

# Documentação do projeto

## Prints

### Métricas:

- cpu and memory by container: Quantidade de CPU e Memória do Host utilizada por cada container;
- cpu load and memory: Quantidade de CPU e Memória do Host utilizada por todos os containers;
- network and I/O: Tráfego de rede e I/O do Host utilizado por todos os containers;
- running containers:
- dash logs, disk and cpu: Quantidade de containers em execução, resumo dos logs de erro e debug;

### Logs

- log error: Query com os logs de Erro que aconteceram no API;
- log info e debug: Query com os logs de Info e Debug que aconteceram no API;
- Logs live trail: Logs de Erro, Info e Debug monitorados em tempo real;

## Arquitetura

O projeto está separado em 4 containers que são responsáveis por:

- Servidor para hospedar, expor e documentar os endpoints da API;
- Servidor Web com pacotes Core da linguagem na API para hospedar os fontes;
- Servidor para Banco de Dados relacional para hospedar os dados;
- Agent da aplicação que monitora e coleta os logs do projeto;

Foi escolhido para o projeto a Linguagem de programação PHP versão 7.2 utilizadas sob o Framework Laravel, versão 6.0. A escolha deu-se pela facilidade da linguagem, sua relevância no mundo Web e a familiaridade da mesma que em conjunto com o Framework, que é amplamente utilizado pela comunidade, fornecem segurança, e padronização nas implementações além de uma abstração que permite ao desenvolvedor concentrar-se apenas nos domínios e regras de negócio da solução.

Foi escolhido para o projeto uma arquitetura de backend e frontend segregados e conectados através de uma API que implementa o padrão REST, que coleta e disponibiliza os dados em endpoints e payloads documentados no Swagger. As páginas web foram implementadas utilizando HTML, CSS que fazem uso do framework Bootstrap em sua versão 4 e JavaScript puro. A independência das camadas de backend e frontend possibilitam reaproveitamento das regras de negócio para diferentes interfaces (ex: mobile e desktop), além de flexibilizar a mudança de frameworks responsáveis pela parte visual da aplicação sem impactar como os dados são consumidos.

