

TICKET MACHINE – INSPEÇÃO DOCUMENTAÇÃO

*Problema formulado pelo Professor Calebe de Paula Bianchini

Alunos: Enzo Ferroni – 10417100

Rafael Neves – 10418316

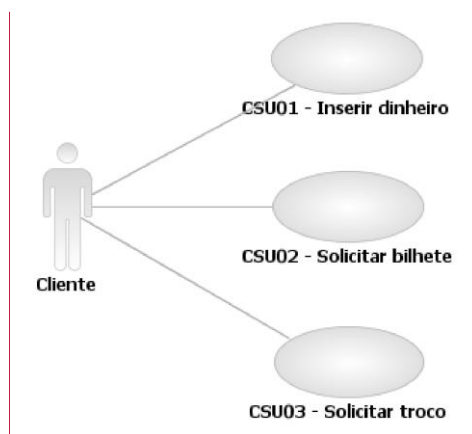
1 Engenharia de Requisitos

As estações de trem frequentemente fornecem máquinas de vender bilhetes que imprimem um bilhete quando um cliente insere a quantia correta para pagar a passagem. As máquinas mantêm uma soma total da quantidade de dinheiro que coletou durante toda sua operação.

1.1 Catálogo dos Atores

Ator	Descrição
Cliente	Usuário da máquina de vender bilhetes.

1.2 Diagrama dos Casos de Uso



1.3 Especificação dos Casos de Uso

Comentado [R01]: Omissão: Os requisitos não informam se existem tipos de bilhetes diferentes (comum, ida e volta, estudante, idoso, etc.)

Comentado [R02]: Inconsistência: Os artefatos gerados se contradizem, uma vez que, está sendo informado que o cliente deve inserir a quantia correta para pagar a passagem, mas tanto no diagrama de casos de uso quanto no protótipo da tela principal existe a opção de selecionar troco.

Comentado [R03]: Omissão: Os requisitos não informam que o cliente tem a opção de solicitar o bilhete, troco ou apenas inserir mais dinheiro.

Comentado [R04]: Ambiguidade: Não está claro se sempre que o cliente inserir dinheiro o sistema automaticamente imprime o bilhete ou o cliente deve solicitar sua emissão.

Comentado [R05]: Omissão: Falta a informação sobre quem é o usuário dessa funcionalidade.

Comentado [R06]: Ambiguidade: O que está sendo considerado como "durante toda sua operação"? Se refere a cada operação de vender um bilhete ou o tempo que a máquina permanece ativa ao longo do dia/semana.

Comentado [R07]: Inconsistência: Qual seria a necessidade de manter a soma total se o cliente sempre colocaria a quantia correta para pagar a passagem, considerando que a "operação" seja vender um bilhete.

Comentado [R08]: Omissão: Falta de outros atores importantes para o Sistema da Máquina de Tickets, sendo um deles o Administrador/Coletor que poderia ser responsável pela coleta do dinheiro.

Comentado [R09]: Omissão: Falta representar o ator Administrador/Coletor, considerando que este utilize a funcionalidade de manter a soma total e coletar o dinheiro acumulado.

Comentado [R010]: Fato Incorreto: Seguindo a lógica dos requisitos, o usuário não deveria ter que solicitar nem o bilhete e nem o troco, dessa forma o desenho do diagrama está incorreto. Deveria considerar o uso da representação <<include>>. Caso o cliente tenha a opção de inserir mais dinheiro, solicitar o troco ou emitir um bilhete, deveria considerar o uso da representação <<extend>>.

Comentado [R011]: Fato Incorreto: Os requisitos informam que o usuário deve fornecer a quantia correta para pagar a passagem, logo não seria necessário troco, porém o Diagrama de Caso de Uso contradiz informando um caso de Solicitar Troco.

1.3.1 CSU01 – Inserir dinheiro

Identificador	CSU01
Nome	Inserir dinheiro
Atores	Cliente
Sumário	Uma quantia de dinheiro é inserida na máquina. É importante ressaltar que essa quantia sempre é representada por uma única nota de papel-moeda.
Complexidade	Médio
Regras de Negócio	N/D
Pré-condições	N/D
Pós-condição	O valor inserido é adicionado ao saldo total disponível.
Pontos de Inclusão	N/D
Pontos de Extensão	N/D

Comentado [RO12]: Fato Incorreto: O caso de uso não especifica uma regra de negócio que envolve a necessidade do sistema de validar o valor mínimo para emissão de bilhete ou devolução de troco.

Comentado [RO13]: Fato Incorreto: Caso o cliente assim que inserir a quantidade necessária para pagar um bilhete, o sistema automaticamente imprime o bilhete, sua emissão deve ser incluída como Ponto de Inclusão.

Comentado [RO14]: Fato Incorreto: Caso o cliente tenha a opção de solicitar a emissão do bilhete deve ser incluído como Ponto de Extensão.

Fluxo Principal	
Ações do Ator	Ações do Sistema
1. O Cliente insere uma nota de papel-moeda.	
	2. O Sistema mostra uma mensagem informando para o Cliente aguardar alguns instantes.
	3. O Sistema valida a nota de papel-moeda inserida.
	4. O Sistema adiciona o valor ao saldo disponível.
	5. O Sistema informa que a nota de papel-moeda foi aceita.
	6. O sistema informa o saldo atual. Caso de Uso é encerrado.

Comentado [RO15]: Fato Incorreto: O sistema deveria informar o cliente de que a nota foi aceita logo após sua validação, portanto esse deveria ser a etapa 4.

Fluxo de Exceção 1: 3a. O Sistema não valida a nota de papel-moeda.	
Ações do Ator	Ações do Sistema
	1. O Sistema devolve a nota de papel-moeda.
	2. Enquanto o Cliente não retirar a nota de papel-moeda, o Sistema informa que a nota de papel-moeda é inválida.
3. O Cliente retira a nota de papel-moeda.	
	4. Volta ao passo 6 do Fluxo Principal.

Comentado [RO16]: Inconsistência: No Fluxo de Exceção 1, o passo 4 indica "Volta ao passo 6 do Fluxo Principal", mas o Fluxo Principal termina no passo 6.

1.3.2 CSU02 – Solicitar bilhete

Identificador	CSU02
Nome	Solicitar bilhete
Atores	Cliente
Sumário	Um único bilhete de transporte é impresso.
Complexidade	Fácil
Regras de Negócio	N/D
Pré-condições	N/D
Pós-condição	Um bilhete é impresso e seu valor é debitado do saldo disponível.
Pontos de Inclusão	N/D
Pontos de Extensão	N/D

Comentado [R017]: Fato Incorreto: Não há uma regra de negócio que especifique que o sistema deve verificar se o saldo é suficiente antes de emitir o bilhete.

Comentado [R018]: Fato Incorreto: O caso de uso não define que o saldo deve ser maior ou igual ao valor do bilhete antes da solicitação.

Comentado [R019]: Fato Incorreto: Se existe a possibilidade do cliente solicitar troco, deveria ser incluído como um Ponto de Extensão.

Fluxo Principal	
Ações do Ator	Ações do Sistema
1. O Cliente solicita impressão do bilhete de transporte.	
	2. O Sistema verifica o saldo e emite o bilhete.
	3. O sistema informa o saldo atual. Caso de Uso é encerrado.

Comentado [R020]: Ambiguidade: Não é informado se o cliente pode escolher o tipo do bilhete (comum, estudante, idoso, etc.).

Fluxo de Exceção 1: 2a. O saldo é insuficiente para emissão do bilhete.	
Ações do Ator	Ações do Sistema
	1. O Sistema informa que o saldo é insuficiente.
	2. Volta ao passo 3 do Fluxo Principal.

1.3.2 CSU03 – Solicitar troco

Identificador	CSU03
Nome	Solicitar troco
Atores	Cliente
Sumário	O troco é devolvido. Vale ressaltar que o troco também é em nota de papel-moeda.
Complexidade	Fácil
Regras de Negócio	N/D
Pré-condições	N/D
Pós-condição	O valor em nota de papel-moeda é devolvido e o saldo disponível é zerado.
Pontos de Inclusão	N/D

Comentado [R021]: Fato Incorreto: Não existe uma regra de negócio para cálculo do troco solicitado.

Comentado [R022]: Fato Incorreto: Para solicitar o troco, o cliente deverá ter saldo disponível.

Pontos de Extensão	N/D
--------------------	-----

Fluxo Principal	
Ações do Ator	Ações do Sistema
1. O Cliente solicita devolução do saldo restante como troco.	
	2. O Sistema verifica o saldo e devolve a quantidade do saldo em notas de papel-moeda.
	3. O Sistema zera o valor do saldo.
	3. O sistema informa o saldo atual. Caso de Uso é encerrado.

Comentado [R023]: Omissão: Não existe um fluxo de exceção do caso de uso Solicitar Troco, considerando que o cliente pode não ter saldo disponível.

Comentado [R024]: Fato Incorreto: A numeração da etapa está errada, causando duplicação da etapa 3.

1.4 Protótipos

1.4.1 Tela principal

Sistema de Emissão de Passagens

Saldo Atual

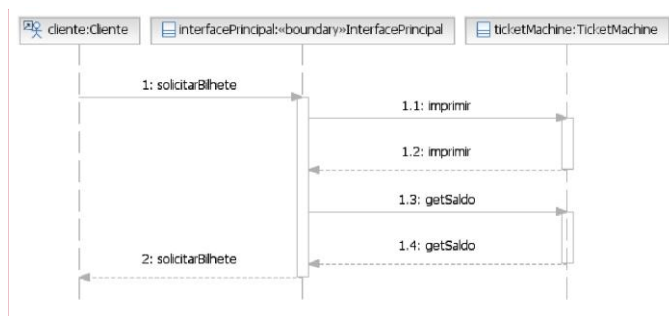
R\$ 2,00

Valor Inserir

Emitir Troco

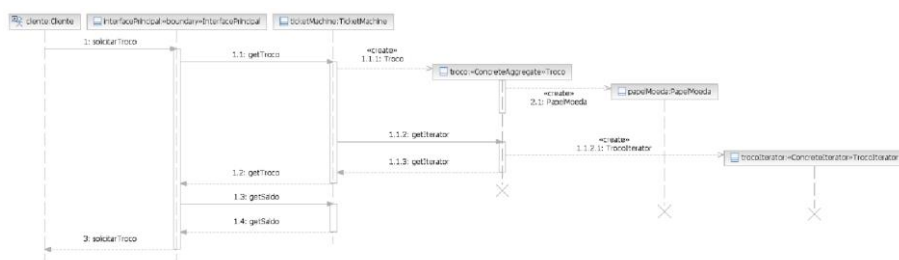
Comentado [R025]: Informação Estranha: Se o sistema irá ler a nota não tem sentido a existência de um campo para informar o valor inserido.

Comentado [R026]: Informação Estranha: De acordo com a lógica de negócio, não tem sentido o cliente poder solicitar diretamente o bilhete ou troco sem inserir o dinheiro antes.



Comentado [R036]: Inconsistência: O diagrama de sequência não está de acordo com o fluxo principal da especificação do caso de uso, faltando etapas como informar para o cliente o saldo restante.

2.5 Diagrama de Sequência – Solicitar troco



Comentado [R037]: Fato Incorreto: Considerando a associação de composição presente no diagrama de classe, o objeto PapelMoeda deveria ser removido juntamente com o objeto Troco ao invés de ser removido posteriormente.

É importante salientar que estes diagramas representam apenas o fluxo principal dos casos de uso.

3 Inspeção

Crie um relatório com o resultado da inspeção deste documento. Liste os defeitos e apresente onde foram encontrados. Classifique os defeitos por categoria (veja tabela abaixo) e quanto à severidade (baixa/média/alta).

Tabela 1. Defeitos de requisitos [Shull 1998].

Tipos de Defeito	Descrição
Omissão	Deve-se à omissão ou negligência de alguma informação necessária ao desenvolvimento do software.
Ambiguidade	Ocorre quando uma determinada informação não é bem definida, permitindo assim uma interpretação subjetiva, que pode levar a múltiplas interpretações.
Fato incorreto	Informações dos artefatos do sistema que são contraditórias com o conhecimento que se tem do domínio da aplicação.
Inconsistência	Ocorre quando duas ou mais informações são contraditórias entre si.
Informação estranha	Informação desnecessária incluída nos requisitos do software que esta sendo desenvolvido

Tabela 2. Defeitos de código [Jones 2009].

Tipos de Defeito	Descrição
Comissão	Ocorre quando existe algum segmento de código que foi implementado incorretamente, i.e., cuja implementação é diferente do que foi especificado.
Inicialização	Ocorre quando se tenta acessar uma variável que não foi inicializada.
Computação	Similar ao defeito de comissão; ocorre quando um valor é definido erroneamente para uma variável.
Desempenho	Algumas rotinas executam comandos ou laços (<i>loops</i>) desnecessários.
Controle	Ocorre quando um comando de desvio condicional é usado de forma incorreta.
Excesso	Existem trechos de código irrelevantes e desnecessários.
Dados	Ocorre quando uma estrutura de dados é manipulada de forma incorreta (por exemplo, quando se tenta acessar um índice inexistente de um vetor/matriz).

4 Referências

Pötter, H.; Schots, M. , 2011. *InspectorX: Um Jogo para o Aprendizado em Inspeção de Software*. Anais do FEES11 - Fórum de Educação em Engenharia de Software, São Paulo-SP, Brasil.

Jones, C., 2009. *Software Engineering Best Practices*, McGraw-Hill Inc., New York, USA.

Shull, F., 1998. *Developing Techniques for Using Software Documents: A Series of Empirical Studies*, Ph.D. Thesis, University of Maryland, College Park.