*Primeiro agradecer por participar dessa entrevista. E aí a primeira coisa, como é que está a questão de trabalhar com o Flutter? Tu tens trabalhado depois do projeto ou já está numa nova linguagem? Como é que está isso? Depois do projeto eu dei uma parada, apesar de querer continuar estudando, porque sempre é bom aprender, ter uma nova ferramenta. Só que agora eu estou trabalhando com o pai, porque eu entrei no estágio. Qual era o teu papel no time? O meu papel era... Eu era o que? Eu fui dar a parte de back-end, digamos assim, da integração do banco de dados do aplicativo. Quais eram as features que tu mais trabalhou no projeto? O que são features? Funcionalidades, recursos do projeto, do aplicativo que tu trabalhou. Consegue lembrar? Deixa eu lembrar aqui. Deixa eu puxar aqui no Trello. Ok. Eu trabalhei na parte de... foram várias, assim. Na parte de criar demanda, de vincular alunos, abrir convocações, encerrar convocação, lista de projetos, detalhes de projetos, lista de demandas, várias coisas. Qual foi a feature que mais te marcou no projeto? O que mais me marcou... Eu acho que nem sei dizer, porque... Mas deixa eu ver, assim, a mais difícil. Eu acho que foi lá pro final, porque eu percebi... Quando eu tava finalizando, eu percebi que eu tinha feito algumas coisas erradas. E aí eu tive que voltar algumas coisas desde o início, das coisas que eu tinha feito anteriormente. Eu acho que talvez tenha sido na parte de visualizar gerente. Beleza. É... tá. Como tu explicaria a CDD pra um colega teu? Eu explicaria que o CDD, ele é uma forma de organizar melhor o teu código. É, através de, digamos assim, pontos que tu dá pra cada tipo de complexidade de uma linguagem. Por exemplo, o if vai ter um ponto, a função vai ter um ponto, os strings vão ter um ponto, enfim. Cada uma dessas vai ter um ponto diferente. E aí vai ter um limite, né? E o objetivo é estar ou igual ao limite ou abaixo do limite. Isso aí serve pra diminuir a complexidade de leitura do código. Pra tornar o código menos complexo de ler. É... tu achou que realizar essas atividades de CDD no projeto foi algo tranquilo? Que tu pegou de primeira? Ou foi um processo gradual? Foi um processo gradual. No início tava meio obscuro, aí depois foi clareando mais. Entendi. Tu acha que o CDD foi útil? Eu acho que em alguns... Eu acho que em algum... algumas... Alguns códigos foi útil, porque realmente estavam muito grandes. E outros códigos nem tanto. Porque, apesar... De... os pontos não corresponderem ao que estava combinado. Mesmo assim dava pra ler bem o código. Tu acha que teve uma etapa do projeto que ele foi mais útil? Ou menos útil? Ah, uma etapa... Eu acho que pro final. No final foi mais útil, no caso. No final? Sim. Entendi. Ahn... Teve alguma etapa que tu mais gostou de fazer? Que tu achou tranquilo? Se identificou mais em fazer? Alguma etapa? Alguma etapa de aplicar o CDD? De aplicar o CDD. De aplicar o CDD? Foi nas... nos projetos... nas features... Os códigos das features que eu fiz. Depois que ficou bem grande. Foi legal refatorar eles. Então tu diria que ele foi... Tu ficou mais... tranquilo de fazer essas atividades durante o processo de refatoração de classes. Aham. É... Olhando pra tabela final. A versão final da tabela de CP. Que o time produziu. Tu concorda com todos os itens que estão ali? Ou tem um item particular que tu acha que não faz sentido? Ou que tu gostaria de adicionar algum outro item? Ah... deixa eu ver... Hum... Eu acho que talvez o acoplamento. Acho que talvez seja um item de necessário. Consegue dizer porque? Porque não é um... difícil assim... não é complexo saber o que tá acontecendo quando tem uma... Uma... um acoplamento de função. Na minha opinião. E alguns caras... E o if, na minha opinião, ele é meio... Cinzento. Porque tem... Partes de if que são muito fáceis de entender. E as vezes... É... É... Chato... ... Principalmente quando eles estão muito... É... encadeados. Pode voltar um pouco a tua fala? Porque travou aqui pra mim os áudios. Ah, sim. É... É... Eu acho que a função de acoplamento... Eu acho que talvez não seja necessário colocar o ICP. Eu acho que na minha opinião não é tão difícil de ler quando tem um acoplamento. E a questão do if-else... Do if-else... Do forno, entanto. Mas mais do if do else. É... Também até... Também do... Do... Do ternário. É... Muito cinzento. Porque tem alguns ifs que são fáceis de entender. Quando é... Tem apenas um if. De... Digamos um if pra mostrar algo numa tela, né? E tem outros ifs que são mais difíceis, na minha opinião. São... Principalmente quando eles estão encadeados. Esses ifs mais simples é... É... É chato tu colocar o ponto de ICP. Mas esses ifs que estão mais encadeados dá pra entender. É... Tem mais ou menos alguma ideia... Quer dizer... Tem um pensamento que diz que você fazer acoplamento numa classe dificulta a sua... Porque você precisa navegar pra outras classes que estejam ali, por exemplo, acopladas nessa classe. É... Hum... Pela tua fala, tu disse que tu não concorda porque um acoplamento não dificulta. É... Então, tu discorda desse... Dessa fala? Se o nome da função for um nome bem... Descritivo do que ela faz, né? Entendi. Acho que... É... É... É... É... É... É... É... É... É... É... É... É... É... É... É... Eu acho que fica fácil de entender. Beleza. É... Tu lembra se no processo ali de construção da tabela houve muita discordância pela equipe na hora de desenvolver, de escolher os itens, de... É... De colocar um custo pra esses itens ou foi algo mais de boa, a equipe foi mais... Entrou em consenso quase sempre? A equipe na maioria das vezes entrou em consenso. Às vezes discordaram um pouco. Mas... na maioria das vezes está de consenso. Lembra de alguma coisa específica? Algum item específico que a equipe colocou que tu discordava? Deixa eu ver. Acho que esse é o complemento. E... E só. O complemento e só. Tem algum item que tu adicionaria que a gente não pensou, não colocou na tabela de sugestão? O que tu pensou depois? Talvez percebeu? Eu acho que talvez seria interessante separar o if, o else do for e do while. Acho que na minha opinião principalmente quando eu precisava fazer alguns filtros do banco de dados, eu acho que era muito mais difícil entender o for do while do que do if e do else. Acho que não fazia sentido. Esse seria o mesmo ponto. Essa é uma pergunta que eu ia fazer também aproveitando para fazer logo. Então, tu acha que a tabela da maneira como ela ficou tendo custos mais específicos faz mais sentido do que padronizar os custos para um, por exemplo? Todo mundo ter o mesmo custo? Eu acho melhor especificar mais. Tá, ok. E aí, no caso, if e else teria um custo menor que o while, por exemplo? Do que do for e do while. Entendi. Quais são os critérios pessoais que tu utilizou para decidir um item ou para determinar o custo de algum desses itens da tabela? Eu determinei a experiência que eu tinha de usar aquela ferramenta da linguagem, por exemplo provider, block, block, o future, o stream e o stream builder que eram coisas que eu não estava acostumado a mexer. Então, isso interferiu na forma que eu escolhesse os pontos da tabela. E o custo? Como é que tu definiu o custo? Se eu acho uma por exemplo, se eu acho um stream builder difícil de usar, eu coloco alguns pontos a mais. Por exemplo, se eu não tenho tanta habilidade com o flutter block, eu coloco também pontos mais complexos. Entendi. Faz comparação entre os itens que tem na tabela? Sim. Depois de feita, construída a tabela inicial, e aí vocês começaram o processo de codar as features ali do sistema, esses requisitos. Durante esse processo, qual era a frequência que tu usava a tabela? Tu olhava para ela? Foi um mapa constante que tu utilizava enquanto estava já codando ali? Ou era mais um guia que tu olhava de vez em quando? Eu usava a tabela toda vez quando eu terminava de fazer uma feature. Então, eu terminava de fazer uma tela, eu terminava de fazer uma funcionalidade do aplicativo, e logo na sequência eu olhava a tabela e tentava colocar os pontos, e se passasse dos pontos, eu tentava repatorar. Então, no teu processo de codar, tu não pensava em CDD? Só pensava no final? Isso. No final, só pensava no final. Ok. Tá. A gente teve três versões da tabela. Teve uma versão inicial, uma versão que mudou ali no meio, e essa última versão que foi usada na repatoração. Tu acha que essas constantes reuniões que tiveram toda semana para fazer um refinamento da tabela foi algo que mais ajudou ou mais atrapalhou? Eu acho que ajudou. Porque, por vir na tabela, a gente tinha alguns pontos a mais de CDD, e isso depois foi corrigido. Então, eu acho que ajudou. Entendi. Tá. Então, é... Como foi o teu processo para identificar, apontar o ICP e calcular ele no final? Como foi o processo de quê mesmo? Para identificar o ICP dentro do teu código, apontar ele, né, dizer, colocar alguma coisa para mostrar onde ele está, e depois calcular. Como foi esse processo? Tu fazia no começo, durante, no final? Eu fazia sempre no final. Terminava lá de codar, e aí depois eu pegava a tabela, analisava, pegava a tabela, né, e analisava em cada categoria que tinha colocado, em cada widget diferente, e depois eu ia dando os pontos e no final ia somando. Teve alguma classe que fazer isso foi difícil? Teve. Lembra, pode falar algum exemplo? Deixa eu ver aqui no Travel. Eu acho que foi a classe de visualizar projetos, que ela era muito grande. E a classe de visualizar demandas, que ela também era bem grande. E aí depois a gente tinha que estourar dois pontos e precisava repatorar elas. Entendi. Então, com que frequência tu ultrapassou esse limite nesse processo de codagem? Nesse processo de codar, quando tu ia lá e fazia a contagem, era frequente você passar do limite permitido? Não, era bem raro passar do limite. Só ficou frequente depois que a gente decidiu na terceira versão diminuir os pontos máximos de CP, que aí foi praticamente todas. Tu acha que a partir disso mudou o teu estilo de programar ou se manteve? Eu acho que se tivesse continuado o projeto, provavelmente teria mudado sim. Como mudaria? Eu acho que mudaria que eu tentaria criar mais funções, tentar diminuir o código, e tentaria criar mais classes também para diminuir o código. Entendi. Quando tu estava nesse processo de equilibrar os CPs da classe, quando essa classe ultrapassava o limite e quando tu ia refatorar, o que tu decidia que era mais crítico de resolver? Pode repetir a pergunta? Quando estava no processo de refatorar a classe para colocar ela de volta para o limite quando ela ultrapassava o limite permitido, como tu decidia o que era mais crítico para resolver? Bom, eu pegava uma feature e, por exemplo, tinha parte dessa feature que cada classe resolvia um problema dela. E aí eu tentava, eu refatorava, tirava essa classe, aí eu transformava numa só e fazia um acompanhamento disso. Eu fazia dessa forma e diminuía o código. Era uma espécie de componentização? É, eu fazia isso, era uma componentização. Entendi. Quais eram os itens que tu tentava geralmente extrair das classes? Tinha item específico que tu mirava? Ou, dependendo do contexto, ou tu ia em alguma outra estratégia? Eu ia mais na questão de CP mesmo. Eu tentava mesclar na questão de CP e tentava mesclar na questão de funcionalidades da classe. Por exemplo, se uma classe fazia uma coisa só, eu não costumava cortar ela na metade. Eu pegava ela toda, e refatorava ela e colocava numa outra parte do arquivo. Então não tinha ICPs que tu, itens da tabela que tu focava mais para tentar resolver. Dependia do contexto? Era, dependia do contexto. Claro que teve vezes que se eu não tinha ICPs, eu não tinha ICPs. Claro que teve vezes que se eu fizesse só isso, não ia adiantar. Aí teve alguns arquivos que eu fui mais pelo ICP mesmo, de tentar cortar o ICP pela metade. Entendi. Tu acha que o CDD gerou um impacto positivo no projeto? Eu acho que sim, né? Porque que ainda não a gente conseguiu componentizar mais, que em alguns casos, na minha opinião, foi uma componentização boa, em outros casos nem tanto. Porque apesar de ter estourado, às vezes, o número de ICPs, às vezes não era difícil de ler. E aí algumas coisas, eu não tinha muito sentido em refatorar. Tu acha que em todos os casos, a gente tem que fazer uma coisa que é mais simples, que é o CDD gerar uma maturidade? Tu acha que em todo o processo de trabalhar com CDD, teve algum momento que a complexidade daquela classe, ela não refletia tanto a complexidade que estava sendo atribuída e depois teve mudança? Teve. Consegue dizer o momento em que, se é que teve, né? Em que a classe, ela tinha uma complexidade em que o que era atribuído de complexidade através do CDD não representava exatamente aquela complexidade. Eu acho que teve. Acho que na minha opinião, acho que foi numa classe chamada Saim, se não me engano. Eu não estou lembrado. Mas teve, teve. Por exemplo, de acordo com o acordo que a gente tinha feito de limite de ICP, ele tinha estourado, mas não era tão difícil de entender. E acabou que ficou gerando mais um arquivo. Entendi. Tu acha que teve algum impacto negativo em usar CDD no projeto? Negativo, negativo não. Acho que teve mais impacto positivo. Qual seria um indicador que tu mostraria para determinar esse impacto positivo? O indicador seria na organização dos arquivos e na facilitação da leitura dos arquivos. Beleza, entendi. Tem algum benefício que tu conseguiu observar enquanto trabalhava com CDD? Benefício... Eu acho que foi a mesma coisa que eu falei. Uma maior facilidade na leitura. Tá. Em que momento... Em que momento tu diria que o CDD fez realmente a diferença no projeto? Eu acho que eu diria mais para o final, em que a gente conseguiu aplicar mais o CDD, já que durante o projeto a gente quase praticamente não estourava ele. Então foi mais para o final, em que o limite caiu bastante e a gente teve que cortar bastante coisa. Ok. Sobre a qualidade de código que tu produzia, tu acha que o CDD ajudou a melhorar a qualidade do código? Qualidade do código? Eu acho que sim. É, qualidade do código... É, tipo... Depois que eu vi que estourava o limite, depois que eu vi que tinha arquivos muito complexos de ler, eles eram componentizados e eram mais fáceis de ler depois que eu aplicava o CDD. Tu leu o código dos teus colegas durante o processo? Sim. Tu acha que o CDD ajudou nessa legibilidade? Sim. Eu acho que no início, não, porque como eu disse, a gente quase não estourava o limite e acabava que era um código, na minha opinião, complexo de ler. Mas depois, quando a gente foi fazendo isso no final, ficaram mais fáceis de entender. Tá, entendi. Mas eu acho que uma coisa que foi ser bastante na leitura dos arquivos foi a organização do código. Foi a organização de pastas. E dificultou bastante, porque às vezes as coisas ficavam muito espaçadas. Nos arquivos a gente não sabia exatamente onde é que estava. O CDD, ele foi capaz de te indicar um bom caminho na hora de repatorar? Um bom caminho? Teve, teve. Acho que na maioria das vezes, sim. Então facilitou o processo de repatoração, no geral. Facilitou. Tu diria que o CDD foi bem sucedido no projeto? Eu diria que no final, sim. No início ele ficou meio sem sentido, mas no final, sim. Ok. É... E aí, o CDD, ele sempre fez sentido ou ele foi confuso? Em algum momento durante o projeto? Ele ficou confuso no início e no meio. O que é que o tornava ele confuso? Era conceito? Era aplicação? Era mais aplicação mesmo. Porque a gente nunca conseguia estourar o limite e não refletia na hora que a gente ia repatorar. Tu acha que o CDD tornou o projeto mais complicado? Acho que não. Teve alguma dificuldade em utilizar, em lidar com o CDD em qualquer fase do projeto? Em algum momento? A dificuldade, às vezes, era identificar e dar os pontos com as coisas. Saber exatamente onde que a gente ia colocar no widget, assim, etc. Tu teve problema para entender algum item da tabela? Tive. Foi o aqua planeta. O complemento e o widget assim, então. Tem algum... Tem alguma coisa que tu pensou que... Isso aqui podia ser mais fácil durante o processo? De aplicação do CDD? Acho que não. Acho que é tranquilo sempre. Aham. Ok. Ahn... Teve algum momento inadequado do CDD? Inadequado? Acho que também não. Ok. Ahn... Tá. Tu acha que a equipe, ela aceitou bem o CDD? Ela trabalhou bem com o CDD? Na tua visão, teve alguma resistência, ou pessoal, ou de algum membro da equipe? Ou tu acha que a equipe sempre aceitou, sempre foi de boa? Acho que a equipe sempre aceitou. Ok. Tu usaria CDD? Futuramente? Eu acho que antes de usar um CDD, eu acho que eu tentaria ver uma forma melhor de organizar os arquivos. Mas eu usaria CDD se a equipe que eu estivesse trabalhando fosse fácil de usar. Acha que é mais um trabalho em equipe, o CDD? É. Por quê? Porque as outras pessoas vão ler o teu código, tu vai ler o código de outras pessoas, né? Acho que não faz muito sentido tu aplicar isso de forma individual. Se alguém pedisse uma dica, se alguém quisesse utilizar CDD e pedisse uma dica pra ti, o que é que tu diria? Uma dica? Ahn... Eu acho que era tentar sempre refatorar coisas que façam sentido na medida do possível. E verificar, estudar bem. Decidir bem o limite de CDD. Falou que tu aplicaria, mas mudaria algumas coisas. Quais melhorias tu faria na abordagem do CDD pro próximo projeto? Qual ajuste, mudança específica tu implementaria pra poder aplicar CDD no futuro? Ahn... Eu acho que eu só discutiria mesmo essa questão do IF, do ELSE e do FOR. Ahn... Fora isso, acho que só isso. Só isso mesmo. Beleza. Tem alguma coisa, algum pensamento que tu teve sobre CDD? Alguma indicação? Alguma... Algum caminho que tu acha importante falar aqui, que eu não perguntei sobre, mas que tu pensou? Não. Não. Não. Beleza. Então... Então eu agradeço pela entrevista. Acho que todo mundo que participou desse processo tá oferecendo bastante insight pra gente. Pra gente construir um caminho pra melhorar a abordagem pra aplicar no futuro. E eu vou sempre estar disponível lá no WhatsApp pra, caso tu lembrar de alguma coisa, ter algum código que tu queira compartilhar. E algo interessante que tu acha que seja importante pra gente, pode mandar lá no WhatsApp. E, enfim, agradecer por isso, por essa entrevista, que ela vai ajudar muito tanto no meu trabalho, quanto de colegas no futuro, pesquisadores no futuro. Beleza? Beleza. Tu sabe, eu lembro que falaram lá no grupo que ainda teria a possibilidade de apresentar o projeto esse ano. Tu sabe exatamente... Tu sabe se tem alguma previsão?*