**UNIVERSIDADE REGIONAL INTEGRADA DO ALTO URUGUAI E DAS**

**MISSÕES**

**PRÓ-REITORIA DE ENSINO, PESQUISA, EXTENSÃO E PÓS-GRADUAÇÃO**

**CÂMPUS DE ERECHIM**

**DEPARTAMENTO DE ENGENHARIAS E CIÊNCIAS DA COMPUTAÇÃO**

**CURSO DE CIÊNCIAS DA COMPUTAÇÃO**

**AUTORES:**

**ANDRÉ SCHENATO**

**BERNARDO KUJAVA**

**EVELYN RODRIGUES KELIN FABISIAK**

**FELIPE RODIGHERO**

**JOÃO VITOR MENEZES**

**LEONARDO BORGHELOT**

**LUCIANO RODRIGUES DOS SANTOS JUNIOR**

**PATRÍCIA DA SILVEIRA CASTELLAN**

**PEDRO HENRIQUE RILL**

**RUAN SAUGO**

**INGESTÃO DE LOGS + DETECÇÃO DE ANOMALIAS EM LOGS**

**ERECHIM - RIO GRANDE DO SUL**

**2025**

**1. INTRODUÇÃO**

A presente documentação tem por objetivo apresentar a proposta desenvolvida pela equipe, voltada para a ingestão de logs e detecção de anomalias em tempo real. A solução busca reduzir o tempo de resposta a incidentes e aumentar a confiabilidade de sistemas computacionais, uma vez que muitas falhas só são percebidas após lentidão ou indisponibilidade do serviço. Além disso, a análise manual de grandes volumes de logs é um processo trabalhoso e sujeito a erros, o que justifica a necessidade de uma abordagem automatizada.

**2.DESENVOLVIMENTO**

**2.1 Descrição da solução**

A solução consiste em um pipeline on-premise responsável por coletar, padronizar e analisar logs em tempo real. O sistema aplica algoritmos de machine learning e regras heurísticas simples para identificar comportamentos anômalos e emitir alertas imediatos. A proposta inclui ainda um painel de monitoramento desenvolvido em Python, utilizando a biblioteca StreamLit, onde é exibido uma visão geral, utilizando gráficos para apresentação dos dados, do total de requisições por dia, semanal e mensal. Além de possuir um gráfico com top requisições por cada IP e um gráfico de rosca onde possuem os métodos de HTTP presente nas Logs. Possui uma aba onde é possível visualizar todas as anomalias presentes são apresentadas em formatos binários (1 ou -1) e por fim uma tabela completa com toda a log, podendo ser exportada para formato csv.

O público-alvo são equipes de NOC, SRE e DevOps, além de PMEs, órgãos públicos, ISPs e ambientes corporativos de TI que buscam confiabilidade, privacidade e redução de custos.

**2.2 Arquitetura / Design da solução**

A arquitetura da solução é composta por três camadas principais. A primeira é a ingestão de dados, responsável pela coleta e padronização de logs provenientes de diferentes serviços e sistemas. Em seguida, ocorre o processamento e análise, no qual são aplicados algoritmos de detecção de anomalias que combinam técnicas de aprendizado de máquina com regras de negócio pré-definidas. Por fim, há a camada de visualização e alertas, implementada em Python por meio da biblioteca StreamLit. Essa estrutura modular garante escalabilidade, facilidade de manutenção e abre espaço para evolução futura da solução.

**2.3 Instalação e configuração**

O sistema foi desenvolvido para ser executado em ambiente on-premise. Para sua instalação, é necessário possuir Python em versão 3.10 ou superior e as bibliotecas adequadas, como pandas, numpy e scikit-learn, StreamLit para a criação da Dashboard. A configuração inicial envolve a definição das fontes de logs, os parâmetros de análise e as credenciais para os mecanismos de notificação.

**Pré-requisitos:**

* **Python 3.10+**
* **Ambiente virtual recomendado (venv ou conda)**

**Passos:**

**# Clone o repositório**

git clone https://github.com/seu-usuario/codeconquer-logaware.git

cd codeconquer-logaware

**# Crie e ative o ambiente virtual**

python -m venv .venv

source .venv/bin/activate # Linux/Mac

.\.venv\Scripts\activate # Windows

**# Instale as dependências**

pip install -r requirements.txt

**2.4 Guia de Operação**

Após a instalação, o usuário deve iniciar a aplicação, incluir um arquivo de logs, que automaticamente começará a coletar e analisar os logs. A dashboard exibe os indicadores de disponibilidade dos serviços monitorados, o histórico de incidentes detectados, as possíveis causas identificadas pelos algoritmos e as métricas de desempenho do sistema, servindo como um centro de monitoramento acessível e de fácil compreensão para as equipes técnicas.

**2.5 Código-fonte**

O código-fonte do projeto será hospedado integralmente no repositório Forge URI, garantindo acesso transparente ao funcionamento da solução. A organização seguirá boas práticas de programação, com separação em módulos, documentação online por meio de comentários explicativos e utilização de versionamento Git, de modo a assegurar a rastreabilidade das alterações e a colaboração eficiente entre desenvolvedores.

**2.6 Testes e validação**

A validação inicial será realizada a partir de cenários simulados de falhas, utilizando tanto logs reais quanto logs artificiais. Essa etapa permitirá avaliar a capacidade do sistema de identificar eventos anômalos, o tempo de resposta na detecção e a consistência dos alertas emitidos. O vídeo demonstrativo já entregue comprova a funcionalidade do Produto Mínimo Viável, incluindo ingestão de logs, detecção básica de anomalias e visualização em painel.

**2.7 Plano de evolução**

Entre as possibilidades de evolução futura do sistema, destacam-se o aperfeiçoamento dos modelos de detecção por meio de técnicas mais avançadas de aprendizado de máquina, a integração com sistemas capazes de recomendar ações corretivas de forma automática, o suporte à maior volume de dados em ambientes distribuídos e a disponibilização de dashboards personalizados que permitam a adaptação das métricas a diferentes perfis de usuários.

**3.CONCLUSÃO**

A solução proposta apresenta um sistema inovador para detecção de anomalias em logs em tempo real, oferecendo ganhos de confiabilidade, proatividade e redução de custos operacionais. Por meio do uso de Python e utilização de bibliotecas como pandas, StreamLit para a interface, numpy, entre outras. A equipe conseguiu desenvolver um Produto Mínimo Viável, com potencial de evolução para um produto robusto e escalável. A documentação aqui apresentada reforça a viabilidade técnica da proposta e sua relevância para diferentes setores, contribuindo para o avanço das práticas de monitoramento inteligente de sistemas.