

Manual de Instalação - EPI Guard

Sumário

1. [Pré-requisitos](#)
2. [Instalação da API FastAPI](#)
3. [Instalação da Interface Desktop](#)
4. [Configuração Inicial](#)
5. [Primeiro Uso](#)
6. [Solução de Problemas](#)

Pré-requisitos

Sistema Operacional

- **Windows:** Windows 10 ou superior
- **Linux:** Ubuntu 18.04+ ou distribuições equivalentes

Software Necessário

- **Python 3.8+:** Para executar a API FastAPI
- **Node.js 16+:** Para desenvolvimento (opcional)
- **Câmeras IP:** Com suporte RTSP (opcional)

Hardware Mínimo

- **RAM:** 4 GB
- **Processador:** Intel i3 ou equivalente
- **Armazenamento:** 2 GB livres
- **Rede:** Conexão com internet para instalação

Instalação da API FastAPI

Passo 1: Preparar o Ambiente da API

1. **Extrair os arquivos da API:** `bash # Extrair o arquivo
helmet_detection_system.zip unzip helmet_detection_system.zip cd
helmet_detection_system/api`

2. **Criar ambiente virtual:** ``` `bash # Windows python -m venv venv
venv\Scripts\activate`

Linux/macOS `python3 -m venv venv source venv/bin/activate `` ``

1. **Instalar dependências:** `bash pip install -r requirements.txt`

Passo 2: Configurar o Modelo YOLOv8

1. **Verificar o modelo:**
2. Certifique-se de que o arquivo `best.pt` está na pasta `api/`
3. O modelo deve ter aproximadamente 6-14 MB
4. **Testar o carregamento:** `bash python -c "from ultralytics import YOLO; model = YOLO('best.pt'); print('Modelo carregado com sucesso!')"`

Passo 3: Iniciar a API

1. **Executar a API:** `bash python main.py`
2. **Verificar funcionamento:**
3. Acesse: `http://localhost:8000/docs`
4. Teste o endpoint: `GET /api/v1/health`
5. Resposta esperada: `{"status": "healthy", "model_loaded": true}`

Instalação da Interface Desktop

Opção 1: Usar o AppImage (Linux - Recomendado)

1. **Baixar o AppImage:**
2. Arquivo: `EPI Guard-1.0.0.AppImage`
3. Tamanho: ~103 MB
4. **Tornar executável:** `bash chmod +x "EPI Guard-1.0.0.AppImage"`
5. **Executar:** `bash ./"EPI Guard-1.0.0.AppImage"`

Opção 2: Desenvolvimento (Todas as Plataformas)

1. **Extrair o código fonte:** `bash # Extrair epi-guard-electron.zip unzip epi-guard-electron.zip cd epi-guard-electron`
2. **Instalar dependências:** `bash npm install`
3. **Executar em modo desenvolvimento:** `bash npm run electron-dev`

Opção 3: Build Local

1. **Gerar build de produção:** `bash npm run build`
2. **Gerar instalador:** ``` bash # Linux npm run dist-linux`

Windows (requer Wine no Linux) `npm run dist-win ```

Configuração Inicial

Passo 1: Verificar Conectividade

1. **Iniciar a API FastAPI** (se não estiver rodando): `bash cd helmet_detection_system/api source venv/bin/activate # Linux/macOS # ou venv\Scripts\activate # Windows python main.py`
2. **Iniciar a Interface Desktop:**
3. Execute o AppImage ou use `npm run electron-dev`

Passo 2: Configurar a Conexão

1. **Acessar Configurações:**
2. Faça login com: `admin / admin123`
3. Vá para a aba "Configurações"
4. **Configurar API:**
5. **URL Base:** `http://localhost:8000`
6. **Timeout:** `30000 ms`
7. **Tentativas:** `3`
8. Clique em "Testar Conexão"

9. Verificar Status:

- 10. Status deve mostrar: "API Conectada"
- 11. Modelo deve mostrar: "Carregado"

Passo 3: Configurar Câmeras

1. Acessar Gerenciamento de Câmeras:

2. Vá para a aba "Câmeras"

3. Clique em "Adicionar Câmera"

4. Configurar Webcam Local:

5. **Nome:** "Webcam Principal"

6. **URL RTSP:** 0 (para webcam padrão)

7. **Localização:** "Escritório"

8. Clique em "Salvar"

9. Configurar Câmera IP:

10. **Nome:** "Câmera Entrada"

11. **URL RTSP:** rtsp://usuario:senha@ip:porta/stream

12. **Localização:** "Portaria"

13. Clique em "Salvar"

Primeiro Uso

Passo 1: Login no Sistema

1. Credenciais Padrão:

2. **Administrador:** admin / admin123

3. **Técnico:** tecnico / tecnico123

4. Alterar Senhas:

5. Vá para "Usuários" → "Editar"

6. Altere as senhas padrão por segurança

Passo 2: Testar Detecção

1. Teste Manual:

2. Vá para "Câmeras"
3. Clique em "Ver" em uma câmera
4. Clique em "Detectar"
5. Verifique os resultados
6. **Teste Automático:**
7. Ative "Detecção Automática" nas configurações
8. Monitore o dashboard para detecções em tempo real

Passo 3: Configurar Relatórios

1. **Gerar Primeiro Relatório:**
2. Vá para "Relatórios"
3. Selecione período: "Hoje"
4. Clique em "Gerar PDF"
5. **Configurar Exportação:**
6. Configure formatos desejados (PDF, Excel, CSV)
7. Defina periodicidade automática

Solução de Problemas

Problema: API não conecta

Sintomas: Dashboard mostra "API Desconectada"

Soluções: 1. Verificar se a API está rodando: `bash curl http://localhost:8000/api/v1/health`

1. Verificar logs da API: `bash # Verificar arquivo de log na pasta api/
tail -f helmet_detection_api_*.log`
2. Reiniciar a API: `bash # Parar (Ctrl+C) e reiniciar python main.py`

Problema: Modelo YOLOv8 não carrega

Sintomas: `"model_loaded": false` na API

Soluções: 1. Verificar arquivo do modelo: `bash ls -la best.pt # Deve ter 6-14 MB`

1. Verificar dependências: `bash pip install ultralytics torch torchvision`
2. Testar carregamento manual: `bash python -c "from ultralytics import YOLO; YOLO('best.pt')"`

Problema: Câmera não conecta

Sintomas: Erro ao visualizar câmera

Soluções: 1. **Para webcam local:** - Verificar se a webcam está sendo usada por outro programa - Tentar diferentes índices: 0, 1, 2

1. **Para câmera IP:**
2. Verificar URL RTSP: `rtsp://ip:porta/stream`
3. Testar com VLC: `vlc rtsp://ip:porta/stream`
4. Verificar credenciais de acesso
5. **Permissões:** `bash # Linux: adicionar usuário ao grupo video sudo usermod -a -G video $USER`

Problema: Interface não abre

Sintomas: AppImage não executa

Soluções: 1. **Verificar permissões:** `bash chmod +x "EPI Guard-1.0.0.AppImage"`

1. **Verificar dependências:** `bash # Ubuntu/Debian sudo apt install libgtk-3-0 libxss1 libasound2`
2. **Executar em modo debug:** `bash ./"EPI Guard-1.0.0.AppImage" --verbose`

Problema: Performance lenta

Sintomas: Detecção demorada

Soluções: 1. **Otimizar configurações:** - Reduzir intervalo de detecção - Diminuir resolução das câmeras - Ajustar limite de confiança

1. **Verificar recursos:** `bash # Monitorar CPU e RAM htop`

2. Usar GPU (se disponível):

```
bash # Instalar PyTorch com CUDA pip install torch torchvision  
--index-url https://download.pytorch.org/whl/cu118
```

Logs e Diagnóstico

Localização dos Logs

1. API FastAPI:

2. Arquivo: `helmet_detection_api_YYYYMMDD.log`

3. Local: Pasta `api/`

4. Interface Desktop:

5. Console do Electron (F12)

6. Logs do sistema operacional

Comandos Úteis

```
# Verificar status da API  
curl http://localhost:8000/api/v1/health  
  
# Testar detecção via API  
curl -X GET http://localhost:8000/api/v1/detect/test  
  
# Verificar processos  
ps aux | grep python  
ps aux | grep electron  
  
# Verificar portas  
netstat -tlnp | grep 8000
```

Contato e Suporte

Para suporte adicional: - **Email:** suporte@epiguard.com - **Documentação:** Consulte o README.md - **Logs:** Sempre inclua logs relevantes ao reportar problemas

EPI Guard v1.0.0 - Manual de Instalação

Última atualização: Junho 2025