**AVALIAÇÃO DE CONHECIMENTOS QUALITY ASSURANCE**

**ESTRUTURA DO CONTEÚDO PROVA\_QA.ZIP**

* **Pasta controlefinanceiro**
  + **Utilize o index.html para os testes abaixo.**

**FORMA DE ENTREGA:**

* Crie um repositório privado no GitLab com o nome prova\_QA
* Efetue os commits de seu projeto neste repositório
* Não adicione ao repositório a pasta de projeto que sua IDE de desenvolvimento gera de forma automática
* Adicione os seguintes usuários ao repositório privado:
  + No GitLab: [afagundes@kpmg.com.br](mailto:afagundes@kpmg.com.br) e [wmarques@kpmg.com.br](mailto:wmarques@kpmg.com.br) (como “developer”)
* Informe o nome do repositório na resposta do e-mail
* Adicione qualquer outra informação que você julgue relevante no e-mail ou em um arquivo readme.md

**Boa sorte. Tenha criatividade e seja livre para a criação dos testes.**

**ESTAPA 1 – DOCUMENTAÇÃO DOS CASOS DE TESTES**

**User Story:**

Sistema de controle financeiro para registro de receitas e despesas.

Como usuário devo informar o nome e valor de cada receita ou despesa. Todos os registros são adicionados a uma lista de transações que contém o nome, valor e uma label lateral que altera a cor de acordo com o registro, caso seja receita, a label é verde e se for despesa, a label é vermelha.

Para cada registro é feito o balanço no saldo atual. As receitas são somadas ao saldo atual e ao valor total de “Receitas”, já as despesas são subtraídas do saldo atual e somadas ao valor total de “Despesas”.

Todos os registros são armazenados no localStorage do navegador, desse modo eles permanecem salvos mesmo que a página seja recarregada. Para apagar um registo é só passar o mouse em cima dele na lista de transações que aparece um botão no canto esquerdo para removê-lo, após excluído será refeito o balanço do saldo atual, receitas e despesas sem contabilizar o registro excluído.

* **Dada a seguinte *User Story*** , escreva 6 casos de testes entre positivo e negativo.

**ESTAPA 2 – TESTES AUTOMATIZADOS**

**Contexto:** Você está em um time ágil. Foi priorizado pelo P.O. a automatização do sistema de controle financeiro. Foi realizado o refinamento, planejamento, desenvolvimento e agora está nas suas mãos construir os testes automatizados.

**Sobre o cenário abaixo, segue parte do contexto. Seja criativo e livre para criação de outros cenários.**

**Cenário:** cadastrar receita e despesa no sistema de controle financeiro

**Dado** que o usuário inseri um valor de (**receita**/**despesa**)

**Quando** clicado em adicionar

**Então** o valor é adicionado na aba de transações

**REQUISITOS:**

* Editores **Eclipse** ou **IntelliJIDEA**
* Linguagem **Java** e as bibliotecas **Mave, jUnit, Selenium** e **Cucumber**.
* Criar uma pasta com o seu **nome**, dentro da pasta criar o projeto com o nome **controlefinanceiro** do tipo **Maven**.
* Utilizar o arquivo pom.xml com as dependências **Mave, jUnit, Selenium e Cucumber e Selenium WebDriver**.
* Driver a ser utilizado é o **chromedriver**.

**SUGESTÃO:**

* Crie 3 cenários usando técnicas do Cucumber.
* Automatize os processos com base nos cenários criados.