# Entrevista 22 - Entrevistado



Analista de Engenharia Pleno. Eu desenvolvo em Csharp.net, atuo com desenho de arquitetura voltado para a Cláudia WS e manutenção e sustentação do projeto. É uma única equipe com duas frentes e atualmente eu estou somente em um projeto, mas eu já trabalhei nos dois também paralelamente.

Certo. Ok. Dois.

Sou eu e um outro sênior que atua junto comigo. E aí não faço papel de liderança, nem ele também, mas aí nós dois nos conversamos dentro do projeto. Certo.

Supondo que aqui a gente já está falando da metodologia AJA, a gente já tem a feature descrita dentro do board. O primeiro passo é abrir o repositório dentro do GitHub, já contando que ela já está refinada, eu já sei exatamente o que eu preciso fazer. Se eu tiver ela localmente, eu vou descrever o Focus como se eu não tivesse ela local.

Eu faço o clone dela para a minha máquina, eu crio uma branch com base na main, que é a versão que está em produção no momento, e aí eu faço todas as alterações que são necessárias. Faço a parte de testes também, inicialmente em ambiente local. Eu preciso falar os comandos também? Ah, certo.

No caso, você quer que eu detalhe aqui a parte do código... Isso, entendi. Certo, perfeito. Entendi agora.

E aí eu crio a branch com base na main, a branch tem que ter uma estrutura, que ela tem que iniciar com feature e barra, e aí você escolhe o título que você quer colocar da sua branch. Isso é obrigatório para passar pós-request. Após isso, a gente sobe as alterações via git push, e ela vai passar pela primeira pipeline, que é a pipeline dentro do próprio GitHub ainda.

E quando é feito o merge, inicialmente para develop, é feita todas as validações, tem as validações de infra, as validações de cobertura de código, de testes, e aí o ambiente de desenvolvimento pode ser validado, as alterações, com tudo certo, é validado ali para ser validado, é feito o pure quest de ambiente de desenvolvimento, isso aqui é uma escolha da minha equipe, não é obrigatório para todas as equipes escolherem a minha. Para desenvolvimento, a gente não pede code review para aprovação de pure quest, a gente pede o code review da aprovação de desenvolvimento para homologação, que é o ambiente ali que a gente faz os testes realmente integrados. Então, quando eu vou mandar ali da develop para a release, que é para onde vai o ambiente de homologação, eu preciso da aprovação do pure quest e de um code review, e eu tenho que estar disponível para tirar qualquer dúvida que surja ou algo do tipo.

Após o ambiente de homologação, é feito os testes totalmente integrados, nesse ambiente a gente não tem, a develop a gente ainda consegue manipular alguns dados para ver o comportamento, o ambiente de homologação a gente não consegue mais fazer isso. Com tudo certo, a gente abre um pure quest para a main, mas a esteira não é executada nesse momento, a esteira só vai ser executada após o fluxo de gestão de mudança, que é onde a gente cria uma GMUD, a gente preenche o que quer ser feito, a gente atende os horários permitidos da janela de implantação, se tudo dá certo, a GMUD fica em estado de agendar, e no horário que a gente marcou na GMUD tem uma integração, a própria pipeline gera o ID da GMUD, a gente pega quando a GMUD está no horário e já está em agendar, ou seja, com todas as aprovações definidas, ela vai ser executada e vai mandar o código para o ambiente de produção, e é mais ou menos isso o fluxo todo. Em outros projetos sim, já usei, não, era bem diferente mesmo, a gente trabalhava com implantação em pacotes, então não, tinha o GitHub para a gente guardar a versão do último pacote, mas não tinha uma pipeline integrada, a gente pegava o pacote, e aí a gente mandava esse pacote para um software interno deles, e a gente fazia a GMUD, escolhia a data, e na data uma pessoa realmente pegava esse pacote e aplicava em ambiente de produção, então é bem diferente mesmo.

Os benefícios, no caso. Em termos de benefício, eu acredito muito que a parte de a release foi um, eu já trabalhei também quando não tinha branch release, então era só developer e master, e causava um pouco de confusão ali, porque a master não era um ambiente produtivo, ela era um ambiente de homologação, então a gente perdia um pouco de noção do que estava em homologação e do que estava em produção, a gente só conseguia saber isso vendo a aplicação, então isso era uma dor muito grande trabalhar dessa forma, e aí já teve até conflitos de coisas que não deveriam ter ido para o ambiente produtivo, e foi porque estava na branch, então acho que o maior fator positivo para mim é a branch release, que ela cria a versão do que você tem ali, então qualquer alteração que você fizer, você vai ter que criar uma nova versão de release, eu acho que esse é um dos fatores que eu vejo mais benefício no cenário atual. Um fator que não favorece? É um fator que eu acho que não tem relação com o fluxo, mas é as pipelines são bem problemáticas, mas não sei se valeria colocar, porque não tem uma relação com o próprio fluxo, mas lá a gente sofre muito, porque a pipeline é bem estável e ela apresenta muitos problemas, acho que esse é um ponto negativo, não sei se isso é a resposta que você queria, mas eu acho que o ponto negativo do fluxo para mim é a instabilidade da pipeline, levar o código, subir o código para outro ambiente, não, eu gosto do modelo atual bastante, empresa estabelecida, no modelo anterior tínhamos esse modelo atual com release e não, é bem ativo a gente ter problema, não tem realmente.