# Verificar permissões  
chmod 644 templates/\*.pptx

### Problema: Alto uso de memória

**Sintomas:** - Sistema lento - Erros de memória

**Soluções:**

# Verificar uso de memória  
ps aux | grep python  
  
# Limpar cache  
curl -X POST "http://localhost:5000/admin/cache/clear" \  
 -H "X-API-Key: sua-api-key"  
  
# Ajustar configurações no .env  
CACHE\_MAX\_SIZE=500 # Reduzir tamanho do cache  
WORKERS=2 # Reduzir número de workers  
  
# Reiniciar serviço  
./scripts/stop\_ai.sh  
./scripts/start\_ai.sh

### Script de Diagnóstico Automático

# Executar diagnóstico completo  
python scripts/diagnosticar\_integracao\_copilot.py \  
 --url http://localhost:5000 \  
 --full-check  
  
# O script verificará:  
# - Conectividade  
# - Status de agentes  
# - Integridade de dados  
# - Configurações  
# - Logs de erro  
# - Performance

## 🔐 Segurança e Backup

### Configuração de Firewall

**Linux (UFW):**

# Permitir apenas IPs específicos  
sudo ufw allow from 192.168.1.0/24 to any port 5000  
  
# Permitir Copilot Studio  
sudo ufw allow from 20.190.0.0/16 to any port 5000

**Windows:**

# Criar regra de firewall  
New-NetFirewallRule -DisplayName "CORTEX BI" `  
 -Direction Inbound `  
 -LocalPort 5000 `  
 -Protocol TCP `  
 -Action Allow `  
 -RemoteAddress 192.168.1.0/24

### Rotação de API Keys

# Script de rotação automática (executar mensalmente)  
#!/bin/bash  
  
# Gerar nova API Key  
NEW\_KEY=$(python3 -c "import secrets; print(secrets.token\_urlsafe(32))")  
  
# Atualizar no .env  
sed -i "s/API\_KEY=.\*/API\_KEY=$NEW\_KEY/" .env  
  
# Reiniciar serviço  
./scripts/stop\_ai.sh  
./scripts/start\_ai.sh  
  
# Notificar administradores  
echo "Nova API Key gerada: $NEW\_KEY" | mail -s "CORTEX BI - Nova API Key" admin@empresa.com

### Criptografia de Dados Sensíveis

# Configurar criptografia no .env  
ENCRYPTION\_ENABLED=True  
ENCRYPTION\_KEY=sua-chave-de-criptografia-32-bytes

### Auditoria e Compliance

# Gerar relatório de auditoria  
python scripts/gerar\_relatorio\_auditoria.py \  
 --start-date 2025-10-01 \  
 --end-date 2025-10-31 \  
 --output relatorio\_auditoria\_outubro.pdf  
  
# Relatório incluirá:  
# - Todos os acessos ao sistema  
# - Operações realizadas por usuário  
# - Dados acessados  
# - Modificações em configurações  
# - Tentativas de acesso negadas

## 📞 Suporte e Recursos Adicionais

### Documentação Completa

* **README Principal**: /docs/README.md
* **API Reference**: http://localhost:5000/docs
* **Guias de Integração**: /docs/integracao/

### Scripts Úteis

Todos os scripts estão disponíveis em /scripts/:

* install.sh/bat - Instalação automática
* start\_ai.sh/bat - Iniciar servidor
* stop\_ai.sh/bat - Parar servidor
* verificar\_integracao\_copilot.py - Verificar pré-requisitos
* configurar\_integracao\_copilot.py - Configurar integração
* diagnosticar\_integracao\_copilot.py - Diagnosticar problemas
* monitorar\_integracao\_copilot.py - Monitoramento contínuo
* backup.sh - Backup automático
* restore.sh - Restaurar backup

### Contato

* **GitHub Issues**: https://github.com/Rimkus85/cortex-bi/issues
* **Email**: suporte@cortexbi.com
* **Documentação Online**: https://github.com/Rimkus85/cortex-bi

**CÓRTEX BI v2.0** - *Cognitive Operations & Real-Time EXpert Business Intelligence*  
Desenvolvido em parceria com **Manus AI** | Outubro 2025

✅ Sistema pronto para produção  
🚀 Transformando dados em decisões inteligentes