

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?

R: Como são 5 bandeiras, com 2 modalidades de operações, 10 seria o total mínimo considerando apenas os testes dos pares, isso sem contar com testes de caixa preta entre outros.

- 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

R: Alternativa D, pois testa valores antes e depois dos limites dos ranges. Eu ainda adicionaria 232426 para o teste do limite inferior.

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

R: Scrum > Metodologia que divide a carga de trabalho em Sprints, deixando as atividades mais organizadas e tendo um deadline claro onde se sabe o que deve ser esperado e executado. Existem as cerimônias como Daily, Refining e Planning para manter todos a par da evolução da Sprint. Retro e Review para mostrar o que foi feito e o que não foi. Normalmente as Sprints são de 7 dias a 2 semanas a depender do objetivo e projeto, sendo assim os objetivos são claros e o acompanhamento é feito de perto, podendo mudar a estratégia assim que é descoberto uma falha no processo.

- 4) O que é CI/CD?

R: Abreviação de Continuous Integration, Continuous Delivery/Deployment. Tradução para Integração Contínua, Entrega Contínua. É uma prática que busca otimizar o ciclo de vida do desenvolvimento do software, permitindo a detecção precoce de problemas, possibilitando a automação de testes a cada entrega, mantendo assim a qualidade e consistência do produto x equipe.

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

R: São abordagens para garantir a qualidade e a confiabilidade do software durante o desenvolvimento. Cada uma tem um foco e um objetivo específico:

TDD (Test-Driven Development) é uma abordagem onde os testes são escritos antes do código. O processo inclui escrever um teste, ver o teste falhar, escrever o código para passar o teste, executar o teste novamente e refatorar o código. É útil para garantir alta cobertura de testes e código bem testado desde o início.

BDD (Behavior-Driven Development) foca no comportamento do sistema do ponto de vista do usuário final, usando uma linguagem compreensível para descrever o que o

software deve fazer. Utiliza o formato "Given-When-Then" (sintaxe Gherkin) para descrever testes e é útil para garantir que todos os stakeholders compreendam e concordem com os requisitos do software.

ATDD (Acceptance Test-Driven Development) envolve definir e acordar os critérios de aceitação antes do desenvolvimento. Semelhante ao BDD, utiliza uma linguagem acessível para descrever os critérios que o software deve atender, garantindo que o produto final esteja de acordo com as expectativas dos stakeholders. É aplicado em projetos onde a colaboração estreita com o cliente e a validação dos requisitos são essenciais.

Porém, de todas as formas, a escolha é de acordo com as necessidades e expectativas da equipe de planejamento, de testes e de produto.

- 6) [Cite uma heurística de testes utilizada para testes de front e de back e explique-a](#)
R: "Equivalence Partitioning" (Particionamento de Equivalência): é uma técnica de teste que divide o conjunto de entradas possíveis em grupos (ou partições) de dados que devem ser tratados da mesma maneira. A ideia é que se um teste nessa partição passa, os demais passam, mas se um falha, todos falharão.
Exemplo de aplicação no front: Emails válidos (teste@exemplo.com), Emails inválidos (teste@com, t@d.c).
Exemplo de aplicação no back: Números de telefones válidos (+5511999999999), telefones inválidos (+12345, abc)
- 7) [Cite ferramentas de testes automatizados para Desktop, Web, API's e mobile](#)
R: Desktop = Selenium / Web = Cypress / Mobile = Appium, XCUITest