

DASHBOARD DE MONITORAMENTO DE SERVIDORES

JASMIM SABINI E VINICIUS MARTINS

Contexto

Empresas dependem de servidores estáveis e eficientes, tornando essencial o monitoramento contínuo de recursos como CPU, memória RAM, disco e rede. Ferramentas como Grafana e Prometheus são amplamente utilizadas, mas muitas vezes exigem configuração complexa.

- Nosso dashboard visa uma solução simplificada, aplicando conceitos de Sistemas Operacionais, como chamadas de sistema, gerenciamento de processos e monitoração de processos.



Objetivo do Projeto

- Coletar métricas do sistema operacional (CPU, RAM, disco, rede).
- Armazenar dados em um banco de dados leve (SQLite).
- Exibir visualizações em tempo real com gráficos dinâmicos.
- Implementar alertas visuais quando limites forem ultrapassados.



Tecnologias Utilizadas



- **Python:** Linguagem de programação principal para desenvolvimento do backend.
- **Flask:** Microframework Python para desenvolvimento web, utilizado no backend para criar a API REST que disponibiliza as métricas.
- **Psutil:** Biblioteca Python para coleta de métricas do sistema operacional, como uso de CPU, memória RAM, disco e rede.
- **SQLite:** Banco de dados embarcado e leve, utilizado para armazenar métricas históricas de forma persistente.
- **JavaScript:** Linguagem de programação para desenvolvimento frontend.
- **Chart.js:** Biblioteca JavaScript para visualização de dados em gráficos dinâmicos e interativos no frontend.
- **HTML/CSS:** Linguagens de marcação e estilização para a construção da interface web do dashboard.

Estrutura

- Templates:
 - Base: Estrutura básica da página
 - Config: Configuração de alerta da CPU
 - Histórico: Filtra as informações de um período
 - Index: Página inicial do dashboard
- Python:
 - App: Define as rotas, salva no DB e inicia a thread de coleta periodica

```
▼ DASHBOARD-MONITOR
  ▼ static
    ▼ img
      login-bg.png
      logo-nero.png
    # style.css
  ▼ templates
    <> base.html
    <> config.html
    <> historico.html
    <> index.html
    <> login.html
    <> sobre.html
  + app.py
  ≡ metrics.db
  ⓘ README.md
  ≡ requirements.txt
```

Referências

Livros Técnicos:

TANENBAUM, Andrew S.; BOS, Herbert. Sistemas operacionais modernos. 4. ed. São Paulo: Pearson Education do Brasil, 2016.

STALLINGS, William. Sistemas operacionais. 10. ed. Porto Alegre: Pearson, 2018.

Documentação Oficial:

PYTHON SOFTWARE FOUNDATION. Documentação do psutil. 2025. Disponível em: <https://psutil.readthedocs.io/> . Acesso em: 01 ago. 2025.

SQLITE. SQLite Documentation. 2025. Disponível em: <https://www.sqlite.org/docs.html>. Acesso em: 01 ago. 2025.

Ferramentas:

CHART.JS. Official Documentation. 2025. Disponível em: <https://www.chartjs.org/docs/latest/>. Acesso em: 01 ago. 2025.

LOCUST. Locust: User Documentation. 2025. Disponível em: <https://docs.locust.io/>. Acesso em: 01 ago. 2025.

FLASK. Flask Documentation. 2025. Disponível em: <https://flask.palletsprojects.com/>. Acesso em: 01 ago. 2025.

Obrigado!