



# Busca Fonada

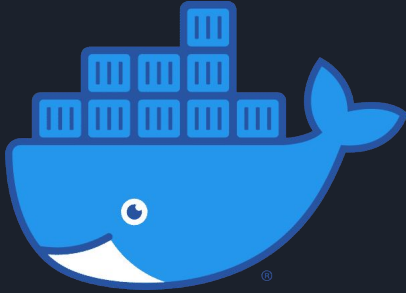
Bernardo Aquino Capello Coelho  
Guilherme Roberto Ferreira Santos  
Gustavo de Castro Nogueira  
Maria Aryene Costa dos Santos  
Thalles Nascimento Carvalho



## Problemática

Segundo um levantamento realizado por funcionários do hospital, são realizadas aproximadamente 300 ligações telefônicas por mês. Sendo todas elas acompanhadas apenas por uma planilha do Excel. Tendo isso em vista isso, os colaboradores buscam soluções na engenharia de software que possam minimizar essa questão.

# Tecnologías utilizadas





# Visão geral do software;

- Funcionalidades principais
- Fluxo geral





# Modelos desenvolvidos

- Diagrama de classes
- Diagrama de casos de uso
- Diagrama de comunicação
- Diagrama de pacotes
- Diagramas de sequência

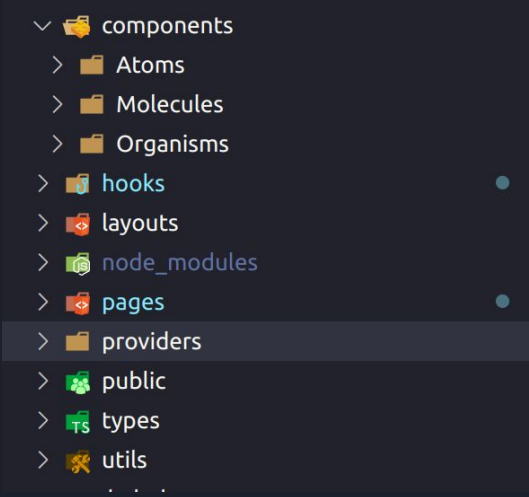


# Análise do projeto

- Arquitetura
- Diagramas
- Melhorias futuras

# Arquitetura - Frontend

- Atomic design
  - Os componentes são separados em camadas definidas como átomos, moléculas e organismos
- Utilização de hooks
  - Funções reutilizáveis para realizar as requisições da aplicação
- Utilização de contextos
  - Utilizado para prover dados de sessão e tema da aplicação
- CSS-in-JS



A screenshot of a file explorer interface showing a directory structure for a frontend application. The structure is as follows:

- components
  - Atoms
  - Molecules
  - Organisms
- hooks
- layouts
- node\_modules
- pages
- providers
- public
- types
- utils
- ...





# Arquitetura - Backend

- Controller
- Services
- Repository
- Model

```
▼ br.com.pucminas.hospital
  > config
  > controller
  > exceptions
  > mapper
  > model
  > repository
  > security
  > services
  HospitalApplication
```



# Diagramas

Os diagramas contribuem para uma melhor projeção do software e deveriam ter sido desenvolvidos durante essa etapa. Entretanto, grande parte desses diagramas foram feitos após a implementação do código, não contribuindo para o projeto, apenas para o entendimento do fluxo já estabelecido.

Outro ponto é que não temos todos os fluxos mapeados nos diagramas, apenas os principais. Em contrapartida podemos dizer que esses estão bem fiéis ao código desenvolvido e seus fluxos.



# Melhorias futuras

- Melhor adequação e entendimento de arquitetura
- Organização dos arquivos do projeto
- Melhor definição de interfaces para cada endpoint
- Padronização dos fluxos (Exemplo: controller -> service -> repository)
- Adequação dos diagramas existentes e desenvolvimento dos faltantes