CENTRO PAULA SOUZA

FACULDADE DE TECNOLOGIA DE FRANCA

“Dr. THOMAZ NOVELINO”

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMA

MICHELLE GEICE GONÇALVES MELETTI  
RAFAEL AFONSO DA SILVA FARIA

Versionamento de código

Utilizando a ferramenta GIT

**FRANCA/SP**

**LISTA DE SIGLAS  
  
LISTA DE TABELAS**

**ABNT –** Associação Brasileira de Normas Técnicas

**SUMÁRIO**

**1 INTRODUÇÃO --**

**2 O QUE É VERSIONAMENTO DE CÒDIGO --**

2.1 QUAIS AS VANTAGENS DE SUA UTILIZAÇÂO --

**3 GIT --**

3.1 O QUE É GIT --

**4 FLUXO DE TRABALHO E SEUS COMANDOS --**

4.1 INICIANDO REPOSITÓRIO --

4.2 CLONAGEM DE REPOSITÓRIO --

4.3 COMMIT DE CÒDIGO (DO LOCAL PARA O REPOSITÓRIO REMOTO) --  
4.4 ATUALIZAR REPOSITÓRIO LOCAL COM CÓDIGO RECENTE.......................--

**5 TAREFAS --**

4.1 O QUE SÃO TAREFAS --

4.2 COMO FUNCIONAM --

**CONDIREÇÔES FINAIS --**

**REFERÊNCIAS --**

1 INTRODUÇÃO

Com base no gerenciamento de projetos da engenharia de software e a evolução do código-fonte no processo de desenvolvimento de sistemas, surgiu a necessidade de uma solução que gerenciasse e controlasse as versões deste código-fonte bem como sua documentação e compartilhamento de trabalho.

Desta forma, surgiram os softwares responsáveis pelo controle de versões destes códigos.

O sistema de controle de versões de código-fonte (VCS – do inglês version control system) tem por objetivo armazenar todas as alterações de código sofridas durante o projeto desde o estado inicial até o estado atual, guardando histórico de todas alterações intermediárias.

Dentre os mais comuns destes sistemas, encontram-se as soluções livres: CVS, Mercurial, Git e SVC, e as comerciais: SourceSafe, TFS, PVCS (Serena) e ClearCase (IBM).

Neste trabalho, abordaremos a ferramenta Git com repositórios em GitHub.

Criado em 2005, pela equipe de desenvolvedores do Linux (em particular Linus Torvalds, o criador do Linux) com o objetivo de oferecer velocidade, design simples, suporte robusto a desenvolvimento não linear (milhares de branches paralelos), totalmente distribuído e capaz de lidar eficientemente com grandes projetos como o Kernel do Linux (velocidade e volume de dados).

**2 O QUE É VERSIONAMENTO DE CÓDIGO**

Um versionamento de código permite gravar todas as alterações feitas em um arquivo ou conjunto de arquivos feitas ao longo do tempo pela equipe de desenvolvimento de forma que possam ser recuperadas versões específicas.  
Permite reverter arquivos para um estado anterior, reverter um projeto completo para um estado anterior, comparar mudanças realizadas ao decorrer do desenvolvimento, verificar quem foi autor das mudanças e quando ocorreram. Utilizar um sistema de versionamento de códigos significa que se ocorreu algum bug ou houve perda de algum arquivo, poderá facilmente reavê-lo.

**2.1 QUAIS AS VANTAGENS DE SUA UTILIZAÇÂO**

Existem várias vantagens de utilização dos sistemas de versionamento de código de código, dentre elas estão:

- Suporte à colaboração: os sistemas de versionamento de código permitem que vários desenvolvedores trabalhem em paralelo em um mesmo arquivo sem sobrescrever o código do outro facilitando o trabalho em equipe e, o gerenciamento do que foi alterado em cada arquivo por cada usuário, fica por conta desta ferramenta.

- Marcação (criação de tags) e resgate de versões estáveis: permite a marcação de documentos que estão em uma versão estável de forma que possa ser resgatado no futuro. As metodologias ágeis buscam a capacidade de implantar uma nova versão do software ao final de cada iteração, portanto a equipe de desenvolvimento deve estar preparada para que, caso a versão atual apresente algum problema eventual, possa retorná-la a versão estável. Este procedimento é possível através do recurso de marcação das ferramentas de controle de versões.

- Rastreabilidade: além da marcação através das tags, o sistema de versionamento de código também atribui um identificador único a cada versão, armazenando o responsável pela criação da nova versão e a data em que foi criada. O responsável pela criação da nova versão também tem a opção de adicionar um comentário e descrever o que foi alterado em relação ao arquivo anterior.  
- Segurança: Apenas pessoas autorizadas e devidamente identificadas podem acessar o arquivo. Evitando assim, corrupções no código fonte

**3 GIT**

Através do projeto *open source* de desenvolvimento do Kernel (núcleo) do Linux, surgiu a ferramenta Git. Durante a maior parte do período de manutenção deste núcleo, as mudanças no software eram repassadas com patches e arquivos compactados e, em 2002, o projeto do Kernel do Linux começou a utilizar um sistema DVCS proprietário chamado BitKeeper.

A relação entre a comunidade de desenvolvimento do Kernel e a empresa BitKeeper se desfez em 2005 e isso levou Linus Tovarlds (criador do Linux) a desenvolver sua própria ferramenta de controle baseado nas lições que aprendeu utilizando o BitKeeper. Com o objetivo de desenvolver um sistema que obtivesse velocidade, design simples, suporte robusto a desenvolvimento não linear, totalmente distribuído e capaz de lidar eficientemente com grandes projetos como o Kernel do Linux.

Assim foi criado o sistema de versionamento de código Git.

**3.1 O QUE É GIT**

É um sistema de versionamento de código que armazena e pensa sobre informação de uma forma totalmente diferente de outros VCSs. Esta diferença está na forma de como o Git trata os dados.

A maior parte dos outros sistemas armazenam informação como uma lista de mudanças por arquivo ao longo do tempo conforme figura 1.

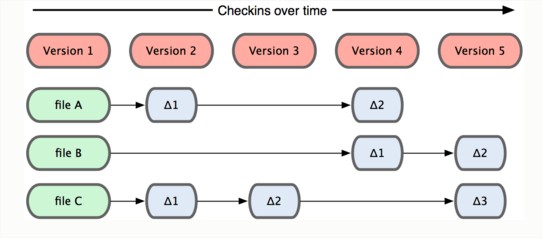


Figura 1.Sistemas que armazenam mudanças em uma versão inicial de cada arquivo

Já no Git, os dados são considerados como um conjunto de *snapshots* (captura de algo em determinado instante, como uma foto) de um mini-sistema de arquivos. Quando salvo ou consolidado um estado do projeto, o Git age como se tirasse uma foto de todos os arquivos naquele momento e armazena uma referência para esta captura, caso nenhuma alteração tenha sido feita, será salvo um link para arquivo idêntico ao anterior que já estava armazenado. Na figura 2 está ilustrado este procedimento.

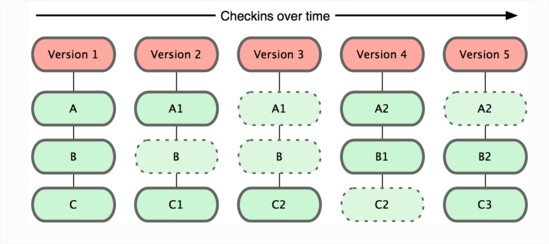


Figura 2.Git armazena dados como snapshots do projeto ao longo do tempo

**4 FLUXO DE TRABALHO E SEUS COMANDOS**

**REFERÊNCIAS**

CODEPROJETS, **Visual representation of SQL joins,** 10/01/2015. Disponível em: <http://www.codeproject.com/Articles/33052/Visual-Representation-of-SQL-Joins>. Acesso em: 05.out.2015.

DATE, C J. **Introdução a sistemas de banco de dados**. 8 ed. Rio de Janeiro: Elsevier, 2003.

ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistema de banco de dados**. 4 ed. São Paulo: Pearson Addison Wesley, 2005.

Não usar transcrição de e-books no trabalho, só deverá ser feito como citação indireta.

**Referências da internet:**

chave, título, data ou *sd* quando nao tiver data. Disponível em <link>. Acesso em 12.jan.12.

No texto, quando for feita a citação da internet deverá constar:

chave, data ou *sd, online*.

[http://nocoesengsw.blogspot.com.br/2010/03/controle-de-versao-de-software.html dia 02/05/17](http://nocoesengsw.blogspot.com.br/2010/03/controle-de-versao-de-software.html%20dia%2002/05/17)

<https://pt.wikipedia.org/wiki/Sistema_de_controle_de_vers%C3%B5es> dia 04/05/17

<https://git-scm.com/book/pt-br/v1/Primeiros-passos-Uma-Breve-Hist%C3%B3ria-do-Git> dia 04/05/17

[http://gnmd.webgrupos.com.br/arquivo\_downloads/Apresenta\_\_oSVN.pdf dia 04/05/17](http://gnmd.webgrupos.com.br/arquivo_downloads/Apresenta__oSVN.pdf%20dia%2004/05/17)

<http://blog.myscrumhalf.com/2011/10/vantagens-do-controle-de-versao-no-desenvolvimento-agil/> dia 04/05/17

https://git-scm.com/book/pt-br/v1/Primeiros-passos-No%C3%A7%C3%B5es-B%C3%A1sicas-de-Git dia 10/05/17 (Figura 1 e 2)