Hoje em dia projetos com uma ou mais pessoas, tendem a ter erros no decorrer de sua execução, visando este problema é onde entram os sistemas de controle de versões (Version Control System ou VCS). Como o próprio nome sugere, ele simplesmente registra o arquivo toda vez que ele é modificado, de tal forma que caso uma alteração seja feita errada e salva por engano, ele permite que essa alteração seja desfeita, voltando o arquivo a versão anterior antes da alteração.

Obter um controle de versão local é um método muito utilizado hoje em dia por ser um processo simples de ser realizado, porém, pode ser muito suscetível a falhas. Por outro lado, existem outros tipos de sistemas de controle de versão, como o centralizado e o distribuído. O centralizado é onde são armazenadas somente as versões dos arquivos, um servidor, do qual não se é editado o arquivo no mesmo, apenas nas outras máquinas, da qual oferece muitas vantagens em saber o que os outros desenvolvedores estão fazendo além dos administradores terem acesso de quem está fazendo o que. Já o sistema de versão distribuído, como o GIT por exemplo, não é realizado apenas a cópia do arquivo e sim do repositório inteiro, tanto online, quanto nos computadores dos contribuintes do projeto, assim, se um servidor ou máquina falhar, tem como acessar esses arquivos em outro lugar.

O projeto será terá um controle de versões utilizando o Git, pois ele tem uma grande diferença entre os demais programas, que é a maneira como ele salva essas versões, além dos branchs. Os arquivos são tratados como Snapshots que em resumo, seria como se o Git tirasse uma foto do projeto todo no momento em que você salva o arquivo, ou seja, ele “fotografa” o projeto todo naquele instante e salva.

Como mencionado no último parágrafo, os branchs também é outra vantagem da ferramenta, do qual será muito utilizado no projeto de desenvolvimento do aplicativo. Quando se inicia um projeto, ele inicia com uma linha de desenvolvimento principal chamada de branca master. Os branchs permitem que o desenvolvedor crie ramificações podendo assim desenvolver separadamente do principal sem ter que alterá-los, caso aconteça algum problema, se acontecer, basta apenas voltar o projeto para o branch master, ou seja, desfazer toda alteração que foi feita a partir do branch criado. Caso sejam criados outros branchs dos quais funcionem e sejam importantes para aplicação, a ferramenta permite que os desenvolvedores possam juntar esses branchs, tornando novamente um só com toda alteração feita.

Essa é uma das vantagens do Git por proporcionar a equipe desenvolvedora que, sejam feitas alterações no mesmo projeto sem ter conflito de arquivos que sejam mexidos na mesma linha de código.

A ferramenta já está sendo utilizada pela equipe, e está auxiliando muito na documentação do projeto através dos commits, que são nada mais nada menos que mensagens deixadas pelo desenvolvedor quando se faz uma alteração, da qual ele deve subir o arquivo alterado e através do commit, resumidamente ou não, dizer o que foi alterado, para que assim que outros desenvolvedores saibam o que foi mexido pela última vez no código.