

## Trabalho de Conclusão de Curso

# Design da solução

### Documento de Arquitetura (Design)

**Projeto:** Loci Exams Reports

**Gerentes de Projeto:** Lucas Bottrel Lopes de Moura

**Grupo:** Lucas Bottrel Lopes de Moura.

### Problema

Os laboratórios que realizam exames têm dificuldade em contratar um serviço que atenda as necessidades de cadastro de exames e geração de relatórios. Esta situação é agravada pelo fato de ser uma aplicação muito específica e que não possui um grande público-alvo.

### Restrições Arquiteturais

As restrições impostas ao projeto que afetam sua arquitetura são:

- O frontend deverá ser desenvolvido em AngularJS / ReactJS / Flutter
- O backend deverá ser adaptado para utilizar um sistema pronto (firebase)
- O banco de dados deverá ser moldado por documentos
- Na empresa do cliente trabalha-se com aplicações web
- O sistema deverá estar hospedado em um serviço gratuito inicialmente

### Design Thinking

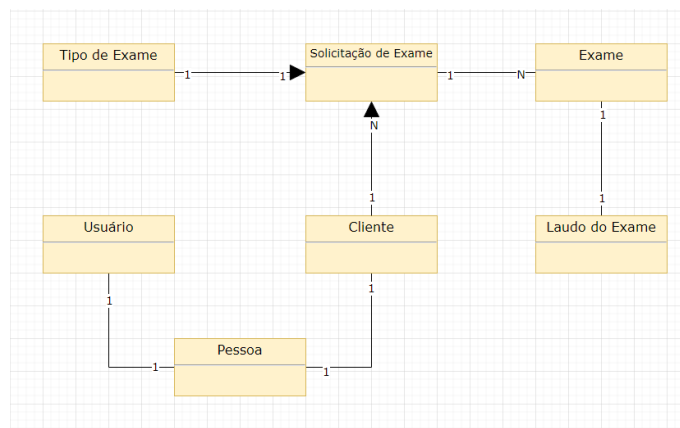
**Definição:** Os laboratórios que realizam exames não possuem um sistema que consiga realizar cadastros e exames de forma personalizada e de acordo com os produtos oferecidos pela empresa.

**Ideação:** 1) Aplicação web que atenderia as dores do cliente, criando um sistema do zero para prover todas as funcionalidades; 2) Sistema local utilizando planilhas do excel para automatizar o processo de solicitação e preenchimento de exames.

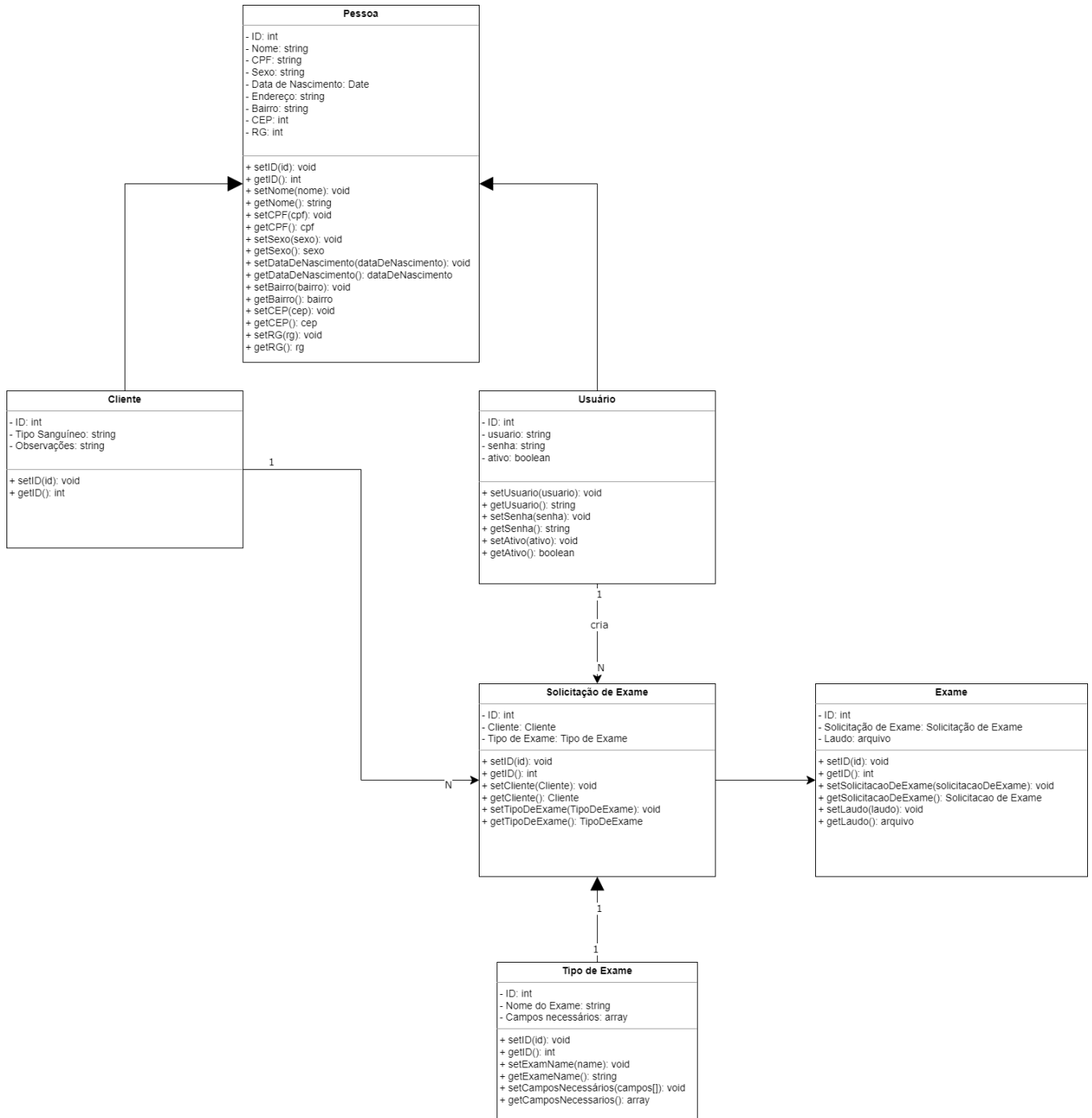
### Mecanismos Arquiteturais

| Análise (quais mecanismos) | Design (solução para o mecanismo)                                  | Implementação (ferramenta, recurso tecnológico) |
|----------------------------|--|---|
| Persistência               | Banco de Dados por documentos                                      | MySQL   |
| Front end                  | Camada responsável pelo contato direto entre usuário e sistema     | AngularJS / ReactJS                             |
| Back end                   | Camada responsável por validar dados e gerir processos             | Spring Boot                                     |
| Integração                 | -  | -   |
| Log do sistema             | Implementação dos recursos de log do componente de persistência    | -   |
| Teste de Software          | Testes automatizados e de caixa preta                              |   |
| Fluxo de Trabalho          | Organização de histórias do projeto e fluxo de trabalho controlado | Teams do Github, Notion                         |

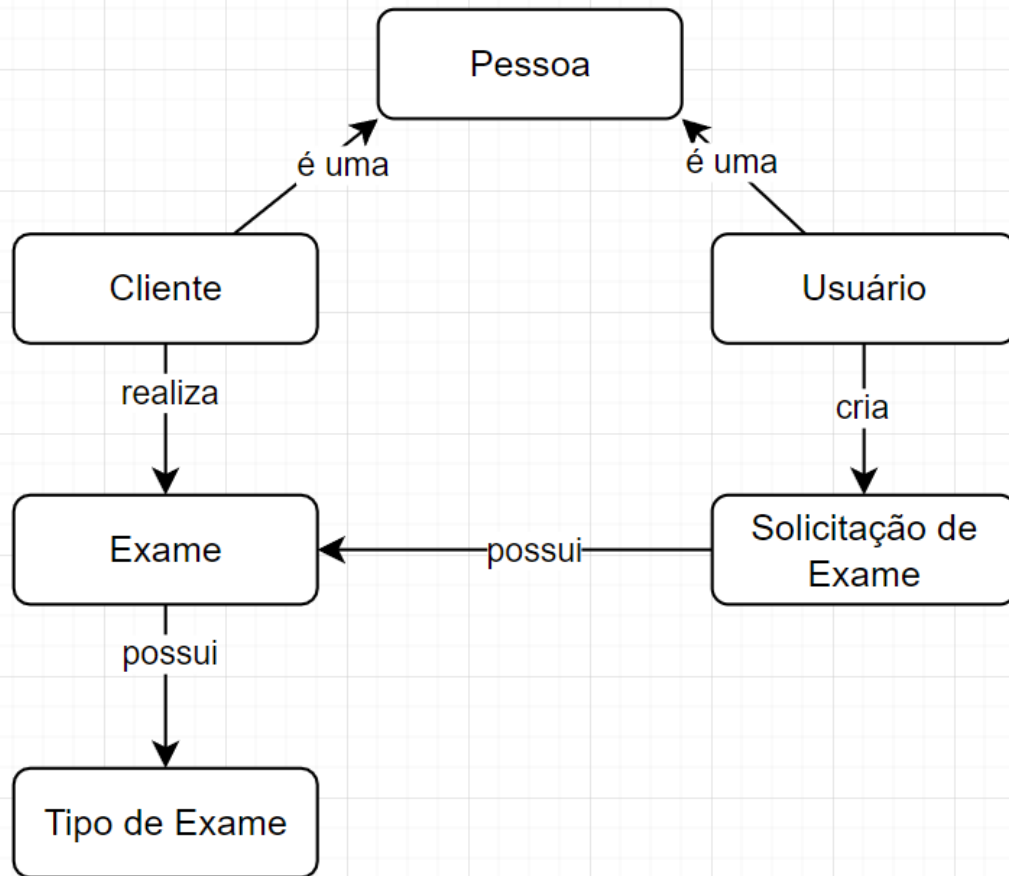
### Diagrama de classes conceitual



## Diagrama de classes Lógico



### Modelo para Banco de dados não relacional



### Casos de uso

#### UC01 – Listar Solicitações de Exames

|                              |  |
|------------------------------|--|
| <b>Descrição</b>             | Listar as Solicitações de Exames cadastradas pelo usuário do sistema |
| <b>Atores</b>                | Sistema  |
| <b>Prioridade</b>            | Alta   |
| <b>Requisitos associados</b> | RF008, RF020   |

|                              |  |
|------------------------------|--|
| <b>Fluxo Principal</b>       | Ir para tela de Solicitações de Exames<br>Listar as Solicitações de Exames |
| <b>Caminhos alternativos</b> | Não listar os itens (erro)   |

#### UC02 – Cadastrar Cliente

|                              |   |
|------------------------------|---|
| <b>Descrição</b>             | Cadastrar Cliente no sistema                                |
| <b>Atores</b>                | Usuário   |
| <b>Prioridade</b>            | Alta  |
| <b>Requisitos associados</b> | RF001, RF013  |
| <b>Fluxo Principal</b>       | Entrar na tela de cadastro<br>Criar um novo Cliente         |
| <b>Caminhos alternativos</b> | Criar um Cliente já existente<br>Dados do Cliente inválidos |

#### UC03 – Cadastrar Solicitação de Exame

|                              |  |
|------------------------------|--|
| <b>Descrição</b>             | Cadastrar uma nova solicitação de exame no sistema   |
| <b>Atores</b>                | Usuário  |
| <b>Prioridade</b>            | Alta   |
| <b>Requisitos Associados</b> | RF005, RF017   |
| <b>Fluxo Principal</b>       | Entrar na tela de cadastro de solicitação de exame<br>Criar uma nova solicitação<br>Preencher dados da solicitação |
| <b>Caminhos Alternativos</b> | Dados inválidos para solicitação de exame  |

#### UC04 – Cadastrar Tipos de Exames

|                  |   |
|------------------|---|
| <b>Descrição</b> | Cadastrar um novo tipo de exame que o sistema poderá utilizar na criação do exame |
|------------------|---|

|                              |   |
|------------------------------|---|
| <b>Atores</b>                | Usuário, Sistema  |
| <b>Prioridade</b>            | Alta  |
| <b>Requisitos Associados</b> | RF031, RF035, RF036   |
| <b>Fluxo Principal</b>       | Ir para tela de criação de tipo de exame<br>Criar o tipo de exame<br>Salvar |
| <b>Caminhos Alternativos</b> | Tipo de Exame já criado<br>Erro ao Salvar                                   |

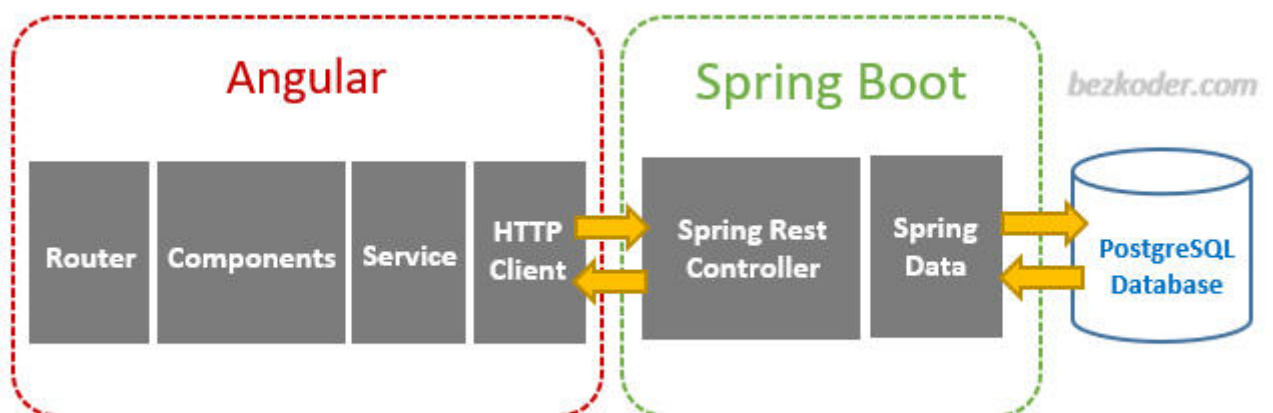
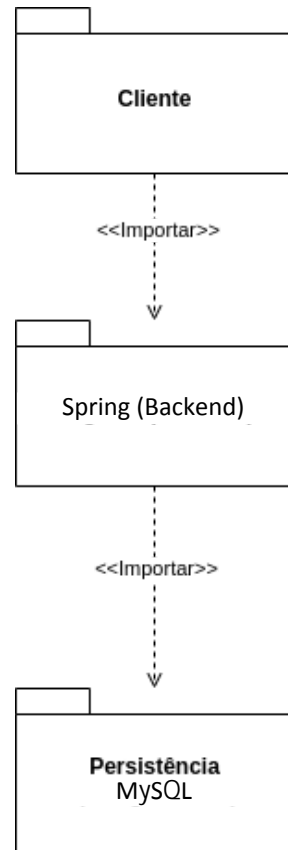
#### UC05 – Gerar Relatório de Exames

|                              |   |
|------------------------------|---|
| <b>Descrição</b>             | Gerar relatório de cada exame realizado na solicitação de exames          |
| <b>Atores</b>                | Sistema   |
| <b>Prioridade</b>            | Alta  |
| <b>Requisitos Associados</b> | RF030   |
| <b>Fluxo Principal</b>       | Gerar relatório a partir de um exame realizado<br>Anexar arquivo no exame |
| <b>Caminhos Alternativos</b> | Erro na geração do relatório<br>Exame preenchido incorretamente           |

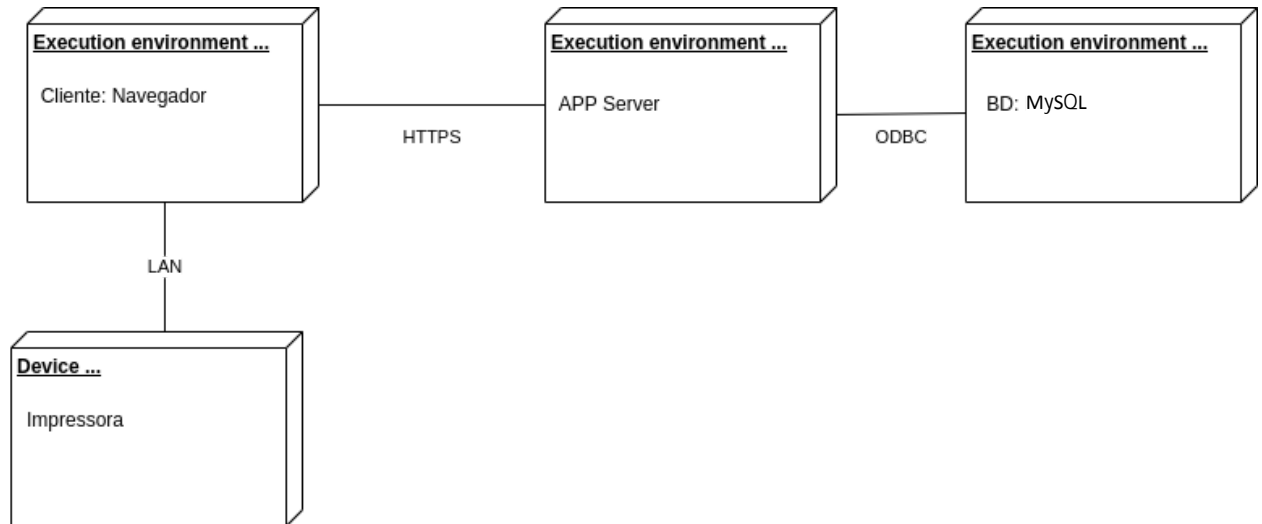
#### UC06 - Registro de atividades

|                              |   |
|------------------------------|---|
| <b>Descrição</b>             | Registrar alterações e atividades realizadas pelo usuário em um LOG |
| <b>Atores</b>                | Sistema   |
| <b>Prioridade</b>            | Média   |
| <b>Requisitos Associados</b> | RF035   |
| <b>Fluxo Principal</b>       | Registrar atividade em cada alteração realizada                     |
| <b>Caminhos Alternativos</b> | Falhar no registro<br>Usuário cancelar alteração                    |

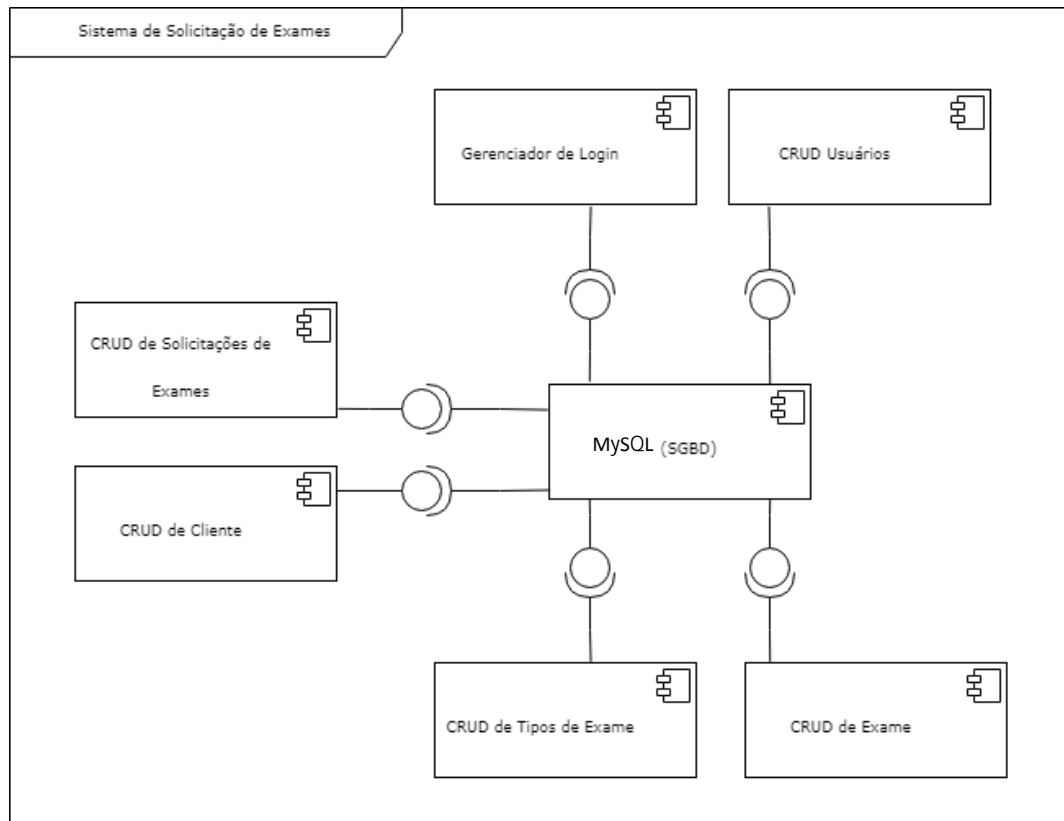
## Modelagem e projeto arquitetural



## Deployment Diagram

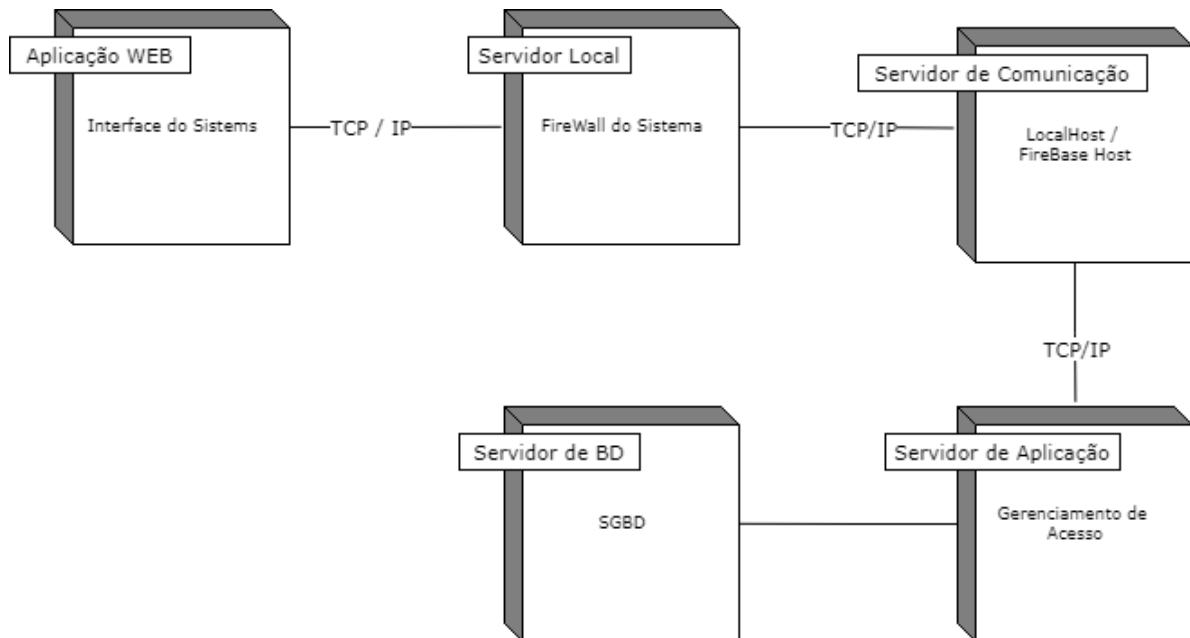


## Diagrama de componentes





### Diagrama de Implantação com Componentes



### Front-end e Back-end

A implementação de referência da camada de apresentação é baseada na utilização dos seguintes frameworks:

- Angular (<https://angular.io/>) – Angular é um framework de desenvolvimento web de código aberto mantido pelo Google. Ele é usado para construir aplicações web modernas e ricas em recursos, seguindo o padrão MVC (Model-View-Controller) ou, mais especificamente, o padrão MVVM (Model-View-ViewModel). O Angular é escrito em TypeScript, que é uma linguagem de programação baseada em JavaScript, com recursos adicionais para adicionar tipagem estática e recursos modernos ao JavaScript.
- Spring Boot (<https://spring.io/projects/spring-boot>) – O Spring Boot é um projeto do ecossistema Spring Framework desenvolvido pela Pivotal Software (anteriormente uma divisão da VMware e, mais tarde, da Dell Technologies). É uma estrutura (framework) que tem como objetivo simplificar o processo de configuração e desenvolvimento de aplicativos Java, especialmente aplicativos baseados em Spring.
- MySQL (<https://www.mysql.com/>) – O MySQL é um sistema de gerenciamento de banco de dados relacional (RDBMS - Relational Database Management System) de código aberto. Ele é uma das opções mais populares e amplamente utilizadas para armazenar, gerenciar e recuperar dados de forma eficiente e segura. Desenvolvido pela empresa sueca MySQL AB (adquirida posteriormente pela Sun Microsystems, que por sua vez foi adquirida pela Oracle Corporation), o MySQL é conhecido por sua confiabilidade, desempenho e facilidade de uso.