**ATIVIDADE DA AULA 2**

**EM GRUPO**

**Grupo**:

Erick Domingues Soares – 82414486

Wellington de Oliveira Sousa – 825240209

Barbara Tavares Tracanella – 824124152

Angelo Rodrigues - 824139676

**Análise do Caso 1 – Aplicativo de Banco Digital**

Para o Caso 1, “Aplicativo de Banco Digital”, que apresenta bloqueios durante transações em horários de pico, as características de qualidade da ISSO/IEC 9126 que devem ser priorizadas são Funcionalidade e Eficiência.

**Funcionalidade**

Esta característica é crucial porque o aplicativo não está cumprindo sua função principal de permitir transações de forma confiável. O bloqueio em horários de pico indica uma falha na capacidade do sistema de executar suas funções sob certas condições.

**Melhorias de Funcionalidade**

​Essas melhorias focam em garantir que as transações sejam concluídas de forma confiável e sem falhas.

​**Realização de Testes de Carga e Estresse**: A equipe de desenvolvimento pode usar ferramentas para simular um grande número de usuários realizando transações ao mesmo tempo. Isso ajuda a identificar o ponto exato em que o sistema começa a travar, permitindo que os desenvolvedores corrijam o problema antes que ele afete os usuários reais.

​**Refatoração do Código Crítico**: O código responsável por processar as transações pode ser revisado e otimizado. Isso significa reescrever partes do código para que ele seja mais rápido e menos propenso a erros, mesmo sob alta demanda.

​**Revisão do Banco de Dados**: A estrutura do banco de dados e as consultas (instruções para buscar ou salvar dados) podem ser otimizadas para lidar com um volume maior de informações. Isso garante que a base de dados não se torne um gargalo durante os horários de pico.

**Sub-características relevantes**:

**Adequabilidade** (Suitability): O aplicativo precisa ser adequado para a finalidade a que se destina, que é realizar transações financeiras. Bloqueios impedem essa adequação.

**Confiabilidade** (Reliability): A falta de confiabilidade é evidente, já que o sistema não se comporta de forma estável e previsível, especialmente quando a demanda aumenta.

**Métricas propostas**:

**Taxa de Sucesso de Transações**: Medir a porcentagem de transações que são concluídas com sucesso em relação ao total de tentativas. O objetivo é ter uma taxa próxima de 100%.

**Tempo Médio para Concluir uma Transação**: Analisar o tempo que leva para uma transação ser processada. O objetivo é ter um tempo baixo e consistente, mesmo em horários de pico.

**Tempo de Recuperação**: Medir o tempo que o sistema leva para se recuperar e voltar a operar normalmente após um bloqueio.

**Melhorias sugeridas:**

**Testes de Regressão e Carga**: Realizar testes de estresse para simular o comportamento do sistema em horários de pico e identificar os pontos de falha antes que ocorram em produção.

**Refatoração de Código**: Melhorar o código para otimizar o processamento das transações.

**Revisão do Banco de Dados**: Analisar a performance e a configuração do banco de dados para garantir que ele suporte o volume de transações.

**Eficiência**

Essa característica se relaciona diretamente com a performance do aplicativo, especialmente a forma como ele utiliza recursos para entregar a funcionalidade. Os bloqueios indicam que o sistema não está sendo eficiente em lidar com o alto volume de requisições.

**Melhorias de Eficiência**

​Essas melhorias se concentram em otimizar o desempenho do aplicativo e a forma como ele utiliza os recursos do servidor.

​**Otimização da Infraestrutura do Servidor**: A infraestrutura que hospeda o aplicativo pode ser melhorada. Isso pode incluir a adoção de balanceamento de carga, que distribui o tráfego entre vários servidores para evitar sobrecarga, ou a migração para uma plataforma de nuvem que permite escalabilidade automática (aumento de recursos do servidor em momentos de pico).

​**Implementação de Caching**: A equipe pode usar técnicas de caching para armazenar temporariamente dados acessados com frequência (como saldo da conta ou extrato recente). Isso reduz a necessidade de fazer consultas demoradas ao banco de dados para cada transação, acelerando o processo.

​**Uso de Ferramentas de Monitoramento**: Ferramentas de monitoramento de performance (APM – Application Performance Monitoring) podem ser instaladas para coletar dados em tempo real sobre o uso de CPU, memória e a velocidade de resposta do servidor. Isso permite que a equipe identifique e resolva problemas de desempenho de forma proativa, antes que os usuários notem os travamentos.

**Sub-características relevantes**:

**Comportamento de Tempo (Time Behaviour)**: O sistema não está respondendo em um tempo aceitável sob carga, o que é a principal causa do problema.

**Utilização de Recursos (Resource Utilization)**: A performance do sistema pode estar sendo prejudicada por uma utilização ineficiente de recursos como memória ou processamento.

**Métricas propostas**:

**Tempo de Resposta do Servidor**: Medir o tempo que o servidor leva para responder a cada requisição. O objetivo é manter esse tempo baixo, mesmo com alto número de acessos.

**Latência Média**: Avaliar o atraso entre o envio de uma solicitação e a recepção de uma resposta.

**Taxa de Erro**: Medir o número de erros de conexão ou de processamento que ocorrem por minuto.

Melhorias sugeridas:

**Otimização do Servidor e da Infraestrutura**: Melhorar a arquitetura de back-end para que possa lidar com um número maior de transações simultâneas. Considerar o uso de servidores mais robustos, balanceamento de carga ou computação em nuvem.

**Otimização do Código do Aplicativo**: Identificar e corrigir gargalos de performance no código-fonte, como loops ineficientes ou consultas demoradas ao banco de dados.

**Uso de Caching**: Implementar técnicas de cache para armazenar dados usados com frequência, reduzindo a necessidade de novas consultas ao banco de dados.

**Monitoramento em Tempo Real**: Usar ferramentas para monitorar o desempenho do aplicativo e da infraestrutura, o que permite identificar e resolver problemas de forma proativa.