

CONCLUSÃO

15. RESUMO E REFORÇO DOS PONTOS PRINCIPAIS

15.1 RECAPITULAÇÃO DA IMPORTÂNCIA DO VERSIONAMENTO.

Como vimos, o Versionamento de Código ou Controle de Versão surgiu com o intuito de facilitar a dinâmica do trabalho entre programadores, possibilitando compartilhar e controlar as diversas alterações de um código raiz visando diminuir problemas e facilitar a execução de um programa ou aplicação.

Neste contexto, os benefícios do versionamento destacam-se pela facilidade de identificar as alterações e compartilhá-las aos colaboradores. Além de possibilitar a recuperação de versões anteriores dos códigos. Por isso, entender as nuances dos tipos de Sistemas de Controle de Versão (VCS ou SCV) se fez fundamental. Nisso vimos que os modelos ou tipos de VCS, como os do tipo Centralizados (CVCS), apresentam uma natureza hierárquica, controle rigoroso, e colaboração controlada, enquanto os do tipo Distribuídos (DVCS) apresentam modelo descentralizado, colaboração descentralizada e gestão de ramificações aprimoradas.

Dentre os Sistemas de Controle de Versão, o mais popular e conhecido entre os programadores como vimos, foi o Git, desenvolvido em 2005 por Linus Torvalds. Comparado a outros SCV, se destacou e ficou bastante conhecido a partir de 2010, por evitar o desgaste de armazenamento das diversas cópias de arquivos no sistema, e por impossibilitar a perda de versões que se queira reutilizar no futuro.

15.2 PRINCIPAIS TÉCNICAS E FERRAMENTAS DISCUTIDAS.

Vamos relembrar?

Falamos muito como o Git é uma ferramenta flexível e dinâmica, e para ilustrar isso iremos relembrar os conceitos ou comandos básicos do Git, e não bastando alguns conceitos que são comuns de se ouvir, iremos destacar o que é snapshot, commit, flags, e staging. Onde um snapshot, é como uma captura de algo em um determinado instante como uma foto. O commit, o comando que leva as mudanças para o repositório no Git, podendo ser instantâneos ou marcos ao longo do desenvolvimento de um projeto Git. O flags, um comando lógico de sinalização, para que finalize o processo de um programa. E o staging, a área que intermedia entre o diretório de trabalho e o repositório git.

Os conceitos ou comandos básicos do Git se dividem nos seguintes tópicos, e em cada tópico serão esclarecidos, sendo eles:

- a. Inicialização e configuração:
 - git init: inicia um novo repositório;
 - git config: configura as opções de instalação e/ou de usuário do git.
- b. Enviar arquivos:
 - git add: antes de fazer o commit (projeto oficial);
 - git commit: submeter as mudanças;
 - git fetch: importar commits;
 - git pull: versão automatizada do git fetch;
 - git push: transfere commits;
 - arquivos.gitignore: para os arquivos que o Git deve ignorar na hora de fazer um commit;
 - git tag: cria etiquetas de estado relevantes.
- c. Verificar informações:
 - git log: verificar as revisões passadas;
 - git diff: mostra alterações;

d. Receber arquivos:

- git clone: cria cópia de um repositório já existente.

e. Alterar Branches:

- git branch: gerencia as branches de um repositório;
- git checkout: muda de Branch ou volta para algum estado do projeto;
- git merge: faz mescla entre branches;
- git rebase: move as branches;
- git stash: arquiva alterações não “commitadas”, volta para o estado do último commit, guardando as alterações adicionais.

Além dos comandos, no Git existem as tags, elas são como fotos de um determinado momento de um repositório, que também servem para ajudar a identificar a origem da falha em um código, e funcionam como apontadores fixos para um commit específico, e não avançam com novos commits.

Vimos também que a plataforma web mais utilizada quando se trata de Git, é a GitHub, devido a facilidade que ela permite ao trabalho em equipe, por qualquer integrante, tendo internet conseguir acessar os arquivos de um determinado projeto sem maiores problemas. E por ser uma espécie de rede social, é possível interagir com pessoas e os seus trabalhos. No GitHub, existe também os releases, que são as versões das aplicações disponíveis, e contribuem para a disponibilização do que seria na linguagem mais popular, a versão atualizada da aplicação.

Além do GitHub, existem outras plataformas como o GitLab e Bitbucket, que também facilitam a interação com os repositórios, pela adição de funcionalidades, e permitem identificar problemas, e rever códigos e também realizar integração continua e entrega continua. Por isso, não é de se estranhar que por essas e outras, o Git só tende a evoluir e revolucionar o mercado de desenvolvedores, pela facilidade de lidar com grandes projetos, por ser dinâmico e robusto.

REFERÊNCIAS BIBLIOGRÁFICAS

3 Versionamento de código. Disponível em:
<https://prdm0.github.io/aulas_computacional/versionamento-de-c%C3%B3digo.html>.
Acesso em: 28 jun. 2024.

ATLASSIAN. Git commit. Disponível em: <<https://www.atlassian.com/br/git/tutorials/saving-changes/git-commit>>. Acesso em: 28 jun. 2024a.

ATLASSIAN. O que é um sistema distribuído? Disponível em:
<<https://www.atlassian.com/br/microservices/microservices-architecture/distributed-architecture>>. Acesso em: 28 jun. 2024b.

MARCELA. Versionamento de código: entenda o que é e porque é importante. Disponível em:
<https://awari.com.br/versionamento-de-codigo/?utm_source=blog&utm_campaign=projeto+blog&utm_medium=Versionamento%20de%20c%C3%B3digo:%20entenda%20o%20que%20%C3%A9%20e%20porque%20%C3%A9%20importante>. Acesso em: 28 jun. 2024.

MARKETING ZUP. Versionamento Semântico. Disponível em:
<<https://www.zup.com.br/blog/versionamento-semantic>>. Acesso em: 28 jun. 2024.

ROSSI, T. GIT: História, Evolução e Aplicações. Disponível em:
<<https://www.dio.me/articles/git-historia-evolucao-e-aplicacoes>>. Acesso em: 28 jun. 2024.

SANTANA, R. Git: entenda conceitos básicos sobre o sistema. Disponível em:
<<https://www.lumis.com.br/a-lumis/blog/git-conceitos-basicos.htm>>. Acesso em: 28 jun. 2024.
Disponível em: <<https://mundododev.com.br/2024/01/02/c-tipos-de-sistemas-de-controle-de-versao-centralizado-vs-distribuido/>>. Acesso em: 28 jun. 2024a.

Disponível em: <<http://www2.ic.uff.br/~ilaim/repeticao1.pdf>>. Acesso em: 28 jun. 2024b.
Disponível em: <<https://napoleon.com.br/glossario/o-que-e-git-staging-area/#:~:text=A%20Git%20Staging%20Area%20%C3%A9,serem%20inclu%C3%ADdos%20no%20pr%C3%B3ximo%20commit.>>. Acesso em: 28 jun. 2024c.

Disponível em: <<https://www.telecom.uff.br/pet/petws/downloads/apostilas/GIT.pdf>>. Acesso em: 28 jun. 2024d.

DAVI FERREIRA SANTIAGO, FELIPE MEIRELES LEONEL, JOÃO VITOR SAADE SIMÃO. GIT INIT - Uma introdução ao controle de versão com Git. Universidade Federal de Minas Gerais, Escola de Engenharia, Bloco 3, Sala 1050.: PETEE-MG, 2023.

Criando release no GitHub. Disponível em: <<https://treinaweb.com.br/blog/criando-release-no-github>>. Acesso em: 28 jun. 2024.

CI/CD. Disponível em: <<https://www.redhat.com/pt-br/topics/devops/what-is-ci-cd>>. Acesso em: 28 jun. 2024.