

# Documentação Técnica - Bot AFIS

Data de Emissão: 14/01/2026

Versão do Sistema: 1.0 (Fase de teste) - Por Carlos Ribeiro

## 1. Objetivo e Escopo

O Bot AFIS foi desenvolvido para mitigar falhas operacionais referentes ao esquecimento do envio de Boletins de Ocorrência (BOs) finalizados para o grupo AFIS. O software atua como um painel de monitoramento passivo ('Dashboard'), permitindo o controle temporal de múltiplos talões simultaneamente, sem interferir na rotina do operador.

## 2. Stack Tecnológico e Bibliotecas

O sistema foi construído em Python 3.12, priorizando o uso de bibliotecas nativas (Standard Library) para garantir segurança, estabilidade e auditabilidade.

### Pacotes Utilizados:

- Tkinter & TTK: Responsáveis pela Interface Gráfica (GUI) nativa do Windows, garantindo leveza e baixo consumo de memória RAM.
- Threading: Implementação de concorrência. Permite que cada BO tenha seu próprio cronômetro rodando em 'background' (Daemon Threads) sem travar a interface principal.
- OS & Tempfile: Manipulação segura de arquivos temporários para logs visuais, garantindo que nenhum dado sensível persista no disco rígido.
- Datetime: Rastreabilidade temporal (Carimbo de tempo) para entrada e saída de ocorrências.

## 3. Arquitetura e Funções Críticas

O software utiliza o padrão de arquitetura orientado a eventos (Event-Driven). Abaixo, as principais funções lógicas desenvolvidas:

- AFISDashboard (Class): Gerencia a árvore visual (Treeview) e o estado da aplicação principal.
- BOMonitor (Class): Instância isolada para cada ocorrência. Cada objeto desta classe roda em uma Thread separada.
- loop\_monitoramento(): Algoritmo de contagem regressiva fracionada. Verifica a cada segundo se houve comando de parada para garantir resposta imediata.
- adicionar\_monitoramento(): Sanitiza os dados de entrada, converte siglas para maiúsculo e instancia o monitor.
- gerar\_txt(): Cria logs voláteis na pasta %TEMP% do Windows para conferência rápida.

## 4. Segurança da Informação e Requisitos

O software foi projetado respeitando as políticas de segurança da infraestrutura de TI, operando sob o princípio de privilégio mínimo.

- Execução Standalone: O software é compilado via PyInstaller (--onefile), não exigindo instalação ou permissões de Administrador.
- Ambiente Offline: O robô não realiza nenhuma conexão com a internet ou rede externa.

## **Documentação Técnica - Bot AFIS**

- Não-Intrusivo: Não realiza leitura de tela, OCR, nem injeção de código em navegadores ou no sistema RDO.
- Privacidade: Não monitora o WhatsApp. Todos os inputs são manuais e conscientes por parte do operador.

### **5. Compilação e Deploy**

O código fonte é encapsulado em binário executável (.exe) para Windows 64-bit. O empacotamento inclui o interpretador Python necessário, tornando o software portátil ('Portable App') que pode ser executado diretamente de pastas locais ou unidades removíveis.