Universidade Regional de Blumenau – FURB  
Departamento de Sistemas e Computação  
Curso de Ciência da Computação  
Projeto de Software

Fair  
Versão 3.0

Breno Spagnolo da Rosa

Felipe Marques Hamann

Gustavo André Bulhmann  
Henrique Haruda Gollnick

Leonardo Cognacco Conceição

Blumenau, 21 de junho de 2022

1. **INTRODUÇÃO**

Para pessoas que buscam adquirir produtos de feira e para feirantes que procuram ampliar seu alcance de venda o Fair é um aplicativo para dispositivos móveis que possibilita que sejam oferecidos produtos desta categoria. Diferente de aplicativos de mercados como da Cooper por exemplo, o Fair é um marketplace, ou seja, não realiza a venda direta de produtos, apenas realizando o contato entre feirante e consumidor.

|  |  |
| --- | --- |
| **É**  - Aplicativo móvel.  - Disponível em Android e iOS.  - Via internet. | **NÃO É**  - Não é uma loja.  - Não é um delivery. |
| **FAZ**  - Anuncia produtos.  - Possibilita encomendas. | **NÃO FAZ**  - Não faz venda direta.  - Não faz revenda de produtos. |

1. **OBJETIVOS**

Facilitar o comércio de produtos de feira.

Aumentar a alcance de venda dos feirantes.

Auxiliar na encomenda de produtos alta demanda.

1. **PERSONAS**

Severino 62 anos agricultor familiar está procurando facilitar o comercio de sua produção de alface.

Maria, 30 anos e coordenadora de infraestrutura de uma empresa do ramo Metalúrgico procura comprar produtos de feira mas não consegue ir até o local de venda em horário comercial.

Renato, 35 anos músico e humorista gostaria de encomendar produtos de feira de maneira semanal.

Cleusa, aposentada de 40 anos que gostaria de comercializar seus produtos, mas não consegue trabalhar presencialmente na feira devido à sua invalidez.

1. **USER STORIES**

|  |  |
| --- | --- |
| **Cliente** | **Comerciante** |
| - Como cliente, quero poder encomendar produtos que em geral tem pouca disponibilidade em feiras.  - Como cliente, quero poder receber os produtos em minha casa, sem a necessidade de ir até as feiras. | - Como comerciante independente, quero ter uma forma de oferecer meus produtos online para conseguir aumentar minhas vendas.  - Como comerciante independente, quero ter uma forma segura de receber o pagamento de meus clientes que efetuaram compras online. |

1. **REQUISITOS FUNCIONAIS**
2. O aplicativo deverá permitir o anúncio de produtos.
3. O aplicativo deverá permitir a venda de produtos.
4. O aplicativo deverá permitir a busca por produto ou por comerciante.
5. O aplicativo deverá permitir a compra de produtos.
6. O aplicativo deverá permitir a encomenda de produtos.
7. O aplicativo deverá permitir a compra programada de produtos com periodicidade definida pelo usuário.
8. **REQUISITOS NÃO-FUNCIONAIS**

O comerciante poderá colocar uma pequena descrição sua, dando um breve resumo dos produtos oferecidos e outras informações que julgar necessário.

O aplicativo é disponibilizado de forma gratuita aos clientes.

O aplicativo deverá ter uma interface simples e intuitiva tanto aos usuários, tanto cliente quanto às pessoas que estiverem anunciando produtos.

O aplicativo deverá possuir uma tela de login.

As telas voltadas aos vendedores e as telas voltadas aos clientes deverão estar no mesmo aplicativo.

O pagamento será mediado pelo vendedor

1. **REGRAS DE NEGÓCIO**

O comerciante não terá acesso direto aos dados pessoais dos usuários. Estas informações são fornecidas apenas ao aplicativo.

Será possível retirar os documentos diretamente com o comerciante ou receber por delivery.

O aplicativo é exclusivo para comerciantes independentes, não permitindo pessoas jurídicas que não sejam MEI (“Microeemprendedor Individual”) ou restaurantes, por exemplo.

1. **LEAN CANVAS**

Table

Description automatically generated

1. **PROTÓTIPO**

Graphical user interface, application

Description automatically generated

1. **DIAGRAMA DE CASOS DE USO**

Diagram

Description automatically generated

1. **DIAGRAMA DE FLUXO**

Diagram

Description automatically generated

1. **DESCRIÇÃO DE CASOS DE USO**

Este caso de uso inicia quando o consumidor abre o aplicativo Fair e busca por um produto de seu interesse. Após a busca, o sistema mostra o produto em específico que foi pesquisado juntamente com produtos similares. É apresentado então ao consumidor possíveis vendedores juntamente com seus preços e suas distâncias relativas à posição atual deste. O cliente seleciona então a quantidade, data e horário da entrega e seleciona que não vai ser uma compra agendada, após isto confirma o pedido. O vendedor aprova o pedido, e envia ao consumidor, O sistema envia uma notificação, ao consumidor informando que o pedido saiu para entrega.

Este caso de uso inicia quando o vendedor abre o aplicativo Fair com o intuito de cadastrar um produto para vender. O Sistema apresenta uma tela com as informações do produto, com dados como: tipo, nome, peso, validade e foto. O vendedor é apresentado então com uma tela de confirmação, que ao confirmar, cadastra o este na lista de produtos do vendedor.

Este caso de uso se inicia quando o consumidor efetua um pedido e solicita que seja uma entrega recorrente. O Sistema então apresenta uma tela solicitando a periodicidade da recorrência, juntamente com o número de entregas que serão efetuadas. O Sistema então apresenta uma tela de confirmação. O cliente confirma o pedido. O Sistema enviará uma notificação ao cliente, com informações sobre o pedido um dia antes de cada entrega.

1. **DIAGRAMA DE CLASSES**

**Diagram

Description automatically generated**

1. **DIAGRAMA DE ESTADOS**

Diagram

Description automatically generated

1. **ARQUITETURA**

A arquitetura do sistema visa possuir uma estrutura de servidor/cliente, onde o processamento de dados é efetuado totalmente através do servidor, enquanto o cliente fica responsável por ser uma interface amigável para o input e output da informação provida pelo servidor.

Chart, diagram

Description automatically generated

O “fair­-client” será um aplicativo feito em Flutter, atuando somente como ponte para o servidor, através da utilização de requisições HTTPS. Deste modo, o cliente é a ferramenta de input e output do sistema em relação aos usuários.

Ao mesmo tempo, o “fair-server” será uma Rest API utilizando Java e Spring Boot. A estrutura dentro do servidor será a recomendada pela documentação presente no site do Spring, seguindo o modelo demonstrado a seguir:

* Controller: responsável por receber as chamadas HTTP e responder com o STATUS correspondente.
* Service: responsável por fazer o processamento de dados dentro da aplicação, além de em muitos casos, efetuar a ligação direta com o banco de dados.
* Model: cria um modelo das tabelas dentro do backend, visando ter os dados orientados a objeto.
* Repository: sobrescreve e implementa os métodos de acesso ao banco.

1. **BANCO DE DADOS**

O Banco de Dados do projeto será feito em MariaDB, sendo que ao mesmo tempo será utilizado a tecnologia Hibernate presente dentro do Spring Boot, para a criação e alteração das tabelas. A estrutura de dados utilizada será a demonstrada a seguir:

**Diagram

Description automatically generated**

Ressalta-se as seguintes informações sobre as tabelas:

* A tabela “Pedido” cuja qual possui o atributo periodicidade: demonstra se um pedido em questão será recorrente ou não, caso seja, possuirá o intervalo de dias entre as entregas, caso não, o valor será zero.
* Dentro da tabela Produto existe o atributo “peso”, este é utilizado para produtos com peso fixo, caso este tenha de ser pesado, ele será definido como “0”.
* Relativo à tabela Produtos\_Pedido, o atributo quantidade será utilizado para quantificar produtos com peso fixo, sendo que produtos com peso variável terão a quantidade “1” e por consequência serão caracterizados pelo atributo “peso”.
* A tabela Produtos\_Loja tem o objetivo de demonstrar o preço que certo produto possuí em uma loja específica.

1. **FUTURAS IMPLEMENTAÇÃO**

Carrinho de compras: A arquitetura do sistema já suporta compras com múltiplos produtos, porém ainda será implementado esta funcionalidade.

Funcionários: Será implementado a possibilidade de múltiplos usuários gerenciarem uma loja.

Sistema de mensagens: possibilitar que usuários e lojas se comuniquem por chat.

Geolocalização: Permitir que o usuário use sua localização atual como endereço.

Imagens: Possibilitar a inserção de imagens nos produtos e perfis.

**GLOSSÁRIO**

Agricultura familiar - É o cultivo da terra realizado por pequenos proprietários rurais, tendo como mão de obra, essencialmente, o núcleo familiar.

API - Interface de programação de aplicações, um conjunto de definições e protocolos para criar e integrar softwares de aplicações.

App – Sigla para a palavra aplicativo.

Client - A interface entre o software e o usuário responsável por fazer a ligação com o servidor.

Artesanato - Artesanato é o próprio trabalho manual, utilizando-se de matéria-prima natural, ou produção de um artesão.

Feira – Uma feira é um evento em um espaço público em que as pessoas, em dias e épocas predeterminados.

Feirante - O Feirante é o profissional responsável por trabalhar nas feiras livres, comercializando produtos alimentícios e afins.

HTTP - Protocolo que permite a obtenção de recursos, como documentos HTML.

Produtos agroecológicos – Produtos que são produzidos em um ambiente com princípios agroecológicos que contemplam o uso responsável da matéria prima.

Produtos coloniais - Produtos tradicionalmente processados no estabelecimento agrícola pelos agricultores de origem italiana e alemã - os “colonos”.

Servidor - Computador que inclui programas necessários para atender coletivamente aos requisitos de uma empresa, em vez de um usuário individual, unidade ou aplicativo específico