

De forma simples

- **Para que serve o Git:** é um software de controle de versão(vcs). É uma ferramenta de versionamento de software, códigos. De forma bem mais complexa, o git é uma versão, várias versões da mesma coisa, pois ao invés de criarmos vários arquivos compactados com diversos nomes, ou enviar tudo para o drive, usamos o git, dessa forma a cada nova alteração/versão do projeto fica mais fácil e mais útil de se trabalhar.
- **Para que serve o GitHub:** é uma rede social para programadores, um repositório remoto para armazenar seus programas e até mesmo um repositório de armazenamento de sites básicos que tenham html, css e javascript.

CVS: é um software de controle de versão centralizado, ou seja, ele precisa de um servidor, ele é open source e na sua época era o mais popular, porém ele tinha problemas de consistência, velocidade, não deixava renomear um repositório, etc.

SVN: é um software centralizado e open source, e muito parecido com CVS, ele veio para tentar resolver os problemas que o CVS apresentavam, porém eles tinham um outro problema que era de cópias, tudo eles queriam copiar do CVS. Ele é um projeto que está ativo até hoje.

BitKeeper: é um software distribuído, ele não era open source, e em suas novas versões tinha várias limitações na versão gratuita.

Git: software criado pelo mesmo desenvolvedor do linux, ele é distribuído, open source, o objetivo do criador do git era que o software não fosse igual aos anteriores, ele quis que o software fosse focado em performance, então, enquanto o CVS levava aproximadamente 3 minutos para fazer um commit, o SVN levava entre 1.5/2 minutos, o git por ser um repositório local levava 1 a 2 segundos

Aba issues do GitHub: ela serve como um tipo de fórum, nela você pode fazer perguntas para o dono do repositório, e em outros casos se surgir um erro você pode clonar o repositório, modificar algo e mostrar ao dono, caso ele goste ele adiciona seu código como colaborador.

Versionamento de projetos: de uma forma simplista, o versionamento de projetos se refere a cada etapa que seu projeto avança, ou seja, a cada nova atualização é criado uma nova versão dele, isso é ótimo, pois se a nova versão der errado, você pode voltar a versão anterior.

Aprendendo linguagem Markdown

- **Deixar texto em negrito:** podemos colocar dois * ou dois _ ex: ****texto em negrito**** ou **__texto em negrito__**.
- **Deixar texto em itálico:** usamos um * ou um _ ex: **texto em itálico** ou *_texto em itálico_*
- **Deixar texto grifado:** usamos ~~ ex: **~~Texto grifado~~**
- **Criar título:** #Título, criar título 2 ex: ##Título
- **Criar uma linha:** Podemos usar 3 - ou 3 *, ex: **—linha** ou *****linha**

- **Criar uma lista numerada:** usamos o número 1 e um ponto, só funciona com o 1, ex: 1.primeiro, 1.segundo, 1.terceiro
- **Lista demarcada:** usamos o *
- **Para criar lista de tarefas usamos:** []primeira tarefa, []segunda tarefa, após isso para marcar a tarefa como concluída colocamos o x dentro ex: [x] primeira tarefa.
- **Criar tabela:** Num | Nome | Nota
— | — | —
1 | Éric | 9,5

Branches no GitHub: As branches de forma traduzida significa arbustos, ou melhor dizendo na área de T.I elas são árvores de estruturas, no caso do GitHub, elas são para dividir etapas de um projeto, ex: um projeto tem várias etapas de desenvolvimento, e cada etapa ficará para uma pessoa, então temos a branch principal que é a master, ela é o que chamamos de estrutura, conectada a ela está as outras branches que podem ser nomeadas conforme as necessidades, após o final do desenvolvimento da branch, se estiver tudo ok, conectamos ela a master, então elas passara a fazer parte da estrutura principal.