



PROVAS DE TI
TUDO PARA VOCÊ PASSAR

VERSIC – Em exercícios

Prof. Rodrigo Macedo



Conceitos Gerais



- Conjunto de tecnologias para se **controlar mudanças** em diversos arquivos.
- Permite recuperar versões anteriores de mudanças dos arquivos.
- Permite que múltiplas pessoas manipulem um arquivo.
- Pode ser utilizado além do desenvolvimento de software:
 - Processo de escrita de livros.
 - Legendagem de séries.



git

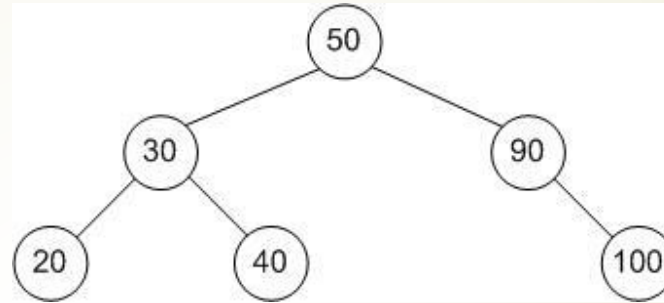


SUBVERSION

Operações Centralizada e Distribuída

- **Commit**(checkin): Realiza uma mudança de um arquivo no repositório.
- **Update**(checkout)/ Clone: Atualiza a cópia de um workspace conforme está versionada no repositório.
- **Diff**: Serve para identificar mudanças em arquivos de diferentes versões.

- **Branch:** Isola alterações numa linha separada do desenvolvimento. Também conhecido por prover ramificações dentro do processo de versionamento.



- **Tag:** Marcam revisões particulares para um conjunto de funcionalidades que precisam ou não ser alteradas.



- **Revision:** Representa um determinado momento de um repositório. Geralmente, se atribui uma numeração para distinguir cada modificação.



- **Merge:** Fusão de versões automática através da comparação de arquivos distintos e mesclando as alterações. **Pode ser feito em branches diferentes.**
 - **Merge com conflito:** Quando houver diferentes versões de um mesmo arquivo, será necessário resolver o conflito manualmente.



Quadro Comparativo – GIT e SVN

Centralizado	Distribuído	Descrição
Checkout	Clone	criação da cópia de trabalho/repositório
Commit	Commit	envia alterações para o repositório, criando uma revisão
Update	Update	atualiza a cópia/área de trabalho em uma revisão
	Pull	importa revisões feitas em outro repositório
	Push	envia revisões locais para outro repositório



Q1- [FCC SEMEF 2019] No repositório SVN, o comando usado para criar uma cópia de trabalho do repositório para poder editar é o

- a) Checkout.
- b) Diff.
- c) Perform.
- d) Commit.
- e) Update.

Q2 – [CS UFG 2017] ClearCase, SourceSafe, Git e SVN são exemplos de ferramentas que automatizam atividades diretamente ligadas

- a) ao gerenciamento de configuração e mudança de software.
- b) à engenharia de requisitos de software.
- c) à engenharia econômica de software.
- d) ao gerenciamento de riscos de software.

Q1- [FCC SEMEF 2019] No repositório SVN, o comando usado para criar uma cópia de trabalho do repositório para poder editar é o

a) **Checkout.**

b) Diff.

c) Perform.

d) Commit.

e) Update.

Q2 – [CS UFG 2017] ClearCase, SourceSafe, Git e SVN são exemplos de ferramentas que automatizam atividades diretamente ligadas

a) **ao gerenciamento de configuração e mudança de software.**

b) à engenharia de requisitos de software.

c) à engenharia econômica de software.

d) ao gerenciamento de riscos de software.

Q3- [FCC SEMEF 2019] Ao tentar postar uma nova versão do código em desenvolvimento no SVN, o técnico obteve a informação de que não seria possível postar, pois o arquivo com o código estava desatualizado. Nesta situação, o técnico deve

- a) fazer a revisão do código utilizando o comando DIFF e depois fazer a postagem utilizando o comando UPDATE.
- b) postar o código utilizando o comando COMMIT ao invés do CHECKOUT.
- c) fazer a atualização do código utilizando o comando CHECKOUT e depois fazer a postagem utilizando o comando UPDATE.
- d) postar o código utilizando o comando COMMIT seguido de UPDATE.
- e) fazer a atualização do código utilizando o comando UPDATE, fazer as correções e depois fazer a postagem.

Q3- [FCC SEMEF 2019] Ao tentar postar uma nova versão do código em desenvolvimento no SVN, o técnico obteve a informação de que não seria possível postar, pois o arquivo com o código estava desatualizado. Nesta situação, o técnico deve

- a) fazer a revisão do código utilizando o comando DIFF e depois fazer a postagem utilizando o comando UPDATE.
- b) postar o código utilizando o comando COMMIT ao invés do CHECKOUT.
- c) fazer a atualização do código utilizando o comando CHECKOUT e depois fazer a postagem utilizando o comando UPDATE.
- d) postar o código utilizando o comando COMMIT seguido de UPDATE.
- e) **fazer a atualização do código utilizando o comando UPDATE, fazer as correções e depois fazer a postagem.**

Q4- [FCC SEMEF 2019] Um Técnico em Web Design tem a incumbência de verificar a possibilidade de uso de um Gerenciamento de Repositório (SVN), sendo que em sua pesquisa esse técnico verificou que, nesse repositório, a operação de

- a) salvar no repositório mudanças feitas em uma cópia de trabalho, corresponde à operação denominada *commit*.
- b) criar um repositório corresponde à operação de *checkout*.
- c) descartar modificações feitas no repositório corresponde à operação denominada *status*.
- d) sincronizar uma cópia de trabalho com o repositório corresponde a uma operação de revisão.
- e) criar uma cópia de trabalho, a partir do repositório, corresponde à operação denominada *commit*.

Q4- [FCC SEMEF 2019] Um Técnico em Web Design tem a incumbência de verificar a possibilidade de uso de um Gerenciamento de Repositório (SVN), sendo que em sua pesquisa esse técnico verificou que, nesse repositório, a operação de

- a) salvar no repositório mudanças feitas em uma cópia de trabalho, corresponde à operação denominada *commit*.**
- b) criar um repositório corresponde à operação de *checkout*.
- c) descartar modificações feitas no repositório corresponde à operação denominada *status*.
- d) sincronizar uma cópia de trabalho com o repositório corresponde a uma operação de revisão.
- e) criar uma cópia de trabalho, a partir do repositório, corresponde à operação denominada *commit*.

Glossário GIT

- git push : envia alterações locais para o servidor (remote)
- git reset --hard HEAD~1 : desfaz commit e perde as alterações nos arquivos
- git stash : armazena alterações na pilha para uso posterior
- git clone: obtém repositório do servidor para a maquina local
- git reset --soft HEAD~1: desfaz commit e não perde alterações nos arquivos
- Git commit --amend: possibilita a alteração da mensagem do último commit



Operações **Apenas** Repositório Distribuído

- **Pull**(Puxar): Atualiza o repositório local com as alterações feitas num outro repositório.



- **Push**(Empurrar): Envia alterações do repositório local para um outro repositório.



Q5- [CCV UFC 2019] Depois de inicializar um novo repositório Git com o comando "*git init*", e criar um arquivo chamado **git-teste.html**, qual dos seguintes comandos **não irá surtir o efeito desejado** se for executado?

- a) Git add.
- b) Git status.
- c) git branch -a.
- d) git add git-teste.html.
- e) git commit -m "arquivo git teste adicionado".

Q5- [CCV UFC 2019] Depois de inicializar um novo repositório Git com o comando "*git init*", e criar um arquivo chamado **git-teste.html**, qual dos seguintes comandos **não irá surtir o efeito desejado** se for executado?

- a) Git add.
- b) Git status.
- c) git branch -a.
- d) git add git-teste.html.
- e) **git commit -m "arquivo git teste adicionado".**

Q6- [IADES CFM 2018] Um dos sistemas mais utilizados para o controle de versão de código é o git. Considere hipoteticamente que um programador inadvertidamente realizou um commit na branch errada utilizando o git. É necessário que ele remova esse commit sem perder o trabalho que ele realizou localmente. O comando que resolverá o problema é

- a) git push
- b) git reset --hard HEAD~1
- c) git stash
- d) git clone
- e) git reset --soft HEAD~1

Q6- [IADES CFM 2018] Um dos sistemas mais utilizados para o controle de versão de código é o git. Considere hipoteticamente que um programador inadvertidamente realizou um commit na branch errada utilizando o git. É necessário que ele remova esse commit sem perder o trabalho que ele realizou localmente. O comando que resolverá o problema é

- a) git push
- b) git reset --hard HEAD~1
- c) git stash
- d) git clone
- e) **git reset --soft HEAD~1**

Q7- [CCV UFC 2019] Qual arquivo é necessário ser configurado para especificar intencionalmente que determinados arquivos não sejam rastreados (tracked) e que o Git deve ignorar no repositório Git local?

- a) ignore.git
- b) .gitignore
- c) git.ignore
- d) gitignore.txt
- e) .gitignore.txt

Q8 – [CESPE SLU DF 2019] No que diz respeito a ferramentas de desenvolvimento, julgue o item a seguir.

Para obter uma cópia de um repositório Git existente, é correto utilizar o comando `git clone <url>`.

Q7- [CCV UFC 2019] Qual arquivo é necessário ser configurado para especificar intencionalmente que determinados arquivos não sejam rastreados (tracked) e que o Git deve ignorar no repositório Git local?

- a) ignore.git
- b) .gitignore**
- c) git.ignore
- d) gitignore.txt
- e) .gitignore.txt

Q8 – [CESPE SLU DF 2019] No que diz respeito a ferramentas de desenvolvimento, julgue o item a seguir.

Para obter uma cópia de um repositório Git existente, é correto utilizar o comando `git clone <url>`. CERTO.

Q9- [CCV UFC 2019] Utilizando Git, qual erro será obtido ao tentar enviar (*git push*) mudanças no *master-branch* para um repositório remoto, sendo que o repositório remoto contém trabalho que você não tem localmente?

- a) 404
- b) 500
- c) Rejected
- d) Access denied
- e) 400 Bad Request

Q9- [CCV UFC 2019] Utilizando Git, qual erro será obtido ao tentar enviar (*git push*) mudanças no *master-branch* para um repositório remoto, sendo que o repositório remoto contém trabalho que você não tem localmente?

- a) 404
- b) 500
- c) **Rejected**
- d) Access denied
- e) 400 Bad Request

Q10- [COMPERVE URFN 2018] Repositórios de controle de versão facilitam o controle do trabalho realizado por diferentes desenvolvedores de uma equipe. Acerca das operações suportadas pelo sistema *git* de controle de versão, é correto afirmar:

- a) *checkout* é a operação realizada para criar um repositório novo a partir de dados de um servidor.
- b) a operação *clean* inicializa um novo repositório vazio na máquina local.
- c) *fork* é utilizado para se realizar uma cópia de um repositório existente, visando geralmente submeter as alterações ao repositório original.
- d) a operação *showTree* apresenta a árvore de versionamento de um dado arquivo.

Q10- [COMPERVE URFN 2018] Repositórios de controle de versão facilitam o controle do trabalho realizado por diferentes desenvolvedores de uma equipe. Acerca das operações suportadas pelo sistema *git* de controle de versão, é correto afirmar:

- a) *checkout* é a operação realizada para criar um repositório novo a partir de dados de um servidor.
- b) a operação *clean* inicializa um novo repositório vazio na máquina local.
- c) ***fork* é utilizado para se realizar uma cópia de um repositório existente, visando geralmente submeter as alterações ao repositório original.**
- d) a operação *showTree* apresenta a árvore de versionamento de um dado arquivo.

Q11- [COMPERVE URFN 2018] O controle de versão de arquivos permite a realização de um melhor trabalho por diferentes desenvolvedores de uma equipe. Sobre o funcionamento do sistema *git* de controle de versão, é correto afirmar:

- a) a sincronização de um repositório local pode ocorrer com um repositório remoto utilizado como servidor central, excluindo repositórios locais em máquinas de outros desenvolvedores.
- b) embora o controle de versão no *git*, usualmente, seja feito de forma distribuída, o *git* pode funcionar apenas com uma versão local do repositório.
- c) o versionamento do *git* funciona para arquivos textos, ignorando arquivos de imagens, vídeos, etc.
- d) existe uma limitação técnica de no máximo 1024 usuários por projeto em um mesmo repositório *git*.

Q11- [COMPERVE URFN 2018] O controle de versão de arquivos permite a realização de um melhor trabalho por diferentes desenvolvedores de uma equipe. Sobre o funcionamento do sistema *git* de controle de versão, é correto afirmar:

- a) a sincronização de um repositório local pode ocorrer com um repositório remoto utilizado como servidor central, excluindo repositórios locais em máquinas de outros desenvolvedores.
- b) embora o controle de versão no *git*, usualmente, seja feito de forma distribuída, o *git* pode funcionar apenas com uma versão local do repositório.**
- c) o versionamento do *git* funciona para arquivos textos, ignorando arquivos de imagens, vídeos, etc.
- d) existe uma limitação técnica de no máximo 1024 usuários por projeto em um mesmo repositório *git*.

Q12- [CCV UFC 2019] Em um repositório Git, eventualmente é necessário realizar a correção de alguma mensagem de commit. Qual dos comandos abaixo possibilita a alteração da mensagem do último commit?

- a) git log.
- b) git checkout
- c) git reset HEAD
- d) git diff –cached
- e) git commit –amend

Q13 – [CESPE EBSE RH 2018] Acerca de infraestrutura de TI, julgue o item subsequente.

O comando git clone, seguido do caminho do repositório, pode ser usado para criar uma cópia desse repositório para o diretório atualmente em uso em uma máquina Linux.

Q12- [CCV UFC 2019] Em um repositório Git, eventualmente é necessário realizar a correção de alguma mensagem de commit. Qual dos comandos abaixo possibilita a alteração da mensagem do último commit?

- a) git log.
- b) git checkout
- c) git reset HEAD
- d) git diff --cached
- e) **git commit --amend**

Q13 – [CESPE EBSE RH 2018] Acerca de infraestrutura de TI, julgue o item subsequente.

O comando git clone, seguido do caminho do repositório, pode ser usado para criar uma cópia desse repositório para o diretório atualmente em uso em uma máquina Linux. CERTO.

Q14- [CS UFG SANEAGO 2018] Uma desenvolvedora está trabalhando em um projeto cujo repositório de arquivos está em um servidor que usa o modelo descentralizado GIT. Ela não possui internet, mas deseja consolidar as modificações que fez para posteriormente enviá-las para gravação na nuvem assim que tiver conexão à rede. Para executar este versionamento, a desenvolvedora deve

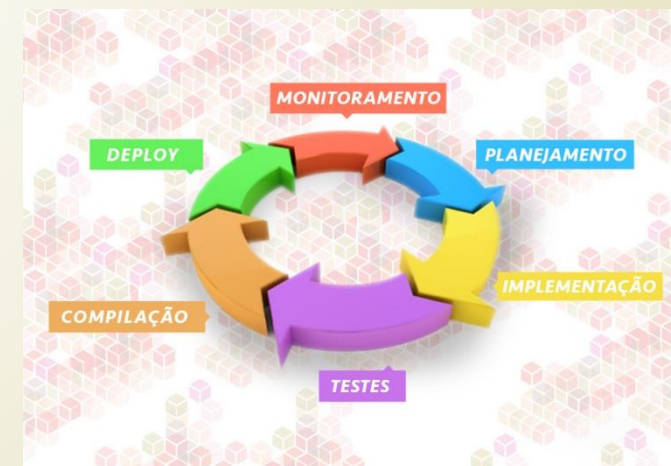
- a) realizar um commit local e, posteriormente, fazer o envio da versão nova para o servidor principal.
- b) fazer uma cópia dos arquivos em uma nova pasta, renomeando esse novo diretório com número de versão acima do atual.
- c) fazer solicitação de commit para o servidor, de maneira que esta solicitação fique pendente e, assim que existir conexão à rede, o código será automaticamente enviado para o sistema.
- d) criar uma nova ramificação do repositório local, colocando todos os arquivos da nova versão na pasta *trunk*.

Q14- [CS UFG SANEAGO 2018] Uma desenvolvedora está trabalhando em um projeto cujo repositório de arquivos está em um servidor que usa o modelo descentralizado GIT. Ela não possui internet, mas deseja consolidar as modificações que fez para posteriormente enviá-las para gravação na nuvem assim que tiver conexão à rede. Para executar este versionamento, a desenvolvedora deve

- a) **realizar um commit local e, posteriormente, fazer o envio da versão nova para o servidor principal.**
- b) fazer uma cópia dos arquivos em uma nova pasta, renomeando esse novo diretório com número de versão acima do atual.
- c) fazer solicitação de commit para o servidor, de maneira que esta solicitação fique pendente e, assim que existir conexão à rede, o código será automaticamente enviado para o sistema.
- d) criar uma nova ramificação do repositório local, colocando todos os arquivos da nova versão na pasta *trunk*.

Integração Contínua

- Prática utilizada no desenvolvimento de software em que os membros integram seus trabalhos frequentemente.
- Presente na metodologia XP. E no DevOps.
- “Integração Contínua é uma pratica de desenvolvimento de software onde os membros de um time integram seu trabalho frequentemente, geralmente cada pessoa integra pelo menos diariamente – podendo haver **múltiplas integrações por dia**. Cada integração é verificada por um build automatizado (incluindo testes) para detectar **erros de integração** o mais rápido possível. Muitos times acham que essa abordagem leva a uma significativa redução nos problemas de integração e permite que um time desenvolva software coeso mais rapidamente” – Martin Fowler



Pré - Requisitos

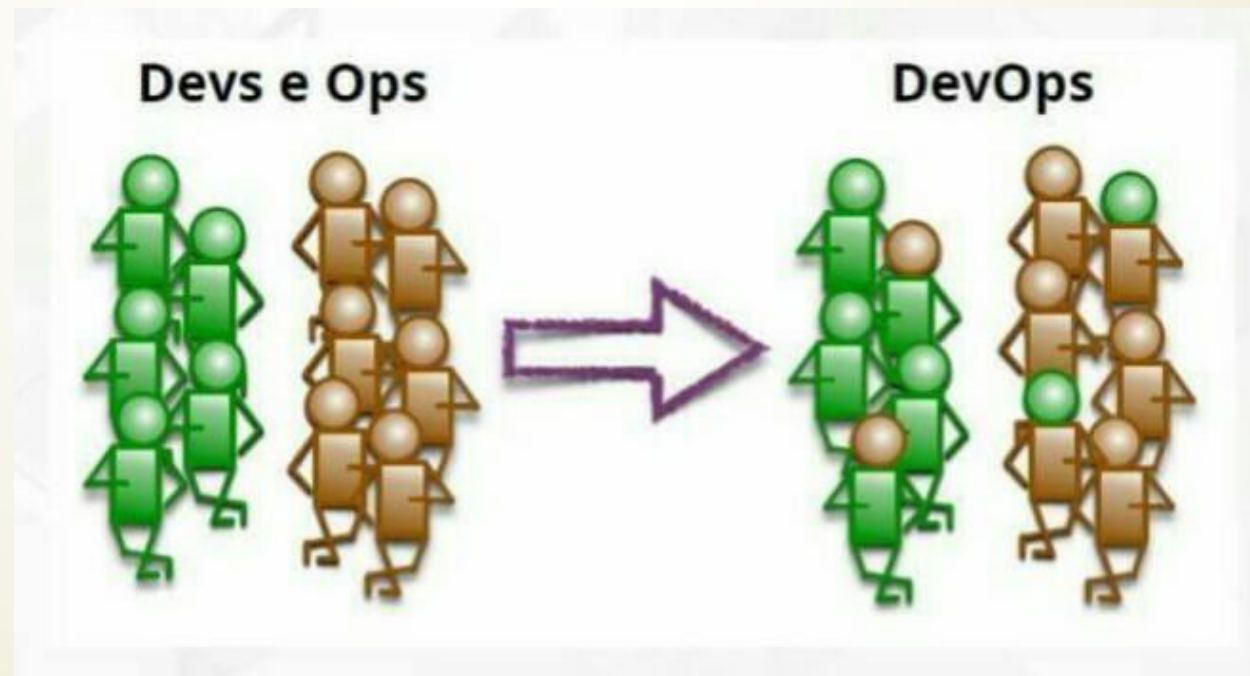
- Utilizar alguma ferramenta para controle de versão.
- Utilização de ferramentas para criação de testes automáticos(TDD).
- Utilização de ferramentas que venham prover o processo de build(compila, roda os testes, etc) automático

JUnit



Entrega Contínua

- Todo o processo de um software só tem valor ao cliente quando é **entregue**.
- “Entrega Contínua é uma **disciplina** de desenvolvimento na qual software é construído de tal maneira que o mesmo pode ser colocado em **produção** a qualquer momento” – Martin Fowler.
- “Todos envolvidos” no processo de entrega.



Principais Erros

- ▶ Entregar software só após a conclusão de todo o processo do software.
- ▶ Pouca colaboração entre os diferentes integrantes de uma equipe.



Q15- [CCV UFC 2019] Sobre *Jenkins*, assinale a alternativa correta.

- a) Jenkins é uma ferramenta de testes unitários.
- b) *Jenkins* é uma IDE de desenvolvimento Java.
- c) *Jenkins* é uma ferramenta de automação de compilação.
- d) *Jenkins* é uma ferramenta de integração contínua e entrega contínua.
- e) *Jenkins* possui suporte apenas aos sistemas de controle de versão SVN, Git e Mercurial.

Q15- [CCV UFC 2019] Sobre *Jenkins*, assinale a alternativa correta.

- a) Jenkins é uma ferramenta de testes unitários.
- b) *Jenkins* é uma IDE de desenvolvimento Java.
- c) *Jenkins* é uma ferramenta de automação de compilação.
- d) ***Jenkins* é uma ferramenta de integração contínua e entrega contínua.**
- e) *Jenkins* possui suporte apenas aos sistemas de controle de versão SVN, Git e Mercurial.

Q16- [CESPE SLU DF 2019] Com relação a desenvolvimento de *software*, julgue o item a seguir.

Na prática de integração contínua para desenvolvimento de *software*, vários colaboradores criam e mantêm o código de forma organizada e controlada, utilizando ferramentas como Git (controle de versão), Junit (testes), Hudson e Jenkins (*deploys* em ambientes de desenvolvimento e produção), o que reduz a geração de erros de integração.

Q16- [CESPE SLU DF 2019] Com relação a desenvolvimento de *software*, julgue o item a seguir.

Na prática de integração contínua para desenvolvimento de *software*, vários colaboradores criam e mantêm o código de forma organizada e controlada, utilizando ferramentas como Git (controle de versão), Junit (testes), Hudson e Jenkins (deploys em ambientes de desenvolvimento e produção), o que reduz a geração de erros de integração. CERTO.

GABARITO

Q1 – LETRA A.

Q2 - LETRA A.

Q3 – LETRA E.

Q4 - LETRA A.

Q5 - LETRA E.

Q6 - LETRA E.

Q7 – LETRA B.

Q8 – CERTO.

Q9 - LETRA C.

Q10 - LETRA C.

Q11 – LETRA B.

Q12 - LETRA E.

Q13 - CERTO.

Q14 – LETRA A.

Q15 – LETRA D.

Q16 - CERTO.