1.A área de Quality Assurance possui diversas premissas e ideologias a serem seguidas para o bom ato da carreira, a fim de trazer o significado de seu título aos projetos onde o QA está envolvido. Conte-nos sobre suas experiências passadas como QA, suas ambições e suas principais características da profissão.

Dentro da área de tecnologia possuo 12 anos de experiência e com QA possuo 10 anos de experiência profissional. Comecei atuando com testes manuais, documentações e regras de negócios. Após um 1 ano, tive meu primeiro contato com automação de testes, atuando com a linguagem java. Trabalhei com a linguagem java, testes de back e front end também. Em 2018 comecei a entender mais a parte técnica, comecei a escrever meus estes com BDD, utilizando o framework Cucumber. Além disso atuei na área da gestão da configuração e da mudança, onde fui responsável pelas liberações de versões. Também fiz algumas visitas piloto no cliente. Acompanhando o dia a dia do funcionamento de uma agência.   
 Minha ambição é continuar estudando e melhorando meus conhecimentos, todos os dias. Pois tenho muita vontade de me tornar um líder técnico em breve e a longo prazo me tornar gerente de projetos.   
 Minhas principais características dentro da profissão é ser uma pessoa curiosa, sempre em busca de melhorias, gosto de entender o processo, regras dos sistemas e gosto de testar estas regras e até mesmo fazendo o processo ao contrário. Acho muito válido, garantir a qualidade de um produto. Tentando trazer uma melhor versão deste produto ao consumidor final. Pois acredito que na busca do conhecimento, compartilhando informações com o time, com muita comunicação e entendimento, podemos realizar uma excelente entrega para o cliente.

2. O Desenvolvedor de Testes Automatizados trabalha paralelamente com a Qualidade e, em muitos casos, o próprio desenvolvedor é mutuamente o QA da equipe. Conte-nos sobre suas experiências passadas como Dev-Tester: (sinta-se à vontade para comentar quais os tipos de testes na qual já trabalhou).

Durante as minhas experiências, tanto como testes manuais, testes automatizados, testes de performance e exploratórios, puder perceber muitos pontos. O principal deles, é que sempre irá existir uma versão melhor. Estamos sempre na busca de uma melhoria, de correção de um bug. Muitos dos casos, sempre atuei próximo ao desenvolvimento e a parte mais legal, foi que pude aprender muita coisa na área de desenvolvimento de software e além de sempre sugerir melhorias para os sistemas para qual já trabalhei. Sempre eu mostrava ao desenvolvedor, outro caminho, pelo qual ele não estava esperando. Isso garantiu muitas melhorias e pudemos evitar alguns bugs no sistema. Por outro lado, durante o tempo de automação, eu já conhecia a regra de negócio melhor, então eu explicava para alguns desenvolvedores como melhorar alguns testes automatizados. Eu já desenvolvia alguns scripts com apoio dos desenvolvedores, porém eu tinha o conhecimento da regra de negócio mais focado, por executar este passo manual anteriormente.

3. Explique a Pirâmide de Testes, o que a literatura diz sobre o assunto. Disserte seu ponto de vista a favor e/ou contra sobre o conceito. Qual o valor dos testes de cada camada?

A função da pirâmide de testes é basicamente definir níveis de testes e dar um norte quanto à quantidade de testes que deve ser feito em cada um desses níveis.

No topo da pirâmide, temos os testes de ponta a ponta (end to end). O objetivo é imitar o comportamento do usuário final nas aplicações (seja ele uma pessoa, uma api, ou qualquer outro tipo de cliente).

Na base, temos os testes de unidade, onde verificamos o funcionamento da menor unidade de código testável da nossa aplicação.

Entre essas duas camadas, temos os testes de integração. A ideia é verificar se um conjunto de unidades se comporta da maneira correta, só que de forma menos abrangente do que os testes de ponta a ponta.

Os valores destes testes em cada camada é assegurar que o produto tenha uma melhor qualidade e que os clientes finais fiquem mais satisfeitos.

A importância é dividir e principalmente, fazer testes durante o ciclo de desenvolvimento e não apenas no final.

Também podem auxiliar na manutenção e evolução do sistema, pois com uma boa suíte de testes, desenvolvedores com menos experiência na regra de negócio do projeto terão uma segurança muito maior ao realizar o desenvolvimento de novas funcionalidades.

4. Explique o TDD e o BDD. Disserte seu ponto de vista a favor e/ou contra.

“TDD” é o Desenvolvimento Guiado por Testes. E “BDD” é o Desenvolvimento Guiado por Comportamento.

O TDD busca garantir a qualidade do código, sempre pensando em 100% de cobertura de testes, melhorar o que acabou de ser feito e nunca escrever uma linha de código sem antes pensar em como garantir que aquilo irá funcionar.

E o BDD trabalha para definir como uma demanda chega ao desenvolvedor, integrar diferentes áreas da empresa e pensar a partir do ponto de vista do comportamento esperado de uma funcionalidade pelo usuário. Por consequência, ele acaba influenciando em como os testes são planejados e escritos.

Segundo as definições acima, eu sou a favor, pois podemos aplicar ambos os métodos em conjunto ou apenas um deles. Os dois buscam melhor o desenvolvimento de software e são excelentes ideias para se ter em qualquer projeto ou empresa. Já fazem uns 2 anos que eu venho utilizando BDD e na minha opinião, está sendo muito mais útil e mais fácil para entender.

5. Evidências são parcialmente polêmicas entre equipes distintas de QAs. Seu valor é notável, porém a aplicabilidade pode ser desfavorável dada a demanda de sua Squad. Qual o valor de uma evidência, no que ela pode agregar e como podemos executá-la em um projeto de testes automatizados. (sinta-se à vontade para exemplificar uma situação).

A evidência, pode ser a maior prova do nosso trabalho com ela podemos servir de insumo e métricas, documentar melhor o software, rastrear problemas e alimentar o aprendizado do time.

Apesar de algumas pessoas acharem que evidência poderá ser para descobrir culpados, expor e apontar erros e pessoas, não vejo desta forma. E sim, agregar qualidade. Um exemplo clássico, é que dependendo o ambiente utilizado, tal sistema poderá funcionar em algum momento e em outros ambientes poderá ocorrer algumas falhas. Com isso a importância pela evidência e justificar como a falha ocorreu. Em um ambiente de automação, o bom é que podemos deixar os testes rodando e após um certo tempo, podemos verificar todos os "steps" destes testes automatizados e verificar se ainda ocorre falha em algumas rotinas que foram aplicadas o teste de regressão. Com isto, ganhando tempo para o time explorar e testar novos caminhos.

6. Conhecemos os valores dos Testes, mas ao analisarmos métricas e caminhos, preocupante seria decidir a melhor escolha. Quando que o teste automatizado vai agregar mais valor que um teste manual, e quando um teste manual vai agregar mais valor que o automatizado. Explique e dê um exemplo.

O teste automatizado vai agregar mais valor quando, utilizamos o mesmo para realizar testes de regressão (testes repetitivos), geralmente a automação agrega valor e tempo nestas tarefas, fazendo assim com que o teste manual possa agregar valor nos testes exploratórios e dar início em novas rotinas de testes. Podendo realizar novos testes manuais e descobrindo novas falhas. Enquanto o teste automatizado é executado automaticamente e explorando as mesmas rotinas já testadas manuais anteriormente, o teste manual sempre é uma boa opção para validar novos caminhos e descobrir novas falhas.

7. Conte-nos sobre Testes de Desempenho e Testes de API, quais são os valores desses testes e como podemos executá-los. (Explique um exemplo).

Os testes de desempenho mostram que determinada aplicação precisa para ser melhorada, antes de ser disponibilizada para o público. Sem a aplicação desses testes, o software pode apresentar diversos problemas, como exemplo, lentidão com uma grande quantidade de usuários acessando ao mesmo tempo; apresentar inconsistências entre plataformas diferentes; possuir falhas e funcionalidades faltando ou inoperantes.

De outro lado, o teste de API é igualmente importante como qualquer outro tipo de teste de aplicativo, pois ajuda a garantir o funcionamento, o desempenho e a confiabilidade de diferentes aplicativos e sistemas baseados em dados, ao certificar as trocas e comunicações entre aplicativos, sistemas, bancos de dados e redes.

Os testes de API´s são altamente necessários, pois ajudam a cobrir cada vez mais os requisitos de testes de APIs e implementações de aplicativos de segurança e qualidade.

Nos testes de API´s, já atuei com testes funcionais utilizando as ferramentas Postman e Swagger. Com testes automatizados já utilizei o framework restassured.

8. Existem artigos que comentam sobre o “fim” dos QAs e dos Testers. Para ambas as profissões o novo título dado a essa evolução é o DevTestOps. Como o próprio nome já diz, trata-se de um profissional que entende sobre QA, Dev.Tester e DevOps. Comente sua opinião sobre o assunto, e explique o valor do CI-CD em um projeto de Testes Automatizados, e nos dê um exemplo de como pode ser criado. Links sugestivos:

https://www.linkedin.com/pulse/what-devtestops-chedy-missaoui

https://gaea.com.br/entenda-como-os-devops-estao-mudando-os-qa/

Acredito que o papel do testador, não é somente testar aplicações, tanto como manual e automatizadas. Na verdade, o papel de um testador, vai muito além disto. O profissional pode atuar tanto na parte de processos, metodologias ágeis, documentação de regras do sistema, entender o cliente, prestar algum eventual suporte, além de validar as regras do sistema e garantir a qualidade em todas as áreas do projeto ou de uma empresa. Pois garantindo todas estas partes, acredito que podemos entregar um produto com a qualidade melhor. A integração contínua e entrega contínua já vem fazendo parte deste cenário, com isto asseguram um deploy automatizado, agregam e reportam bugs permanentemente. Por isso é sempre importante investir no conhecimento técnico, pois a automação cada vez mais, vem se tornando ágil e realizando tarefas que antes eram feitas manuais. Acredito que é possível trabalhar com DevOps e em parceria com outras metodologias, como a organização por Sprints trazida pelo Scrum ou até com alguns aspectos das metodologias tradicionais. Chamado de TI Bimodal, estrutura híbrida que deve estar presente nas organizações.