alestrante 1

Beleza, então agora já está gravando? É aí. A primeira sessão da entrevista é sobre algumas questões mais demográficas. Então, a primeira pergunta é: qual que é a sua posição oficial atualmente na empresa?

Palestrante 2

Um Full Stack Developer.

Palestrante 1

Full Stack Developer, beleza. Então, basicamente, você está desenvolvendo o código. Você não está na parte de design de arquitetura, é desenvolvimento.

Palestrante

Projetos.

Palestrante 1

Atividades, então, do seu dia a dia, basicamente. Correto.

Palestrante 2

Tem. No nosso contexto, temos bastante parte de planejamento. Então a parte do design do sistema também vem dos devs. Obviamente existe um quórum de aprovação para cada situação, mas a maior parte também é planejar o que vai ser codado, planejar em termos técnicos e não de produto.

Palestrante 1

Ah, entendi. Legal, perfeito. Quantos anos de experiência você tem nessa posição?

Palestrante 2

6.

Palestrante 1

6 anos?

Palestrante 3

É.

Palestrante 1

Entendi, perfeito. E você tem algum papel específico no time, de ser líder ou PO ou alguma coisa assim, ou é desenvolvedor mesmo, nada específico?

Palestrante 2

Legal, legal.

Palestrante 1

Então, beleza. Passando dessa parte mais demográfica, de informação mais geral, a gente vai para avaliação do bot em específico. Ela é dividida basicamente em 2 seções. A primeira parte é para analisar a utilidade do bot: sua percepção de quão útil o bot pode ser e em quais contextos. A segunda parte é sobre quão fácil você percebe que seria a utilização dele — facilidade de aprender e de operar o bot.

Então, uma pergunta mais geral: como foi a sua experiência usando o bot? De maneira geral, você teve alguma dificuldade? Ficou com alguma dúvida?

Palestrante 2

Cara, não. A documentação que você me forneceu era bem completa. E a utilização dele muito direta. Depois de configurado, não há muito setup a ser feito. Então, do ponto de vista de um desenvolvedor que pega um repositório que já está pronto, ele não vai ter que gastar tempo tentando entender, fazendo setups, etc. É muito simples de utilizar e ir direto ao ponto que ele visa resolver.

Palestrante 1

Legal, legal. A principal função que o bot tem, no caso, é prover essas labels para issues. Você acha que com essa funcionalidade o bot te ajudaria de alguma forma a identificar e monitorar itens de dívida técnica? Te auxiliaria de alguma forma no gerenciamento das issues?

Palestrante 2

Eu penso que sim. Acho que ele pode ajudar nesse contexto. Ajuda, na verdade. Talvez, do ponto de vista prático, se eu fosse colocar onde eu trabalho hoje, a única coisa que eu teria que entender seria como ele classifica uma dívida técnica ou não. Porque uma coisa pode ser considerada dívida técnica e outra não. Acho que talvez seria necessário gastar tempo entendendo como isso funciona. Não sei se isso acaba sendo parametrizado também, do ponto de vista de colocar o limiar um pouco mais para cá ou para lá. Mas acredito que sim, é aplicável e utilizável no meu contexto. Aliás, não seria porque nós não utilizamos issues no GitHub, mas, no geral, se utilizássemos, seria utilizável sim.

Palestrante 1

Legal. No contexto do teu time, vocês têm algum entendimento sobre a questão de dívida técnica que você usaria para identificar na issue ou não?

Palestrante 2

Então, nós utilizamos Jira. Normalmente tudo acaba ficando lá, sendo categorizado lá. A gente traz aqui para o GitHub só os pull requests. Mas o time tem sim um entendimento do que é dívida técnica e normalmente isso acaba sendo identificado de forma manual pelos próprios devs. Vez ou outra acontece de pegarmos um problema de performance e conseguimos identificar que é uma dívida técnica, mas normalmente isso acaba sendo feito manualmente.

Palestrante 1

Entendi, entendi. Legal. E você acha que se tivesse usando uma ferramenta como esse bot, conseguiria monitorar e identificar mais itens nas issues, gerenciar mais issues e identificar mais itens de dívida técnica?

Palestrante 2

Eu penso que sim. Acho que isso poderia acontecer. Porque acaba ficando fácil de categorizar o que é e o que não é dívida técnica e, talvez, priorizar uma coisa ou outra. Então acredito que sim, seria útil.

Palestrante 1

Você acha que essa melhora na produtividade seria em qualquer contexto ou tem algum em que o bot ajudaria mais ou menos?

Palestrante 2

No meu contexto, neste momento, ele não seria muito útil porque nós não utilizamos as issues do GitHub. Mas em um projeto que utiliza, eu vejo que poderia sim ser útil.

Palestrante 1

Uhum.

Palestrante 2

Seria útil principalmente onde toda a discussão acontece no GitHub. E aí os pull requests são criados a partir das issues. Eu acredito que isso ajudaria bastante.

Palestrante 1

Entendi. E a última pergunta: você acha que o bot ajudaria a identificar os itens corretos da dívida técnica? Ajudaria a identificar o que é dívida técnica ou não? Ou acha que isso seria mais responsabilidade do desenvolvedor?

Palestrante 2

Eu acho que seria um bom ponto de partida. Só que primeiro precisaria, entre aspas, ganhar a confiança do time para entendermos o que o bot considera como dívida técnica e se bate com o que nós entendemos como dívida técnica. Ali eu criei 3 issues, todas intencionalmente para serem dívidas técnicas, mas uma delas não foi categorizada como tal, que foi uma do ESLint.

Palestrante 1

Entendi.

Palestrante 2

Aí, talvez num contexto real, isso poderia trazer confusão. Tipo: por que não classificou? Talvez pudesse dar um contexto, explicar por que foi categorizado ou não como dívida técnica. Também ajudaria a priorizar. Porque uma vez que entende o que é dívida técnica, talvez consiga identificar criticidade: se é crítico, se é minor, se é irrelevante. Então acredito que, além da classificação sim/não, seria possível classificar também o nível de criticidade.

Palestrante

Sim.

Palestrante 2

Isso ajudaria ainda mais. Poderíamos falar: temos dívidas técnicas, mas essas aqui são mais prioritárias.

Palestrante 1

Perfeito. Ótimo feedback, ótima sugestão. Da parte de utilidade era isso que a gente ia tocar. Agora passamos para a parte de facilidade de uso. A primeira questão seria: quão fácil você acha que foi aprender como o bot opera, como funciona?

Palestrante 2

Numa escala de 1 a 5, foi muito tranquilo. Utilizando os comandos, foi bem fácil mesmo.

Palestrante 1

Entendi. Você acha também que as descrições das opções que estão no arquivo de configuração estão claras? Teve alguma dificuldade de entender alguma opção ou foi tudo tranquilo?

Palestrante 2

Não. Eu não cheguei a alterar nenhuma configuração. Uma que achei interessante foi a do payout, acho. Achei uma configuração bastante interessante, mas no geral tudo foi tranquilo, meio que autoexplicativo.

Palestrante 1

Legal, legal. Então nessa linha, seria tranquilo configurar esse bot em outros repositórios.

Palestrante 2

Sim, sim.

Palestrante 1

Entendi. E obviamente nem sempre temos a documentação aberta para consultar. Você acha que, da forma como leu, seria fácil lembrar como configurar o bot em outros momentos, mesmo sem acesso à documentação?

Palestrante 2

Lembrar, eu diria que não. Acho que poderia ter dificuldade, não porque é difícil, mas porque às vezes ficamos com muita coisa na cabeça. Talvez teria dificuldade em lembrar. Mas toda vez que uma issue é criada, o bot já manda aquele primeiro comentário dizendo que está ativo no repositório, e ali tem o help. O help ajuda bastante. Então, sei lá, uma vez por semana eu teria que usar o help porque esqueci como autocategorizar ou autorrotular. Depois que uso o help, consigo trabalhar normalmente. Vez ou outra, se eu esquecer, o help ajudaria porque é simples também.

Palestrante 1

Perfeito, perfeito. E por fim: você acha que a estrutura da documentação, como está organizada, é fácil para consultar em caso de dúvida ou tem algo que não ficou bem organizado?

Palestrante 2

Eu achei que está clara. Talvez senti falta de um link levando para um exemplo. Se tivesse um screenshot direto na documentação poderia ajudar, porque temos que abrir outro link. Mas achei bem organizada e tranquila.

Palestrante 1

Perfeito, legal. E por fim: você sente falta de alguma outra feature que o bot poderia ter? Além de atribuir labels, teria alguma outra funcionalidade que poderia ser agregada?

Palestrante 2

Talvez a classificação de criticidade. É um pouco mais complexo, mas seria interessante. Outra coisa: não sei se é possível, mas talvez categorizar com base na discussão inteira. Porque às vezes uma issue começa com um título e descrição, mas ao longo do tempo a discussão muda o entendimento. Não sei se o bot lê só a primeira descrição ou a thread inteira. Se ler a thread inteira, pode mudar a categorização inicial. Isso seria interessante.

Também, talvez, nos comandos de classificar como dívida técnica ou não, poderia haver um campo para dizer a qual parte do sistema aquilo se refere. Tipo: página de produtos, módulo de pedidos, etc.

Palestrante 1

Sim, sim.

Palestrante 2

Então, pedir para o bot: classifica como dívida técnica ou não, e já rotula que é referente a tal módulo.

Palestrante 1

Legal, seria como colocar mais um nível de classificação: não só se é dívida técnica ou não, mas também o módulo ou feature relacionada.

Palestrante 2

Isso. Não precisa ser automático. Eu posso informar manualmente. Se conseguir classificar automático, melhor. Mas sei que pode ser difícil para o bot.