**FACULDADE SENAC GOIÁS**

**ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**2º PERÍODO**

**PESQUISA**

**SOFTWARES DE CONTROLE DE VERSÃO MAIS UTILIZADOS NO MERCADO**

**LUANA FERREIRA DE ALMEIDA E KAREN BOMFIM**

**GOIÂNIA-GOIÁS**

**2019**

**SOFTWARES DE CONTROLE DE VERSÃO MAIS UTILIZADOS NO MERCADO**

**LUANA FERREIRA DE ALMEIDA.**

Trabalho solicitado pela Professora Ana Flavia da disciplina de Controle de Versão do 2º período do curso de Análise e Desenvolvimento de Sistemas da Faculdade Senac-Goiás.

**GOIÂNIA-GOIÁS**

**2019**

**SOFTWARES DE CONTROLE DE VERSÃO MAIS UTILIZADOS NO MERCADO**

**CVS**

A CVS é uma das ferramentas de controle de software mais antigas no mercado. A primeira versão dela foi desenvolvida em 1968. Essa ferramenta possui como maior desvantagem o fato de ser considerada como uma tecnologia antiga. Porém, ainda é bastante utilizada por [equipes de desenvolvedores](https://gaea.com.br/afinal-o-desenvolvedor-de-software-deve-seguir-um-codigo-de-etica/).

É muito simples de ser operada. Isso significa que a sua equipe pode aprender rapidamente como usar todas as funcionalidades da CVS com eficiência.

**SUBVERSION**

No meio corporativo, o Subversion é uma ferramenta de controle de versão de software bastante utilizada. Ela é bastante rápida na execução das funcionalidades do sistema e ainda se mostra como uma das mais simples de ser empregada. Isso significa que com um conhecimento básico de conceitos relacionados ao controle de versão de software é possível executar comandos na ferramenta. A aprendizagem da equipe também é rápida nesse aspecto.

Um dos problemas do Subversion são as críticas relacionadas à eficácia do software. No passado, essa ferramenta apresentou problemas na hora de executar as principais funções de um controle de versão de software eficiente. Porém, as últimas versões lançadas parecem ter solucionado tudo que foi apontado como desvantagem do programa.

**TFS**

O TFS sigla para Team Foundation Server é uma outra ferramenta de controle de versão de software que pode ser utilizada na sua empresa. Ele traz uma série de características interessantes, principalmente se você utiliza metodologias ágil no setor de TI da sua empresa.

Isso porque ele possibilita a gestão de projetos por meio de SCRUM ou CMMI. Também permite a utilização de forma centralizada ou distribuída, sendo adequado tanto para equipes que compartilham o mesmo espaço físico quanto aquelas que trabalham à distância. Outra vantagem apresentada por esse sistema é o fato de não possuir limitações de crescimento e ter integração direta com o Microsoft Office.

**GIT**

O GIT é uma das ferramentas de controle de versão de software mais populares, principalmente em [projetos](https://gaea.com.br/guia-sobre-gerenciamento-agil-de-projetos/)[open source](https://gaea.com.br/softwares-open-source-ate-onde-vale-a-pena/). Isso se deve, principalmente, pela popularidade do GitHub, uma plataforma para hospedagem de códigos. Apesar de ela poder ser utilizada em outras ferramentas, o GIT é a que ganhou mais repercussão nesse aspecto.

As principais vantagens dessa ferramenta são o design interno e interface, a eficácia e o desempenho do software. Isso significa que ele é agradável de ser utilizado, consegue atingir todos os objetivos de um bom controle de software e é rápido. Porém, apresenta desvantagens relacionadas à simplicidade de utilização. Apesar de ser a principal ferramenta de controle de versão de software disponível no mercado, possui controles um pouco mais complexos quanto comparado a outros softwares.

**MERCURIAL**

O Mercurial é a ferramenta de controle de versão de software utilizada por grandes empresas como o Facebook e Google. Ela é bastante eficiente, o que significa que consegue desempenhar bem as funções básicas de um bom controle de software.

O Mercurial é uma ferramenta bastante rápida na execução dos comandos e ainda funciona muito bem para [equipes](https://gaea.com.br/talentos-tecnologicos-saiba-como-recrutar-os-melhores/) grandes, nas quais os desenvolvedores não estão todos trabalhando no mesmo local. Isso porque ela é uma ferramenta de controle de versão distribuída.