

O controle de versão, também conhecido como controle de fonte, é a prática de rastrear e gerenciar as alterações em um código de software.

## 1. Para que Serve o Controle de Versão?

O Controle de versão apoia o desenvolvimento de diversas maneiras:

### Registro do Histórico:

Registra toda a evolução do projeto, cada alteração sobre cada arquivo. Com essas informações sabe-se quem fez o que, quando e onde. Além disso, permite reconstruir uma revisão específica do arquivo sempre que desejado;

### Colaboração Concorrente:

O controle de versão possibilita que vários desenvolvedores trabalhem em paralelo sobre os mesmos arquivos sem que um sobrescreva o código de outro, o que traria reaparecimento de defeitos e perda de funcionalidades;

### Variações no Projeto:

Mantém linhas diferentes de evolução do mesmo projeto. Por exemplo, mantendo uma versão 1.0 enquanto a equipe prepara uma versão 2.0.

## Como Funciona o Controle de Versão?

O controle de versão é composto de duas partes: o repositório e a área de trabalho. O repositório armazena todo o histórico de evolução do projeto, registrando toda e qualquer alteração feita em cada item versionado.

O desenvolvedor não trabalha diretamente nos arquivos do repositório. Ao invés disso, usa uma área de trabalho que contém a cópia dos arquivos do projeto e que é monitorada para identificar as mudanças realizadas. Essa área é individual e isolada das demais áreas de trabalho.

**linguagem de servidor**, ou *Server-side scripting*, é a linguagem que vai rodar "por trás dos panos", fornecendo a lógica principal da aplicação. Funciona da seguinte maneira: sempre que o usuário faz um *HTTP request* (entra numa página, clica num link, etc), o pedido é enviado para o servidor. A linguagem server-side recebe o requerimento (*Request*) e faz o processamento. Depois, transforma o resultado final em um *XHTML* e envia para o *navegador*. É a linguagem server-side que vai verificar se o usuário está logado, vai buscar informações no *banco de dados* etc.<sup>[1]</sup>

Como a linguagem server-side processa as requisições antes de enviar para o navegador, isso significa que uma vez que a página foi enviada ao navegador do usuário, não há mais nada que a linguagem server-side possa fazer até que um novo "Request" seja enviado. Ou seja, não é possível usar estas linguagens para manipular a página do usuário em tempo real. Para o usuário, a linguagem server-side não importa, e ele nem tem como descobrir qual linguagem está sendo usada.<sup>[1]</sup>

## Exemplos de Linguagens Server-Side

- *PHP*
- *ASP*
- *.NET*

- .NET
- Oracle
- Perl
- Java (linguagem de programação)
- Aptana
- Python
  - Django
  - Flask
  - Bottle
- Ruby
  - Ruby on Rails
- Smalltalk
- ColdFusion
- Lotus Domino
- Websphere
- Node.js