

Apostila de introdução à controle de versões e Git

Evaldo Junior Bento

April 10, 2011

Resumo

Pequenas e grandes equipes de desenvolvimento de *software* sofrem com problemas de versionamento de código fonte. Quem fez? O que fez? Quando fez? Quais linhas foram alteradas? Estas são perguntas comuns no dia a dia de equipes que não usam sistemas de controle de versão.

Esta apostila foi desenvolvida para ajudar iniciantes e interessados em sistemas de controle de versões à entender os conceitos de controle de versões e por que fazê-lo. O **Git**, desenvolvido por Linus Torvalds, criador do kernel Linux, é o software de controle de versões usado nesta apostila para exemplificar os conceitos.

Sumário

0.1	Introdução	1
0.2	Controle de versões	1
0.3	Git	1
0.4	Fluxo comum de trabalho em um projeto de software	1
0.4.1	Os problemas começam	2
0.4.2	A solução aparece no horizonte	2

0.1 Introdução

Desenvolver *software* sem utilizar um sistema de controle de versões...

0.2 Controle de versões

O que é?

0.3 Git

Desenvolvido por Linus Torvalds

0.4 Fluxo comum de trabalho em um projeto de software

O fluxo "normal" de trabalho, em um projeto de *software*, pode ser resumido em:

- Pegar o código atual;
- Editar;
- Salvar;

- Devolver para o centralizador.

Até tudo bem. O desenvolvedor abre o código fonte, faz seu trabalho e então devolve o resultado para uma base central. Em projetos individuais ou de equipes bem pequenas, como duplas ou trios, esse método pode até funcionar, mesmo assim esta não é uma boa solução.

0.4.1 Os problemas começam

Imagine alterar um arquivo, colocar em produção e depois de um tempo ver que suas alterações simplesmente desapareceram. Isso pode acontecer quando outro desenvolvedor também fizer alterações no mesmo arquivo e enviar as suas alterações após o primeiro, sem antes verificar se os arquivos do projeto continuavam iguais aos que ele pegou antes de alterar.

Veja o tempo que se perde por ter que verificar os arquivos antes de cada atualização. E o tempo maior ainda por ter que refazer algo que já havia sido feito, fora a frustração e o desânimo que esse tipo de situação geralmente causa.

0.4.2 A solução aparece no horizonte

E se existisse algum tipo de software que ajudasse a verificar os arquivos, saber quem alterou, que linhas alterou e quando alterou?

Sim, isso seria muito legal. Mas espere, isso existe sim! São os Sistemas de Controle de Versão¹.

¹SCM - Source Code Management, em inglês.