

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA



**Aplicativo para monitoramento da qualidade do ar
em ambiente hospitalar.**

Documentação de Concepção.

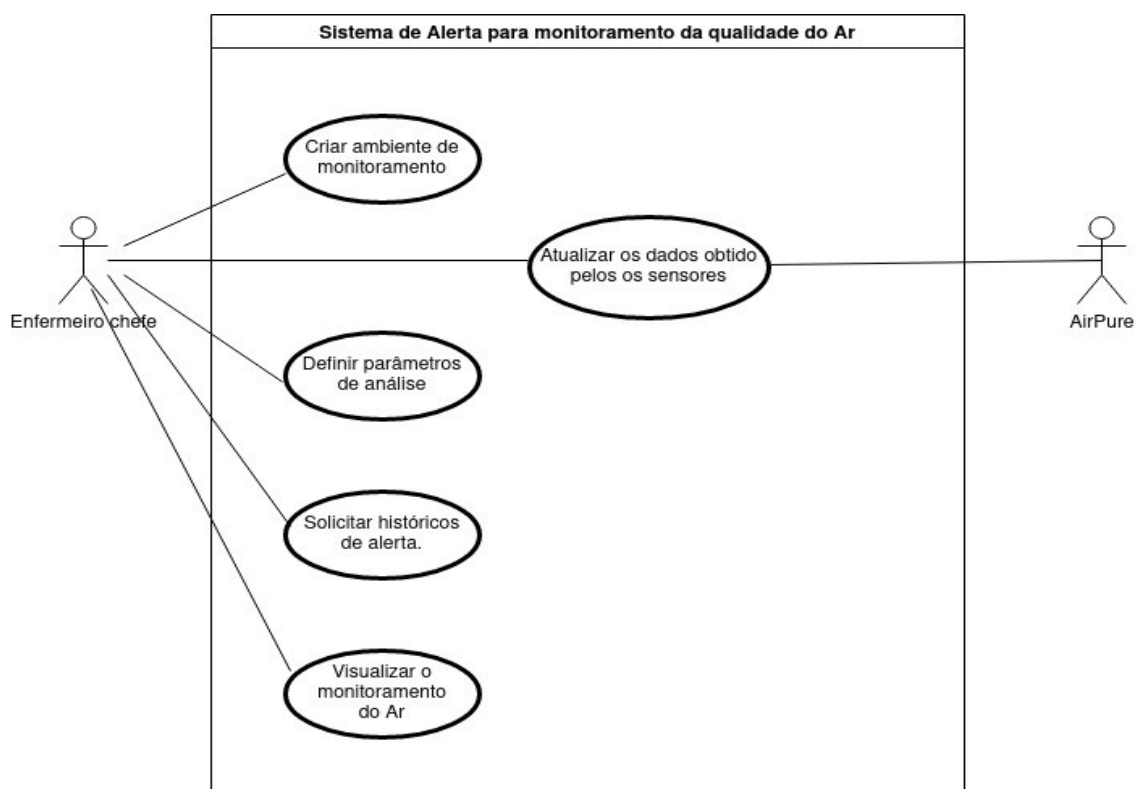
Alunos: Luciano Mendes.

1. Conceitos operacionais do sistema

1.1. Entidades externas identificadas

Nesse sistema de monitoramento da qualidade do ar foram identificados e documentados os seguintes autores.

- **Enfermeiro chefe:** Usuário responsável pelo gerenciamento e monitoramento do sistema.
- **Plataforma Airpure:** Plataforma responsável por coletar e fornecer os dados através de sensores.



1.2. Solução proposto do software

O NeoNatAir é um sistema projetado para aprimorar os cuidados no ambiente UTI por meio do monitoramento preciso das condições dos ambientes hospitalares. Com a API do Airpure fornecendo os dados em tempo real sobre a luminosidade, umidade, nível de ruído, presença de compostos orgânicos voláteis (VOCs) e temperatura, essenciais para garantir um ambiente seguro e saudável. Todas essas informações são facilmente acessíveis aos profissionais responsáveis pela UTI por meio de uma plataforma web intuitiva, oferecendo uma visão abrangente e atualizada das condições do ambiente. Com o

NeoNatAir os profissionais podem tomar decisões mais rápidas, proporcionando o melhor cuidado possível aos pacientes de alto risco.

2. Ambiente de trabalho do projeto

2.1. Gerência de configuração

Ferramentas para apoio no desenvolvimento do sistema.

- **Github** para controle de mudança e versão do projeto, o repositório consiste em duas branch. A branch principal do projeto *MAIN* e uma branch de homologação *Development*.
<https://github.com/lucianomenddes/neoNatAir>
- **Trello** para acompanhamento das fases, atividades e sprints do projeto.

2.2. Infraestrutura tecnológica

Para o desenvolvimento do sistema foram definidas algumas ferramentas para a construção do mesmo.

- **Frontend:** Javascript, biblioteca React, framework nextjs, material ui.
- **Backend:** Node js, express js, axios.
- **Banco de dados:** Postgresql.
- **Hospedagens:** Heroku, netlify.
- **Repositório remoto:** Github.
- **Testes da aplicação:** jest, selenium.
- **Prototipação:** Figma.
- **Ferramenta de modelagem:** Google diagrams.

3. Arquitetura do Software

3.1. Especificação de requisito e restrições arquiteturais

3.1.1. Requisitos funcionais

RF-01: O sistema deve permitir que o administrador crie ambientes virtuais para monitoramento da qualidade do ar para cada ambiente hospitalar.

RF-02: O sistema deve permitir que o administrador configure para cada ambiente individual, os parâmetros mínimo e máximo de cada dado obtido através dos sensores de temperatura, umidade e ruído, CO2, composto orgânico voláteis, fornecidos pela plataforma air pure.

RF-03: O sistema deve comparar em tempo real os dados obtidos pelo air pure com os parâmetros definidos previamente e notificar com alerta em caso de não conformidade.

RF-04: O sistema deve armazenar os alertas de não conformidades em um banco de dados.

RF-05: O sistema deve gerar relatórios dos alertas de não conformidades e permitir consultar por data específica.

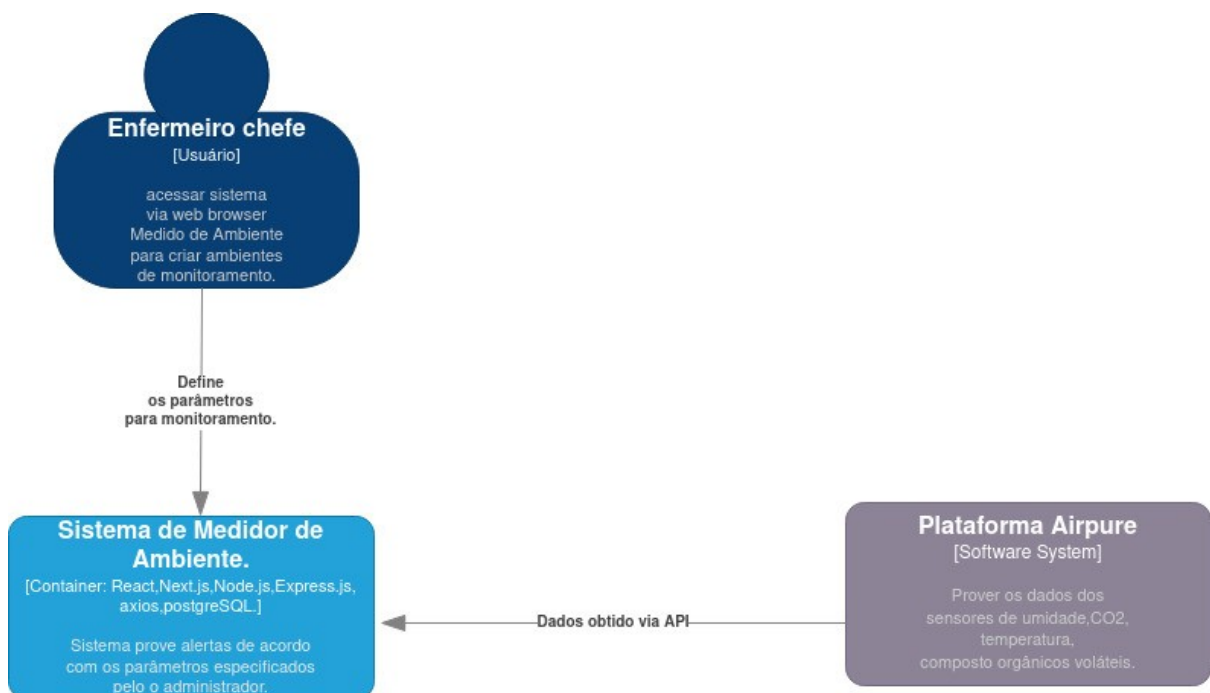
3.1.2. Requisitos não funcionais

RNF-01: O sistema deve exibir os dados do ambiente criado em dashboard .

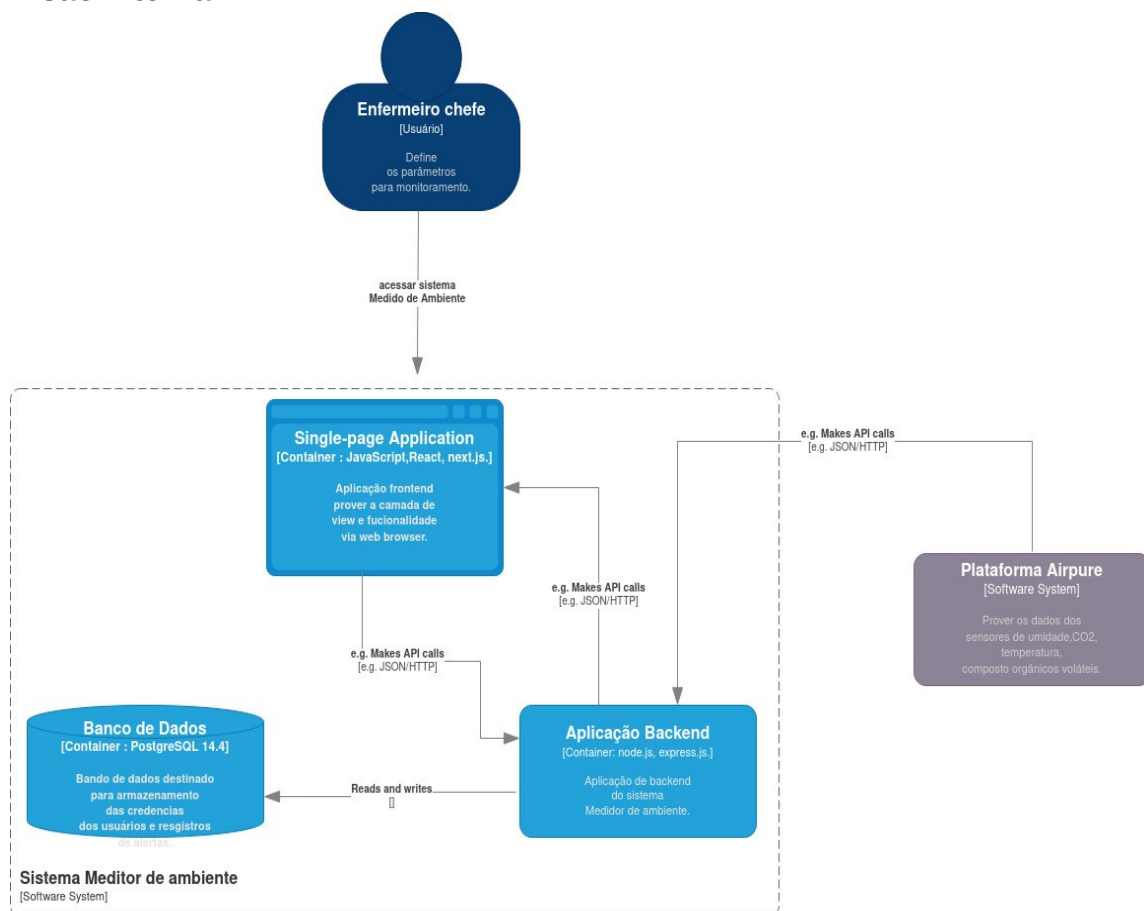
RNF-02: O sistema deve ser desenvolvido com tecnologias web suportadas pelos navegadores Firefox, google chrome.

RNF-03: O sistema deve permitir configurar os alertas de não conformidades com push notification e alarmes sonoros.

3.2. Visão externa



3.3. Visão interna



4. Projeto de Desenvolvimento do Software

4.1. definição do ambiente de trabalho

4.2. cronograma

Etapas do projeto	Artefatos	Data início	Data final	Entrega
Concepção do SW	termo de abertura, doc de concepção	16/06/2022	07/07/2022	
Elaboração do SW	Doc especificação de requisitos	23/06/2022	07/07/2022	
Construção do SW	Definição de design detalhado de software	14/07/2022	04/08/2022	
Transição do SW	Implantação e disponibilização do software	11/08/2022	08/09/2022	
Entrega final		08/09/2022	15/09/2022	

4.3. outra atividades e tarefas