



**Ministério da Previdência Social**  
Secretaria do Regime Geral da Previdência Social  
Departamento de Perícia Médica Federal

---

*Coordenação-Geral de Assuntos Corporativos e Disseminação de Conhecimento*

**Gustavo Magalhães Mendes de Tarso**

## **RELATÓRIO INSTITUCIONAL SISTEMA atestmed-defender**

### **1 Introdução**

O presente relatório visa documentar o desenvolvimento e a implementação de um sistema automatizado para a gestão de análises de tarefas no contexto de sistemas públicos. Este sistema foi concebido e desenvolvido exclusivamente por Gustavo Magalhães Mendes de Tarso, Coordenador-Geral de Assuntos Corporativos e Disseminação de Conhecimento. O sistema, cujo nome é derivado diretamente da pasta raiz do projeto, busca otimizar a extração e análise de dados complexos provenientes do Portal PMF, facilitando a filtragem de profissionais conforme indicadores-chave de desempenho (KPIs).

### **2 Contexto do Problema**

O desafio central enfrentado era a extração de dados do Portal PMF, que se apresentavam em arquivos CSV complexos. Essa complexidade dificultava a filtragem eficaz de profissionais com base em KPIs, essenciais para a gestão estratégica e operacional. A necessidade de um sistema que pudesse automatizar a filtragem, cálculo de indicadores e geração de relatórios era evidente, dada a quantidade e complexidade dos dados envolvidos.

### **3 Soluções Desenvolvidas**

Para enfrentar o desafio, foi desenvolvido um sistema que automatiza a filtragem de dados, o cálculo de indicadores e a geração de relatórios. O sistema utiliza scripts em Python e R para processar os dados, calcular indicadores como ICRA, IATD e Score Final, e gerar relatórios detalhados. A automação desses processos não apenas reduz o tempo necessário para a análise de dados, mas também aumenta a precisão e consistência dos resultados.

## 4 Indicadores Estratégicos

Os KPIs utilizados no sistema incluem o Índice de Conformidade de Risco de Análise (ICRA), o Índice de Análise de Tarefas Diárias (IATD) e o Score Final. O ICRA mede a conformidade das análises realizadas, enquanto o IATD avalia a eficiência das análises diárias. O Score Final é uma métrica composta que integra múltiplos indicadores para fornecer uma visão abrangente do desempenho. Esses indicadores são cruciais para a gestão estratégica, pois permitem o monitoramento em tempo real e o suporte à decisão baseado em dados objetivos e acionáveis.

## 5 Importância para a Gestão Estratégica

O sistema desenvolvido é de extrema importância para a gestão estratégica, pois permite o monitoramento em tempo real das análises de tarefas, transformando dados brutos em indicadores objetivos e acionáveis. Isso facilita a tomada de decisões informadas e a implementação de estratégias baseadas em evidências, promovendo uma gestão mais eficiente e eficaz.

## 6 Importância do Uso de Inteligência Artificial

A utilização de inteligência artificial no sistema é um diferencial significativo, permitindo a automação das análises e a geração de relatórios institucionais de forma inovadora. A IA contribui para a inovação na administração pública, automatizando processos complexos e permitindo uma análise mais profunda e precisa dos dados.

## 7 Impactos Institucionais

O impacto institucional do sistema é significativo, proporcionando melhorias tanto para a população quanto para o governo. Para a população, o sistema garante uma análise mais rápida e precisa das tarefas, melhorando a qualidade dos serviços prestados. Para o governo, a automação dos processos resulta em economia de recursos e aumento da eficiência operacional.

## 8 Conclusão

Em conclusão, o sistema desenvolvido por Gustavo Magalhães Mendes de Tarso representa um avanço significativo na gestão de análises de tarefas em sistemas públicos. Ao automatizar processos complexos e integrar inteligência artificial, o sistema não apenas melhora a eficiência e precisão das análises, mas também contribui para uma gestão estratégica mais informada e eficaz.

## 9 Arquivos e Saídas Utilizados pelo Relatório

O arquivo `graphs_and_tables/compare_fifteen_seconds.py` gera gráficos comparativos de análises de tarefas com duração inferior a quinze segundos. Os principais parâmetros utilizados por `reports/make_report.py` incluem `--start`, `--end`, `--threshold`, e `--mode`. Os artefatos gerados são salvos em formato PNG.

O arquivo `graphs_and_tables/compare_indicadores_composto.py` calcula indicadores compostos e gera gráficos comparativos. Os parâmetros principais são `--start`, `--end`, `--top10`, e `--mode`. Os resultados são salvos em PNG e ORG.

O arquivo `graphs_and_tables/compare_motivos_perito_vs_brasil.py` analisa os motivos de não conformidade, gerando gráficos e tabelas. Utiliza parâmetros como `--start`, `--end`, e `--perito`, salvando os resultados em PNG e MD.

O arquivo `graphs_and_tables/compare_nc_rate.py` compara taxas de não conformidade, gerando gráficos e relatórios. Os parâmetros incluem `--start`, `--end`, e `--perito`, com saídas em PNG e ORG.

O arquivo `graphs_and_tables/compare_overlap.py` analisa sobreposições de tarefas, gerando gráficos e relatórios. Os parâmetros principais são `--start`, `--end`, e `--mode`, com saídas em PNG e ORG.

O arquivo `graphs_and_tables/compare_productivity.py` calcula a produtividade dos profissionais, gerando gráficos e relatórios. Utiliza parâmetros como `--start`, `--end`, e `--threshold`, com saídas em PNG e ORG.

O arquivo `r_checks/01_nc_rate_check.R` verifica a taxa de não conformidade, gerando gráficos em PNG. Os parâmetros principais são `--start`, `--end`, e `--perito`.

O arquivo `r_checks/02_le15s_check.R` verifica a duração das análises, gerando gráficos em PNG. Os parâmetros incluem `--start`, `--end`, e `--threshold`.

O arquivo `r_checks/03_productivity_check.R` verifica a produtividade, gerando gráficos em PNG. Os parâmetros principais são `--start`, `--end`, e `--threshold`.

O arquivo `r_checks/04_overlap_check.R` verifica sobreposições de tarefas, gerando gráficos em PNG. Os parâmetros incluem `--start`, `--end`, e `--perito`.

O arquivo `r_checks/05_motivos_chisq.R` realiza testes qui-quadrado sobre motivos de não conformidade, gerando gráficos em PNG e relatórios em MD. Os parâmetros principais são `--start`, `--end`, e `--perito`.

O arquivo `r_checks/06_composite_robustness.R` verifica a robustez dos indicadores compostos, gerando gráficos em PNG. Os parâmetros incluem `--start`, `--end`, e `--perito`.

O arquivo `r_checks/g01_top10_nc_rate_check.R` verifica a taxa de não conformidade dos 10 piores, gerando gráficos em PNG. Os parâmetros principais são `--start`, `--end`, e `--min-analises`.

O arquivo `r_checks/g02_top10_le15s_check.R` verifica a duração das análises dos 10 piores, gerando gráficos em PNG. Os parâmetros incluem `--start`, `--end`, e `--min-analises`.

O arquivo `r_checks/g03_top10_productivity_check.R` verifica a produtividade dos 10 piores, gerando gráficos em PNG. Os parâmetros principais são `--start`, `--end`, e `--min-analises`.

O arquivo `r_checks/g04_top10_overlap_check.R` verifica sobreposições de tarefas dos 10 piores, gerando gráficos em PNG. Os parâmetros incluem `--start`, `--end`, e `--min-analises`.

O arquivo `r_checks/g05_top10_motivos_chisq.R` realiza testes qui-quadrado sobre motivos de não conformidade dos 10 piores, gerando gráficos em PNG e relatórios em MD. Os parâmetros principais são `--start`, `--end`, e `--min-analises`.

O arquivo `r_checks/g06_top10_composite_robustness.R` verifica a robustez dos indicadores compostos dos 10 piores, gerando gráficos em PNG. Os parâmetros incluem `--start`, `--end`, e `--min-analises`.