## Atividade 2 - Parte 1 (Documento, Requisitos e Casos de Uso)

### Checklist: Casos de Uso, especificação e documentação

Na seção inicial da engenharia de requisitos, o documento não deixa claro algumas informações permitindo uma interpretação subjetiva. Na frase ‘’ As máquinas mantêm uma soma total da quantidade de dinheiro que coletou durante toda sua operação. ’’, está ambígua, pois não fica claro se todo esse dinheiro coletado é o saldo de um único cliente ou se é o saldo de todos os clientes que solicitaram/inseriram o dinheiro. Além disso, na especificação dos requisitos, deveria ser informado se o sistema só irá imprimir o bilhete caso o cliente for inserir o dinheiro, mesmo que ele ainda tenha saldo suficiente para imprimir o bilhete. Ou seja, inserir o dinheiro é um requisito obrigatório para solicitar o bilhete e é preciso usar o caminho <<include>> nesta associação? E se o saldo já for suficiente para imprimir o bilhete, será que é necessário inserir dinheiro nesta situação? Deveria ser melhor explicado que ele só precisará inserir o dinheiro caso o saldo dele na máquina estiver zerado ou insuficiente. Isto classifica-se como omissão da informação e fica mais claro ainda na apresentação da tela principal onde ele tem a opção se emitir o bilhete, solicitar troco e inserir o dinheiro, além da informação do salto atual. Classifica-se como inconsistência do sistema e fato incorreto. O documento não explica muito bem esse mecanismo do saldo. A questão do troco também deveria ser melhor discutida na especificação dos requisitos de que o cliente pode solicitar a devolução do saldo como troco. Além disso na descrição do caso de uso ‘’ solicitar troco’’, o ação 3 do sistema ’’o sistema zera o valor do saldo’’ deveria ser revisto, pois o troco deveria ser calculado em função do valor do dinheiro que o cliente inseriu e não ser simplesmente zerado.

Ao analisar o UC do documento, percebe-se algumas inconsistências, como por exemplo, a figura do ator que deseja atingir uma meta. A figura que representa o ator deveria respeitar as normas da UML. Além disso, a linha que liga o ator ao caso de uso que ele vai interagir, não apresenta o sentido, ou seja, falta o sentido da associação.

O caso de uso ‘’ inserir dinheiro ‘’ não faz muito sentido neste cenário, pois passa a impressão de que ele vai apenas gastar o seu dinheiro sem nenhum objetivo e voltar pra casa sendo que na verdade ele está inserindo dinheiro para adicionar ao saldo total ou imprimir para o bilhete caso ele não tenha saldo suficiente, pois o documento não deixa muito claro isso. O mais correto deveria ser criado o caso de uso ‘’ solicitar o bilhete’’ e associa-lo ao caso de uso em um caminho opcional <<extends>> ao caso de uso ‘’inserir dinheiro’’ e esse último ser associado ao caso de uso em um caminho opcional ‘’ solicitar troco’’. Isto é apenas uma ideia, pois o documento não deixa muito claro se o ‘’inserir dinheiro’’ é um caminho obrigatório ou não para solicitar bilhete. Tem partes que passa a impressão de que é obrigatório e em outras parece que é opcional, como por exemplo na descrição dos 3 casos usos e na tela principal causando ambiguidade, inconsistências e fatos incorretos entre os requisitos.

### Checklist: código

Ao analisar o código, não foi encontrada a classe principal(main) para testar o código para checar se o sistema está funcionando.

O pacote que implementa as exceções do sistema está incompleto, pois faltou o tratamento de exceções como o try e catch por exemplo, para mostrar ao usuário uma mensagem na tela informando que o saldo é insuficiente, ou que o papel moeda foi invalidado pelo sistema, afinal, eram duas ações do sistema que foram descritas nos casos de uso, portanto deveria ser melhor trabalhada no código. Classifica-se como comissão.

### Conclusão

O documento precisa ser revisto e melhor trabalhado para atender ás boas características da qualidade de software.