

PRODUÇÃO TEMÁTICA PSI SCORE CESOA - Assistente Master TI

Ronaldo Francisco Alves da Silva - c093307-4

Bloco I Pergunta 2: Um bug crítico foi ... repita?

1. **Tente reproduzir o bug** em um ambiente de desenvolvimento ou homologação. Isso ajuda a identificar qual parte do cálculo de juros está incorreta.
2. **Escreva testes automatizados** que cubram os cálculos de juros em diferentes cenários. Isso ajudará a garantir que o cálculo esteja correto e a detectar quaisquer alterações involuntárias no futuro.
3. **Use ferramentas de depuração** e inspeção de código para entender o fluxo de execução e os valores das variáveis no momento em que o bug ocorre.
4. **Verifique o histórico de commits** relacionados ao cálculo de juros para identificar quais alterações podem ter introduzido o bug.
5. Depois de identificar a causa raiz do problema, **corrija o código de forma clara e concisa**, mantendo as boas práticas de programação e estilo de código.
6. **Execute os testes automatizados** criados no primeiro passo para garantir que a correção funcione conforme esperado e que o bug não se repita em outros cenários.
7. Depois de verificar que a correção está funcionando corretamente, **implante-a em produção** para mitigar os efeitos do bug.
8. Após a implantação, **monitorar o sistema** para garantir que o bug foi resolvido e que não haja efeitos colaterais.
9. **Registre o bug**, sua causa raiz, a correção aplicada e os testes realizados no sistema, para que outros desenvolvedores possam se beneficiar da sua análise e resolução.

Bloco I Pergunta 3: Uma rotina de processamento ... problema de desempenho?

1. A primeira etapa é **identificar as partes específicas do código ou processo que estão causando o gargalo**. Isso pode ser feito usando ferramentas de perfis que medem o tempo gasto por diferentes partes do código.
2. Uma vez que os gargalos tenham sido identificados, a próxima etapa é **otimizar o código** para melhorar seu desempenho. Destaco as seguintes táticas:
 - Se dados estiverem sendo repetidamente recuperados ou calculados, **armazenar em cache** os resultados.
 - Se o processamento puder ser dividido em tarefas menores e independentes, **o processamento paralelo** pode ser usado para acelerar o tempo de processamento geral.

- Se o gargalo estiver no banco de dados, **otimizar as consultas** do banco de dados ou **indexar as tabelas** pode ajudar.
 - **Simplificar o código**, remover laços desnecessários e reduzir o número de chamadas de função também podem ajudar.
3. Depois de otimizar o código, é importante testar e medir o desempenho para garantir que as alterações tenham tido um impacto positivo.
 4. Uma vez que as alterações tenham sido implementadas, **é importante monitorar o sistema** para garantir que as melhorias de desempenho sejam sustentadas ao longo do tempo.

Bloco II Pergunta 4 - Crie um módulo de autenticação e autorização ... Apresente o código-fonte e explique a implementação

Explicação da Implementação

1. Configuração do Banco de Dados:

- `ApplicationDbContext.cs` configura o Entity Framework Core com suporte ao Identity.

2. Modelos:

- `ApplicationUser.cs` estende `IdentityUser` para incluir um campo `FullName`.

3. Registro e Login:

- `Login.cshtml.cs` e `Register.cshtml.cs` gerenciam a autenticação de usuários.
- Após o login bem-sucedido, os usuários são direcionados para a página do respectivo Role, haverá a validação do Role com a tag `[Authorize(Roles = "EspecialUser")]`. Na aplicação foram criados 3 roles - Admin, User e EspecialUser

4. Gerenciamento de Usuários:

- `ManageUsers.cshtml` e `ManageUsers.cshtml.cs` permitem que administradores visualizem usuários cadastrados. Ali é possível alterar Roles e elevar de privilégios.

5. Layout:

- `_Layout.cshtml` define a estrutura comum, incluindo navegação e partial para login.

6. Autorização:

- `ManageUsersModel` é protegido por `[Authorize(Roles = "Admin")]` para garantir que apenas administradores acessem a página.

Este módulo fornece autenticação e autorização básica com controle de acesso baseado em roles para uma aplicação web. O código fonte completo está disponível em <https://github.com/ronaldofas/user-role-example>.