

**Star**  
**Documento de Arquitetura de Software**

**Versão 1.0**

Star	Versão: 1.0
Documento de Arquitetura de Software	Date: 2022

#### Histórico da Revisão

Data	Versão	Descrição	Autor
29/05/2022	1.0	Início da elaboração da documentação	Caroline, Felipe, Brendon e Nicássio
02/06/2022	1.1	Edição e elaboração de campos	Caroline, Brendon, Felipe e Nicássio
28/06/2022	1.2	Edição do documento	Caroline, Brendon
03/07/2022	1.3	Finalização do documento	Caroline, Felipe

Star	Versão: 1.0
Documento de Arquitetura de Software	Date: 2022

## Índice Analítico

1.	Introdução	4
1.1	Finalidade	4
1.2	Escopo	4
1.3	Definições, Acrônimos e Abreviações	4
1.4	Visão Geral	4
2.	Representação Arquitetural	4
3.	Metas e Restrições da Arquitetura	4
4.	Visão de Casos de Uso	5
5.	Visão Lógica	5
5.1	Visão Geral	5
5.2	Pacotes de Design Significativos do Ponto de Vista da Arquitetura	6
6.	Visão de Processos	6
7.	Visão de Implantação	6
8.	Visão da Implementação	6
8.1	Visão Geral	7
8.2	Camadas	7
9.	Visão de Dados (opcional)	7
10.	Tamanho e Desempenho	7
11.	Qualidade	7

Star	Versão: 1.0
Documento de Arquitetura de Software	Date: 2022

# Documento de Arquitetura de Software

## 1. Introdução

### 1.1 Finalidade

Este documento oferece uma visão geral arquitetural abrangente do sistema Star, usando diversas visões arquiteturais para representar diferentes aspectos do sistema. O objetivo deste documento é capturar e comunicar as decisões arquiteturais significativas que foram tomadas em relação ao sistema.

### 1.2 Escopo

Este Documento de Arquitetura de Software se aplica ao sistema Star, que será desenvolvido pela equipe: Felipe, Caroline, Brendon e Nicássio. O documento auxilia os envolvidos no projeto a captar aspectos arquiteturais do sistema que são necessários para o desenvolvimento de uma solução que atenda às necessidades dos usuários finais. Além de auxiliar no entendimento do sistema por novos membros da equipe.

### 1.3 Definições, Acrônimos e Abreviações

MVC – Padrão de arquitetura de software onde M significa modelo sendo responsável pela parte de regras de negócio, V a visualização responsável pela parte de interfaces e C a parte de controle dos dados.

### 1.4 Visão Geral

São apresentados ainda neste documento diferentes visões arquiteturais de como o sistema deve se comportar em diferentes processos, como deve ser implantado é implementado e restrições de desempenho e qualidade.

## 2. Representação Arquitetural

Esta seção descreve a representação arquitetural do projeto, sendo:

- **Visão de Casos de Uso** - Apresenta as funcionalidades arquiteturais importantes e os usuários do sistema.
- **Visão Lógica** - Descreve as classes e sua organização e apresenta o padrão de arquitetura que deverá ser utilizado para o desenvolvimento do sistema.
- **Visão de Processos** - Mostra o padrão de comportamento do sistema diante de diferentes ações do usuário.
- **Visão de Implantação** - Descreve a estrutura do ambiente onde o software será instalado.
- **Visão de Implementação** - Ilustra a distribuição do processamento em um conjunto de nós do sistema, incluindo a distribuição física dos processos e threads.

## 3. Metas e Restrições da Arquitetura

Esta seção descreve os requisitos de software e restrições que impactam significativamente na arquitetura do projeto.

- Estrutura MVC;
- Linguagem Typescript;
- Framework NodeJs, React;
- O sistema em questão deverá ser multiplataforma;

Star	Versão: 1.0
Documento de Arquitetura de Software	Date: 2022

## 4. Visão de Casos de Uso

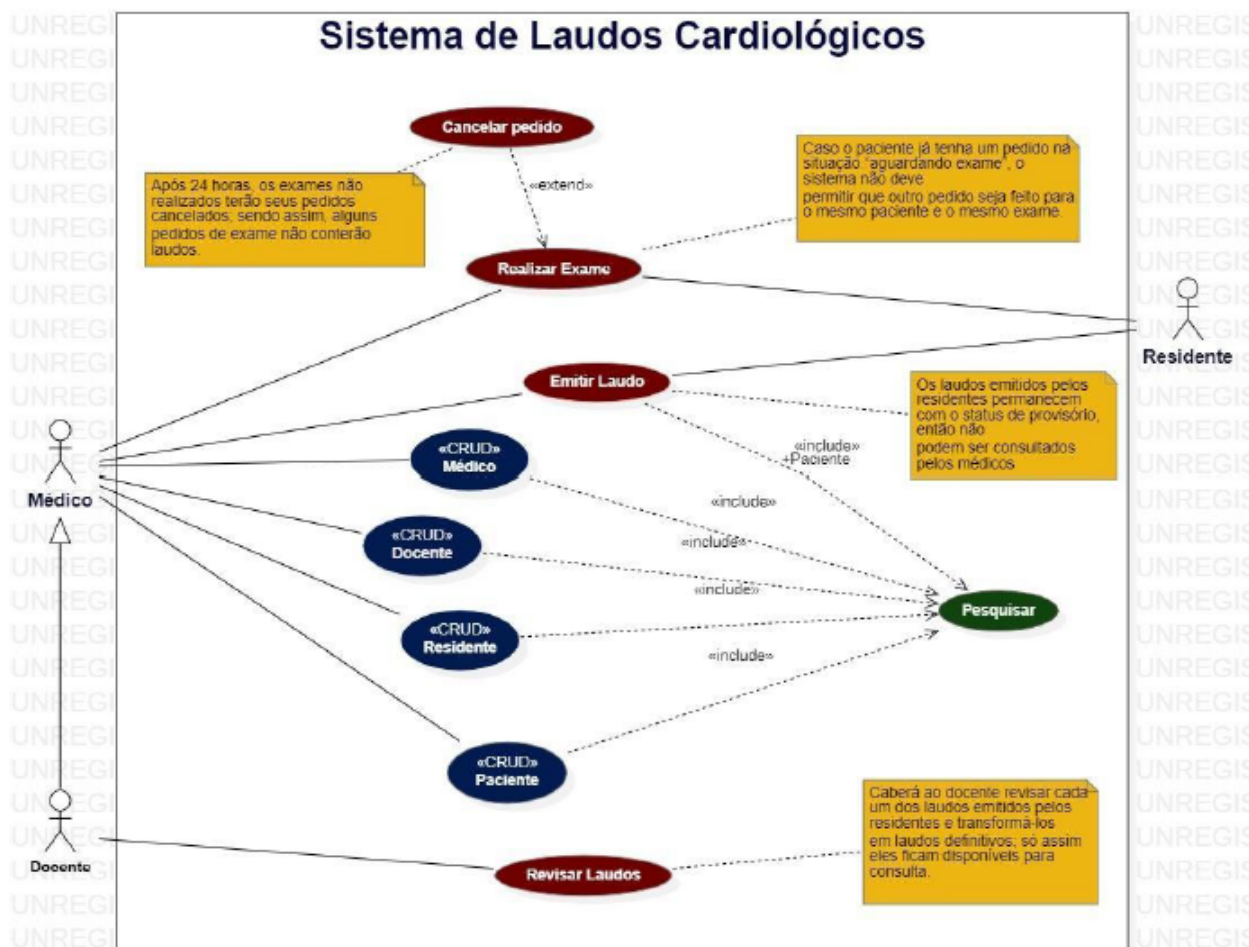


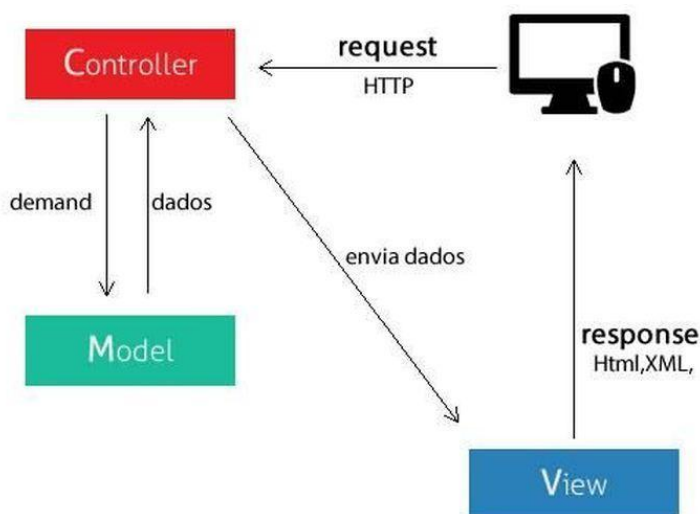
Figura 1 - Casos de uso

## 5. Visão Lógica

### 5.1 Visão Geral

A visão lógica define a estrutura da arquitetura. Abaixo será especificado o padrão utilizado para o desenvolvimento do sistema, no caso, MVC. O padrão foi escolhido, pois possui melhor nível de sustentabilidade, facilitando assim a manutenção da aplicação. Proporciona também uma melhor performance do projeto, separando a estrutura geral em camadas. Permite que os desenvolvedores e designers responsáveis pelo projeto consigam trabalhar em paralelo, assim como, partes da aplicação podem ser modificadas sem a necessidade de alterar outras.

Star	Versão: 1.0
Documento de Arquitetura de Software	Date: 2022

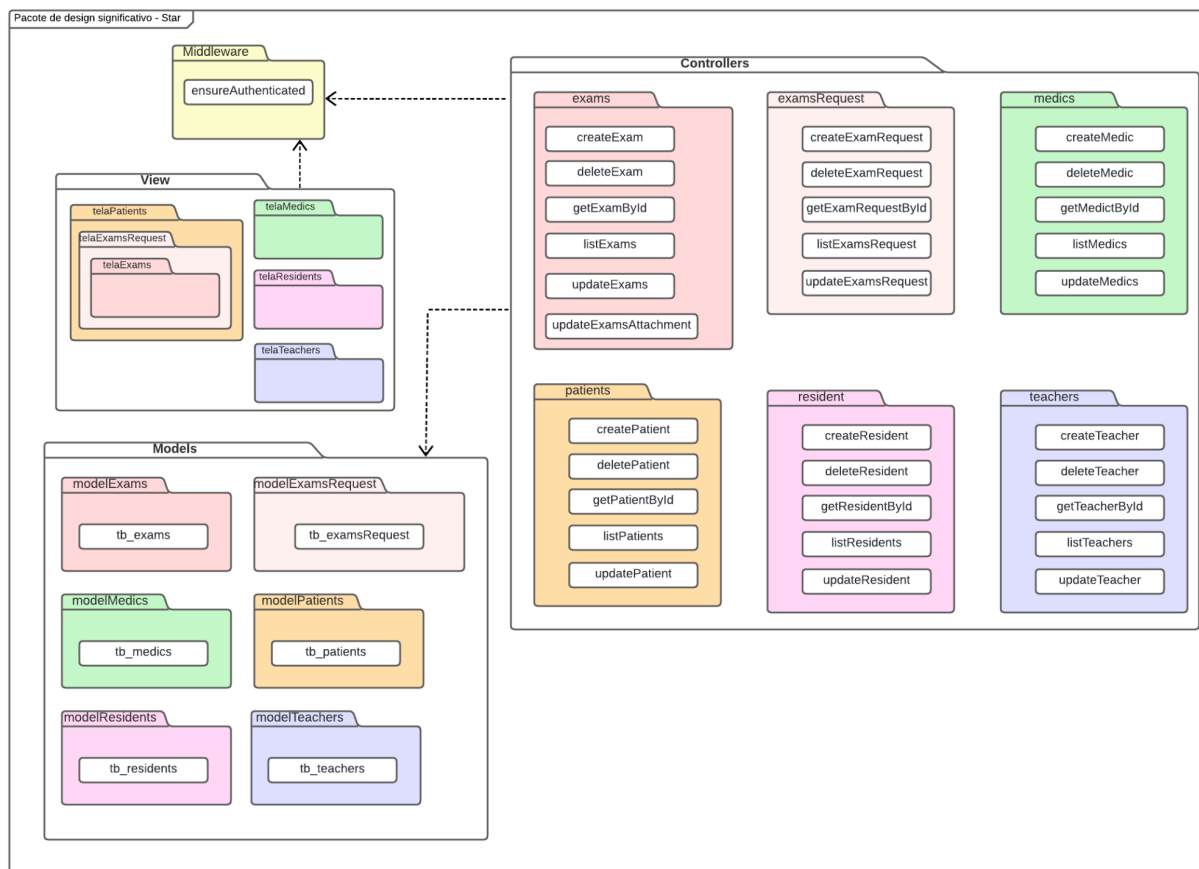


**Figura 2 - Representação padrão MVC**

- **View:** também conhecida como cliente-side. View é a camada de interface com o usuário e tudo que engloba a representação dos dados do projeto.
- **Model:** é a camada usualmente portada em JSON, que contém a estrutura de dados atrás de uma parte específica da aplicação. É responsável pela leitura, manipulação e validação de dados, aos quais são obtidos e traduzidos em informações relevantes para serem exibidas pela View. Também notifica a view e controller associados quando há mudança em seu estado.
- **Controller:** é a camada de controle que exerce o controle de qual Model deverá ser aplicado e qual View será mostrado ao usuário. O controller manipula e roteia as requisições dos usuários. Também interpreta as requisições submetidas pelo usuário e traduz em comandos que são enviados para o Model e/ou para o View. Podemos dizer que esta camada faz uma gerência das outras duas camadas.

Star	Versão: 1.0
Documento de Arquitetura de Software	Date: 2022

## 5.2 Pacotes de Design Significativos do Ponto de Vista da Arquitetura



## 6. Visão de Implantação

O Star é uma aplicação WEB o qual seguirá um padrão de Cliente-Servidor, toda sua implantação é composta em três fases: Cliente, que faz acesso da aplicação; Servidor da Aplicação, onde a partir de uma requisição HTTP/HTTPS ele faz a interação com o Star assim como suas regras de negócios, tendo acesso aos recursos do mesmo e a camada de Banco de dados que é o local onde estarão armazenados todos os dados que nele foi salvo, todos se utilizam da camada TCP/IP para comunicação.

## 7. Visão da Implementação

O Star contará com 3 níveis de acesso, e cada um deles estão divididos em dois níveis arquiteturais o primeiro segue-se no modelo cliente-servidor, onde segue as regras de apresentação, regras de negócio e dados, e o segundo nível de arquitetura é o MVC, o qual contará com os padrões de boas práticas do NodeJs, seguindo as regras das rotas e as três camadas view, model e controller. As rotas responsáveis em direcionar os usuários as view's requisitadas através de sua comunicação com o controle que estará sempre em comunicação com o model que é o responsável pelas persistências com o banco de dados.

## 8. Tamanho e Desempenho

O desempenho do Star está relacionado a velocidade do sistema e o retorno para o usuário, fazendo com o que o usuário final não espere muito tempo para ter suas respostas. O desempenho está relacionado a:

Star	Versão: 1.0
Documento de Arquitetura de Software	Date: 2022

1. **Tempo de resposta:** deverá retornar o resultado no tempo médio de 1 segundo podendo chegar no máximo 3 segundos para concluir uma transação.
2. **Taxa de Transferência:** Os processos não podem ultrapassar 6 segundos para realizar qualquer ação.

## 9. Qualidade

1. **Estrutura MVC:** Desenvolvido na arquitetura com as camadas de Model, View e Controle;
2. **Linguagens:** Desenvolvido em Typescript;
3. **Banco de dados:** Modelado para banco Postgres;
4. **Frameworks:** NodeJs e React;
5. **IDE:** Visual Studio Code.