**Aluno: Bruno Camargo Manso**

Os Sistemas de Controle de Versão ou Source Code Management têm a finalidade de gerenciar diferentes funções de um documento qualquer. Amplamente utilizado em desenvolvimento de softwares e com a capacidade de gerar um histórico de alterações, gerenciamento de usuários e respectivamente de suas devidas permissões, também pode ser armazenado em um servidor garantindo assim uma descentralização de um processo de criação qualquer, sendo assim, um colaborador remoto poderá ter acesso às versões permitidas com seus repositórios envolvidos. Bastante presente em instituições que, de alguma forma, produzem software ou que apenas utilizem um sistema atualizável, modificável na (medida do possível) e colaborativo, relacional.

As vantagens não se restringem ao controle de histórico das versões ou o trabalho em equipe, mas também se estendem: a possibilidade de efetuar marcações e resgates de versões estáveis; às Ramificações de projetos ou os chamados *forks*, permitindo a divisão do projeto em diversas linhas de pensamento podendo até mesmo ser modificadas paralelamente entre colaboradores; em relação a segurança pois cada software de controle usa mecanismos que evitam agentes deletérios externos ao projeto; à possibilidade de rastreamento das versões, permitindo assim que qualquer um com permissão, depurar informações sobre o quem, onde e como foi feita uma modificação; também a presença de interfaces visuais, organização cartesiana, que permitem a visualização global de um projeto em desenvolvimento ou não; sobretudo em relação a confiança depositada no uso dos repositórios remotos que evitam a perda de dados, ou até mesmo no desenvolvimento de outro projeto com base em um precedente, porém sem danificar seu original.

Um dos sistemas mais famosos é o Git, de Linus Torvalds, pai da plataforma Linux, se diz sempre ter sido um lobo solitário em relação a seu sistema criado. elaborando todo O.S em prol de seu conforto e de suas necessidades, nunca teve preocupações em estender seu sistema a outrem, no entanto, devido a genialidade da plataforma, e consequentemente o imenso interesse por parte de entusiastas, usuários finais, desenvolvedores etc, foi necessário um escalonamento para garantir que todos seus processos fossem devidamente distribuídos, permitindo assim a criação de repositórios de terceiros, colaboração com outros desenvolvedores de O.S baseados em linux entre outras necessidades intrínsecas a um software com magnitude global, e mais: Open Source, ou seja livre, de graça e perfeitamente hackeável. Só tinha ocorrido, até então, por parte de Linus, apenas um esforço próprio no desenvolvimento de suas próprias versões, despretensiosamente esboçou o Git. Após esse período de ostracismo, quando houve de fato a necessidade de colaboradores para o escalonamento e descentralização da plataforma Linux, situação que era segundo ele desagradável, por ter que se relacionar, alinhar ideias com pessoas que agora estavam tão envolvidas quanto ele em sua plataforma, Linus precisou também escalonar a forma que o Git funcionava, tornando assim o sistema que é hoje, incluindo o imenso repositório colaborativo e OpenSource chamado GitHub.

Logicamente existem diversos outros programas tão importantes quanto o Git, também em versões Livres como o CVS, Mercurial, SVN, no entanto o Git vem substituindo algumas delas como o SVN (que uma vez foi sucessora do CVS). Também existem as versões comerciais que garantem um amplo suporte técnico, garantia, que evitaria perda de dados de alguma forma, voltadas para grandes instituições. São eles: Clear Case da IBM, SourceSafe, TFS, PVCS (Serena), Team Foundation Server (da Microsoft). Mesmo assim as versões livres podem ter segurança e desempenho mais satisfatórios!

Órgãos como SPICE e CMMI são diretamente responsáveis por desenvolver certificações para a devida existência de um software de controle de versão, garantindo assim, a normalização e fidedignidade ao objetivo proposto em cada um desses softwares.