

Prompt 1:

Quero criar um podcast, sugira um nome de podcast sobre programação que tenham a ver com star wars

Prompt 2:

Gostei da referência "Que a força esteja com você", sugira nomes para o meu podcast com essa referência

Prompt 3:

Decidi "Que os commits estejam com você". Agora sugira um subtítulo para o meu podcast. Depois crie uma introdução para eu falar toda vez que começar o podcast

Prompt 4:

Crie o roteiro do 1º podcast, que será justamente sobre o git e o github. Esse podcast vai explicar o que é cada um dos dois, pra que servem, a importância do desenvolvedor aprender os dois e como usa-los (principais comandos, etc)

Prompt 5:

agora prepare as falas para os temas abaixo, com piadas pontuais: "1) O que é o Git? (2–3 min) Pontos a abordar: Criado por Linus Torvalds em 2005 Sistema de controle de versão distribuído Permite acompanhar histórico do código Trabalhar em diferentes funcionalidades sem quebrar o projeto Possibilidade de voltar no tempo (como "bootar um backup da Força") Exemplo simples: "Imagina que seu código é um sabre em construção. Cada ajuste pode melhorar... ou explodir tudo. O Git guarda cada etapa da evolução." 2) O que é o GitHub? (2–3 min) Pontos: Plataforma / serviço online Hospeda repositórios Git + recursos sociais Colaboração, pull requests, issues, CI/CD Empresas e open source dependem dele Frase síntese: "O Git é a ferramenta. O GitHub é o lugar onde a galáxia se encontra." (Se quiser: mencionar alternativas como GitLab, Bitbucket — breve) 3) Por que aprender Git e GitHub? (3–4 min) Requisito básico em vagas de desenvolvimento Trabalho em equipe → evitar conflitos e perdas de código Contribuir em projetos open source Portfólio público e rastreável Segurança: nada some se o computador pifar Ganchos: "Sabe aquele famoso ctrl+z? O Git é um ctrl+z multiplicado por mil com histórico organizado." "Seu GitHub é seu currículo vivo. Ele mostra o que você sabe fazer." 4) Como usar na prática — o básico dos básicos (8–10 min) Você pode ir ritmando exemplos como se estivesse no terminal: a) Criar um repositório git init — "Começamos um repositório Git na pasta atual." OU clonar um projeto existente git clone <url> b) Ver status e histórico git status git log c) Ciclo de versionamento Explicar área de staging brevemente. git add . git commit -m "mensagem clara e objetiva" d) Criar e trocar de branch git branch minha-nova-feature git checkout minha-nova-feature (ou o novo comando unificado) git switch -c minha-nova-feature Explicação temática: Branches são como trilhas alternativas no hiper-espço: você testa coisas sem explodir a nave principal. e) Conectar ao GitHub git remote add origin <url-do-repo> f) Enviar alterações para o GitHub git push origin main g) Baixar atualizações do servidor git pull h) Fundir mudanças git merge minha-nova-feature 👉 Comentar brevemente sobre conflitos de merge e como faz parte da vida dev: "É como um duelo de sabres no código. Mas no final, dá pra resolver com diálogo... e um editor de texto." 5) Boas práticas rápidas (1–2 min) Mensagens de commit curtas e claras Commits pequenos e frequentes Nomear branches de forma legível Atualizar o repositório sempre antes de começar a programar 6) Conclusão (1–2 min) Resumo leve: "O Git te traz segurança. O

GitHub te conecta à galáxia. Juntos, eles transformam devs aprendizes em Jedi da colaboração.” Convite: “Se você ainda não tem perfil no GitHub, esse é o seu desafio da semana! Cria um repositório, faz seus primeiros commits e marca o dia em que você começou sua evolução.” 7) Chamado para o próximo episódio (30s) Dizer que no próximo episódio pode falar de: “Branches avançados e Pull Requests” ou “Como montar portfólio no GitHub” ou “Front-end x Back-end — por onde começar?” 🎤 Fechamento Usa sua tagline: “E lembre-se: a Força guia os jedi... mas quem guia os devs são os commits. Até a próxima, e que os commits estejam com você!”