**Faculdade de Tecnologia da Zona Leste – Fatec ZL**

**Análise e Desenvolvimento de Sistemas - 4º Semestre**

**Programação para Dispositivos Móveis**

**Aluno(s)**:

Lucas Sardinha de Arruda

**Diretrizes do Trabalho**

Cada aluno receberá um tema e deverá, de acordo com os conhecimentos de Engenharia de Software I, II, III e IHC, fazer um levantamento, dentro do escopo passado individualmente para um aplicativo que resolva 1 problema daquele escopo com tecnologias Android mínimas, como definida abaixo.

A entrega deverá ser:

* 1 Documento descrevendo o problema encontrado naquele escopo e a proposta de solução;
* 1 Código de App Android com a solução do problema utilizando o escopo tecnológico mínimo;
* 1 Vídeo Público ou Não listado (Youtube ou MSStream) de até 5 minutos demonstrando o App no AVD.

São considerados elementos mínimos, podendo ter mais elementos, no App. Saliente-se que a ausência desses elementos acarretará descontos:

* Aspectos de Orientação a Objetos
* Pelo menos uma classe com herança;
* Pelo menos uma classe abstrata;
* Pelo menos uma interface;
* Pelo menos um tratamento de Exceção;
* Pelo menos uma sobrescrita que não seja do toString;
* Todos os objetos devem ter encapsulamentos;
* Todos os objetos devem fazer sobrescrita de toString.
* Aspectos de Arquitetura de Sistemas
* O aplicativo deve ser feito em arquitetura de camadas MVC ou BCE com camada de persistência. Considera-se que a camada view está na raiz do App, sendo as Activities.
* Aspectos de Bancos de Dados
* CRUD de pelo menos 1 objeto / entidade com SQLite.

Serão avaliados a qualidade da descrição do problema, a adequação da solução, a observância aos elementos tecnológicos mínimos e a usabilidade do App. Apps iguais ou com grande espetro de semelhança serão desconsiderados.

**Análise e Design de uma Aplicação Mobile de Encomenda de Produtos**

Apresenta-se de maneira clara e objetiva os requisitos e modelos empregados durante o desenvolvimento de uma aplicação mobile com foco na encomenda e entrega de produtos (alimentos e bens domésticos) para lojas próximas.

No modelo de comércio local/regional, vemos uma necessidade contínua do varejista em realizar viagens contínuas a um representante atacadista para reposição de seu estoque, limitando suas opções de produtos a oferta do dia e uma possível disponibilidade de estoque.

Atuando nesse nicho de mercado, a aplicação “Orion SC” (Orion Supply Chain) visa atuar diretamente na cadeia de suprimento como uma solução mobile, fazendo a ponte entre os dos pequenos vendedores locais e possíveis fornecedores primários, reduzindo assim possiveis impactos de disponibilidade de estoque e flutuações de preço. Utilizando uma interface minimalista e de fácil utilização, o aplicativo mobile permite o controle de clientes; gerenciamento de um banco de dados de produtos; e organização de pedidos, permitindo ao representante local (usuário) trabalhar de qualquer lugar.

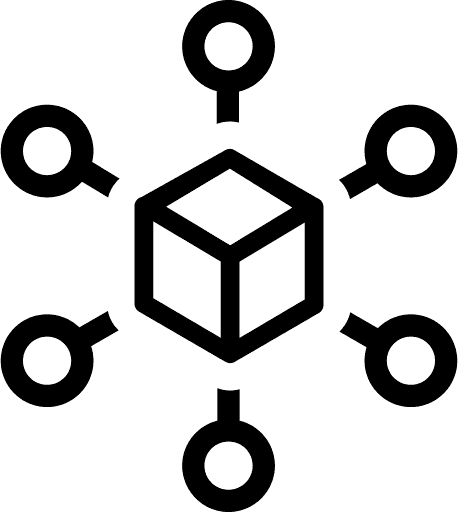


Figura 1. Logomarca/Icone do aplicativo “Orion SC”. [Fonte: UXWing]

Nesse contexto, apresenta-se a seguir uma lista de requisitos; regras de negócio deste projeto.

**Requisitos Funcionais**

RF01: O sistema deve permitir a manutenção do cadastro de lojas/clientes.

RF02: O sistema deve permitir a inserção e manutenção de novos produtos no catálogo, respeitando a diferenciação entre alimentos e bens domésticos.

RF03: O sistema deve permitir que as lojas/clientes criem listas de produtos e tenham acesso as listas já existentes.

RF04: Deve-se poder pesquisar as listas de cada cliente, assim como seu respectivo conteúdo.

**Requisitos Não-Funcionais**

RNF01: A aplicação deve ser intuitiva e não requerer treinamento prévio. (Usabilidade)

RNF02: A aplicação deve apresentar certa disponibilidade, ter uma baixa frequência de falhas críticas (crashes).

RN03: Ser de fácil manutenibilidade.

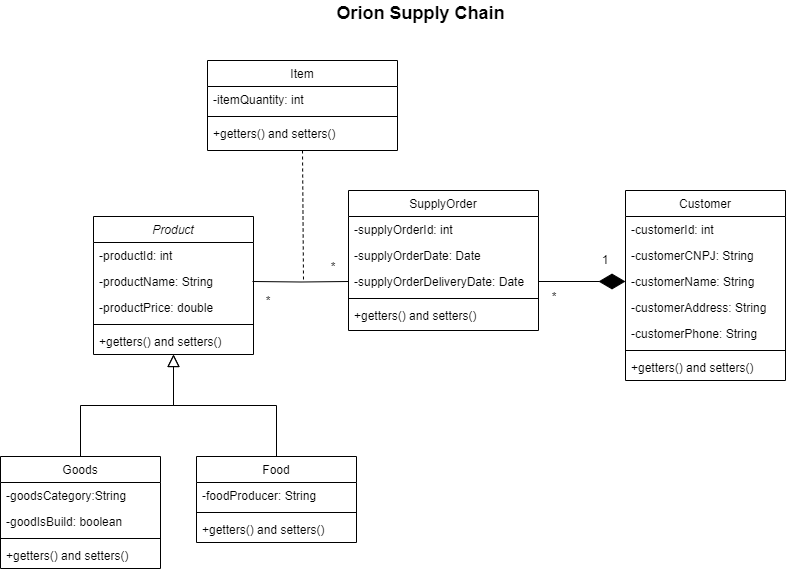
**Regras de Negócio**

RN01: O cadastro de usuário deve conter informações básicas como nome da loja/cliente, CNPJ, telefone e endereço.

RN02: Comercializa-se dois tipos de produto: Alimentos e Bens Domésticos, ambos devendo apresentar Nome e Preço.

**Diagrama de Classe**

A fim de se detalhar o funcionamento da aplicação, apresenta-se a seguir o diagrama de classes utilizado tanto para configuração do bando de dados, como para modelagem das classes de entidade.

Figura 2. Diagrama de classes da aplicação “Orion SC”. [fonte: autor]

**Banco de Dados**

Apresenta-se o script utilizado para geração do bando de dados da aplicação, onde é possível observar a aplicação do comando “ON DELETE CASCADE” para representar a remoção das entidades representadas via relacionamento de composição.

CREATE DATABASE orionsc;

USE DATABASE orionsc;

CREATE TABLE customer (

customerId INTEGER NOT NULL,

customerCNPJ CHAR(14) NOT NULL,

customerName VARCHAR(100) NOT NULL,

customerAddress VARCHAR(100) NOT NULL,

customerPhone VARCHAR(13) NOT NULL,

PRIMARY KEY (customerId)

);

CREATE TABLE product (

productId INTEGER NOT NULL,

productName VARCHAR(100) NOT NULL,

productPrice DECIMAL(10,2) NOT NULL,

PRIMARY KEY (productId)

);

CREATE TABLE food (

foodProductId INTEGER,

foodProducer VARCHAR(100) NOT NULL,

FOREIGN KEY (foodProductId) REFERENCES product(productId)

ON DELETE CASCADE

);

CREATE TABLE goods (

goodsProductId INTEGER NOT NULL,

goodsCategory VARCHAR(100) NOT NULL,

goodsIsBuild BOOLEAN NOT NULL DEFAULT 'FALSE',

FOREIGN KEY (goodsProductId) REFERENCES product(productId)

ON DELETE CASCADE

);

CREATE TABLE supplyOrder (

supplyOrderId INTEGER NOT NULL,

supplyOrderCustomerId INTEGER,

supplyOrderDate CHAR(8) NOT NULL,

supplyOrderDeliveryDate CHAR(8) NOT NULL,

PRIMARY KEY (supplyOrderId),

FOREIGN KEY (supplyOrderCustomerId) REFERENCES customer(customerId)

ON DELETE CASCADE

);

CREATE TABLE item (

itemSupplyOrderId INTEGER,

itemProductId INTEGER,

itemQuantity INTEGER NOT NULL,

PRIMARY KEY (itemSupplyOrderId, itemProductId),

FOREIGN KEY (itemSupplyOrderId) REFERENCES supplyOrder(supplyOrderId)

ON DELETE CASCADE,

FOREIGN KEY (itemProductId) REFERENCES product(productId)

ON DELETE CASCADE

);

**Aplicação**

Finalmente, apresenta-se algumas imagens da aplicação em funcionamento, visando enfatizar suas principais funções.

