

## **Tipos e técnicas de teste**

Além dos testes automatizados de interface, a aplicação também pode ter testes unitários, para validar as regras de negócio desenvolvidas no back-end, testes de API, para validar os serviços por trás das telas e testes de performance, para verificar o comportamento da aplicação diante de uma determinada situação.

Caso seja objetivo da empresa validar a aceitação da interface, ainda podem ser realizados testes alfa e beta.

Para maximizar o valor dos testes automatizados, podemos utilizar de técnicas de desenvolvimento de software (BDD e TDD) e ainda coloca-los para executar em uma ferramenta de integração contínua (Jenkins).

Os testes de unitários podem ser executados antes do merge do código desenvolvido com o código disponível no repositório remoto, verificando se a alteração não gerou impacto nas demais partes da aplicação.

Os testes de API podem ser executados várias vezes ao dia, para verificar o comportamento dos serviços e se alguma alteração não impactou nos seus resultados.

Os testes de interface podem ser executados pelo menos uma vez por dia, fazendo a regressão da aplicação.

A execução de todos estes testes em uma ferramenta de integração faz com que o time tenha um feedback rápido e constante sobre as alterações que estão sendo realizadas, não cria dependência sobre uma pessoa para executa-los e ainda guarda histórico de todas as execuções.

## **Demais considerações**

Julgo ser importante em um projeto de testes automatizados, que ele seja capaz de notificar as pessoas responsáveis pela aplicação em caso de falhas, para que as medidas sejam tomadas o mais rápido possível.

Também é importante ter um relatório que sintetize os resultados obtidos que possa ser consumido por todo o time.