

# Projeto de Desenvolvimento de Aplicações Distribuídas

## Banco de Alimentos



Participantes:  
Amanda Paloma  
Elaine Melo

Orientador:  
Luiz Alberto F.  
Gomes

- I. Etapa 1 – Definição Conceitual da Solução
  - 1. Introdução
    - a. Problema
    - b. Meta
  - 2. Requisitos Funcionais
  - 3. Requisitos Não Funcionais
  - 4. Restrições
  - 5. Mecanismos Arquiteturais
- II. Etapa 2 – Modelagem e projeto arquitetural
  - 1. Diagrama de contexto
  - 2. Diagrama de contêiner
  - 3. Diagrama de componentes
  - 4. Abordagens e ferramentas de testes
  - 5. APIs

# Definição Conceitual da Solução



Apesar de figurar entre os principais provedores de alimentos do mundo, o Brasil possui profundas desigualdades sociais que levam parte de sua população a situações de insegurança alimentar (IA).

## ● Problema

- Falta de gerenciamento no cadastro de beneficiários para programas relacionados a insegurança alimentar em um município.
- Carência de controle no sistema de distribuição de cestas, para os cadastrados.

## ● Meta

- Desenvolver um sistema de cadastro de beneficiários, que possa inserir dados das famílias em bancos e que elas possam ser avisadas quando sua cesta estiver disponível.



# Requisitos funcionais



ID	Descrição do Requisito	Prioridade
RF-001	Permitir que o usuário faça login de entrada	ALTA
RF-001	Permitir que o usuário faça logoff do sistema	ALTA
RF-003	Permitir que o administrador cadastre os usuários do sistema	ALTA
RF-004	Permitir que o administrador consulte, altere e exclua o cadastro dos usuários do sistema	ALTA
RF-005	Permitir que o usuário cadastre os beneficiários	ALTA
RF-006	Permitir que o usuário consulte, altere e exclua o cadastro dos beneficiários	ALTA
RF-007	Permitir que o usuário registre as retiradas de cestas pelo beneficiário	ALTA
RF-008	Restringir a retirada de cesta, conforme periodicidade estabelecida pelo município	ALTA

ID	Descrição do Requisito	Prioridade
RF-009	Permitir que o usuário visualize histórico de retirada dos beneficiários	MÉDIA
RF-0010	Permitir que o usuário realize a redefinição de senha	MÉDIA
RF-0011	Consultar data de recebimento de cesta	BAIXA
RF-012	Consultar local de retirada de cesta	ALTA
RF-013	Consultar recebimento de cesta	ALTA
RF-014	Permitir que após o usuário gravar os dados de retirada de cesta, mensagens assíncronas sejam enviadas para o histórico de consulta de retiradas do portal do Beneficiário.	ALTA
RF-015	Permitir que mensagens de erro sejam enviadas ao usuário quando a mensagem não for enviada para o histórico de consulta do portal do Beneficiário	ALTA
RF-016	Fornecer a outros sistemas parceiros informações de famílias beneficiadas através da consulta pelo CPF	ALTA
RF-017	Permitir o que o usuário cadastre os dependentes dos beneficiários no sistema	MÉDIA



## Usabilidade

RNF-001	O sistema deve ser pensado para o usuário
Estímulo	O Usuário realiza ações específicas e objetivas no sistema.
Fonte do estímulo	Usuário.
Ambiente	Sistema com carga normal.
Artefato	Sistema como um todo.
Resposta	Fluxo do usuário sem interrupções e com fácil navegabilidade.
Medida de resposta	O usuário consegue finalizar as tarefas específicas, do início ao fim, no sistema.

## Disponibilidade

RNF-002	O sistema deve estar disponível em horário integral, durante 24h por dia e sete dias por semana.
Estímulo	Usuário acessa sistema no domingo no período da madrugada.
Fonte do estímulo	Usuário.
Ambiente	Sistema com carga normal.
Artefato	Sistema como um todo.
Resposta	O sistema deve permitir o acesso ao usuário em qualquer hora, seja do dia ou noite.
Medida de resposta	Todas tentativas de acesso e requisição de informações ao sistema devem ser respondidas.

## Desempenho

RNF-003	A latência total do sistema não deve ultrapassar 10 segundos
Estímulo	Usuário autenticado acessa uma funcionalidade na interface gráfica do sistema.
Fonte do estímulo	Usuário autenticado.
Ambiente	Sistema com carga normal.
Artefato	Módulos de informações cadastrais.
Resposta	O sistema processa a solicitação e retorna informação em até 10 segundos para o usuário.
Medida de resposta	Resposta dos sistema com até 10 segundos de duração.



## Modificabilidade

RNF-004	O sistema deve ser modular e componentizado, utilizando orientação a objetos e ser construído com arquitetura distribuída.
Estímulo	Construção e integração de um novo módulo com outros módulos da arquitetura distribuída.
Fonte do estímulo	Usuário autenticado.
Ambiente	Sistema com carga normal.
Artefato	Módulos de informações cadastrais.
Resposta	O sistema processa a solicitação e retorna informação em até 10 segundos para o usuário.
Medida de resposta	Qualquer intervenção de um ou mais módulos não gera reflexos generalizados no ambiente como um todo e a disponibilização de um novo módulo não interrompe os outros em funcionamento.

## Portabilidade

RNF-005	O sistema deve ser executado nos principais navegadores (Chrome, Mozilla e Edge)
Estímulo	Usuário acessando o sistema.
Fonte do estímulo	Usuário.
Ambiente	Sistema com carga baixa, normal ou além do normal.
Artefato	Módulos de informações cadastrais.
Resposta	A interface gráfica do projeto apresenta navegabilidade nas ações disponíveis ao usuário.
Medida de resposta	O usuário consegue ter acesso a interface gráfica do sistema.
RNF-006	O sistema deve ser responsivo para adaptação em dispositivos móveis
Estímulo	Usuário logado utilizando a interface gráfica do sistema.
Fonte do estímulo	Usuário logado.
Ambiente	Sistema com carga baixa, normal ou além do normal.
Artefato	Módulos de informações cadastrais.
Resposta	A interface gráfica do projeto apresenta navegabilidade e objetividade nas ações disponíveis ao usuário.
Medida de resposta	O usuário realiza suas ações rapidamente e sem desvios indesejados para outras funcionalidades.

## Segurança

RNF-007	O sistema deve exigir autenticação prévia de usuários
Estímulo	Usuário realiza ação de carregamento de uma página protegida ao qual necessita de login e senha para ser visualizada.
Fonte do estímulo	Usuário autenticado.
Ambiente	Sistema com carga normal.
Artefato	Módulo de autenticação.
Resposta	Entrada de usuário no sistema e acesso as páginas do sistema.
Medida de resposta	Todas as requisições com usuários e senhas corretos devem dar acesso ao usuário a área protegida do sistema.
RNF-008	O sistema deve possuir controle de acesso baseado em role
Estímulo	Usuário com perfil de administrador realiza cadastro de um novo usuário do sistema.
Fonte do estímulo	Usuário administrador.
Ambiente	Sistema com carga baixa, normal ou além do normal.
Artefato	Módulo de autenticação.
Resposta	Cadastro de usuário no sistema.
Medida de resposta	Ao verificar o banco de dados, o novo usuário deverá ter o perfil de acordo com o que foi cadastrado.



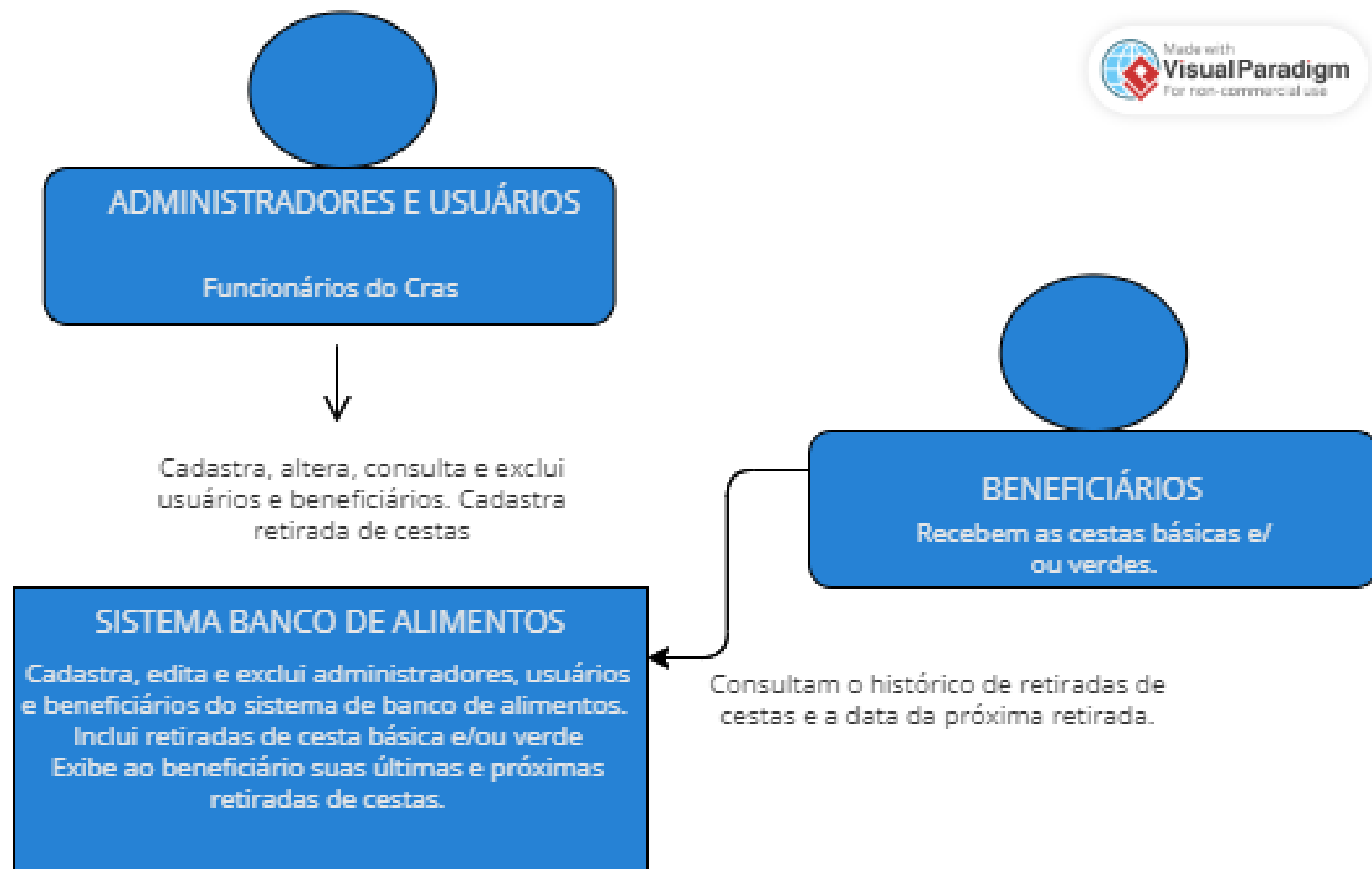
ID	Restrição
01	O projeto deverá ser entregue até o final do semestre
02	O front end deve ser desenvolvido com a linguagem de programação Javascript e para isso deve ser utilizado o framework React na versão mais recente.
03	Deve ser utilizado o framework Bootstrap para padronização de interface do usuário.
04	Como linguagem de folha de estilos deverá ser utilizado o CSS.
05	O sistema deve fazer uso de Media Queries e do sistema de Grid do Bootstrap para ajustar de forma responsiva em aparelhos de diferentes resoluções.
06	O sistema deverá expor funcionalidades e interoperar com outras aplicação legadas através do protocolo HTTP por meio de APIs RESTful.
07	O transporte de informações se dará por meio do formato JSON.
08	O mecanismo de autorização entre as APIs deverá ser feito por meio do JWT.
09	O sistema deve ser construído de forma modular para facilitar a implantação.
10	Os módulos que compõem os microsserviços no back end devem ser desenvolvidos na linguagem de programação C# utilizando .NET como ambiente de execução do lado do servidor.
11	Deve ser utilizado como ferramenta de versionamento o GIT e cada módulo deve possuir repositório distinto para melhor controle das modificações do código.
12	O sistema deve possuir características de aplicação distribuída: abertura, portabilidade, uso de recursos de rede.
13	Os módulos do sistema devem fazer uso do banco de dados relacional SQLServer.
14	O sistema deve ser desenvolvido utilizando recursos de gestão de configuração, com interação contínua, para isso deve ser utilizada a plataforma como serviço (PAAS) denominada Heroku.

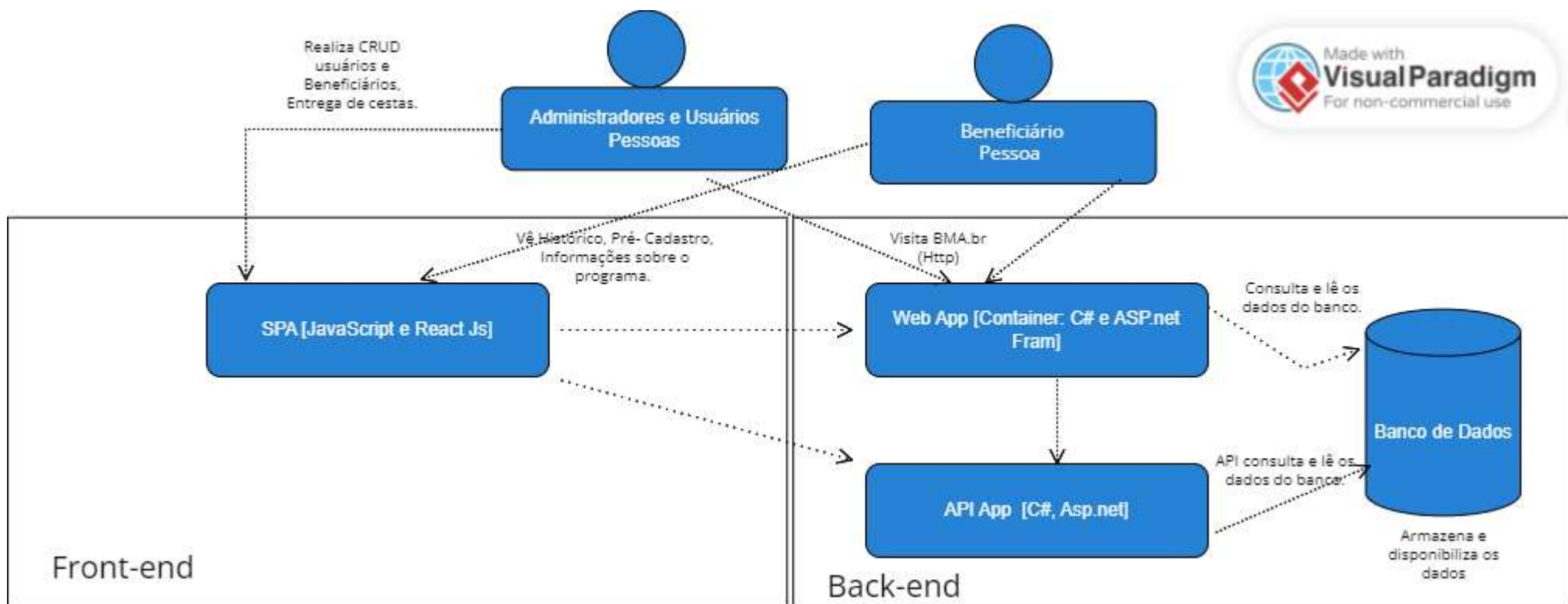
# Restrições

Ambiente	Design	Prioridade
Persistência	Framework ORM	Entity Framework
Persistência	Banco de dados relacional	SQLServer
Front end	Framework de desenvolvimento	React
Front end	Linguagem de programação	JavaScript
Front end	Framework de folhas de estilos	Bootstrap
Folhas de estilo	Linguagem de folha de estilos	CSS
Deploy	Plataforma de automatização de deploy	Heroku
Back end	Linguagem de programação	C#
Back end	Framework de desenvolvimento	ASP.NET Core
Back end	Ambiente de execução	.NET
Back end	Serviço de execução de código	-
Back end	Gerenciador de tráfego	-
Autenticação	Padrão de formato para autenticação	JWT
Autenticação	Serviço de autenticação	-
Integração	Formato de transporte de informações	JSON
Integração	Protocolo de comunicação	HTTP
Integração	Estilo de arquitetura	REST
Versionamento	Controle e armazenamento de código fonte	GitHub
Versionamento	Versionamento de código fonte	Git
Automação	Ferramenta de workflow	-
Build	Ferramenta de build de componentes do software	NPM



# Modelagem e projeto arquitetural





# Diagrama de contêiner

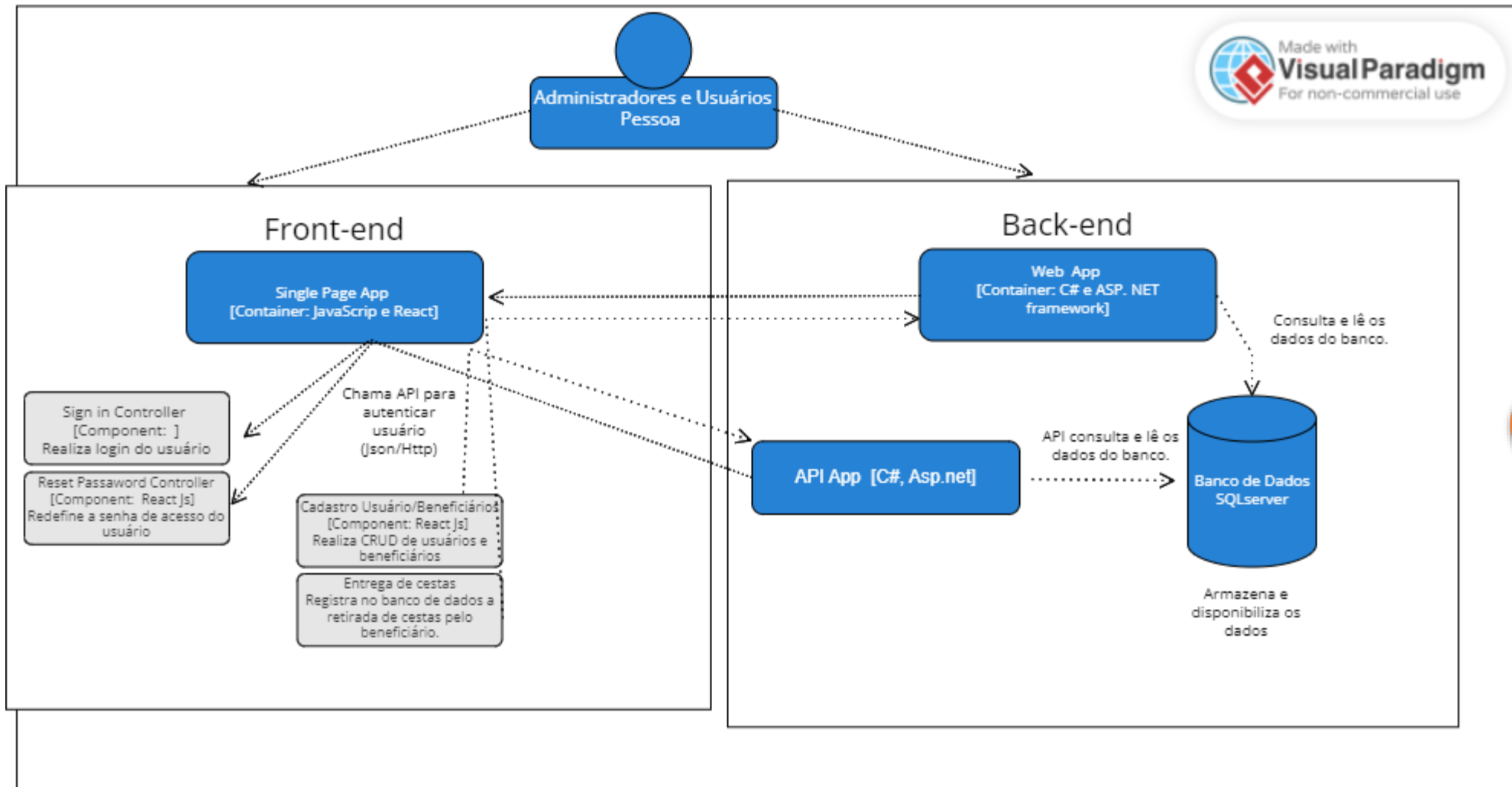


Diagrama de componentes



tipo de teste	Funcional
subtipo de teste	Integração
objetivo do teste	Teste para garantir que o software esteja interagindo corretamente com o banco de dados e que as informações estejam sendo salvas corretamente.
requisitos que motivaram esse teste	RF-03 Permitir que o administrador cadastre os usuários do sistema.
	RF - 05 Permitir que o usuário cadastre os beneficiários.
	RF - 04 Permitir que o administrador consulte, altere e exclua o cadastro dos usuários do sistema.

tipo de teste	Não funcional
subtipo de teste	Usabilidade
objetivo do teste	Testa a facilidade de uso do software para garantir que os usuários possam utilizá-lo facilmente
requisitos que motivaram esse teste	RFN - 01 O sistema deve ser pensado para o usuário

tipo de teste	Não funcional
subtipo de teste	Portabilidade
objetivo do teste	Testa o software em diferentes ambientes e dispositivos para garantir que funcione em todas as plataformas suportadas
requisitos que motivaram esse teste	RFN - 06 O sistema deve ser responsivo para adaptação em dispositivos móveis
	RFN - 05 O sistema deve ser executado nos principais navegadores (Chrome, Mozilla e Edge)

tipo de teste	Não funcional
subtipo de teste	Performace
objetivo do teste	Verifica se o software atende aos requisitos de desempenho especificados, como tempo de resposta e velocidade de processamento.
requisitos que motivaram esse teste	RFN - 03 A latência total do sistema não deve ultrapassar 10 segundos

tipo de teste	Não funcional
subtipo de teste	Segurança
objetivo do teste	Testa a segurança do software contra vulnerabilidades e ameaças de segurança
requisitos que motivaram esse teste	RFN - 08 O sistema deve possuir controle de acesso baseado em perfil

Usuário

HTTP Verb	Endpoint	Ação
GET	/user	Obter todos os usuários do sistema
POST	/user	Criar novo usuário (somente Admin)
GET	/user/{id}	Obter um usuário específico
PUT	/user/{id}	Atualizar um usuário específico (somente Admin)
DELETE	/user/{id}	Excluir um usuário específico (somente Admin)

## Beneficiário

HTTP Verb	Endpoint	Ação
GET	/beneficiario	Obter todos os beneficiários cadastrados do sistema
POST	/beneficiario	Criar novo cadastro de beneficiário
GET	/beneficiario/{id}	Obter um beneficiário específico
PUT	/beneficiario/{id}	Atualizar um cadastro de beneficiário específico
DELETE	/beneficiario/{id}	Excluir um cadastro de beneficiário específico

## Dependente

HTTP Verb	Endpoint	Ação
GET	/benef/{id}/dependente	Obter todos os dependentes do beneficiários cadastrados do sistema
POST	/benef/{id}/dependente	Criar novo cadastro de dependente vinculado a um beneficiário
GET	/benef/{id}/dependente/{id_dependente}	Obter um dependente específico
PUT	/benef/{id}/dependente/{id_dependente}	Atualizar um cadastro de dependente específico
DELETE	/benef/{id}/dependente/{id_dependente}	Excluir um cadastro de dependente específico



### Cesta Básica

HTTP Verb	Endpoint	Ação
GET	/benef/{id}/cesta	Obter o registro de todas a cestas básicas de um beneficiário
POST	/benef/{id}/cesta	Adicionar uma entrega de cesta para um beneficiário específico
DELETE	/benef/{id}/cesta/{id_cesta}	Excluir o cadastro uma cesta de um beneficiário específico (somente Admin)

### Cesta Verde

HTTP Verb	Endpoint	Ação
GET	/benef/{id}/cestaverde	Obter o registro de todas a cestas verde de um beneficiário
POST	/benef/{id}/cestaverde	Adicionar uma entrega de cesta verde para um beneficiário específico
DELETE	/benef/{id}/cestaverde/{id_cestaverde}	Excluir o cadastro uma cesta verde de um beneficiário específico (somente Admin)

# Prova de Conceito e Protótipo Arquitetura

# Banco de Alimentos

v1 OAS3

https://localhost:7273/swagger/v1/swagger.json

Esse é um Sistema de cadastro baseado nas especificações OpenAPI 3.0. O Banco de Alimentos é um sistema com o objetivo de facilitar e agilizar o processo de cadastro de entrega de cestas básicas para famílias que encontram-se em situação de vulnerabilidade social. Projeto criado e desenvolvido pelas alunas: Amanda Paloma e Elaine Souza para o 4º período do curso de Sistemas para Internet, ministrado na Puc Minas EAD.

Apache 2.0

## Beneficiario

Criar, ler, atualizar e excluir cadastros de Beneficiários

GET	/api/Beneficiario	Listar todos os cadastros de Beneficiários no sistema	⌵
POST	/api/Beneficiario	Criar cadastro Beneficiário no sistema	⌵
GET	/api/Beneficiario/{id}	Listar cadastro de Beneficiário específico no sistema	⌵
PUT	/api/Beneficiario/{id}	Atualizar cadastro de Beneficiário específico no sistema	⌵
DELETE	/api/Beneficiario/{id}	Excluir cadastro de Beneficiário específico no sistema	⌵

Schemas ⌵

Beneficiario >

SituacaoEnum >

TipoCestaEnum >

Ativar o Windows  
Acesse Configurações

Prova de conceito

## Schemas

```
Beneficiario {
  id integer($int32)
    readOnly: true
    Identificador do cadastro

  nome* string
    minLength: 1
    Nome completo

  cpf* string
    minLength: 1
    Cadastro de Pessoa Física

  dataNascimento* string($datetime)
    Data de Nascimento

  email string
    nullable: true
    E-mail, se requerido

  telefone* string
    minLength: 1
    Telefone para contato

  cep* string
    minLength: 1
    Endereço

  logradouro* string
    minLength: 1
  numero* string
    minLength: 1
  complemento string
    nullable: true
    Complemento, se presente

  bairro* string
    minLength: 1
  cidade* string
    minLength: 1
  uf* string
    minLength: 1

  situacao* SituacaoEnum integer($int32)
    Situação do cadastrado no Sistema
    Enum:
      > Array [ 2 ]

  tipoCesta* TipoCestaEnum integer($int32)
    Tipo de cesta para retirada
    Enum:
      > Array [ 3 ]
}
```

# Schemas

**SituacaoEnum** integer(\$int32)

Situação do cadastrado no Sistema

Enum:

> Array [ 2 ]

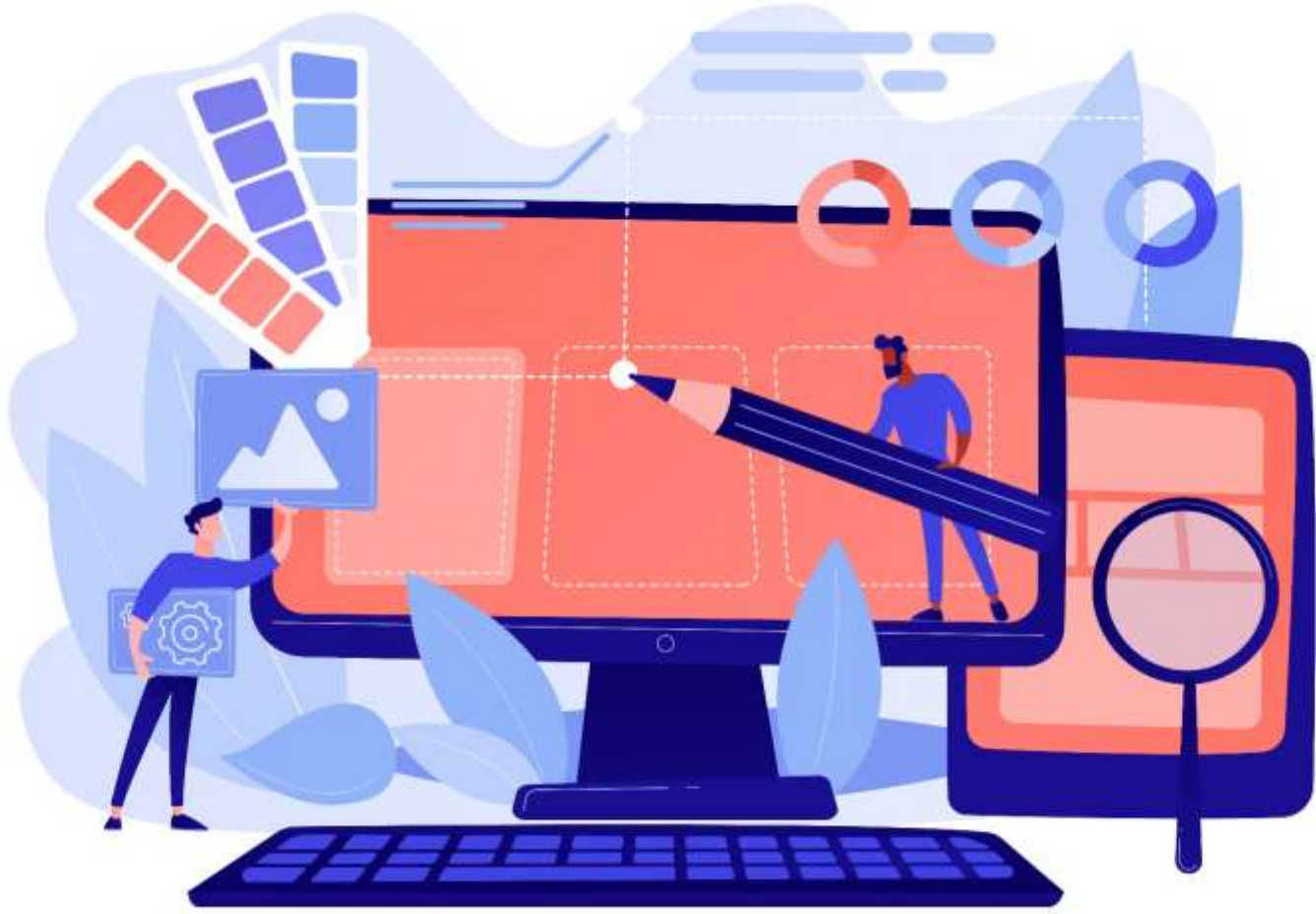
**TipoCestaEnum** integer(\$int32)

Tipo de cesta para retirada

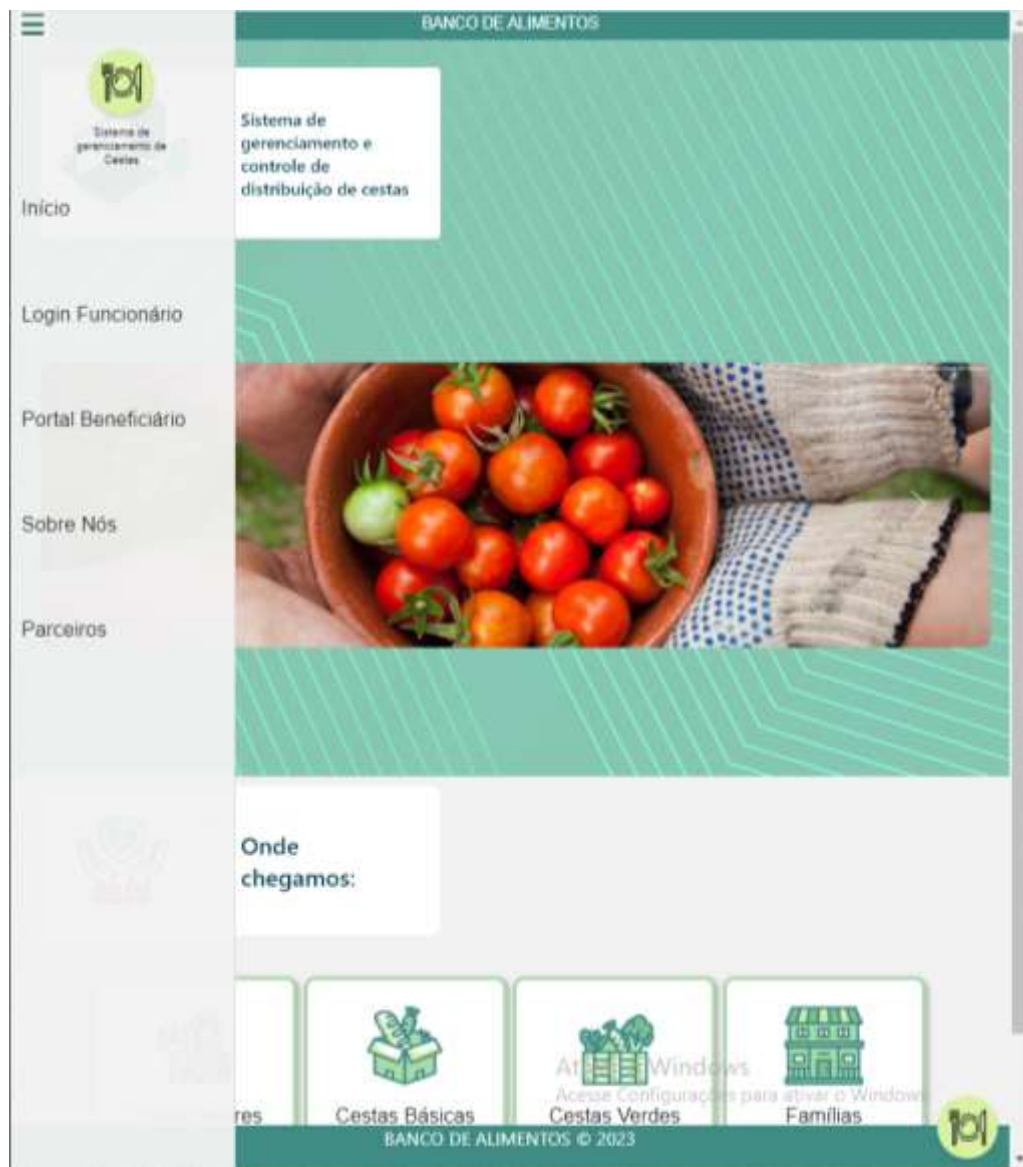
Enum:

> Array [ 3 ]





Front-End



BANCO DE ALIMENTOS

Acesso Funcionário

Login

Usuário

Senha

Entrar

[Esqueci minha senha](#)

Ativar o Windows

Ativar o Windows

BANCO DE ALIMENTOS © 2023

BANCO DE ALIMENTOS

Área Administrativa

Cadastrar Beneficiário

Pesquisar Beneficiário

Cadastrar Funcionário

Buscar Funcionário


Cadastrar Retirada

Ativar o Windows

Ativar o Windows

BANCO DE ALIMENTOS © 2023



  
Sistema de gerenciamento de Cestas

[Início](#)

[Login Funcionário](#)

[Portal Beneficiário](#)

[Sobre Nós](#)

[Parceiros](#)

BANCO DE ALIMENTOS

## Cadastro de Beneficiários

[Voltar](#)

### Dados Pessoais

Nome\*

CPF\*

Data de Nascimento\*  
dd/mm/aaaa

Telefone\*

E-mail

CEP\*

Logradouro\*

Número\*

Bairro\*

Complemento\*

Cidade\*

UF\*


Situação  
☒ Ativo ☐ Inativo


Tipo de Cesta\*  
☒ Básica ☐ Verde ☐ Básica e Verde

\* Campo obrigatório

[Sistema de gerenciamento de Cestas](#)  
[Contato: \(11\) 3061-1000](#)  
[E-mail: contato@bancoalimentos.org.br](#)

BANCO DE ALIMENTOS © 2023





BANCO DE ALIMENTOS

## Pesquisar Beneficiário Cadastrado

[Voltar](#)


ID de busca:

Resposta:

Id	Nome	Cpf	Data de Nascimento	E-mail
15	Pietra Ester Ferreira	32989238697	1977-01-15T00:00:00	
16	Gabriela Mari Sophie Peixoto	61739581636	1977-04-01T00:00:00	gabrielamaripe

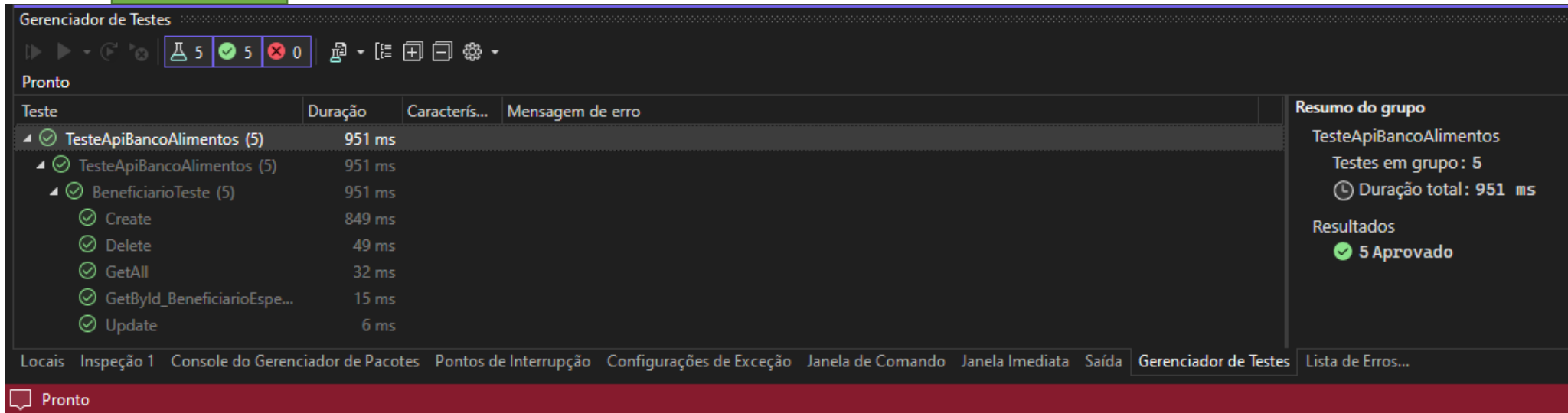
[Sistema de gerenciamento de Cestas](#)  
[Contato: \(11\) 3061-1000](#)  
[E-mail: contato@bancoalimentos.org.br](#)

BANCO DE ALIMENTOS © 2023

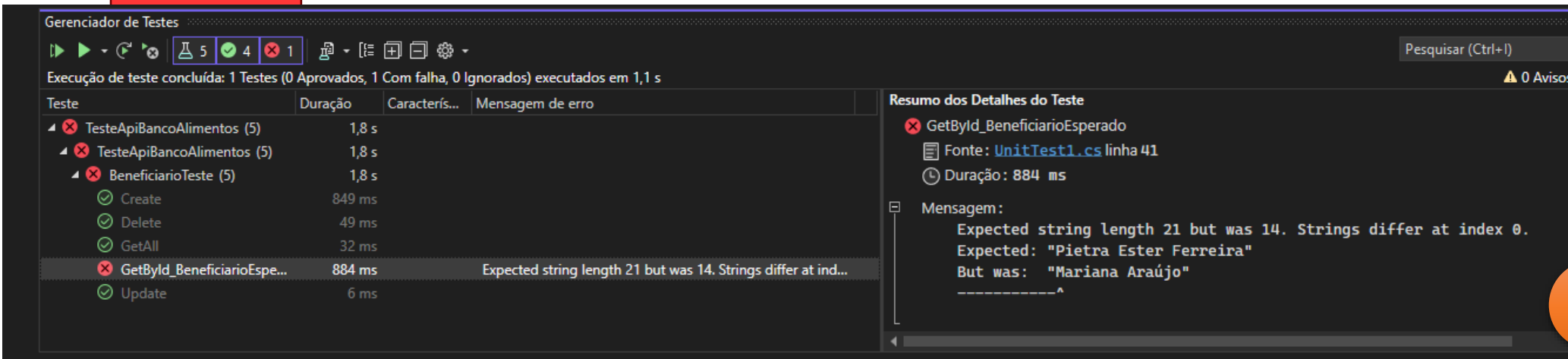




Aprovado



Reprovado



# Teste de Software



# Método: Get by Id

```
namespace TesteApiBancoAlimentos
{
    [TestFixture]
    0 referências
    public class BeneficiarioTeste
    {
        private AppDbContext _dbContext;
        private BeneficiarioController _controller;

        [SetUp]
        0 referências
        public void Setup()
        {
            //Cria banco em memória para o teste
            var options = new DbContextOptionsBuilder<AppDbContext>()
                .UseInMemoryDatabase(databaseName: "TestDatabase").Options;

            //Cria uma instância do banco e do controlador para serem testados
            _dbContext = new AppDbContext(options);
            _controller = new BeneficiarioController(_dbContext);
        }

        [TearDown]
        0 referências
        public void TearDown()
        {
            //Serve para limpar o banco criado em memória após execução
            _dbContext.Database.EnsureDeleted();
            _dbContext.Dispose();
        }
    }
}
```

```
[Test]
0 referências
public async Task GetById_BeneficiarioEsperado()
{
    //Declara um método assíncrono com retorno Task

    //Arrange
    //Define os dados do teste
    var id = 15;
    string actualNome = "Pietra Ester Ferreira";
    string actualCpf = "32989238697";

    var beneficiarioEsperado = new Beneficiario {
        Id = id,
        Nome = "Pietra Ester Ferreira",
        Cpf = "32989238697",
        DataNascimento = new DateTime(1977, 01, 15),
        Email = "",
        Telefone = "33997690408",
        Cep = "39880972",
        Logradouro = "Rua Carneirinho Antonio Soares 157",
        Numero = "838",
        Complemento = "",
        Bairro = "Água Quente",
        Cidade = "Águas Formosas",
        Uf = "MG",
        situacao = api_web_service_bma.Enum.SituacaoEnum.Ativo,
        tipoCesta = api_web_service_bma.Enum.TipoCestaEnum.BASICA
    };

    //Adiciona o objeto ao conteto do banco e salva as alterações
    _dbContext.Beneficiarios.Add(beneficiarioEsperado);
    _dbContext.SaveChanges();

    //Act
    //Testa o método GetById
    var result = await _controller.GetById(id);

    //Assert
    //Verifica se o resultado da ação é um objeto
    //quando o resultado da ação é um HTTP '200 ok'
    Assert.IsInstanceOf<OkObjectResult>(result);
    var okResult = result as OkObjectResult;
    var actualModel = okResult.Value as Beneficiario;

    Assert.IsNotNull(actualModel);
    Assert.That(actualModel.Id, Is.EqualTo(beneficiarioEsperado.Id));
    Assert.That(actualNome, Is.EqualTo(beneficiarioEsperado.Nome));
    Assert.That(actualCpf, Is.EqualTo(beneficiarioEsperado.Cpf));
}
```