

Documento de Requisitos para Desenvolvimento de Software

Sistema de PDV de Alta Usabilidade

Expo USJT/2019

***Prof. Dr. Elcio Abrahão
Laboratório de Práticas Integradas
2019 / 2***

Aprendizado Baseado em Projetos (PBL)

Sumário

1) Introdução	3
2) Escopo	3
2.1) Requisitos Funcionais	3
2.2) Requisitos não funcionais	4
3) Detalhamento dos Requisitos	4
3.1) Entidade Mercadoria	5
3.2) Entidade Pedido	5
3.3) Entidade Venda	5
3.4) Descrição do fluxo de vendas	6
3.5) Descrição do CRUD de mercadoria	7
3.4) Descrição da exportação de vendas	7
4) Cronograma e prazos	7
5) Apresentação e Avaliação	8

1) Introdução

O presente documento visa especificar os requisitos funcionais e não funcionais para o desenvolvimento de software para um Ponto de Venda (PDV) de alta usabilidade, considerando técnicas avançadas de interface humano computador cuja finalidade é agilizar a venda de produtos em pequenos pontos de venda caracterizados pela presença de no máximo dois funcionários o que demanda uma extrema agilidade e simplicidade de interface.

2) Escopo

2.1) Requisitos Funcionais

O presente software deve implementar as seguintes funcionalidades obrigatórias:

- a) Implementação no modelo cliente-servidor;
- b) Ser multiplataforma e multi sistema operacional;
- c) Utilizar banco de dados MySQL;
- d) Ser totalmente implementado em um notebook e não utilizar recursos de internet ou ligação em rede de qualquer tipo (100% autônomo).
- e) Utilizar reconhecimento de código de barras com leitor manual e/ou câmera do notebook;
- f) Implementar telas para CRUD de produto;
- g) Implementar tela para venda de produto;
- h) Implementar exportação de relatório de vendas por data no formato excel (planilha);
- i) O sistema deve permitir cadastro e venda de produtos sem código de barras (exceção);

As seguintes funcionalidades podem ser implementadas no contexto de melhoria (não obrigatória)

- j) Controle de estoque de produtos;

2.2) Requisitos não funcionais

O software deverá implementar obrigatoriamente os seguintes requisitos não funcionais:

- a) A interface deve considerar como princípio básico a “*facilidade de utilização pelo operador*”;
- b) O produto deve ser cadastrado em menos do que 20 segundos;
- c) Um produto deve ser adicionado a lista de compras em menos do que 5 segundos;
- d) Na adição de produtos na lista de compras o usuário não deve utilizar o teclado nem o mouse;
- e) O sistema deve ser facilmente reinicializado, de preferência com utilização de temporizador e sem acesso a teclado ou mouse;
- f) Ações que dependem de CRUD devem ser utilizado de preferência a interface via *teclado* em detrimento do *mouse*;
- g) O usuário do sistema não pode fazer operações de inicialização do sistema como por exemplo estartar banco de dados ou inicializar serviços do sistema operacional;
- h) O escaneamento deve ser preciso e não pode gerar leitura falsa ou recusa de leitura;
- i) Produtos sem código de barra devem apresentar alternativas de leitura de código de barras o digitação via teclado em regime de exceção;
- j) A leitura do código de barras deve permitir ao operador liberdade de movimentação e utilização de somente uma das mãos para posicionamento do produto.

3) Detalhamento dos Requisitos

Na presente seção serão detalhados es requisitos funcionais e não funcionais. Todas as considerações iniciais feitas neste documento devem ser observadas. Modificações podem ser feitas mediante negociação com o cliente e com anuência do mesmo.

O cliente manterá um repositório no GITHUB onde deve ser mantido o código fonte do sistema em regime de controle de versão. Os desenvolvedores devem baixar o código fonte do sistema e manter o mesmo atualizado a cada nova funcionalidade adicionada ao sistema.

Os desenvolvedores devem testar o sistemas e manter o código fonte sempre funcionando, ou seja, não se deve fazer commit de código quebrado no repositório do projeto.

3.1) Entidade Mercadoria

A entidade **mercadoria** deve conter os seguintes campos:

id: integer
codigo: string (13)
descricao: string (200)
unidade: string (10)
preco: double (15,2)
dataAtualizacao: dateTime (dd/mm/aaaa hh:mm:ss)
ativo: boolean

3.2) Entidade Pedido

A entidade **pedido** destina-se a registrar as pedidos para um determinado cliente. Nota-se que não haverá a identificação do cliente que se manterá anônimo. Um pedido identifica um conjunto de vendas para um determinado cliente. O total de itens é a contagem de mercadorias em um pedido e o total do pedido é a soma do preço de cada mercadoria vendida.

id: integer
data: dateTime
totalItens: integer (dd/mm/aaaa hh:mm:ss)
totalPedido: double (15,2)
cancelado: boolean

3.3) Entidade Venda

A entidade **venda** destina-se a identificar as mercadorias de um pedido. Cada venda representa uma mercadoria individual que foi escaneada pelo sistema. Ao se escanear a mercadoria o preço atual da mercadoria deve ser transferido para o registro de venda para registrar o valor real da venda na data em que o pedido foi feito. Isso evita que uma mudança de preço da mercadoria venha a alterar pedidos feitos anteriormente em outro preço.

id: integer
idPedido: integer
idMercadoria: integer
quantidade: integer
preco: double (15,2)

3.4) Descrição do fluxo de vendas

O diagrama da figura 1 descreve o fluxo de vendas do ponto de vista do operador do sistema.

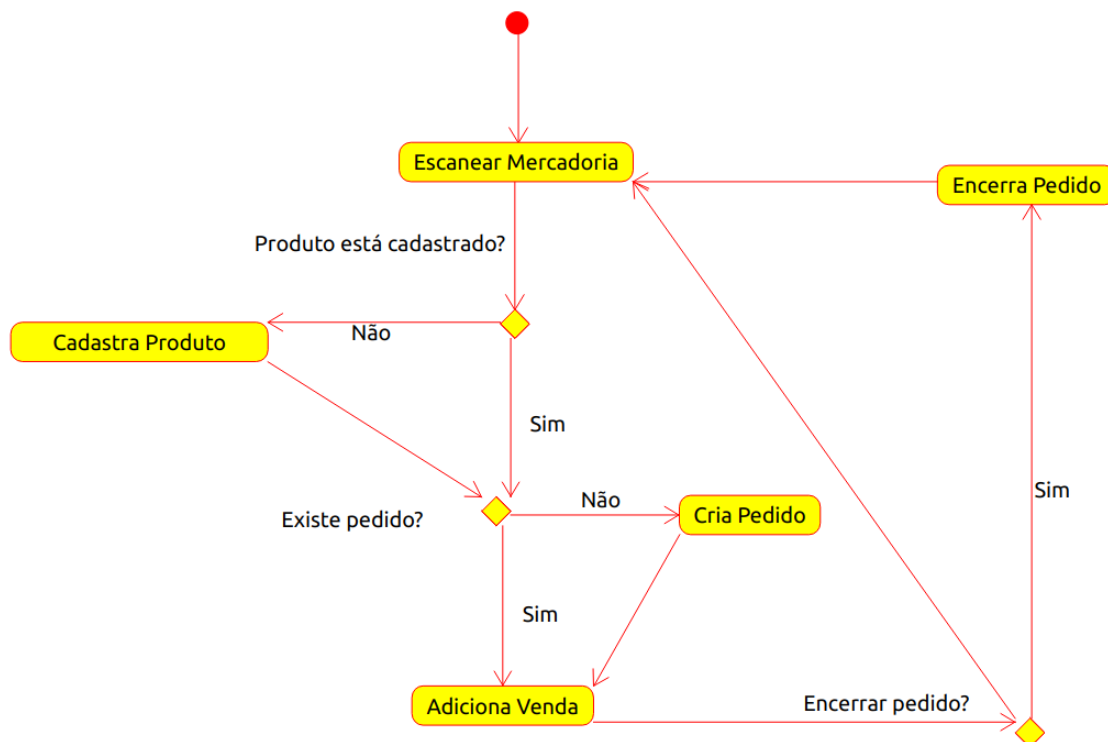


Figura 1 - Diagrama de fluxo de operação. Fonte: autor

Ao se inicializar o sistema a tela de escaneamento deve ser apresentada e o sistema deve estar pronto para o primeiro escaneamento de mercadoria.

Caso a mercadoria escaneada não esteja na base de dados como mercadoria já cadastrada deve-se apresentar a tela de cadastro de mercadoria permitindo ao operador o cadastro imediato da mercadoria.

Caso a mercadoria já esteja cadastrada deve-se verificar se existem um pedido aberto. Se o pedido não está aberto deve-se criar um novo pedido e deve-se adicionar uma venda para a respectiva mercadoria.

Por padrão cada escaneamento deve gerar uma venda de uma unidade de produto.

Após gerar a venda o sistema deve voltar a tela de escaneamento e aguardar pelo escaneamento de uma nova mercadoria ou permitir o encerramento do pedido.

Ao se encerrar o pedido deve-se mostrar uma tela com a lista de mercadorias, quantidades e preços do pedido e o total do pedido. Além disso deve-se exibir sugestões de troco para pagamento em dinheiro dos valores cheios mais comuns imediatamente acima do preço total do pedido.

3.5) Descrição do CRUD de mercadoria

Além da sugestão de cadastro via fluxo de vendas o sistema deve prover um menu para acesso a telas de cadastro de nova mercadoria, alteração de mercadoria existente, consulta de mercadoria por código de barras, consulta de todas as mercadorias em ordem alfabética e desativação de mercadoria.

3.4) Descrição da exportação de vendas

O sistema deve provar a exportação dos pedidos e vendas realizadas no sistemas a partir de uma determinada data gerando um arquivo no formato de planilha .xls ou .ods.

O sistema deve gerar a planilha e perguntar ao usuário o local para gravação do arquivo no sistema.

4) Cronograma e prazos

O sistema deve estar pronto na última semana de novembro: 29/11/2019 e deve estar totalmente funcional, testado e instalado em um notebook.

As aulas de LPI devem ser utilizadas para o desenvolvimento do projeto e eventualmente as aulas de POO e BD com a devida anuência dos professores das respectivas disciplinas.

5) Apresentação e Avaliação

Os alunos serão divididos em grupos com as seguintes responsabilidades:

- Grupo A: Requisitos, projetos e documentação;
- Grupo B: Banco de Dados, Dev Java para Banco de Dados;
- Grupo C: Interface gráfica e Usabilidade (telas swing)
- Grupo D: Programação e integração de sistemas;

Todos os alunos devem ter acesso ao repositório do GITHUB. No início de cada seção de trabalho os alunos devem fazer uma reunião de todos os grupos para planejamento das atividades, divisão de tarefas e detecção de impedimentos e pesquisa de soluções a problemas.

O grupo deve eleger um líder de projetos entre os alunos que será responsável pela execução das reuniões de trabalho.

O professor atuará como tutor e facilitador, orientando as atividades e ajudando na solução de problemas.

O alunos serão avaliados conforme sua participação nas atividades do projeto, frequência nas aulas e trabalhos entregues durante todo o semestre para a disciplina com os seguintes pesos:

Participação no projeto: 50%;

Frequência nas seções de trabalho: 25%;

Trabalhos entregues no semestre: 25%.

Eventuais dúvidas e itens não mencionados nesse documento serão tratados a critério do professor e seguirão as normas de projeto vigentes na Expo São Judas e nos regimentos internos da USJT.