**ESTUDO DE CASO**

*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? Como podemos ser rigorosos em um cenário como este?

Alem de usar as funcionalidades do GitHub para não quebrar o codigo fonte, também acho necessário o uso de testes.

## Por que é importante elaborar testes de unidade? Que situações semelhantes à do miniexercício apresentado na seção de abertura você acredita que enfrentará quando estiver no trabalho? Por quê?

É importante para prevenir que outras funções do codigo sejam afetadas, e que não o corra erros.

Jack é um desenvolvedor java júnior que está frustrado por ter aplicado mudanças em seu projeto e interrompido o funcionamento de todos os testes. Mesmo com suas diversas tentativas de interpretar o código por horas, alguns dos conceitos e bibliotecas ultrapassam sua experiência. Ele está muito frustrado e não sabe o que fazer.

* Por que a persistência é necessária para resolver a questão?

Porque com essas tentativas e erros irao ajudar ele a obter mais conhecimento e auxiliarão a não cometer o mesmo erro novamente.

* De que forma a proatividade pode ser útil?

Se ele pedir ajuda alem de aprender cmo outra pessoa pode economiza tempo.