

Teste - Analista de Desenvolvimento Júnior - Suporte

Questão 1: Considere a seguinte estrutura de tabelas:

Pacientes	Atendimento	Prontuário
id_paciente	id_atendimento	id_prontuario
nome_paciente	id_paciente	id_paciente
tel_paciente	data_atendimento	id_atendimento
email_paciente	hr_atendimento	descricao_prontuario
status_paciente	nome_unidade	
data_nascimento	endereco_unidade	
	cd_especialidade	

- a) Escreva uma consulta SQL que retorne todos os atendimentos do paciente com a especialidade "Neurologia" nos últimos 6 meses.

```
SELECT * FROM Atendimento
WHERE cd_especialidade = 'Neurologia' AND
data_atendimento >= NOW() - INTERVAL '6 MONTHS';
```

- b) Suponha que você precise obter todos os prontuários associados ao paciente de ID 123, incluindo os detalhes do atendimento relacionado a cada prontuário. A consulta deve retornar o nome do paciente, a descrição do prontuário e a data do atendimento correspondente.

```
SELECT p.nome_paciente, pr.descricao_prontuario, a.data_atendimento
FROM prontuario AS pr
JOIN pacientes AS p ON p.id_paciente = pr.id_paciente
JOIN atendimento AS a ON a.id_atendimento = pr.id_atendimento
WHERE pr.id_paciente = 123;
```

Questão 2: Um cliente relata que está enfrentando lentidão em um determinado recurso do sistema. Como você abordaria esse problema para identificar a causa raiz e propor uma solução?

Primeiramente é necessário obter mais informações do cliente, sobre o problema. Fazendo perguntas como, qual recurso específico do sistema está enfrentando lentidão, quais são os principais problemas apresentado pelo o sistema, quando o problema começou e se afeta apenas este usuário e/ou outros também. Quanto mais informações forem obtidas neste primeiro contato, melhor será de entender a natureza do problema.

Além disto, é necessário coletar dados relevantes do sistema, como logs, métricas de desempenho e tempos de resposta do sistema, onde ajudará a ter uma visão mais objetiva da situação. Outra opção é analisar as métricas de desempenho para buscar identificar qualquer padrão de comportamento anormal. Ou seja, Verificar o uso de CPU, memória, recursos de rede e disco. Isso pode ajudar a identificar qualquer recurso que esteja sendo sobrecarregado.

Durante o processo de investigação ser possível comparar as versões anteriores com as atuais, pois pode ocorrer possíveis causa de lentidão nas consultas (queries) do banco de dados, que podem ser analisadas pelo inspecionar, através do network, também é possível observar os logs do sistema em busca de mensagens de erro, avisos ou eventos incomuns, também pode auxiliar na descoberta, pois pode mostra alguma mensagem importante sobre o problema que pode estar causando a lentidão.

Com base nas informações coletadas, agora é possível eliminar possíveis causas raiz. Usando a abordagem de eliminação, até chegar à causa raiz. Após identificar a causa raiz, é possível propor uma solução para resolver o problema. Isso pode envolver otimizações de código, ajustes de configuração, aquisição de recursos adicionais, etc. Além disto, antes de implementar a solução em produção e sempre bom testar em um ambiente de desenvolvimento. E por fim, é só monitorar/acompanhar o sistema para garantir que a lentidão tenha sido resolvida.

Questão 3: Um usuário informa que está recebendo uma mensagem de erro ao tentar realizar uma operação específica no sistema. Explique como você investigaria esse problema e quais etapas seguiria para solucioná-lo.

Quando um usuário relata que está recebendo uma mensagem de erro ao tentar realizar uma operação específica no sistema, acredito ser importante seguir algumas etapas como:

1. Obter Detalhes Precisos do Erro:

Solicitando ao usuário o fornecimento detalhado sobre a mensagem de erro, em que momento ela ocorre, qual ação ele estava tentando realizar e se o erro é consistente ou intermitente. Se possível o fornecimento de imagem ou vídeo que possa ajuda na compreensão/reprodução do erro.

2. Replicar do Erro:

Tentar replicar o erro no ambiente de acordo com as informações obtida pelo usuário, para entender as condições exatas em que ocorre o erro.

3. Verificar os Logs do Sistema:

Examinar os logs do sistema para verificar se há mensagens de erro ou exceções que possam estar relacionadas ao problema. Isso pode fornecer informações essenciais sobre a causa do erro.

4. Analisar o inspecionar

Inspecionar o sistema web na aba network e console para investigar a possível mensagem de erro no ambiente.

5. Pesquisar a Mensagem de Erro:

Pesquisar a mensagem de erro online para ver se outras pessoas já enfrentaram esse problema. Por exemplo conversar com colegas de equipe, pesquisar fóruns e comunidades dentro da empresa.

6. Correção:

Após identificar a causa do erro, propor e aplicar a correção. Testar a correção em um ambiente de desenvolvimento ou de teste para garantir que resolva o problema sem introduzir novos erros.

7. Feedback do Usuário:

É importante acompanhar/observar o usuário se o erro foi corrigido e garantir um bom feedback sobre a correção realizada do erro.

8. Documentar:

A documentação é sempre importante para cada etapa do processo, pois facilita futuras referências e colaborações.

Questão 4: Descreva como você lidaria com uma situação em que um usuário encontra um erro desconhecido.

Ao se deparar com um erro desconhecido relatado por um usuário, é fundamental começar coletando todos os detalhes do usuário sobre o erro, entre eles a mensagem de erro exibida, se existir e o passo a passo que levaram ao este erro. Após isso, tentarei replicar o erro no ambiente para entender a situação.

Depois observarmos as possíveis causas, considerando se o erro está relacionado a uma ação específica ou interações entre diferentes partes do sistema. Analiso os logs do sistema em busca de informações relevantes, ser necessário pesquiso em fóruns e comunidades de desenvolvedores por soluções. Verifico se todos os softwares envolvidos estão atualizados, pois muitas vezes erros são corrigidos em versões mais recentes e verifico o inspecionar em busca de mais detalhes do erro.

Por fim, testo gradualmente diferentes abordagens no ambiente de desenvolvimento para identificar a causa raiz do erro. Mantenha uma comunicação constante com o usuário, informando sobre os progressos e etapas tomadas para resolver o problema, demonstrando comprometimento. Se possível, utilizo ferramentas de depuração para rastrear o fluxo de execução do código e identificar onde o erro ocorre.

Se mesmo após diversas tentativas o erro persistir, tento buscar ajuda de colegas e dos desenvolvedores.

Questões de Lógica: Usando uma das seguintes linguagens de programação: C, C++, Java, C#, Javascript ou Python, resolva as questões a seguir.

Questão 1: Faça um programa, em seguida, um programa que solicite a entrada de dois números, em seguida imprime na tela o quadrado do menor número e a raiz quadrada do maior número, se for possível.

```
#include <stdio.h>
#include <math.h>

int main() {
    double numero1, numero2;

    // Solicita a entrada dos dois números
    printf("Digite o primeiro número: ");
    scanf("%lf", &numero1);

    printf("Digite o segundo número: ");
    scanf("%lf", &numero2);

    // Calcula o quadrado do menor número
    double quadradoMenor = (numero1 < numero2) ? pow(numero1, 2) : pow(numero2, 2);

    // Calcula a raiz quadrada do maior número
    double raizQuadradaMaior = (numero1 > numero2) ? sqrt(numero1) : sqrt(numero2);

    // Imprime os resultados
    printf("O quadrado do menor número é: %.2lf\n", quadradoMenor);
    printf("A raiz quadrada do maior número é: %.2lf\n", raizQuadradaMaior);

    return 0;
}
```

Questão 2: Faça um programa que solicite ao usuário o ano em que ele nasceu e verifique se o ano é Bissexto ou não e imprima uma mensagem na tela.

```
#include <stdio.h>

int verificaBissexto(int ano) {
    if ((ano % 4 == 0 && ano % 100 != 0) || (ano % 400 == 0)) {
        return 1; // Ano bissexto
    } else {
        return 0; // Ano não bissexto
    }
}

int main() {
```

```
int anoNascimento;

// Solicita a entrada do ano de nascimento
printf("Digite o ano em que você nasceu: ");
scanf("%d", &anoNascimento);

// Verifica se o ano é bissexto
if (verificaBissexto(anoNascimento)) {
    printf("O ano %d é bissexto.\n", anoNascimento);
} else {
    printf("O ano %d não é bissexto.\n", anoNascimento);
}

return 0;
}
```