

O que é controle de versão?

Os sistemas de controle de versão são softwares que ajudam a acompanhar as alterações feitas no código ao longo do tempo. Conforme o desenvolvedor edita o código, o sistema de controle de versão tira um instantâneo dos arquivos. Em seguida, ele salva esse instantâneo permanentemente para que ele possa ser recuperado mais tarde, se necessário.

Sem controle de versão, os desenvolvedores ficam tentados a manter várias cópias de código em seu computador. Isso é perigoso porque é fácil alterar ou excluir um arquivo na cópia errada do código, potencialmente perdendo trabalho. Os sistemas de controle de versão resolvem esse problema gerenciando todas as versões do código, mas apresentando à equipe uma única versão por vez.

O controle de versão mantém um histórico de alterações à medida que a equipe salva novas versões do código. Esse histórico pode ser revisado para descobrir quem, por que e quando as alterações foram feitas. O histórico dá às equipes a confiança de experimentar, pois é fácil reverter para uma versão boa anterior a qualquer momento. O histórico permite que qualquer pessoa base trabalhe em qualquer versão do código, como para corrigir um bug em uma versão anterior.

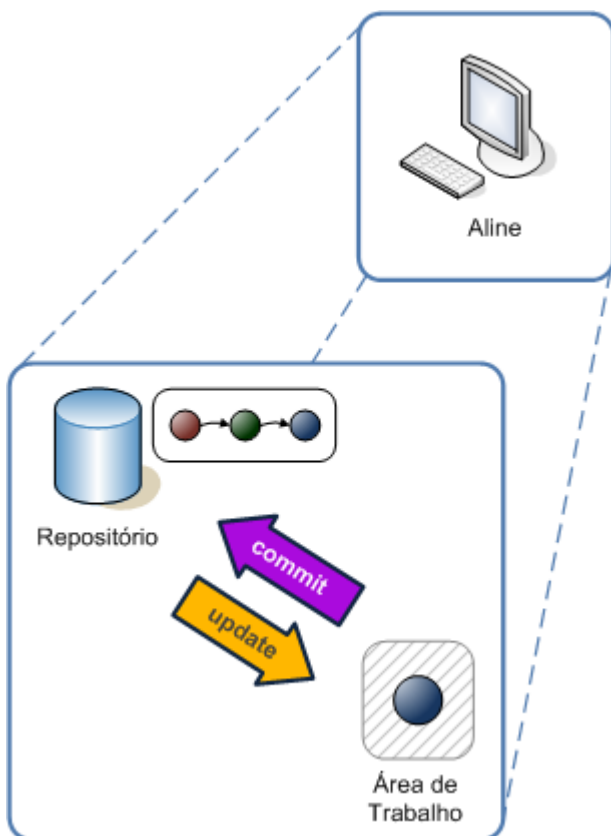
Há dois tipos de controle de versão: **centralizado** ([Subversion](#)) e **distribuído** ([Mercurial](#) e o [Git](#)).

Controle de Versão Centralizado

O controle de versão **centralizado** segue a [topologia em estrela](#), havendo apenas um **único repositório central** mas **várias cópias de trabalho**, uma para cada desenvolvedor. A comunicação entre uma área de trabalho e outra passa obrigatoriamente pelo repositório central.

Controle de Versão Distribuído

São vários repositórios autônomos e independentes, um para cada desenvolvedor. **Cada repositório possui uma área de trabalho acoplada** e as operações **commit** e **update** acontecem localmente entre os dois.



No controle de versão **distribuído** cada desenvolvedor possui um repositório próprio acoplado a uma área de trabalho. A comunicação entre eles continua sendo através de commit e update.

O que é um IDE

Um IDE (ou Ambiente de Desenvolvimento Integrado) é um software que te permite codar com muitos recursos extras além de simplesmente digitar suas linhas de código.

Com IDEs, você pode editar, modificar, debugar, executar e implantar seu código.

Os IDEs vêm com muitos recursos extras, como o preenchimento automático de texto, por exemplo, dando mais flexibilidade para o usuário.

Com a ajuda de IDEs, você pode aumentar a sua produtividade. Existem muitos IDEs na Internet e, felizmente, muitos deles são de código aberto.

IDE's java script

- Eclipse
- Visual StudioCode
- NetBeans
- Atom, um IDE Javascript

IDE Javascript Premium

- Visual Studio

Client-side e Server-side

Client-side

As linguagens client-side são linguagens onde apenas o seu **NAVEGADOR** vai entender. Quem vai processar essa linguagem não é o servidor, mas o seu browser (Chrome, IE, Firefox, etc...).

Significa "lado do cliente", ou seja, aplicações que rodam no computador do usuário sem necessidade de processamento de seu servidor (ou host) para efetuar determinada tarefa.

Server-side

As linguagens server-side são linguagens que o **SERVIDOR** (lado do servidor) entende, ou seja, aplicações que rodam no servidor. Isso quer dizer que você vai escrever um código onde o servidor vai processá-lo e então vai mandar para o seu navegador a resposta.

Por exemplo: se criarmos um script em linguagem back-end (PHP, Asp, Java, Ruby, etc) que apenas calcula a soma de $1 + 1$, será o **SERVIDOR** (ou back) que calculará este resultado.

Se fizermos esse cálculo em alguma linguagem front-end, como o JavaScript, quem calculará essa conta é o **BROWSER** do usuário. Por isso o termo client (ou front).

Os profissionais que trabalham na interface do usuário são chamados de Desenvolvedores front-end, já aqueles que trabalham no core da aplicação, fazendo uma programação que somente o servidor irá entender são chamados de Desenvolvedores back-end.

Exemplos de linguagens

Essas linguagens são linguagens client-side:

HTML
CSS
JavaScript
AJAX

Essas são linguagens server-side:

PHP
ASP
Java
Ruby
Python