Daniela M. Arantes

Generation

ESTUDO DE CASO – JUnit

Esta é a primeira semana de trabalho em seu novo emprego como desenvolvedor java júnior. Você integrará uma equipe de 10 pessoas, elaborando códigos e implementando novas funções no mesmo banco de códigos. Na primeira tarefa para a qual foi escalado, você corrigiu um erro referente à exclusão de credenciais do usuário após execução de logout. Depois de enviar o código, você percebeu que, embora a função de logout esteja funcionando corretamente, uma outra função deixou de funcionar e, consequentemente, os usuários foram impedidos de efetuar login.

- Como podemos garantir que as alterações feitas não interrompam a funcionalidade anterior em grandes sistemas?

Aplicando o que vimos hoje, utilizado os recursos do JUnit em uma classe de teste, assim teremos tempo hábil para identificar e corrigido futuros erros, tornando nosso código mais confiável.

- Como podemos ser rigorosos em um cenário como este?

Primeiramente, antes de enviar o código devemos realizar todos os testes para garantir que ele executará perfeitamente e manter a comunicação com a equipe, solicitar que um algum colega de trabalho revise antes de concluir.

- Por que é importante elaborar testes de unidade?

Para não ocorrer erros parecidos ao do exercício, o JUnit indica qual melhor método de execução, tornando o código mais ágil.

- Que situações semelhantes à do miniexercício apresentado na seção de abertura você acredita que enfrentará quando estiver no trabalho? Por quê?

Acredito que pode acontecer com qualquer pessoa iniciante, as vezes por receio de pedir para alguém conferir antes de entregar ou até mesmo para demonstrar eficiência, mas precisamos ser humildes para pedir ajuda e sempre saber trabalhar em equipe.