**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?**

RESPOSTA:

São 5 bandeiras e 2 tipos de operações. Para validar a máquina de cartão de crédito que aceita múltiplas bandeiras e executa duas operações para cada uma, será necessário considerar todas as combinações possíveis de bandeiras e operações.Acredito que10 testes para validar a máquina de cartão de crédito com as bandeiras e operações especificadas. A técnica de Particionamento de Equivalência e Combinação de Valores é a mais adequada para garantir que todas as combinações de bandeiras e operações sejam testadas. Consiste em dividir os dados de entrada em partições (ou classes) de dados que são tratadas da mesma maneira pelo sistema. Neste caso, cada combinação de bandeira e operação constitui uma partição de equivalência. As combinações a serem feitas são as seguintes:

1. Visa – Débito
2. Visa – Crédito
3. Master – Débito
4. Master – Crédito
5. Elo – Débito
6. Elo – Crédito
7. Amex – Débito
8. Amex – Crédito
9. Hiper – Débito

10. Hiper – Crédito

Além disso, para garantir que todas as combinações sejam cobertas, a técnica de Combinação de Valores é adequada. Nesta técnica, todas as combinações possíveis de um conjunto de variáveis são testadas. Para este caso, as variáveis são as bandeiras e as operações.

### Exemplo de Caso de Teste

Para cada combinação, devemos criar um caso de teste que inclui:

1. Pré-condições:
   * Máquina de cartão configurada e conectada;
   * Cartão da bandeira específica disponível.
2. Passos:
   * Inserir/Passar o cartão na máquina;
   * Selecionar o tipo de operação (débito ou crédito);
   * Inserir o valor da transação;
   * Confirmar a operação.
3. Resultados Esperados:
   * A máquina deve processar a transação corretamente;
   * A transação deve ser aprovada ou rejeitada conforme o saldo e as regras do cartão;
   * A impressão do comprovante.

**2) 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?**

**a) 230000,232425,232460**

**b) 0,232425,232460,240000**

**c) 232425,232460**

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

RESPOSTA:

Para garantir que o novo range de bin’s está funcionando corretamente, o conjunto de testes deve incluir valores dentro do intervalo, nos limites do intervalo e fora do intervalo. A opção mais abrangente que cobre esses aspectos é: d) 232424, 232425, 232459, 232460, 232461

Esta opção garante a validação dos seguintes aspectos:

232424: Fora do intervalo (abaixo do limite inferior)

232425: Dentro do intervalo (limite inferior)

232459: Dentro do intervalo (próximo ao limite superior)

232460: Dentro do intervalo (limite superior)

232461: Fora do intervalo (acima do limite superior)

Portanto, a resposta correta é letra d.

**3) Cite uma abordagem desenvolvimento ágil e explique-a, responda em suas palavras.**

RESPOSTA:

Scrum é uma abordagem ágil eficaz para o desenvolvimento de software que promove a colaboração, a adaptação rápida às mudanças e a entrega contínua de valor ao cliente. Ao dividir o trabalho em Sprints e realizar reuniões regulares de inspeção e adaptação, as equipes podem manter o foco nas prioridades do cliente e melhorar constantemente seu processo de desenvolvimento. Scrum é uma metodologia que facilita a colaboração e a gestão de projetos de software de forma ágil, promovendo entregas incrementais e iterativas. Nela, temos bem definidos as personas envolvidas no projeto(papéis), temos as cerimônias definidas e cada uma com seu objetivo bem claro e definido e também, temos os artefatos entregáveis.

Com o uso do Scrum, são possíveis adaptações rápidas às mudanças nos requisitos e prioridades do cliente. Ao final de cada Sprint, há um incremento de produto que pode ser revisado e potencialmente liberado. Todos os envolvidos no projeto têm visibilidade sobre o progresso e as dificuldades enfrentadas. As retrospectivas incentivam a equipe a continuamente melhorar processos e práticas de trabalho.

**4) O que é CI/CD?**

RESPOSTA:

CI/CD são práticas que permitem integrar, testar e lançar software de forma mais eficiente e rápida. A integração contínua ajuda a detectar erros cedo, enquanto a entrega e a implantação contínua garantem que o código possa ser lançado com segurança e frequência. Essas práticas são essenciais para equipes de desenvolvimento ágil que buscam entregar valor contínuo aos seus usuários. Integração contínua (CI) e a entrega contínua (CD) visam melhorar a eficiência, a qualidade e a velocidade do desenvolvimento de software, automatizando partes do processo de desenvolvimento e implantação.

**5) O que é TDD, BDD e ATDD e quando são aplicadas?**

RESPOSTA:

* TDD: Focado em garantir que o código funcione conforme esperado em nível de unidade. Usado principalmente por desenvolvedores para garantir a qualidade do código.
* BDD: Focado em garantir que o comportamento do sistema atenda aos requisitos de negócios. Promove a colaboração entre desenvolvedores, QA e partes interessadas.
* ATDD: Focado em garantir que o software atenda aos critérios de aceitação definidos pelo cliente. Facilita a validação dos requisitos de negócios antes e durante o desenvolvimento.

Cada uma dessas abordagens pode ser aplicada dependendo das necessidades do projeto e da equipe, e muitas vezes são usadas em conjunto para garantir a qualidade e a satisfação do cliente no desenvolvimento de software.

**6) Cite uma heurística de testes utilizada para testes de front e de back e explique-as**

RESPOSTA:

#### Heurística de Teste para Backend: "F-I-R-S-T": A heurística FIRST (Fast, Independent, Repeatable, Self-checking, Timely) é utilizada para assegurar a eficácia dos testes automatizados, particularmente em testes de backend.

#### Heurística de Teste para Frontend: "C-R-U-D": A heurística CRUD (Create, Read, Update, Delete) é comumente utilizada para testar a funcionalidade de interfaces de usuário que interagem com dados. Essa abordagem assegura que todas as operações básicas de dados sejam verificadas na aplicação frontend.

**7) Cite ferramentas de testes automatizados para Desktop, Web, API´s e mobile**

**RESPOSTA:**

### Ferramentas de Testes Automatizados para Desktop

1. SikuliX: Utiliza imagens para identificar e interagir com elementos da interface do usuário.
2. AutoIt: Uma linguagem de script voltada para a automação de tarefas de Windows GUI.
3. WinAppDriver: Um serviço baseado no WebDriver da Microsoft para automação de aplicativos do Windows.
4. TestComplete: Uma ferramenta abrangente que suporta testes de desktop, web e mobile, incluindo automação de UI para aplicativos de desktop.

### Ferramentas de Testes Automatizados para Web

1. Cypress: Uma ferramenta moderna para testes end-to-end em aplicações web. É conhecida por sua simplicidade e integração com o ecossistema JavaScript.
2. Selenium: Uma das ferramentas mais populares para automação de navegadores. Suporta vários navegadores e sistemas operacionais.
3. Puppeteer: Uma biblioteca Node.js que fornece uma API de alto nível para controlar o Chrome ou Chromium.
4. Playwright: Desenvolvida pela Microsoft, suporta testes em múltiplos navegadores e é similar ao Puppeteer, mas com mais funcionalidades.

### Ferramentas de Testes Automatizados para APIs

1. Postman: Além de ser uma ferramenta de API popular, Postman também suporta a criação de testes automatizados para APIs.
2. RestAssured: Uma ferramenta poderosa para testar APIs RESTful em Java.
3. SoapUI: Uma ferramenta que suporta testes de APIs SOAP e REST, incluindo funcionalidades avançadas para testes de carga e segurança.
4. Karate: Uma ferramenta de teste de APIs que também suporta testes de integração e é baseada em BDD (Behavior-Driven Development).

### Ferramentas de Testes Automatizados para Mobile

1. Appium: Uma ferramenta de automação de código aberto para testes de aplicativos móveis nativos, híbridos e web.
2. Espresso: Uma estrutura de teste de UI para aplicativos Android desenvolvida pelo Google.
3. XCUITest: Uma framework de teste de UI para aplicativos iOS desenvolvida pela Apple.
4. Calabash: Uma ferramenta para automação de testes de aplicativos móveis que suporta tanto Android quanto iOS, utilizando Cucumber para a definição de testes.