

Estoquestrador

ARQUITETURA DE SISTEMA DE GESTÃO DE ESTOQUE

Participantes:

Rafael Alves Rabelo

DTI

Histórico de Revisões			
Data	Autor		Versão
[18/12/2019]	Rafael	Alves	[NDF]
	Rabelo		

SUMÁRIO

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

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)

1.2 Objetivos

Este visa entregar uma definição de organização de sistema distribuído, cuja a finalidade é gerenciar estoque em volumes.

2 Requisitos

- O sistema deve ser escalável; Arquitetura de microserviços.
- O sistema deve utilizar arquitetura e tecnologias de camadas.
- O sistema de utilizar padrões orientados a serviço (REST).
- O sistema deve se comunicar via HTTP(REST) com Objetos Json.
- O sistema deve Suportar de fácil manutenibilidade.
- O sistema deve separar responsabilidades em camadas MVC
- Os endpoints/serviços devem ter capilaridade/granularidade de tamanho médio ou pequena.

2.1 Restrições Arquiteturais

- O Backend deve ser desenvolvido em linguagem C# .

- O Front-end deve ser desenvolvido em tecnologia React.
- O Backend deve utilizar padrão REST WebApi(Restfull) para troca de mensagens.
- O sistema gerenciar as persistências(DB) utilizando por algum framework de persistência.
- O sistema deve utilizar um SGDB relacional MSSQL.

2.2 Mecanismos Arquiteturais

Análise	Projeto (Design)	Implementação
Persistência	Lib	RebelQuery Package
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.

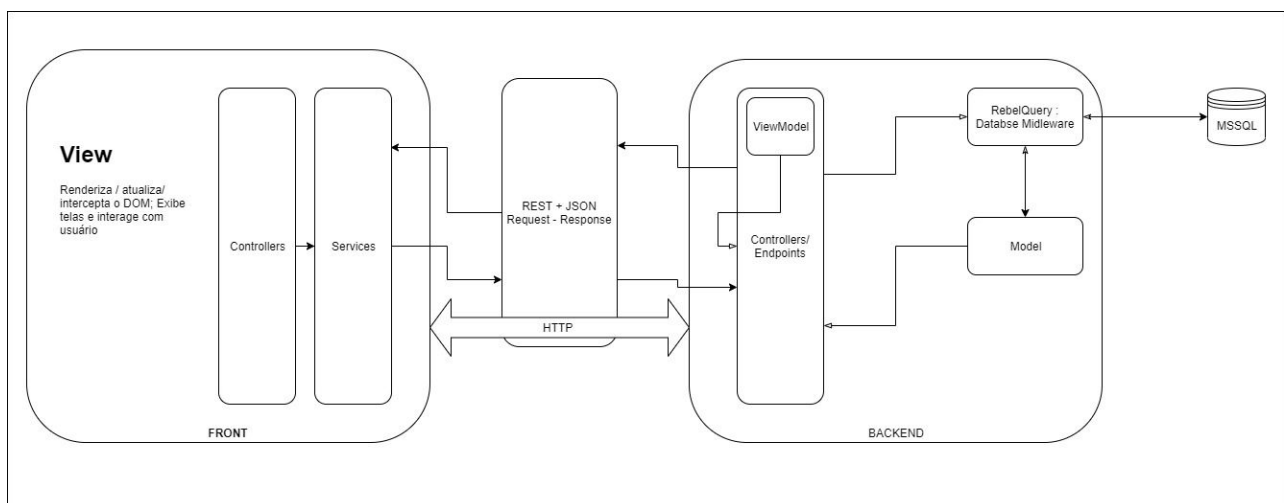


Figura 1 - Visão Geral da Solução

3.1 Visão de Casos de Uso

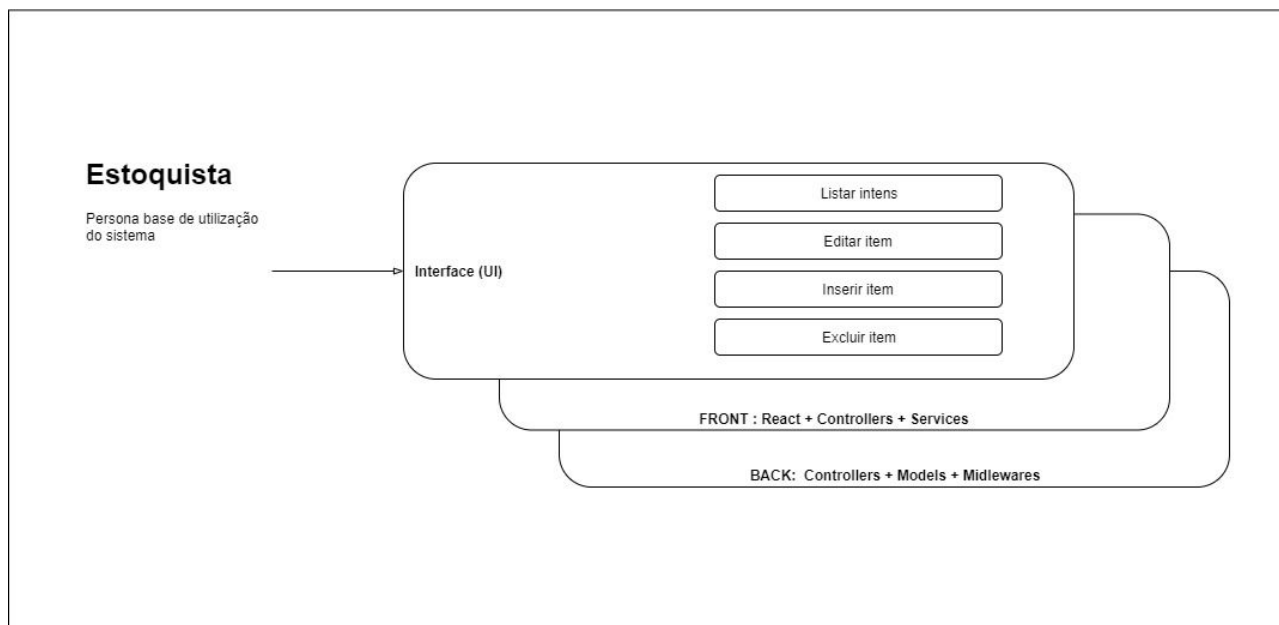


Figura 2 – Diagrama de Caso de Uso – Usuário

3.1.1 Descrição resumida dos casos de uso

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

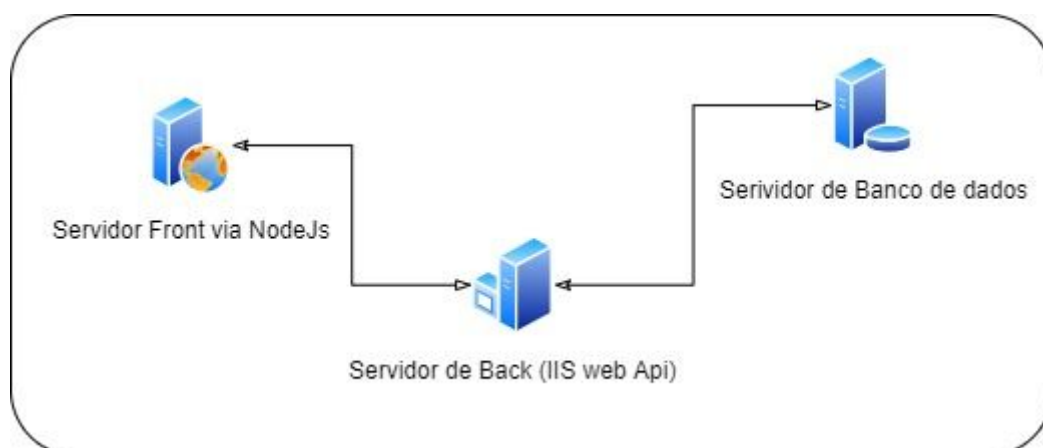
UC02 – Excluir item	
Descrição	O usuário seleciona excluir um item na lista de itens
Atores	Estoquista
Prioridade	Alta
Requisitos associados	
Fluxo Principal	Na lista de itens o ator clica em excluir > o autor confirma > o sistema executa a exclusão;

UC03 – Editar item	
Descrição	O usuário seleciona editar um item na lista de itens
Atores	Estoquista

Prioridade	Alta
Requisitos associados	
Fluxo Principal	Na lista de itens o ator clicka em editar > o autor altera o formulário de dados exibido> o autor confirma as alterações clickando em salvar > O sistema atualiza os dados

UC03 – Inserir item	
Descrição	O usuá seleciona editar um item na lista de itens
Atores	Estoquista
Prioridade	Alta
Requisitos associados	
Fluxo Principal	Na tela de lista de itens o ator clicka em adicionar > o autor preenche oo 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