

DEFINIÇÃO DE ARQUITETURA E DESIGN DO PROJETO MEDICAMENTOS

Contexto e Descrição Geral do Projeto

Cada município oferece em sua rede, medicamentos específicos em determinadas quantidades e em pontos geográficos diferentes. O objetivo deste projeto é fornecer uma plataforma que forneça de forma unificada um catálogo dos medicamentos oferecidos por determinado município.

Pontos importantes para definição da Arquitetura

De forma geral, foi dado como requisito que o software deverá ser escalável, configurável por município e possuir alta disponibilidade. Isso implica em alguns pontos importantes que podem ser importantes para definição e planejamento da arquitetura:

- **Escalável:** isso implica que o software deve estar aberto à extensão de funcionalidades do software no futuro, bem como a possibilidade de aumentar a implantação e viabilização do software para novos municípios.
- **Disponibilidade de dados:** os dados referentes aos estoques de medicamentos devem ser possíveis de serem consumidos tanto pelo usuário final quanto por sistemas externos.
- **Alta disponibilidade:** por se tratar de um sistema da saúde, é desejável que o software tenha alta tolerância à falhas e uma infraestrutura segura que permita que o sistema minimize ao máximo o número de interrupções no tráfego e indisponibilidade.
- **Integração com sistemas externos:** o software deve ser capaz de se integrar com sistemas e bases de dados externas a modo de obter informações sobre disponibilidade de remédios de cada município.

Visões da Arquitetura e Design que serão desenvolvidas

Por meio deste documento, serão definidas algumas visualizações importantes para definição e documentação da arquitetura do software. Sendo eles:

- **Diagrama de Componentes:** este diagrama tem como objetivo explicitar os elementos do software, bem como suas interfaces e fronteiras entre si e com possíveis sistemas ou elementos externos.
- **Diagrama de Infraestrutura/Implantação:** este diagrama visa explicitar e especificar a infraestrutura do software a nível de componentes de hardware (redes, máquinas virtuais).

- **Modelagem de banco de dados:** este diagrama tem como objetivo descrever as coleções e atributos que o banco de dados do software deve possuir.

Elementos do Sistema de Software

Para atender os requisitos levantados, foram pensados alguns elementos para satisfazer as necessidades e complexidades inerentes ao software, ambas são detalhadas com maior profundidade. Vejamos:

- **Back-end Core:** este componente será responsável pela integração com a base de dados unificada do sistema, bem como a integração com possíveis sistemas externos.
- **Sistema Web:** este componente é um sistema front-end web destinado ao usuário final, a fim de disponibilizar o catálogo de medicamentos.
- **Back-end for Front-end:** este componente será responsável por mediar a integração entre o front-end web e o core da aplicação, de modo a disponibilizar informações customizadas para o sistema web.

Ferramentas para Desenvolvimento do Software

Descrição de ferramentas, linguagens de programação, frameworks e serviços para implementação dos elementos de software.

Linguagens de programação:

- **Javascript/Typescript:** linguagem moderna utilizada tanto para construção de aplicações server-side e client-side, com uma comunidade ativa e muitas opções de ferramentas disponíveis.

Frameworks e ferramentas para construção:

- **Next.js:** construção de páginas web baseadas em server-side rendering, utilizando como base a biblioteca ReactJS.
- **Nodejs:** construção de aplicações back-end.
- **Docker e Kubernetes:** containerização, scaling e gerenciamento das instâncias da aplicação, bem como o provisionamento de ferramentas e ambiente eficaz para ambiente de desenvolvimento.
- **Kong API Gateway:** provisionamento, autenticação e gerenciamento das APIs fornecidas pelo sistema.
- **DynamoDB (MongoDB, talvez?):** banco utilizado para armazenamento de informações relativas ao sistema.
- **Redis:** banco utilizado para armazenamento de cache de dados.

Ferramentas de versionamento:

- **Git e Github:** para versionamento e armazenamento de código fonte.

Ferramentas para integração e entrega contínua:

- **Drone CI:** ferramenta utilizada para construção de pipelines de integração contínua.
- **Argo CD:** ferramenta para realização de pipelines de entrega contínua em infraestruturas baseadas em Kubernetes.
- **ESLint:** ferramenta para linting de código Javascript/Typescript

Serviços:

- **Utilização dos seguintes serviços na Cloud AWS:**
 - **AWS VPC:** criação de redes internas.
 - **AWS EKS:** administração de cluster Kubernetes.
 - **AWS EC2:** disponibilização de máquinas em cloud.
 - **ElastiCache:** provisionamento de instâncias de Redis gerenciado na cloud.
 - **DynamoDB:** banco NoSQL gerenciado na cloud AWS.
 - **AWS S3 e CloudFront:** serviços de distribuição de conteúdo estático web.
 - **Route 53:** serviço para fornecimento de DNS.