Aula #7: O que é um versionador?

O que é um versionador de Código?

Segundo o livro Pro Git, um versionador de código pode ser entendido como uma ferramenta ou sistema que armazena as mudanças de um ou mais arquivos ao longo do tempo, possibilitando que essas alterações possam ser desfeitas no futuro.

Ou seja, um versionado é uma ferramenta que utilizamos para armazenar ou guardar os estados dos arquivos de um projeto de software, possibilitando assim que possamos controlar as alterações feitas, e, em caso de necessidade, voltar para um estado anterior.

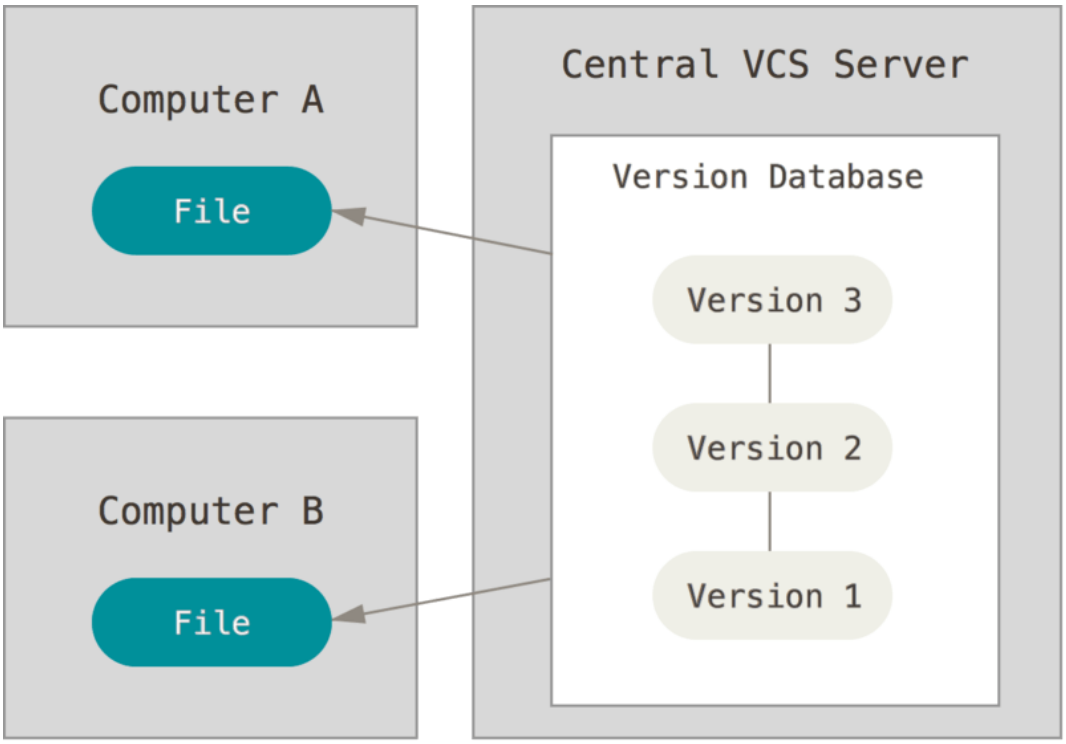
Um versionador de código é uma ferramenta amplamente utilizada pelos profissionais que precisam escrever código, pois possibilita não só armazenarmos os estados da aplicação ao longo tempo, como também melhora a interação e o trabalho entre os membros da equipe.

Imagine que, antes da utilização dos versionadores, os arquivos alterados eram enviados por e-mail, como fazíamos nos nossos trabalhos de escola. O que gerava muitos problemas, uma vez que se um arquivo fosse alterado por dois usuários ao mesmo tempo, não havia uma maneira de fazer o controle de qual das alterações era a correta.

Por tanto, os versionadores de códigos surgiram para facilitar esse mecanismo de trabalho, melhorando o compartilhamento de arquivos entre desenvolvedores.

Como funcionavam os versionadores antigos?

Antes de o Git ser construído, a ferramenta de versionamento mais utilizada era o CVS para fazer o versionamento dos arquivos de um projeto. O principal problema com o CVS era que ele era uma ferramenta de versionamento centralizada.



O que significa que todos os arquivos estão dentro de um único “servidor”, e que, caso um desenvolvedor precisasse utilizar um determinado arquivo para trabalhar,

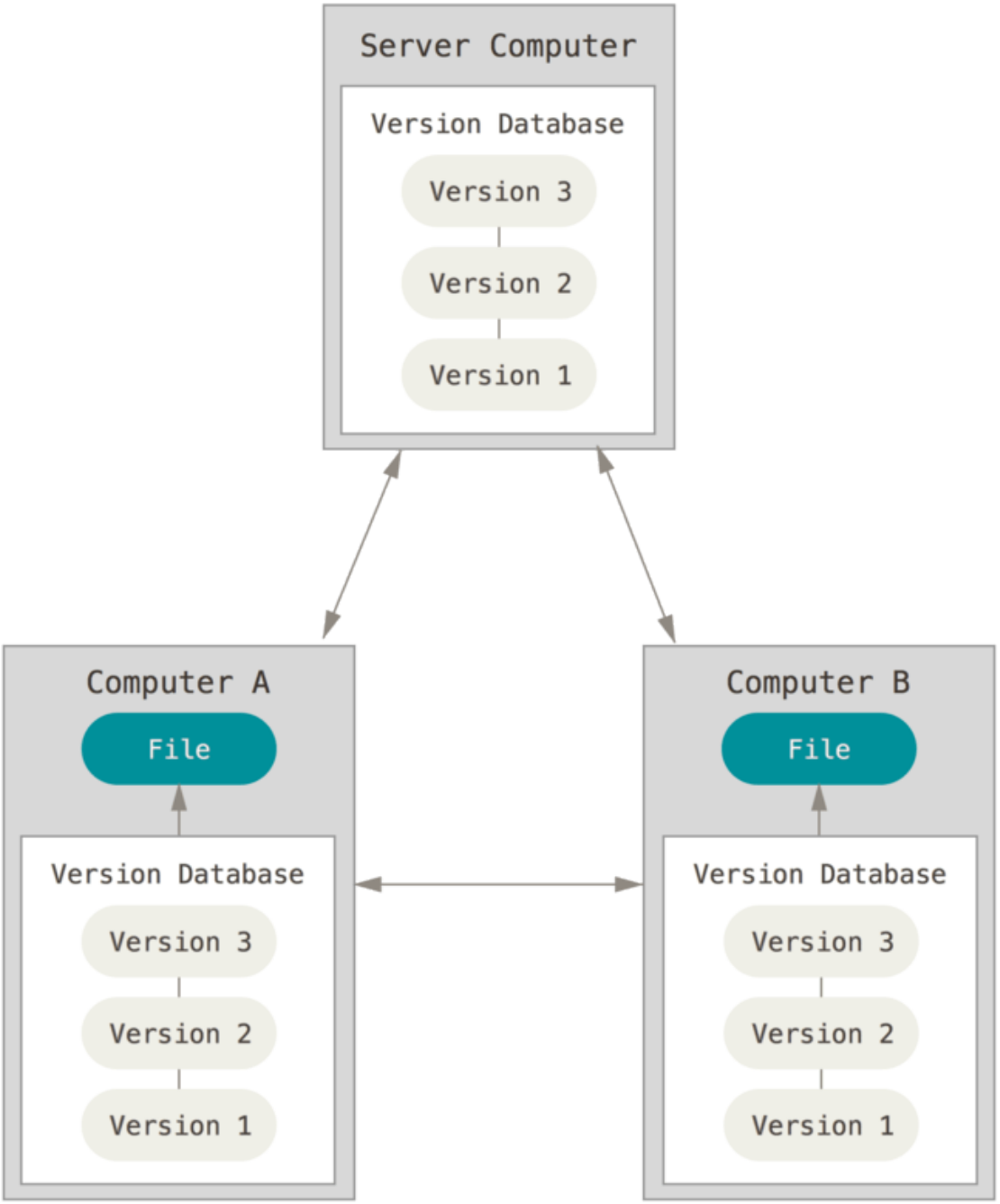
se um segundo desenvolvedor precisasse desse mesmo arquivo, ele teria que esperar o primeiro terminar para então fazer o “lock” desse arquivo. Ou seja, toda vez que um arquivo estivesse em estado de “lock”, ou travado para edição, ele só poderia ser editado por uma outra pessoa caso o estado de “lock” dela fosse liberado.

Esse mecanismo de trabalho apresentava vários problemas principalmente em equipes grandes e internacionais. Imagine que você, que trabalha no Brasil, fez o “lock” de um arquivo para trabalhar nele. Ao final do expediente, você se esqueceu de desfazer o “lock”, e esse arquivo ficou travado para edição. Nisso, seu colega que trabalha na Índia, 8 horas de diferença com o brasil, começou o expediente dele e precisava do arquivo que você esqueceu de desfazer o “lock” para trabalhar. Nesse caso, ele não conseguiria trabalhar até que você fizesse a liberação do arquivo.

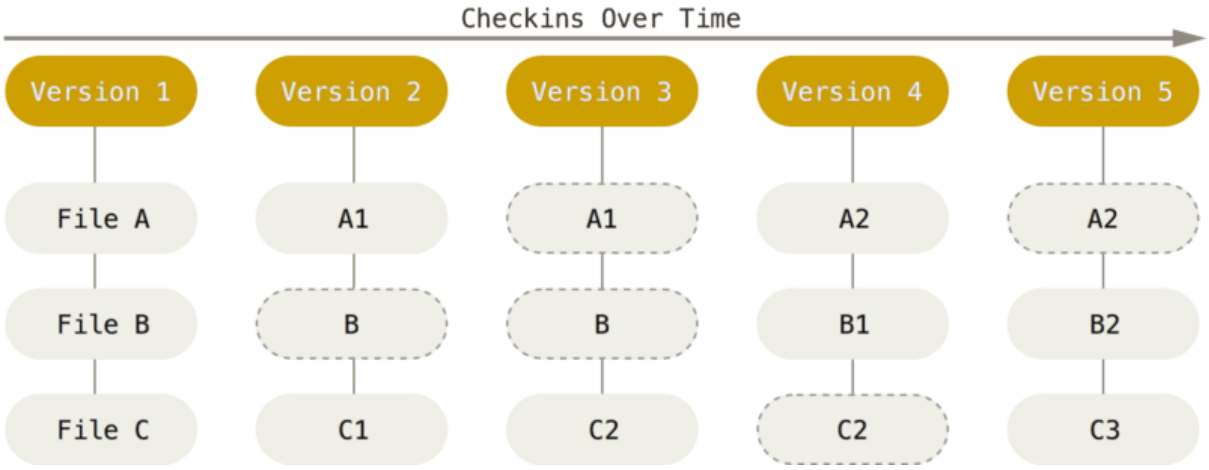
Como funciona o Git?

Nesse tempo, o criador do kernel do Linux, Linus Torvalds, passava por esse e outros problemas com o CVS, que era a ferramenta utilizada para versionar o kernel do Linux, até que um dia ele resolveu criar a própria solução de versionamento: o Git.

O Git é uma ferramenta de versionamento distribuída. O que significa que agora, cada desenvolvedor possui uma cópia das versões dos arquivos em sua máquina local, possibilitando assim que ele possa trabalhar sem interferir no trabalho de um segundo ou terceiro desenvolvedor. Fora isso, os desenvolvedores compartilhavam um servidor mútuo para fazer o upload de seus códigos, facilitando assim o compartilhamento do trabalho feito entre eles.



Além disso, o Git, diferentemente de outras ferramentas de versionamento, salva as alterações somente dos arquivos que foram alterados, sendo que dos arquivos que não foram alterados é criado um link que aponta para esse arquivo sem alteração.



Por exemplo, observe a imagem acima. Nela a Version 1 é o primeiro commit feito dentro do repositório, assim como fizemos na aula passada. Ou seja, é o estado inicial dos arquivos File A, File B e File C. A Version 2 é o estado da primeira alteração, que foi realizada dentro dos arquivos Fila A e File C. Como nessa primeira alteração o arquivo File B não sofreu alterações, é gerado um link simbólico apontando para ele.

Isso permite que o espaço para armazenar o histórico das alterações seja pequeno se comparado a outras ferramentas de versionamento, dando assim velocidade tanto no upload quanto no Download de novas versões.

Outro ponto extremamente importante, se compararmos o Git com outras ferramentas de versionamento, é que o Git é extremamente seguro ao realizar o versionamento dos arquivos. O que significa que quando estamos trabalhando com o Git, não corremos o risco de um commit dar problema e corromper um arquivo, ou ainda de, no processo de download das modificações feitas por um outro desenvolvedor, baixarmos um arquivo corrompido.

Além disso, por ser uma ferramenta de versionamento distribuída, facilita e incentiva o trabalho em equipe, de forma que um desenvolvedor consegue executar o seu trabalho sem atrapalhar o trabalho alheio.

Exercícios

Link para o formulário de exercícios de fixação de conteúdo: Exercícios

Próxima Aula

Na próxima aula, entender melhor o que é um versionador, como o Git funciona e qual o problema que ele resolveu.

Fontes e Links Complementares

Livro Pro Git - Capitulo 1.1 - Começando com Controle de Versões

Livro Pro Git - Capítulo 1.2 - Uma breve história do Git

Livro Pro Git - Capitulo 1.3 - O que é o Git?