**TI QUALIDADE**

Responda as questões a seguir:

1. O cliente adquiriu um novo modelo de máquina de cartão de crédito e ela aceita as bandeiras Visa, Master, Elo, Amex e Hiper e executa duas operações para cada bandeira débito e crédito. Quantos testes serão necessários para validação dessa máquina e qual técnica a ser utilizada?

**Para validar a nova máquina de cartão de crédito que aceita as bandeiras Visa, Master, Elo, Amex e Hiper, e que executa duas operações para cada bandeira (débito e crédito), precisamos considerar todas as combinações possíveis de bandeiras e tipos de operação.**

**Cálculo dos Testes Necessários**

**Número de Bandeiras: 5 (Visa, Master, Elo, Amex, Hiper)**

**Tipos de Operação: 2 (débito e crédito)**

**Portanto, o número total de testes necessários é:**

**[ \text{Número de Bandeiras} \times \text{Tipos de Operação} = 5 \times 2 = 10 ]**

**Técnica a Ser Utilizada**

**A técnica mais adequada para este cenário é a técnica de particionamento de equivalência. Esta técnica envolve dividir os dados de entrada em diferentes classes de equivalência que podem ser tratadas da mesma forma. Cada classe representa um conjunto de condições que devem ser testadas para garantir que a máquina de cartão de crédito funcione corretamente para todas as bandeiras e tipos de operação. Além disso, a análise de valor limite pode ser aplicada para testar os limites das transações, como valores mínimos e máximos permitidos para cada operação de débito e crédito.**

**Passos para a Validação**

**Planejamento dos Testes: Definir os casos de teste para cada combinação de bandeira e tipo de operação.**

**Execução dos Testes: Realizar os testes em um ambiente controlado para verificar se a máquina processa corretamente todas as transações.**

**Documentação dos Resultados: Registrar os resultados de cada teste, incluindo qualquer falha ou comportamento inesperado.**

**Análise dos Resultados: Avaliar os resultados para identificar possíveis problemas e áreas que necessitam de correção.**

**Revisão e Repetição: Ajustar a máquina conforme necessário e repetir os testes até que todos os casos passem com sucesso.**

1. A bandeira Visa disponibilizou um novo range de bin´s no mercado que possui seis dígitos 232425 a 232460. Quais os testes deverão ser feitos desse range de bins para confirmar que está funcionando?
2. 230000,232425,232460
3. 0,232425,232460,240000
4. 232425,232460
5. 232424,232425,232459,232460,232461

**Para confirmar que o novo range de BINs da bandeira Visa (232425 a 232460) está funcionando corretamente, é importante testar os valores dentro e nos limites desse intervalo. A opção que melhor cobre esses critérios é:**

**d) 232424, 232425, 232459, 232460, 232461**

**Justificativa:**

**232424: Testa um valor fora do intervalo (abaixo do limite inferior).**

**232425: Testa o valor no limite inferior do intervalo.**

**232459: Testa um valor dentro do intervalo.**

**232460: Testa o valor no limite superior do intervalo.**

**232461: Testa um valor fora do intervalo (acima do limite superior).**

**Essa abordagem garante que você está verificando tanto os valores dentro do intervalo quanto os valores nos limites e fora dele, o que é essencial para uma validação completa.**

1. Cite uma abordagem desenvolvimento ágil e explique-a, responda em suas palavras.

**Uma abordagem de desenvolvimento ágil é o Scrum. Scrum é um framework que facilita a colaboração em projetos complexos. Ele divide o trabalho em ciclos curtos chamados sprints, que geralmente duram de duas a quatro semanas. Cada sprint começa com uma reunião de planejamento onde as tarefas são priorizadas e atribuídas. Durante o sprint, a equipe realiza reuniões diárias de stand-up para discutir o progresso e resolver impedimentos. No final do sprint, há uma revisão para demonstrar o trabalho concluído e uma retrospectiva para identificar melhorias para o próximo sprint. O objetivo do Scrum é entregar incrementos funcionais do produto de forma contínua e iterativa.**

1. O que é CI/CD?

**CI/CD significa (Integração Contínua/Entrega Contínua). CI é a prática de integrar código de diferentes desenvolvedores em um repositório compartilhado várias vezes ao dia, com cada integração sendo verificada por uma build automatizada e testes. CD é a prática de automatizar a entrega de novas versões do software para ambientes de produção, garantindo que o software possa ser lançado de forma rápida e segura. Juntas, essas práticas ajudam a detectar e corrigir erros rapidamente, melhorar a qualidade do software e reduzir o tempo de entrega.**

1. O que é TDD, BDD e ATDD e quando são aplicadas?

**TDD : Desenvolvimento Orientado a Testes. É uma prática onde os testes são escritos antes do código funcional. O ciclo TDD é: escrever um teste, ver o teste falhar, escrever o código mínimo necessário para passar o teste, e refatorar o código. É aplicado para garantir que o código atenda aos requisitos desde o início.**

**BDD: Desenvolvimento Orientado a Comportamento. Foca no comportamento do software do ponto de vista do usuário final. Utiliza uma linguagem comum para descrever os testes, como Gherkin, que facilita a comunicação entre desenvolvedores, testadores e stakeholders. É aplicado para garantir que o software atenda às expectativas do usuário.**

**ATDD: Desenvolvimento Orientado a Testes de Aceitação. Similar ao TDD, mas os testes são escritos com base nos critérios de aceitação definidos pelo cliente ou usuário final. É aplicado para garantir que o software entregue atenda aos requisitos de aceitação do cliente.**

1. Cite uma heurística de testes utilizada para testes de front e de back e explique-as

**Front-end: Heurística de Consistência e Padrões. Esta heurística verifica se a interface do usuário é consistente em termos de design, comportamento e terminologia. Por exemplo, botões com a mesma função devem ter a mesma aparência e comportamento em todas as telas.**

**Back-end: Heurística de Validação de Dados. Esta heurística verifica se os dados recebidos e enviados pelo sistema são válidos e consistentes. Por exemplo, ao testar uma API, deve-se verificar se os dados de entrada são validados corretamente e se as respostas são formatadas** conforme esperado.

1. Cite ferramentas de testes automatizados para Desktop, Web, API´s e mobile

**Desktop: Selenium, TestComplete**

**Web: Selenium, Cypress, Puppeteer, Playwright, Robot Framework**

**APIs: Postman, SoapUI, RestAssured**

**Mobile: Appium, Espresso (Android), XCUITest (iOS)**

**Prática de API**

Desenvolva o script da automação seguindo as informações a seguir:

Documentação = https://developers.getnet.com.br/api

URI = https://api-homologacao.getnet.com.br/

1. Validar o script de "TOKENZAÇÃO" método "POST” cobertura de testes em API
2. Validar cobertura de Status Code, Campos obrigatórios e Contrato
3. Desenvolver com POJOs.
4. Obs: Para utilizar estes endpoints é necessário realizar a autenticação conforme as instruções de documentação.

Chaves de Autenticação:  
        Client ID: 67823c6d-58de-494f-96d9-86a4c22682cb  
        Client Secret: c2d6a06f-5f31-448b-9079-7e170e8536e4

! Atenção! Obrigatoriamente utilizando a linguagem JavaScript com o framework Cypress, postman ou newman.

Além da automação dos passos descritos acima, serão avaliados itens como design das classes, organização, manutenibilidade do código, orientação a objeto, arquitetura da automação, clean code, seletores utilizados para identificar os elementos web (xpath, css, id) e etc.

OBS1: Enviar o link do código no GitHub