

Documentação da Arquitetura do Sistema



Entrega do Projeto de Bloco:

Helpify

10/06/2025

1. Introdução

a. Objetivo

O objetivo deste documento é descrever de forma detalhada a arquitetura do sistema “Helpify” documentando suas interações, tecnologias adotadas e padrões de desenvolvimento utilizados.

b. Escopo

O sistema descrito neste documento visa atingir o setor de suporte e através de um sistema de chamados permitir a comunicação eficiente entre a pessoa necessitando de ajuda e o profissional de suporte responsável por seu chamado.

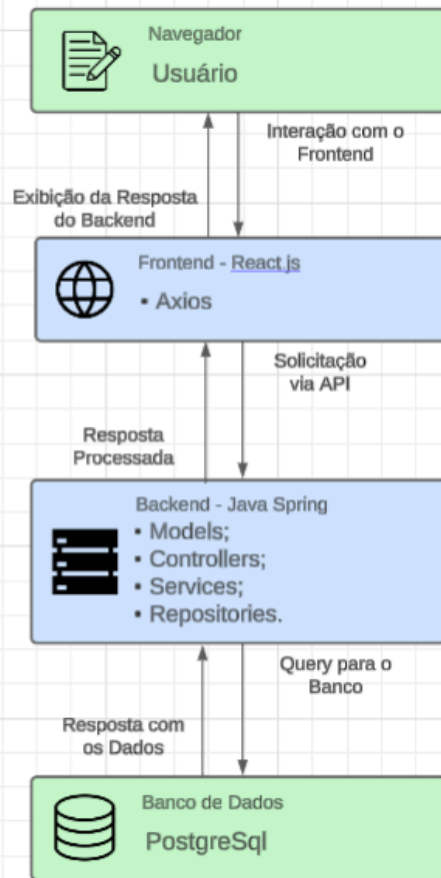
2. Visão Geral

a. Estilo Principal Adotado

Arquitetura em camadas;

b. Diagrama de Alto Nível

Diagrama de Alto Nível: Helpify



3. Componentes

a. Camada de Apresentação

- Tecnologias:** Html, Css, Javascript com React.
- Responsabilidades:** Renderização da interface, interações com o usuário e comunicação com a API no backend.
- Comunicação:** A comunicação é feita com a API Restful que está rodando no backend.

b. Camada de Aplicação

- Tecnologia:** Java com Spring Boot
- Responsabilidades:** Processamento da Lógica do Sistema; Regras de Negócio; Comunicação com o Banco de Dados; Autenticação.
- Padrões Adotados:** MVC; RestFul, Arquitetura em Camada;

c. Banco de Dados

- i. **Tipo:** Relacional - PostgreSQL
- ii. **Principais Entidades:**
 - 1. Users;
 - 2. Tickets:
 - a. TicketAttachments;
 - b. TicketComment.
 - 3. KnowledgeBase:
 - a. KnowledgeBaseCategory;
 - b. KnowledgeBaseTag.
 - 4. Feedback
 - 5. Category
- iii. **ORM:** Hibernate

4. Requisitos Funcionais

RF-01: O sistema deve permitir cadastro e login de usuários.

RF-02: O sistema deve manter um histórico de chamados atrelado ao perfil de cada usuário.

RF-03: O sistema deve permitir que usuários autenticados abram novos chamados.

RF-04: O sistema deve permitir que os chamados sejam classificados por categorias.

RF-05: O sistema deve permitir o envio de anexos.

RF-06: O sistema deve permitir que os chamados sejam atribuídos manualmente ou automaticamente a atendentes específicos.

RF-07: O sistema deve permitir que os atendentes visualizem todos os chamados abertos e em andamento.

RF-08: O sistema deve permitir que os chamados sejam classificados por nível de prioridade.

RF-09: O sistema deve permitir que os atendentes respondam diretamente aos chamados oferecendo soluções.

RF-10: O sistema deve notificar o usuário sempre que o status do chamado for alterado.

RF-11: O sistema deve permitir que o usuário acompanhe em tempo real ou por histórico os seus chamados.

RF-12: O sistema deve disponibilizar gráficos e dashboards para análise do desempenho da equipe de suporte.

RF-13: O sistema deve permitir que o usuário forneça feedback após a finalização do chamado.

RF-14: O sistema deve oferecer um recurso de chat em tempo real entre o atendente e o usuário.

RF-15: O sistema deve manter um histórico de todos os chamados anteriores, incluindo suas interações.

RF-16: O sistema deve enviar um email ao usuário quando houver alteração no status do seu chamado.

RF-17: O sistema deve alertar os atendentes quando seus chamados estiverem próximos do prazo de resolução.

RF-18: O sistema deve notificar o usuário sobre o fechamento de um chamado e solicitar feedback.

RF-19: O sistema deve disponibilizar uma base de conhecimento para que os atendentes possam consultar.

RF-20: O sistema deve permitir a criação e publicação de uma seção de perguntas frequentes (F.A.Q.).

RF-21: O sistema deve identificar chamados sobre problemas comuns e enviar automaticamente artigos ou vídeos ao usuário para ajudá-lo a solucionar o problema.

5. Requisitos Não-Funcionais

RNF-01: O sistema deve ter uma cobertura mínima de 80% do código-fonte por testes automatizados, verificado por uma ferramenta de análise de cobertura.

RNF-02: O sistema deve incluir o uso de workflows automatizados para realizar checagens no código-fonte antes de cada merge ou deploy.

RNF-03: O sistema deve garantir um tempo médio de resposta menor ou igual a 2 segundos quando estiverem sendo trocadas mensagens entre o usuário e o suporte de forma síncrona.

RNF-04: O código-fonte deve ser escrito fazendo uso de boas práticas de desenvolvimento e mantendo o processo bem documentado.

RNF-05: O sistema deve estar disponível 99% do tempo, com exceção das manutenções programadas.

RNF-06: O sistema deve ser compatível com os principais navegadores modernos.

6. Diagramas

a. Diagrama de Tabelas e Relações

https://miro.com/app/board/uXjVlkB5u_M=/?share_link_id=289588221483