

# (Teoria) Aprendendo Sistemas de Controle de Versão (SCV)

Sistemas de controle de versão, também conhecidos como Sistemas de Versionamento ou SCV, são sistemas que existem com o objetivo de controlar as alterações feitas em um software. Softwares evoluem, **e ter um software que controla essa evolução torna o projeto mais organizado**. O uso de SCVs nos ajudam em diversas situações, e uma hora vamos precisar entendê-los para darmos conta de projetos de grande escala.

Basicamente, um sistema de versionamento funciona da seguinte forma: A cada alteração feita em um projeto, enviamos ela até um banco de alterações, onde vão ficar todo o histórico de versões, chamamos ele de **repositório**. Dessa forma, podemos ver toda a evolução do nosso programa.

Sendo assim, podemos por exemplo, resgatar uma versão anterior do repositório para analisá-la e deduzir **o porquê da nossa versão atual não estar funcionando corretamente**.

Nesses sistemas, existem algumas operações úteis que podemos aplicar às versões. Como por exemplo, a operação de comparação (DIFF) que apresenta a diferença entre uma versão e outra. Também existe a operação contrária à essa (PATCH).

## Qual a diferença entre Git, e Github, e o que são eles?

O Git, é um SCV criado em 2005 por Linus Torvalds, o mesmo criador do Kernel Linux. O Git é um software que pode ser instalado localmente em qualquer máquina e funciona por command-line, ou seja, ele é usado no CMD (se for o Windows) ou pelo terminal (Mac ou Linux). E o mais interessante, ele é um software open source, o que significa que qualquer um pode dar uma olhada no código desse software.

Já o GitHub, apesar de ter Git no nome, não é um sistema de versionamento.

Na realidade, ele é um serviço de hospedagem de código na nuvem, que pode ser utilizado por qualquer um, desde que você tenha uma conta cadastrada no site. (<http://github.com/>) foi criado em 2008 e hoje em dia está no domínio da Microsoft, por ter comprado o serviço 10 anos depois de seu lançamento, em 2018.

Mas o “Git” no nome não é em vão. Esse serviço trabalha em conjunto com o Git, permitindo que o usuário armazene as alterações em um repositório **remoto**. O que ajuda na segurança, pois dessa forma teremos uma garantia que não perderemos o repositório caso ocorra problemas com relação ao computador do usuário. Apesar disso, não é necessário ter o Git instalado ou entender de Git para hospedar arquivos no GitHub. Existem outros serviços como o GitHub, Bitbucket e GitLab que são os que mais valem citar.

## Qual é a história dos SCV?

Nos anos 70/80, surgiram os primeiros sistemas. **SCCS** e **RCS**. Sendo que o SCCS surgiu em 1972, e RCS 10 anos depois, em 1982. Nesses sistemas, o repositório era local. Ou seja, permanecia na máquina do usuário. E existiam duas operações que podiam ser feitas.

**Commit** - Envio de informações ao repositório.

**Checkout** - Resgate de informações do repositório.

Já nos anos 80/90, surgiram **CSV (1986)** e **Subversion (2000)**, que funcionavam adotando um sistema cliente-servidor.

Nesse sistema, existe um computador que armazena o repositório (servidor). E os usuários (clientes) a partir das operações de Checkout e Commit, se comunicavam com o repositório no servidor.

E, mais recentemente, no ano de 2005, apareceram **Git** e **Mercurial**. Que adotam o sistema Peer-To-Peer, que é diferente dos sistemas já abordados aqui.

Nesse tipo de SCV, temos um usuário que tem consigo seu repositório local. E ele pode se comunicar com outros repositórios de outros usuários, e vice-versa.

Apesar de ser peer-to-peer, eles tipicamente são adotados de forma que funcionem individualmente (apenas utilização do repositório local), ou de forma cliente-servidor. Pois o peer-to-peer permite esse tipo de utilização. Ainda existem outros modelos, como por exemplo o de gerência de patentes. Onde as alterações são primeiro enviadas a um repositório de um gerente, que depois de analisar as alterações, finalmente as envia a um repositório central.