

Sistema Bazar Alto Oeste

DOCENTE:

Alysson Filgueira Milanez

DISCENTES:

Arthur Alves de Abreu - 2020011069

Alan Almeida da Silva - 2020010195

1. DEFINIÇÃO GERAL

O Bazar Alto Oeste é um sistema de anúncios de produtos novos ou seminovos, que permite que os usuários divulguem seus itens. Os principais recursos incluem login, adição, visualização e remoção de produtos e atualização de dados de conta. O cadastro requer informações como nome, idade, endereço, telefone e e-mail para facilitar a localização e comunicação. O objetivo é proporcionar uma experiência segura e conveniente para divulgação de produtos a preços acessíveis.

Facilita a compra e venda de produtos: Proporciona uma plataforma onde os usuários podem encontrar uma variedade de produtos novos ou seminovos que atendam às suas necessidades, enquanto também oferece a oportunidade para aqueles que desejam se desfazer de itens não desejados.

Promove a sustentabilidade: Ao facilitar a venda de produtos seminovos, o sistema ajuda a prolongar a vida útil dos itens, reduzindo assim o desperdício e promovendo práticas mais sustentáveis.

Oferece uma experiência segura: Ao solicitar informações essenciais durante o cadastro, como nome completo, endereço e número de telefone, o sistema promove a segurança dos usuários, facilitando a comunicação e a localização em caso de necessidade.

Estimula a economia local: Ao promover transações entre usuários da mesma região, o Bazar Alto Oeste pode fortalecer a economia local, incentivando o comércio entre pessoas da mesma comunidade.

Desse modo, o Bazar Alto Oeste desempenha um papel importante ao oferecer uma plataforma conveniente e segura para compra e venda de produtos, promovendo a sustentabilidade e fortalecendo a economia local.

2. REQUISITOS

2.1. REQUISITOS FUNCIONAIS

RF001 - LOGIN NO SISTEMA: O sistema deve conter as opções de login do usuário com as credenciais de CPF e senha, caso o usuário não tenha cadastro, haverá uma opção de cadastrar-se.

//Pré-condições:

Contract.Requires(!string.IsNullOrEmpty(cpf), "CPF não pode ser nulo ou vazio.");

Contract.Requires(!string.IsNullOrEmpty(senha), "Senha não pode ser nula ou vazia.");

//Pós-condições:

Nenhuma pós-condição identificada.

RF002 - CADASTRO: O sistema deverá conter uma opção para cadastrar um novo usuário, serão necessários nome completo, idade, CPF e senha, além dos campos de endereço, como cidade, cep, rua, numero da residência, além também das opções de contato, como número de telefone, e e-mail.

//Pré-condições:

Contract.Requires(usuario != null, "Usuário não pode ser nulo.");

Contract.Requires(!string.IsNullOrEmpty(usuario.CPF), "CPF do usuário não pode ser nulo ou vazio.");

Contract.Requires(!UsuariosContemCPF(usuario.CPF), "Usuário com o mesmo CPF já cadastrado.");

//Pós-condições:

Contract.Ensures(usuarios.Contains(usuario), "O usuário cadastrado deve estar presente na lista de usuários após o cadastro.");

RF003 - MENU INICIAL: O sistema deve conter um menu, onde nele terão as opções de feed geral, adicionar produtos, visualizar seus produtos, alterar dados da conta e logout.

//Pré-condições:

Contract.Requires(usuarioLogado != null, "É necessário estar logado para acessar o menu inicial.");

//Pós-condições:

Contract.Ensures(usuarioLogado != null, "Após acessar o menu inicial, o usuário logado deve continuar logado.");

RF004 - FEED GERAL: O sistema deve conter um feed (em formato de lista) com todos os produtos que estão em oferta.

//Pré-condições:

Contract.Requires(usuarioLogado != null, "É necessário estar logado para acessar o feed geral.");

//Pós-condições:

Nenhuma pós-condição identificada.

RF005 - ADICIONAR PRODUTO: O sistema deve conter a opção de adicionar um produto, com a opção de adicionar uma descrição incluindo o preço do produto e uma opção de comentários para os usuários que o visualizarem.

//Pré-condições

Contract.Requires(id > 0, "ID do produto deve ser maior que zero.");

Contract.Requires(!string.IsNullOrEmpty(descricao), "Descrição do produto não pode ser nula ou vazia.");

Contract.Requires(preco > 0, "Preço do produto deve ser maior que zero.");

//Pós-condições

Contract.Ensures(produtos.Contains(produto), "O produto adicionado deve estar presente na lista de produtos após a adição.");

RF006 - REMOVER PRODUTO: O sistema deve conter a opção de o usuário visualizar os produtos adicionados e dentro dessa opção ele poderá também removê-los.

//Pré-condições:

```
Contract.Requires(id > 0, "ID do produto não deve ser menor que zero.");  
Contract.Requires(ProdutosContemId(id), "Produto com o ID especificado não encontrado.");
```

//Pós-condições:

```
Contract.Ensures(!ProdutosContemId(id), "O produto removido não deve estar mais presente na lista de produtos após a remoção.");
```

RF007 - ALTERAR DADOS DA CONTA: O sistema deve conter a opção de alterar os dados da conta que foram realizados durante o cadastro.

//Pré-condições:

```
Contract.Requires(usuario != null, "Usuário não pode ser nulo.");  
Contract.Requires(!string.IsNullOrEmpty(novoNomeCompleto), "Novo nome completo não pode ser nulo ou vazio.");  
Contract.Requires(novaIdade > 0, "Nova idade do usuário deve ser maior que zero.");  
Contract.Requires(novoEndereco != null, "Novo endereço não pode ser nulo.");  
Contract.Requires(novoContato != null, "Novo contato não pode ser nulo.");
```

//Pós-condições:

```
Contract.Ensures(Contract.OldValue(usuario.NomeCompleto) != novoNomeCompleto ||  
Contract.OldValue(usuario.Idade) != novaIdade || Contract.OldValue(usuario.Endereco) !=  
novoEndereco || Contract.OldValue(usuario.Contato) != novoContato, "Pelo menos um dos dados da  
conta do usuário deve ser alterado após a alteração.");
```

RF008 - EDITAR PRODUTO: O sistema deve conter uma opção de alterar dados do produto ofertado, como preço e descrição.

//Pré-condições:

```
Contract.Requires(id > 0, "ID do produto deve ser maior que zero.");  
Contract.Requires(!string.IsNullOrEmpty(novaDescricao), "Nova descrição do produto não pode ser nula ou vazia.");  
Contract.Requires(novoPreco > 0, "Novo preço do produto deve ser maior que zero.");  
Contract.Requires(ProdutosContemId(id), "Produto com o ID especificado não encontrado.");
```

//Pós-condições:

```
Contract.Ensures(Contract.OldValue(produtos.Find(p => p.Id == id).Descricao) != novaDescricao ||  
Contract.OldValue(produtos.Find(p => p.Id == id).Preco) != novoPreco, "O produto editado deve ter  
uma nova descrição ou um novo preço após a edição.");
```

RF009 - LOGOUT: O sistema deve conter uma opção de saída do sistema.

//Pré-condições:

```
Contract.Requires(usuarioLogado != null, "É necessário estar logado para deslogar do sistema.");
```

//Pós-condições:

Contract.Ensures(usuarioLogado == null, "Após o logout, o usuário logado deve ser nulo.");

2.2. REQUISITOS NÃO FUNCIONAIS

RNF001 - SEGURANÇA: O sistema deve apenas permitir o cadastro de usuário de idade acima dos 18 anos.

RNF002 - INTEGRIDADE: O sistema deve suportar até 11 milhões de requisições por segunda.

3. CENÁRIO DE USO

Usuário: Ana, uma jovem que deseja vender algumas roupas e acessórios que não utiliza mais através no Bazar Alto Oeste.

Cadastro (RF002):

Ana realiza seu cadastro no sistema Bazar Alto Oeste.

Login no Sistema (RF001):

Ana acessa o Bazar Alto Oeste e realiza o login usando seu CPF e senha previamente cadastrados.

Menu Inicial (RF003):

Após o login, Ana é direcionada para o menu inicial, onde pode ver as opções disponíveis, como feed geral, adicionar produtos, editar produtos, remover produtos, visualizar seus produtos, alterar dados da conta e logout.

Adicionar Produto (RF005):

Ana decide adicionar um vestido que deseja vender. Ela seleciona a opção "Adicionar Produto" no menu e preenche os campos necessários, incluindo nome, descrição e preço do vestido, conforme solicitado pelo RF005.

Editar Produto (RF008):

Após adicionar o vestido, Ana percebe que cometeu um erro na descrição. Ela utiliza a opção "Editar Produto" para corrigir a descrição do vestido, conforme o RF008.

Visualizar e Remover Produto (RF006):

Ana decide verificar quais produtos ela já adicionou para venda. Ela seleciona a opção "Visualizar Seus Produtos" no menu e, dentro dessa opção, ela pode ver todos os produtos que adicionou. Caso queira remover algum, ela utiliza a opção fornecida para remover o produto, conforme o RF006.

Alterar Dados da Conta (RF007):

Ana decide atualizar seu número de telefone, que mudou recentemente. Ela seleciona a opção "Alterar Dados da Conta" no menu e faz as alterações necessárias em seu perfil, de acordo com o RF007.

Logout (RF009):

Após concluir suas atividades no sistema, Ana seleciona a opção "Logout" no menu para sair do sistema, conforme o RF009.