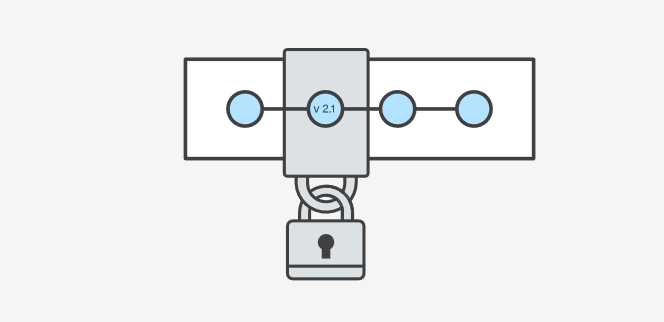
**1 – O que é Git?**

****Git é um sistema de controle de versão, então primeiro precisamos entender que tipo de sistema é esse.

O controle de versão é um sistema que registra as mudanças feitas em um arquivo ou um conjunto de arquivos ao longo do tempo de forma que você possa recuperar versões específicas.

Se não houver um sistema de versão, imagine o caos entre duas pessoas abrindo o mesmo arquivo ao mesmo tempo. Uma das aplicações do git é justamente essa, permitir que um arquivo possa ser editado ao mesmo tempo por pessoas diferentes. Por mais complexo que isso seja, ele tenta manter tudo em ordem para evitar problemas para nós desenvolvedores.

Quando não usamos um controle de versão, corremos o risco de não acompanhar a evolução do nosso trabalho, perder histórico, desconhecer o que foi modificado nos nossos arquivos e corremos o risco também de perde nossos arquivos e não ter back-up.

Os principais modelos de controle de versão são:

Centralizado:



Esses sistemas, como por exemplo o CVS, Subversion e Perforce, possuem um único servidor central que contém todos os arquivos versionados e vários clientes que podem resgatar (check out) os arquivos do servidor. Por muitos anos, esse foi o modelo padrão para controle de versão.

Distribuído:



Em um DVCS (tais como Git, Mercurial, Bazaar ou Darcs), os clientes não apenas fazem cópias das últimas versões dos arquivos: eles são cópias completas do repositório. Assim, se um servidor falha, qualquer um dos repositórios dos clientes pode ser copiado de volta para o servidor para restaurá-lo. Cada checkout (resgate) é na prática um backup completo de todos os dados

As diferenças entre os dois, nós ainda vamos discutir.

Vantagens de se usar um controle de versão:

* **Histórico completo de modificações**. Cada mudança que qualquer pessoa tenha feito irá ser salva como histórico. Ter a história completa permite à equipe retornar a alguma versão anterior, caso haja algum erro ou bug na atual versão do projeto.
* **Ramificar e mesclar.** Sistema de controle de versão nos permite criar ramificações do nosso projeto, assim fazendo diversos fluxos diferentes pra cada desenvolvedor. Assim, evitamos conflito de mudanças de arquivos, pois cada ramo terá seu próprio ciclo de vida. O que facilita a visualização das mudanças feitas, permitindo que mais pra frente seja feito a junção das mudanças sem que haja grandes problemas
* **Rastreabilidade.** A possibilidade de rastrear cada modificação no dá várias oportunidades. Desde escrever mensagens em logs para descrever o que foi feito, até integrar essas mudanças com outros sistemas que irão validar nossos envios para checar se existem erros

Na prática, você pode usar um VCS com qualquer tipo de documento. Imagine que você é um web designer, você poderá versionar suas criações através desse sistema. Ele permite reverter arquivos para um estado anterior, reverter um projeto inteiro para um estado anterior, comparar mudanças feitas ao decorrer do tempo, ver quem foi o último a modificar algo que pode estar causando problemas, quem introduziu um bug e quando, e muito mais. Usar um VCS normalmente significa que se você estragou algo ou perdeu arquivos, poderá facilmente reavê-los. Além disso, você pode controlar tudo sem grandes esforços.

**O que é git?**

