

# Estorquestrador

## ARQUITETURA DE SISTEMA DE GESTÃO DE ESTOQUE

### Participantes:

Rafael Alves Rabelo

DTI

Histórico de Revisões			
Data	Autor	Descrição	Versão
[18/12/2019]	Rafael Alves Rabelo	Define a base de construção/organização do sistemas	[NDF]
[20/12/2019]	Rafael Alves Rabelo	Atualiza definições de persistência de dado	



## SUMÁRIO

<b>Apresentação</b>	<b>4</b>
Problema	4
Objetivos	4
<b>Requisitos</b>	<b>4</b>
Restrições Arquiteturais	4
Mecanismos Arquiteturais	5
<b>Modelagem e projeto arquitetural</b>	<b>5</b>
Visão de Casos de Uso	5
Descrição resumida dos casos de uso	6
Visão Física	7
Modelo de dados	8
<b>APÊNDICES</b>	<b>8</b>

## **Apresentação**

*Eu como estoquista desejo cadastrar/alterar e excluir produtos no meu sistema de estoque.*

*Desejo as seguintes informações:*

*Nome do produto*

*Quantidade de itens do produto*

*Valor unitário do produto*

*Desejo que seja disponível uma tela de listagem onde eu possa selecionar um produto para exclusão ou*

*edição (exibir uma nova tela para edição).*

### **1.1 Problema**

Gestão de itens de estoque (CRUD) : Facilitar a gestão e controle de estoque

### **1.2 Objetivos**

Este visa entregar uma definição de organização de sistema distribuído, cuja a finalidade é permitir o gerenciamento estoque em volume de forma eletrônica por meio de interface de usuário, além de prover arquitetura para ampliação e persistência dos dados de forma centralizada(banco de dados)

## **2 Requisitos**

- O sistema deve ser escalável;
- O sistema deve ter fácil manutenibilidade;
- O sistema deve permitir integrações;
- O sistema deve ser organizado em camadas;
- O sistema de utilizar padrões orientados a serviço ( REST) integráveis;
- Os endpoints/serviços devem ter capilaridade/granularidade de tamanho médio ou pequena;
- O sistema deve ser testável e reutilizável.
- O sistema deve persistir dados;

## 2.1 Restrições Arquiteturais

- O Backend deve ser desenvolvido em linguagem C# na plataforma .NET CORE;
- O Front-end deve ser desenvolvido em tecnologia React;
- O sistema deve ser otimizado como SPA de acordo com a tecnologia React;
- O Backend deve utilizar padrão REST WebApi(Restfull) para troca de mensagens;
- O sistema deve gerenciar as persistências(DB) utilizando framework de persistência Dapper;
- A comunicação React/Front com o backend deve ser suportada pela dependência Axios.
- O sistema deve ser desenvolvido utilizando o padrão de projeto IOC e MVC no Backend;
- O sistema deve utilizar um SGDB relacional MSSQL.
- O sistema deve se comunicar via HTTP(REST) com Objetos Json;

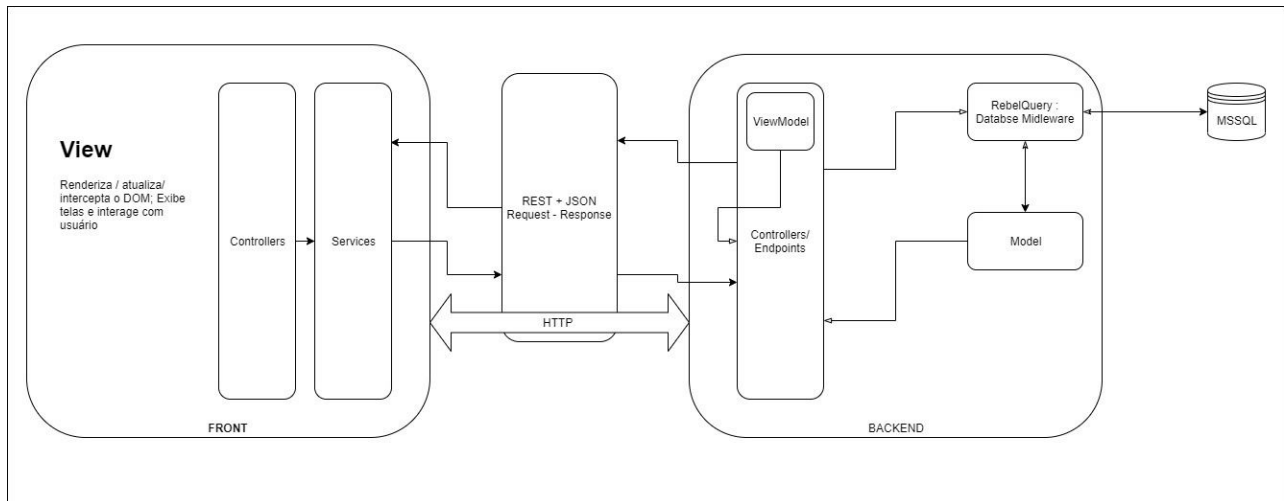
## 2.2 Mecanismos Arquiteturais

Análise	Projeto ( <i>Design</i> )	Implementação
Persistência	ORM	Dapper Frameork
Front end	MVC	JS-Javascript / React
Back end	MVC	.Net Core / C#
Integração	Microserviços	Rest
Transações/Comunicação	HTTP	JSON

Tabela 1: Mecanismos arquiteturais.

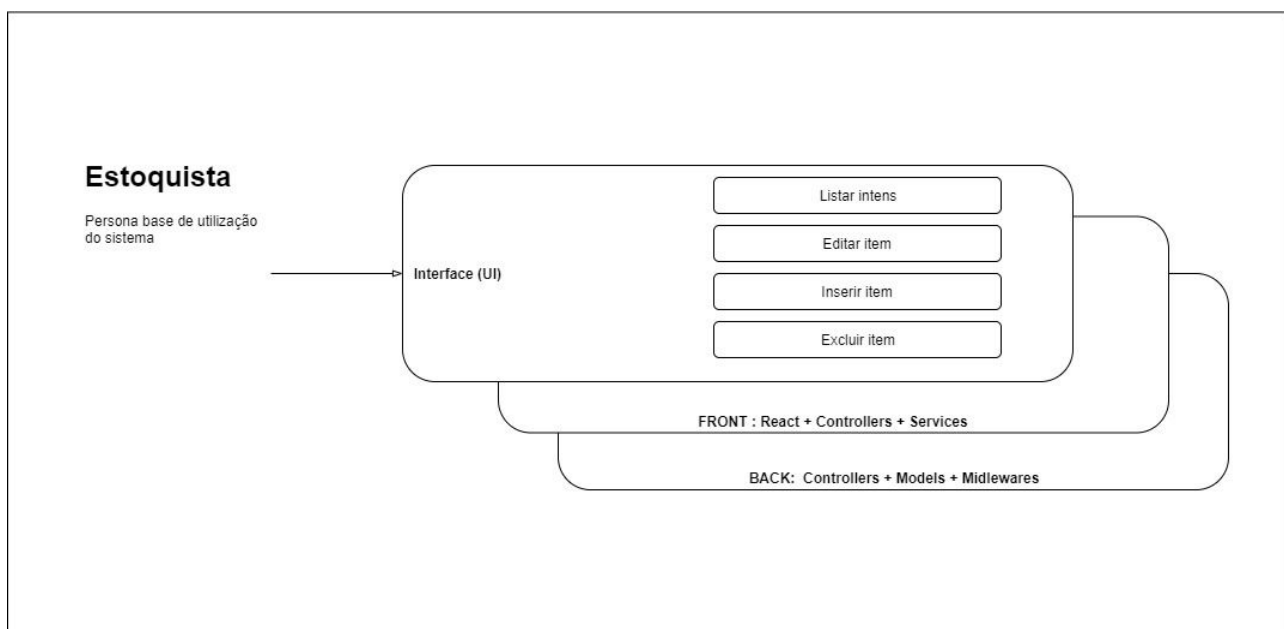
## 3 Modelagem e projeto arquitetural

I.



**Figura 1 - Visão Geral da Solução**

### 3.1 Visão de Casos de Uso



**Figura 2 – Diagrama de Caso de Uso – Usuário**

#### 3.1.1 Descrição resumida dos casos de uso

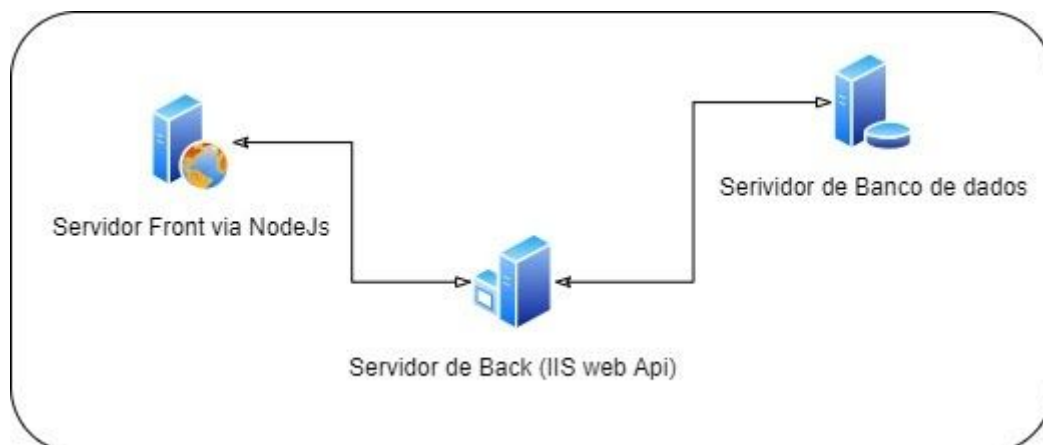
UC01 – Listar itens	
<b>Descrição</b>	O usuário visualiza a lista de itens que lhe permite editar ou excluir
<b>Atores</b>	Estoquista
<b>Prioridade</b>	Alta
<b>Fluxo Principal</b>	O ator abre o sistema > A lista de itens é exibida;

UC02 – Excluir item	
<b>Descrição</b>	O usuário seleciona excluir um item na lista de itens
<b>Atores</b>	Estoquista
<b>Prioridade</b>	Alta
<b>Fluxo Principal</b>	Na lista de itens o ator clica em <b>excluir</b> > o autor <b>confirma</b> > o sistema executa a exclusão;

UC03 – Editar item	
<b>Descrição</b>	O usuário seleciona editar um item na lista de itens
<b>Atores</b>	Estoquista
<b>Prioridade</b>	Alta
<b>Fluxo Principal</b>	Na lista de itens o ator clica em <b>editar</b> > o autor altera o formulário de dados exibido> o autor <b>confirma</b> as alterações clickando em <b>salvar</b> > O sistema atualiza os dados

UC03 – Inserir item	
<b>Descrição</b>	O usuá seleciona editar um item na lista de itens
<b>Atores</b>	Estoquista
<b>Prioridade</b>	Alta
<b>Fluxo Principal</b>	Na tela de lista de itens o ator clica em <b>adicionar</b> > o autor preencheu o formulário de dados exibido> o autor salva o registro clickando em salvar > O sistema insere os dados

### 3.2 Visão Física



### 3.3 Modelo de dados



*Modelo MER - Não se aplica*

*Diagrama DER - Não se aplica*



## **APÊNDICES**

Documento de Especificação : Documento de Especificação.pdf