Vantagens de usar o GITHub?

Salve jovem padawan, hoje estive monitorando o Fórum de nossa comunidade e constatei algo deveras curioso, fiquei intrigado e ao mesmo tempo preocupado. Diversos DEVs postaram pedidos de ajuda, problemas em seu código nos projetos dos LABs da Digital Innovation One, mas algo que era para ser simples e trivial, expos um problema estrutural, algo que me motivou a escrever este artigo.

Recapitulando o tema, um programador deve se preocupar com a qualidade do código, efetuar testes e trabalhar com um código limpo. Correto? Sim! Mas muitas vezes devido a pressão do momento, prazo esgotando e outras mazelas de nossa área. O código desenvolvido não é dos melhores, por isso precisamos de um segundo par de olhos para auxiliar na resolução do problema. Com isso entramos no tema do artigo, o versionamento do código através de criação de um repositório em softwares de controle de versão, tais como o GITHub.

O que é Github?

O GitHub é uma plataforma open-source fantástica, que serve como repositório de projetos, servidor web para html, gestor de versão e versionamento de software, que em parceria ao GIT e ao GIT Desktop, criou uma super ferramenta para auxiliar aos desenvolvedores.

O que é controle de versões?

Para você padawan, novo em nossa comunidade, controle de versões é uma maneira de garantir que as nossas alterações não criem erros catastróficos, pois permite voltar a estágios ou versões de programa em pontos legais, sem erros e funcionais, garantido o trabalho sem perda de produtividade, um paraquedas/Capacete/sinto-de-segurança para garantir a segurança do DEV:

Biblioteca de Projetos

O GitHub e seus repositórios tornam a vida mais fácil para a comunidade DEV, permitindo a troca de código fonte, armazenamento de documentos tais como apostilhas, sebentas, manuais e docs em geral.

Saiba explorar e compartilhe com seus amigos, quanto mais gente participar mais rica será a comunidade e mais pessoas aprendem e democratizam o acesso ao conhecimento, ensinando, aprendendo e interagindo com outros DEVs.

Cooperação de Equipes

Chegamos no ponto que me motivou a escrever este arquivos, muita gente pede ajuda para solucionar problemas em codificação, colocam prints, mas não consigamos ajudar, pois não existe repositórios para interagirmos e auxiliar solucionando o problema, muitas vezes a descrição do problema dificulta o entendimento, vemos o Stake Overflow onde o DEV disponibiliza o código, o tipo de erro e fica mais fácil corrigir e auxiliar.

Caça-bugs

Padawan meu conselho, crie repositórios no Github com todos os seus projetos, em caso de erro compartilhe o link do repositório no Fórum de nossa comunidade e uma equipe fabulosa de BugBusters para ajudar a solucionar problemas no desenvolvimento de código.

Para ajudar quando precisar de auxilio, compartilhe seu código, compartilhe seu repositório e informe todos os problemas que está enfrentando, desta maneira poderemos ajudar com mais precisão e rapidez.

Troca de Ideias.

Além de facilitar a correção de bugs, erros e anomalias, outra ferramenta é o fórum interno que permite troca de mensagens entre equipes de desenvolvedores em projetos, permite a visitantes a deixarem mensagens de apoio e incentivo para o grupo, vale a pena explorar estas ferramentas com tantas funcionalidades.

Melhoria continua

A troca de ideias, ajuda muito, somos criaturas sociais e vivemos em comunidade e conversando, nosso código irá melhorar, pois muitas vezes ficamos presos a pontos de vista, infelizmente não tão eficientes ou eficazes. Um par de olhos frescos ajuda a melhor o código, muitas empresas usam como boa pratica, o programação-dual onde devs trabalham em equipe e corrigem-se e ajudam a melhorar a performance do programa.

Juntos somos mais fortes, este é meu lema, eu olho seu código e você olha o meu, desta maneira um ajuda o outro, pedir auxílio e ao mesmo tempo ajudar o outro, uma boa troca de conhecimento.

Técnica de aprendizado

Outra coisa importante de termos repositório no Github e visitar outros DEVs e aprimorarmos nosso aprendizado, lembre-se sempre, no princípio engatinhamos, depois andamos capenguinhas, tropicando e caindo, aí aprendemos uns passinhos para no final podermos correr.

Em analogia no mundo do desenvolvimento de softwares, um dev iniciante ao conhecermos uma nova tecnologia, copiamos códigos de outros programadores, vemos exemplos em livros, revistas e blogs, com isso vamos aprendendo a usar a linguagem e seus pormenores no dia a dia.

Servidor Web

A parte que mais curto no GitHub e o servidor web, onde podemos compartilhar nossos projetos como páginas Web, usando um domínio semi-personalizavel, principalmente para testarmos código HTML, CSS e Javascript e ao criamos nossos labs na DIO, podemos divulgar para nossa comunidade, podendo incluir referências a bibliotecas externas e referenciar outros sites através de links, hiperlinks e iframes.

Convido o jovem padawan a compartilhar seus projetos, publicar arquivos e páginas web no Github Web Server, qualquer dúvida chama aqui e trocamos umas figurinhas, é superdivertido treinar e ver o resultado final publicado na web.

Github Fork

Uma grande vantagem do GitHub é podermos compartilhar e copiar repositórios de outras pessoas, através do comando FORK, é com isso criamos um repositório espelho, onde podemos manipular, incluindo funcionalidades ou removendas, criando um projeto totalmente novo e ao final, podemos fazer uma solicitação de merge de branches, criando uma solicitação de aprovação do autor owner do repositório.

Github Stars

Uma das atividades lúdicas e extremamente gratificante é atribuir estrelas a projetos de outras pessoas, o proprietário do Repositório, fica muito feliz em receber visitantes na página do projeto e quando alguma alma caridosa atribui uma estrela, melhor impossível, é muito bacana receber estrelas.

Não custa nada e faz uma pessoa feliz, da minha parte sempre que encontro algo interessante, retribuo o trabalho com estrela no projeto.

Tags no repositório

Ajuda bastante atribuir tags de identificação ao repositório, permitindo pesquisar e auxiliando o visitante a entender sobre o que é o projeto, para que serve e quais as tecnologias envolvidas é uma boa pratica também incluir arquivos README.MD em markdown, explicando detalhes do projeto.

Pedidos de Socorro

Mais uma vez voltando ao porquê do artigo, quando necessitar de ajuda, auxilio, help, olhos amigo indique o repositório e o programa com problemas, desta forma a ajuda chega mais rápido, não tenha vergonha de pedir, ninguém nasce sabendo tudo e com a evolução das linguagens de programação, a cada dia surge melhorias e lembre-se um software sempre chegara ao EOL, por melhor que seja o programa, um dia transformara-se em Sistema Legado.

Use das redes sociais, linkedin, twitter, discord e nossa comunidade DIO, somos 700k devs, com certeza alguém irá responder seu apelo e sugerir alguma melhoria. Bora la dev!!!!

Teste de Código

Finalizando o artigo, lembro sempre de criar um plano de testes, testar os pontos críticos e experimentar testes automatizados, a qualidade do software é importante, use variáveis com nomes claros, cuidado com o uso desnecessário de rotinas e variáveis, acesso a rotinas externas. Repetindo-me teste, teste e teste, peque pelo excesso, garantindo a entrega e dando credibilidade a sua carreira de dev.

Conclusão

Este artigo serve de complemento para um artigo anterior sobre GITHUB, mas ao mesmo tempo servindo de alerta para novos DEVs em nossa comunidade, que na inocência dos primeiros passos, pede auxilio na solução de problemas sem indicarem o local correto, dificultando o auxílio.

Use e abuse do GitHub, treine bastante as inúmeras funcionalidades existentes no GitHub, lembrando das versões em linha de comando e desktop, treine as funções de merge, voltar versões, clonar repositórios, subir e descer versões.