

Requisitos Funcionais (RF)

- RF01 – Coleta de dados públicos: O sistema utiliza um conjunto de dados públicos sobre fatalidades de bombeiros obtidos no: [Kaggle](#).
- RF02 – Tratamento de dados: Cumprido. O pré-processamento inicial (limpeza, normalização de datas, tratamento de valores nulos) foi realizado externamente em R, além da criação do Data Base em SQLite(RF03). O script Python (dash.py) também realiza uma segunda camada de tratamento leve para garantir a aplicação dos filtros.
- RF03 – Armazenamento de dados: Os dados tratados e limpos foram carregados num banco de dados SQLite (bombeiros.db).
- RF04 – Dashboard inicial: A aplicação dash.py utiliza a biblioteca Streamlit (para criar uma interface web. Esta interface lê os dados do banco SQLite (RF03) e exibe-os em KPIs, gráficos interativos (barras, linhas, pizza, histograma) e tabelas.

Requisitos Não Funcionais (RNF)

- RNF01 – Usabilidade básica: O dashboard segue princípios de design de interação:
- Visibilidade: O estado do sistema é claro (KPIs no topo, filtros selecionados visíveis).
- Controle do Utilizador: O usuário tem total controle através de filtros interativos na barra lateral.
- Storytelling com Dados: Os gráficos são claros, usam tooltips e contam uma história sobre os dados.
-
- RNF02 – Confiabilidade: O sistema não depende de uma API externa que possa estar fora do ar. Ele lê diretamente do arquivo de banco de dados local , garantindo que o dashboard funcione sempre.

- RNF03 – Desempenho inicial: O carregamento é quase instantâneo. A biblioteca Streamlit utiliza cache de dados (@st.cache_data) para que, após o primeiro carregamento, a leitura do banco de dados não precise ser repetida, garantindo que os filtros sejam aplicados quase que instantaneamente.