



PROVAS DE TI
TUDO PARA VOCÊ PASSAR

Versionamento de código e integração contínua

Rodrigo Macedo – rcbm539@gmail.com

<http://www.itnerante.com.br/profile/RodrigoMacedo>



git



SUBVERSION

Conceitos Gerais

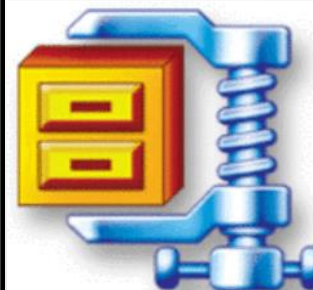
- Conjunto de tecnologias para se **controlar mudanças** em diversos arquivos.
- Permite recuperar versões anteriores de mudanças dos arquivos.
- Permite que múltiplas pessoas manipulem um arquivo.
- Pode ser utilizado além do desenvolvimento de software:
 - Processo de escrita de livros.
 - Legendagem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

'Descontrole' de Versão

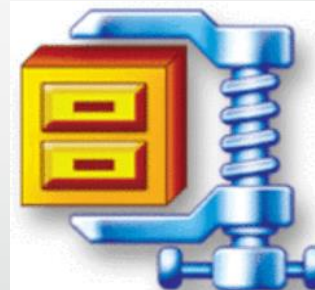


Versionamento de Arquivos: Cópia e cola renomeando arquivos.

V 0.1



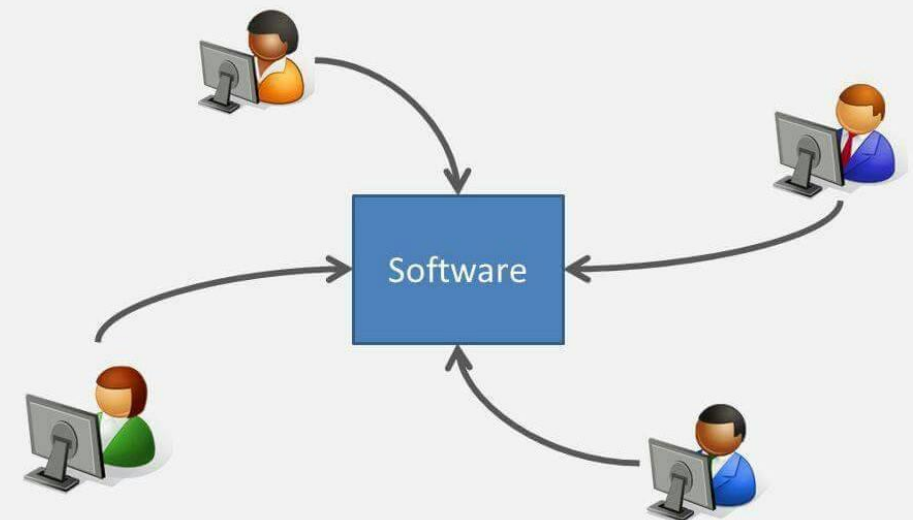
V 0.2



V 1.0

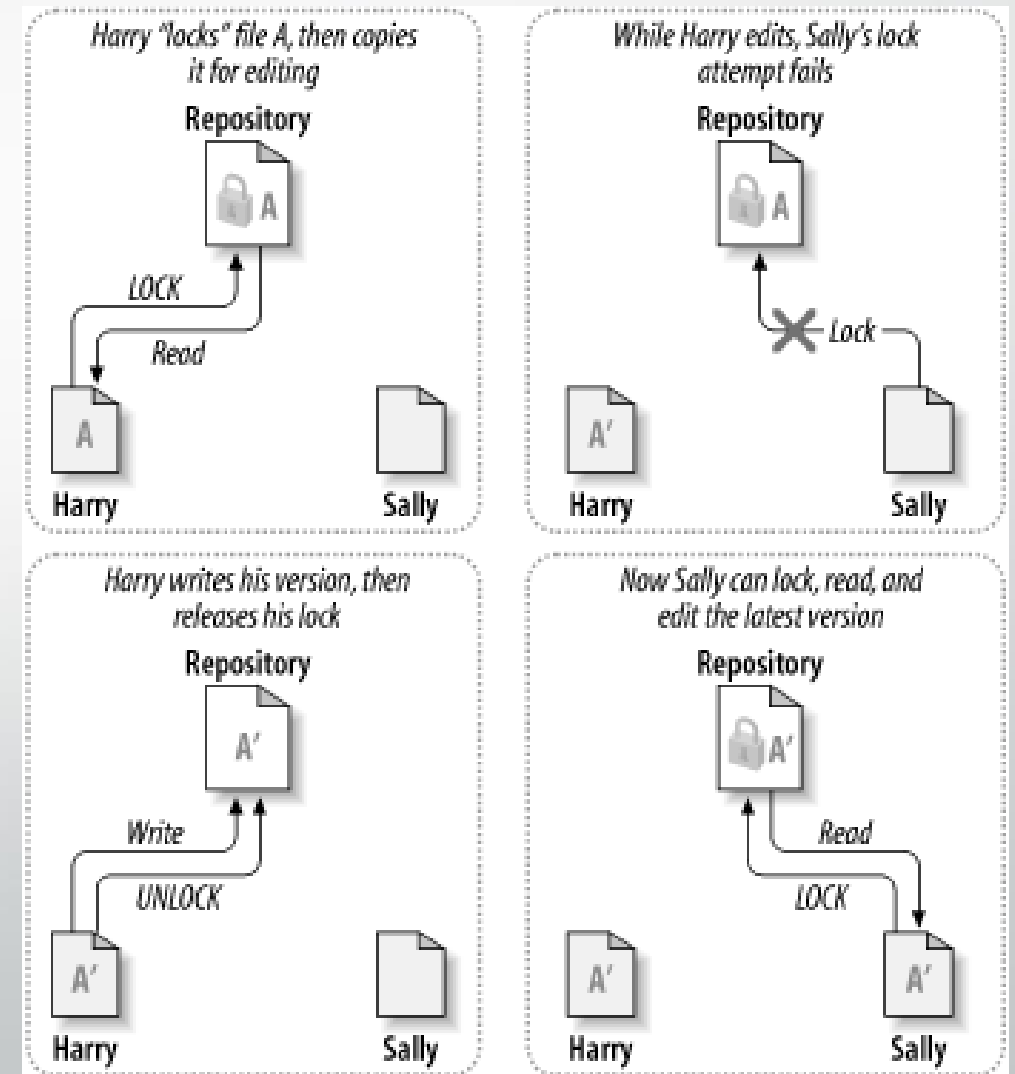


Controle de Versão



Políticas de Versionamento

- **Lock-Modify-Unlock:** Permite que apenas uma pessoa por vez altere um arquivo no repositório.
 - Essa política é gerenciada através de locks.



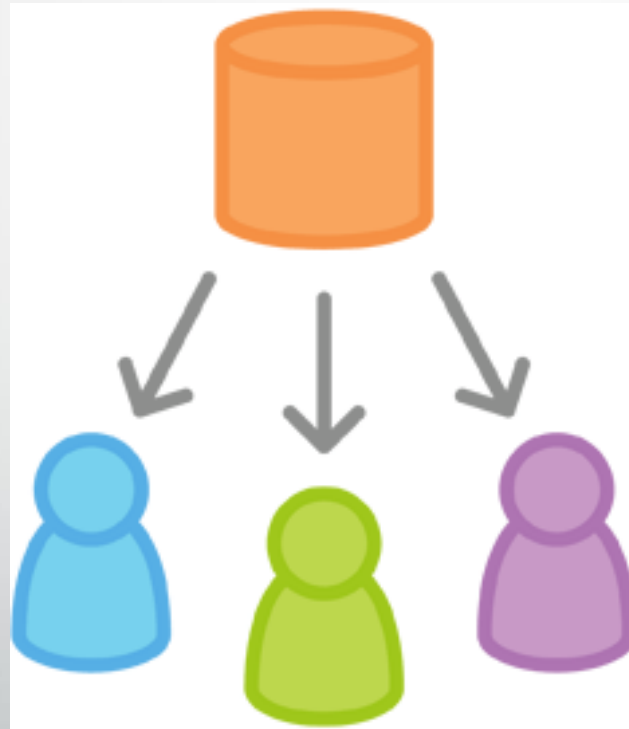
- **Copy-Modify-Merge:** Alternativa ao lock. Política utilizada hoje, nos diversos sistemas de controle de versão.
 - Possibilita trabalho colaborativo.
 - As modificações feitas por diferentes pessoas num mesmo repositório serão mescladas.
 - Caso, as modificações feitas por diferentes pessoas sejam nos mesmos arquivos teremos um merge com conflito, que deve ser resolvido manualmente



Workspace e Repositório

Workspace (cópia de trabalho): Corresponde a localização física de onde os arquivos residem.

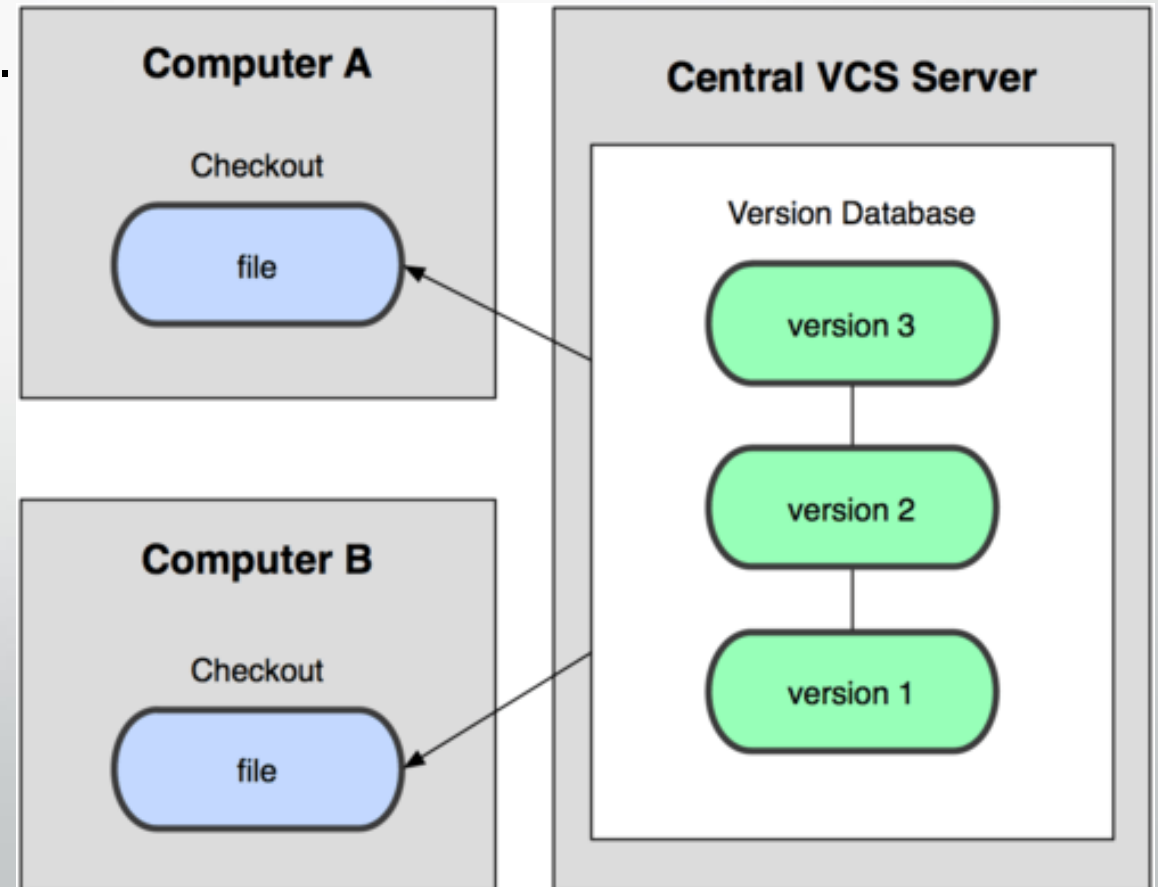
Repositório: Corresponde ao local onde são armazenados os arquivos que estão sendo versionados.



Tipos de Repositório

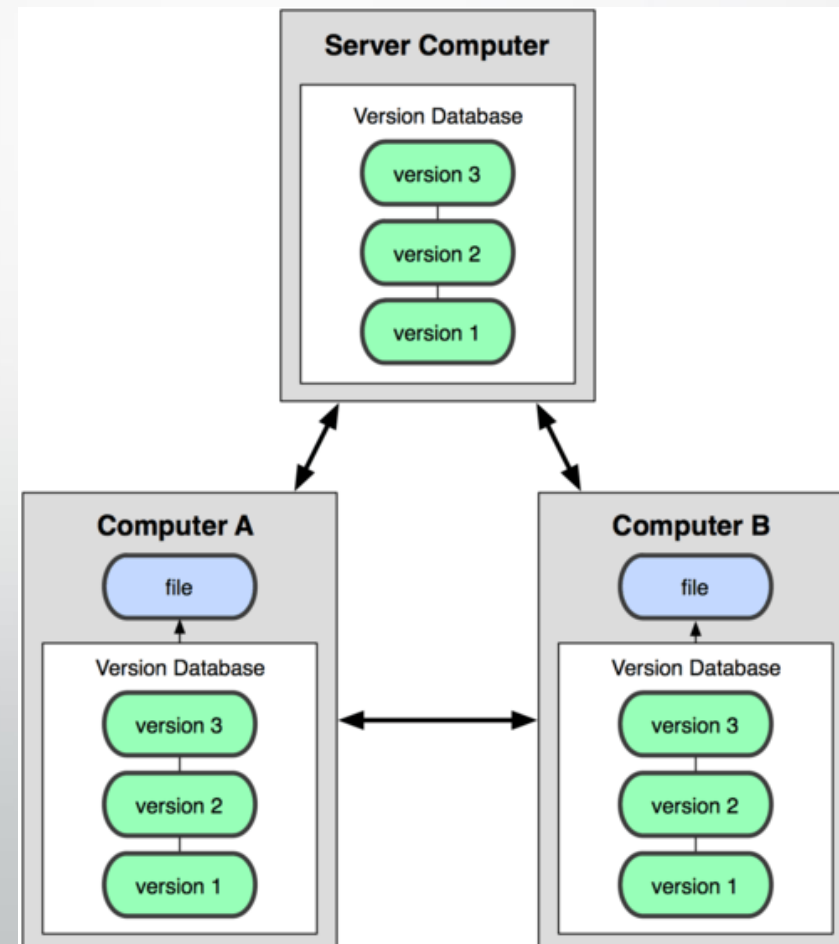
Centralizado:

- Possuem único servidor central que contém todos os arquivos.
- Corre-se o risco de perder os arquivos caso os arquivos do servidor forem corrompidos
- Exemplo: **CVS e Subversion(SVN).**



Distribuído:

- Cada computador possui a cópia de um repositório completo, não apenas das últimas versões.
- Apresenta maior escalabilidade, pois não se limita a um único servidor.
- Exemplo: **GIT e Mercurial**



Primeira Bateria de Questões



Q1 - [FCC TST 2012] Sistemas de controles de versões são ferramentas essenciais na gestão de tecnologia da informação de empresas, em especial em empresas desenvolvedoras de software. Estes sistemas têm o intuito de

- A) alocar recursos específicos para o desenvolvimento de diferentes versões do sistema.
- B) calcular as funcionalidades do sistema, incluindo cálculos de pontos de função.
- C) identificar uma alteração específica efetuada em um código fonte.
- D) controlar as versões dos diversos softwares adquiridos pela empresa.
- E) estimar o custo e tempo de desenvolvimento de uma versão específica de um sistema.

Q1 - [FCC TST 2012] Sistemas de controles de versões são ferramentas essenciais na gestão de tecnologia da informação de empresas, em especial em empresas desenvolvedoras de software. Estes sistemas têm o intuito de

A) alocar recursos específicos para o desenvolvimento de diferentes versões do sistema.

B) calcular as funcionalidades do sistema, incluindo cálculos de pontos de função.

C) identificar uma alteração específica efetuada em um código fonte.

D) controlar as versões dos diversos softwares adquiridos pela empresa.

E) estimar o custo e tempo de desenvolvimento de uma versão específica de um sistema.

Q2 - [INSTITUTO AOCP UFPB 2014] O CVS e o Git são bastante utilizados para dar suporte aos ambientes de desenvolvimento de software. Sabendo disso, assinale a alternativa que apresenta corretamente a funcionalidade a que se destina o CVS e o Git.

- a) Servidores web.
- b) Servidores de backup.
- c) Servidores de aplicação.
- d) Servidores de banco de dados.
- e) Servidores de controle de versão.

Q3 - [CESPE TJDFT 2015] Uma desvantagem dos sistemas de controle de versão que adotam o modelo *lock-modify-unlock* é a baixa produtividade em trabalhos colaborativos: esse modelo não permite que o conteúdo de um arquivo seja alterado por mais de um usuário simultaneamente.

Q2 - [INSTITUTO AOCP UFPB 2014] O CVS e o Git são bastante utilizados para dar suporte aos ambientes de desenvolvimento de software. Sabendo disso, assinale a alternativa que apresenta corretamente a funcionalidade a que se destina o CVS e o Git.

- a) Servidores web.
- b) Servidores de backup.
- c) Servidores de aplicação.
- d) Servidores de banco de dados.
- e) Servidores de controle de versão.

Q3 - [CESPE TJDFT 2015] Uma desvantagem dos sistemas de controle de versão que adotam o modelo *lock-modify-unlock* é a baixa produtividade em trabalhos colaborativos: esse modelo não permite que o conteúdo de um arquivo seja alterado por mais de um usuário simultaneamente.
CERTO.

Q4 - [FCCTRE RN 2011] São exemplos típicos de ferramentas open source para controle de versão no desenvolvimento de um software:

- a) Git, ClearCase e CVS.
- b) CVS, SVN e Git.
- c) SourceSafe, CVS e ClearCase.
- d) SVN, ClearCase e Git.
- e) SourceSafe, ClearCase e SVN.

Q5 - [CESPE ANATEL 2014] As ferramentas de controle de versão Git e SVN oferecem o mesmo grau de confiabilidade no armazenamento das informações e são ambas implantadas conforme o conceito de sistemas de controle de versão distribuído.

Q4 - [FCCTRE RN 2011] São exemplos típicos de ferramentas open source para controle de versão no desenvolvimento de um software:

a)Git, ClearCase e CVS.

b)CVS, SVN e Git.

c)SourceSafe, CVS e ClearCase.

d)SVN, ClearCase e Git.

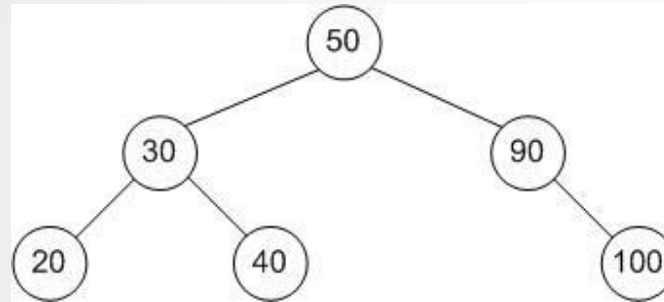
e)SourceSafe, ClearCase e SVN.

Q5 - [CESPE ANATEL 2014] As ferramentas de controle de versão Git e SVN oferecem o mesmo grau de confiabilidade no armazenamento das informações e são ambas implantadas conforme o conceito de sistemas de controle de versão distribuído. **ERRADO.**

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.

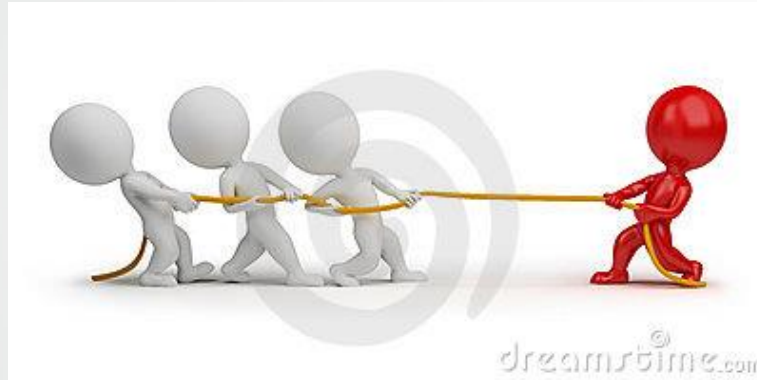


- **Merge:** Fusão de versões automática através da comparação de arquivos distintos e mesclando as alterações. **Pode ser feito em branches diferentes.**
 - **Merge com conflito:** Quando houver diferentes versões de um mesmo arquivo, será necessário resolver o conflito manualmente.

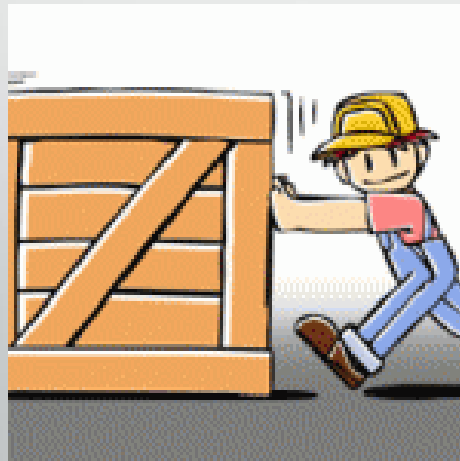


Operações **Apenas** Repositório Distribuído

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



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



Quadro Comparativo

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

Q6 - [FCC MPE AP 2012] Uma ferramenta de controle de versão é utilizada para controlar as diferentes versões de um documento, o qual pode possuir ramificações de sua linha de desenvolvimento original. Um termo técnico comum, utilizado para definir uma ramificação é

- a)trunk.
- b)branch.
- c)head.
- d)mainline.
- e)base.

Q7 - [CESPE TRT ES 2013] Um sistema de controle de versões permite que os documentos, códigos e artefatos possam ser obtidos por meio de uma operação conhecida como check-out, modificados dentro do espaço de trabalho do desenvolvedor e, depois, retornados ao repositório por meio de uma operação conhecida como check-in.

Q6 - [FCC MPE AP 2012] Uma ferramenta de controle de versão é utilizada para controlar as diferentes versões de um documento, o qual pode possuir ramificações de sua linha de desenvolvimento original. Um termo técnico comum, utilizado para definir uma ramificação é

a)trunk.

b)branch.

c)head.

d)mainline.

e)base.

Q7 - [CESPE TRT ES 2013] Um sistema de controle de versões permite que os documentos, códigos e artefatos possam ser obtidos por meio de uma operação conhecida como check-out, modificados dentro do espaço de trabalho do desenvolvedor e, depois, retornados ao repositório por meio de uma operação conhecida como check-in. CERTO.

Q8- [IFPE 2016] Ferramentas de Controle de Versão ajudam a gerenciar mudanças em artefatos de projetos de

software, como código fonte, pastas e arquivos em geral. Com respeito a ferramentas de Controle de Versão é CORRETO afirmar que

- a) SVN e CVS são exemplos de ferramentas de controle de versão distribuído.
- b) GIT e MERCURIAL são exemplos de ferramentas de controle de versão distribuído.
- c) em ferramentas de controle de versão distribuído, o comando clone é utilizado para enviar alterações para o repositório, criando uma versão no repositório central.
- d) em ferramentas de controle de versão centralizado, o comando pull é utilizado para enviar revisões locais para outro repositório.
- e) em ferramentas de controle de versão centralizado, o comando push é utilizado para enviar revisões locais para outro repositório.

Q8 - [IFPE 2016] Ferramentas de Controle de Versão ajudam a gerenciar mudanças em artefatos de projetos de

software, como código fonte, pastas e arquivos em geral. Com respeito a ferramentas de Controle de Versão é CORRETO afirmar que

a) SVN e CVS são exemplos de ferramentas de controle de versão distribuído.

b) GIT e MERCURIAL são exemplos de ferramentas de controle de versão distribuído.

c) em ferramentas de controle de versão distribuído, o comando clone é utilizado para enviar alterações para o repositório, criando uma versão no repositório central.

d) em ferramentas de controle de versão centralizado, o comando pull é utilizado para enviar revisões locais para outro repositório.

e) em ferramentas de controle de versão centralizado, o comando push é utilizado para enviar revisões locais para outro repositório.

Q9 - [FEPESE FATMA 2013] Assinale a alternativa que indica corretamente a ramificação (branch) em um sistema moderno de controle de versão:

- a) Possibilidade de quebrar a linha do desenvolvimento em mais de um caminho.
- b) Apresenta a linha de desenvolvimento conflitada.
- c) Atualiza na cópia local as mudanças enviadas por outro desenvolvedor.
- d) Representa a última versão enviada ao sistema no braço principal.
- e) Uma versão do desenvolvimento que possui alguma anomalia bastante visível e geralmente grave.

Q10 - [CESPE SERPRO 2013] Os sistemas de controle de versões permitem que os artefatos sejam obtidos do repositório por meio da operação conhecida como check-in.

Q9 - [FEPESE FATMA 2013] Assinale a alternativa que indica corretamente a ramificação (branch) em um sistema moderno de controle de versão:

a) Possibilidade de quebrar a linha do desenvolvimento em mais de um caminho.

b) Apresenta a linha de desenvolvimento conflitada.

c) Atualiza na cópia local as mudanças enviadas por outro desenvolvedor.

d) Representa a última versão enviada ao sistema no braço principal.

e) Uma versão do desenvolvimento que possui alguma anomalia bastante visível e geralmente grave.

Q10 - [CESPE SERPRO 2013] Os sistemas de controle de versões permitem que os artefatos sejam obtidos do repositório por meio da operação conhecida como check-in. **ERRADO.**

Q11 – [FCC DPE SP 2015] Um SCV – Sistema de Controle de Versão.

- a) distribuído, trabalha apenas com um servidor central e diversas áreas de trabalho, baseado na arquitetura cliente-servidor. Por ser distribuído, as áreas de trabalho precisam primeiro passar pelo servidor para poderem se comunicar.
- b) localizado, atende apenas equipes de desenvolvimento pequenas e que trabalham em uma rede local sem fio. Este SCV não tem problemas de velocidade para enviar e receber os dados e tem um bom tempo de resposta do servidor.
- c) distribuído, permite que as áreas de trabalho se comuniquem. Mas recomenda-se usar um servidor remoto para hospedar o projeto e centralizar o fluxo de envio, evitando ramificações do projeto e a perda do controle.
- d) centralizado, é recomendado para equipes com muitos desenvolvedores e que se encontram em diferentes filiais. Cada área de trabalho pode ter seu próprio servidor e as operações de push e pull são feitas na própria máquina.
- e) funciona, com duas operações: 1) check-in: permite pegar a versão de outra área de trabalho e mesclar com a do usuário e 2) check-out: permite ao usuário enviar para outra área a sua versão do projeto.

Q11 – [FCC DPE SP 2015] Um SCV – Sistema de Controle de Versão.

- a) distribuído, trabalha apenas com um servidor central e diversas áreas de trabalho, baseado na arquitetura cliente-servidor. Por ser distribuído, as áreas de trabalho precisam primeiro passar pelo servidor para poderem se comunicar.
- b) localizado, atende apenas equipes de desenvolvimento pequenas e que trabalham em uma rede local sem fio. Este SCV não tem problemas de velocidade para enviar e receber os dados e tem um bom tempo de resposta do servidor.
- c) distribuído, permite que as áreas de trabalho se comuniquem. Mas recomenda-se usar um servidor remoto para hospedar o projeto e centralizar o fluxo de envio, evitando ramificações do projeto e a perda do controle.
- d) centralizado, é recomendado para equipes com muitos desenvolvedores e que se encontram em diferentes filiais. Cada área de trabalho pode ter seu próprio servidor e as operações de push e pull são feitas na própria máquina.
- e) funciona, com duas operações: 1) check-in: permite pegar a versão de outra área de trabalho e mesclar com a do usuário e 2) check-out: permite ao usuário enviar para outra área a sua versão do projeto.

Q12 - [FEPESE FATMA 2012] Assinale a alternativa de indica corretamente o comando que envia as alterações da cópia local para o Servidor, através do Cliente, em um sistema moderno de controle de versão:

- a)merge
- b)update
- c)confict
- d)commit
- e)checkout

Q12 - [FEPESE FATMA 2012] Assinale a alternativa de indica corretamente o comando que envia as alterações da cópia local para o Servidor, através do Cliente, em um sistema moderno de controle de versão:

- a)merge
- b)update
- c)confict
- d)commit
- e)checkout

GABARITO

Q1 – LETRA C.

Q2 – LETRA E.

Q3 – C(CERTO).

Q4 – LETRA B.

Q5 – E(ERRADO).

Q6 – LETRA B.

Q7 – C(CERTO).

Q8 – LETRA B.

Q9- LETRA A.

Q10 - E(ERRADO).

Q11 – LETRA C.

Q12 – LETRA D.

Rational ClearCase



- É pago.
- Pertence a IBM.
- Possui todas as funções padrão de um Sistema de controle de versão.
- Há um controle de acesso bem rígido, podendo ser utilizado o protocolo LDAP.
- Pode ser integrado com diversas outras ferramentas da IBM, como o IBM Rational Build Forge, para automatizar o processo de gerenciamento de liberação.

CVS



- Sistema de versão concorrente. Open – Source.
- Um dos primeiros sistema controle de versão consagrado
- Um repositório cvs contém:
 - **Fluxo principal:** chamado **trunk**. Ficam nesse projeto todos os arquivos devidamente funcionais do projeto.
 - **Fluxo alternativo:** chamado **branch**. Onde são implementada mudanças em separado do projeto. Assim que estiverem finalizadas são mescladas ao trunk.
 - **Tags:** Indicam versões estáveis, que não estarão sujeitas a alterações.

Operações Específicas

- **Checkout:** Geralmente utilizado para fazer o primeiro download de um módulo do repositório.
- **Export:** Download de um módulo do repositório que não estarão sob controle do CVS.
- **Import:** Criação de um módulo através do upload de uma estrutura de diretórios.
- **Arquivo .cvsignore:** São adicionados neste arquivo, todos arquivos e pastas que não serão versionados.
- **Arquivos binários:** Todo arquivo que não é legível por humanos.
 - O CVS não permite comparações em arquivos binários.
 - Para que o CVS consiga manipular arquivos binários, é necessário que seja informado o tipo de arquivo em algumas operações como: **importação e adição**. Esse parâmetro é o **-kb**.
 - Exemplo: `cvs add -kb [tipo de arquivo binário]`

Características

- Cada commit altera o seu número de revisão.
- Branch X Trunk



- Merge entre branches:
 - Rebase: Move os arquivos para uma sub-branch.
 - Merge: Mescla os arquivos e versiona na trunk.

Limitações

- Não permite renomear um repositório no cliente, apenas no servidor
- Os arquivos, no cliente, deviam ser removidos e adicionados novamente, caso se quisesse renomear um arquivo sem precisar ir no servidor.
- Não permite também mover arquivos.

Segunda Bateria de Questões





Q13 - [CESPE MEC 2015] Rational ClearCase é um software de soluções de gerência de configuração que simplifica o processo de mudanças e fornece controle de versão.

Q13 - [CESPE MEC 2015] Rational ClearCase é um software de soluções de gerência de configuração que simplifica o processo de mudanças e fornece controle de versão. CERTO.

Q14 - [FGV MEC 2009] O CVS é um sistema de controle de versão Open Source. Acerca das terminologias utilizadas pelo CVS, leia os trechos a seguir:

_____ é o envio das modificações feitas pelo usuário ao repositório CVS.

_____ é a numeração atribuída pelo CVS a cada modificação de um arquivo.

_____ é uma ramificação no desenvolvimento, usada para descrever o processo de divisão dos arquivos de um projeto em linhas de desenvolvimento independentes.

Assinale a alternativa que preencha corretamente, de cima para baixo, as lacunas dos trechos acima:

a)Checkout - Release - Branch.

b)Commit - Revision - Branch.

c)Update - Revision - Merge.

d)Commit - Checkin - Hijack.

e)Update - Checkin - Merge.

Q14 - [FGV MEC 2009] O CVS é um sistema de controle de versão Open Source. Acerca das terminologias utilizadas pelo CVS, leia os trechos a seguir:

_____ é o envio das modificações feitas pelo usuário ao repositório CVS.

_____ é a numeração atribuída pelo CVS a cada modificação de um arquivo.

_____ é uma ramificação no desenvolvimento, usada para descrever o processo de divisão dos arquivos de um projeto em linhas de desenvolvimento independentes.

Assinale a alternativa que preencha corretamente, de cima para baixo, as lacunas dos trechos acima:

a)Checkout - Release - Branch.

b)Commit - Revision - Branch.

c)Update - Revision - Merge.

d)Commit - Checkin - Hijack.

e)Update - Checkin - Merge.

Q15 - [CESPE TRE ES 2011] O CVS é uma ferramenta que implementa as principais funções do processo de controle de versão, armazenando, em um repositório, as modificações realizadas nos arquivos ao longo do tempo. O CVS, ao armazenar um arquivo em um repositório, realiza a compilação ou a interpretação da sintaxe do conteúdo desse arquivo.

Q16 - [CESPE TRE ES 2011] Quando um projeto é transferido do repositório do CVS para a área de trabalho do usuário, a edição do arquivo pode ser iniciada e, uma vez que o projeto conste no CVS, as operações de inclusão ou exclusão de arquivos na área de trabalho do usuário são reconhecidas automaticamente pelo CVS.

Q15 - [CESPE TRE ES 2011] O CVS é uma ferramenta que implementa as principais funções do processo de controle de versão, armazenando, em um repositório, as modificações realizadas nos arquivos ao longo do tempo. O CVS, ao armazenar um arquivo em um repositório, realiza a compilação ou a interpretação da sintaxe do conteúdo desse arquivo. ERRADO.

Q16 - [CESPE TRE ES 2011] Quando um projeto é transferido do repositório do CVS para a área de trabalho do usuário, a edição do arquivo pode ser iniciada e, uma vez que o projeto conste no CVS, as operações de inclusão ou exclusão de arquivos na área de trabalho do usuário são reconhecidas automaticamente pelo CVS. ERRADO.

Q17 -[CESPE TRE ES 2011] A fim de apoiar a execução do ciclo básico de trabalho de equipes de desenvolvimento de software trabalhando sob controle de versões, podem ser usadas, diretamente, as seguintes operações e(ou) ferramentas: update, para atualização das cópias de trabalho; ferramentas de edição, para modificar cópia de trabalho; operações de status, diff e history, para examinar mudanças; merge, para resolver conflitos; e commit, para concluir as mudanças.

Q18 -[CESPE TRE ES 2011] Para se armazenar em arquivos binários no CVS, é necessário informar o tipo do arquivo durante as operações de importação e adição..

Q17 -[CESPE TRE ES 2011] A fim de apoiar a execução do ciclo básico de trabalho de equipes de desenvolvimento de software trabalhando sob controle de versões, podem ser usadas, diretamente, as seguintes operações e(ou) ferramentas: update, para atualização das cópias de trabalho; ferramentas de edição, para modificar cópia de trabalho; operações de status, diff e history, para examinar mudanças; merge, para resolver conflitos; e commit, para concluir as mudanças. CERTO.

Q18 -[CESPE TRE ES 2011] Para se armazenar em arquivos binários no CVS, é necessário informar o tipo do arquivo durante as operações de importação e adição. CERTO.

Q19 -[FUNCAB PRODAM AM 2014] CVS é uma sigla para um software que tem por finalidade

- a) funcionar como ferramenta case no desenvolvimento de sistemas, além de gerar dados de teste de forma automatizada.
- b) gerenciar várias versões de documentos, além de permitir que várias pessoas possam trabalhar no mesmo arquivo.
- c) gerar código-fonte em Java de forma automática, além de produzir os diagramas de classe a partir dos requisitos do software.
- d) compilar programas codificados em Java, além de testar a sua execução por meio de uma base de dados gerada de forma aleatória
- e) facilitar a comunicação entre as pessoas da equipe de desenvolvimento de software por meio de mensagens padronizadas, além de controlar todo o processo de desenvolvimento.

Q19 -[FUNCAB PRODAM AM 2014] CVS é uma sigla para um software que tem por finalidade

a) funcionar como ferramenta case no desenvolvimento de sistemas, além de gerar dados de teste de forma automatizada.

b) gerenciar várias versões de documentos, além de permitir que várias pessoas possam trabalhar no mesmo arquivo.

c) gerar código-fonte em Java de forma automática, além de produzir os diagramas de classe a partir dos requisitos do software.

d) compilar programas codificados em Java, além de testar a sua execução por meio de uma base de dados gerada de forma aleatória

e) facilitar a comunicação entre as pessoas da equipe de desenvolvimento de software por meio de mensagens padronizadas, além de controlar todo o processo de desenvolvimento.

Q20 -[CESPE ABIN 2010] Caso o sistema de build do core engine do PHP gere arquivos com extensões .lo e .la, e o repositório padrão de software usado no desenvolvimento colaborativo desse engine seja o CVS (concurrent versioning system), então é correto que os programadores que desenvolvam módulos (extensions) para o engine coloquem, no diretório do projeto, arquivo com nome .cvsignore, contendo uma lista dos arquivos com terminações .lo e .la, os quais não devem ser copiados para o repositório. É correto, ainda, que os programadores listem outras terminações de arquivo específicas do projeto que não estejam sujeitas ao controle de versões.

Q20 -[CESPE ABIN 2010] Caso o sistema de build do core engine do PHP gere arquivos com extensões .lo e .la, e o repositório padrão de software usado no desenvolvimento colaborativo desse engine seja o CVS (concurrent versioning system), então é correto que os programadores que desenvolvam módulos (extensions) para o engine coloquem, no diretório do projeto, arquivo com nome .cvsignore, contendo uma lista dos arquivos com terminações .lo e .la, os quais não devem ser copiados para o repositório. É correto, ainda, que os programadores listem outras terminações de arquivo específicas do projeto que não estejam sujeitas ao controle de versões. CERTO.

Q21 - [CS-UFG AL-GO 2015] O responsável pela gerência de configuração de um projeto que utiliza repositório centralizado deparou-se com uma organização de arquivos que dividia os artefatos em três diretórios: trunk, branches e tags. De acordo com a terminologia utilizada em controle de versões de artefatos, esses diretórios correspondem, respectivamente, à

- a) versão de software em fase de testes, às fotografias da linha base de desenvolvimento e às versões liberadas para produção.
- b) linha principal do desenvolvimento, às cópias divergentes da linha base de desenvolvimento e às fotografias da linha base de desenvolvimento.
- c) linha principal do desenvolvimento, às fotografias da linha base de desenvolvimento e às cópias divergentes da linha base de desenvolvimento
- d) versão de software em fase de testes, às versões liberadas para produção e às cópias divergentes da linha base de desenvolvimento

Q21 - [CS-UFG AL-GO 2015] O responsável pela gerência de configuração de um projeto que utiliza repositório centralizado deparou-se com uma organização de arquivos que dividia os artefatos em três diretórios: trunk, branches e tags. De acordo com a terminologia utilizada em controle de versões de artefatos, esses diretórios correspondem, respectivamente, à

a) versão de software em fase de testes, às fotografias da linha base de desenvolvimento e às versões liberadas para produção.

b) linha principal do desenvolvimento, às cópias divergentes da linha base de desenvolvimento e às fotografias da linha base de desenvolvimento.

c) linha principal do desenvolvimento, às fotografias da linha base de desenvolvimento e às cópias divergentes da linha base de desenvolvimento

d) versão de software em fase de testes, às versões liberadas para produção e às cópias divergentes da linha base de desenvolvimento

GABARITO

Q13 - C(CERTO).

Q14 - LETRA B.

Q15 - E(ERRADO).

Q16 - E(ERRADO).

Q17 - C(CERTO).

Q18 - C(CERTO).

Q19 - LETRA B.

Q20 - C(CERTO).

Q21 - LETRA B

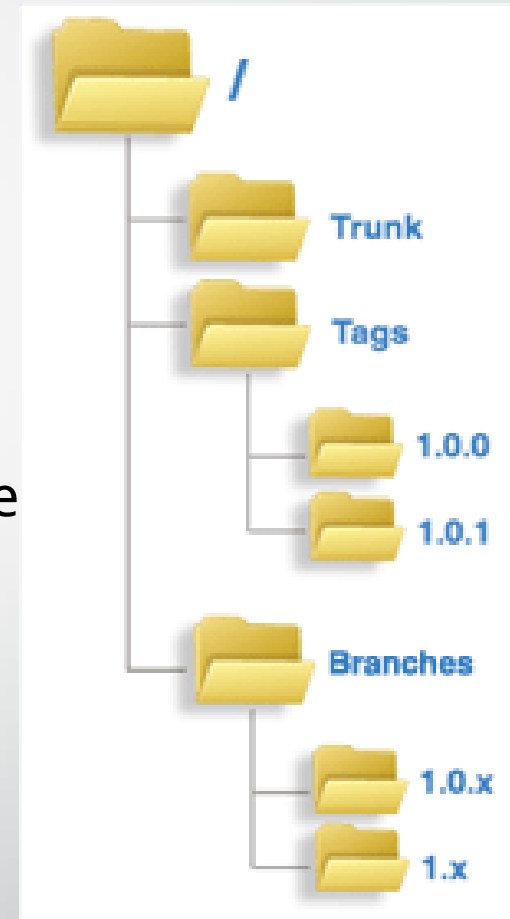
Subversion(SVN)

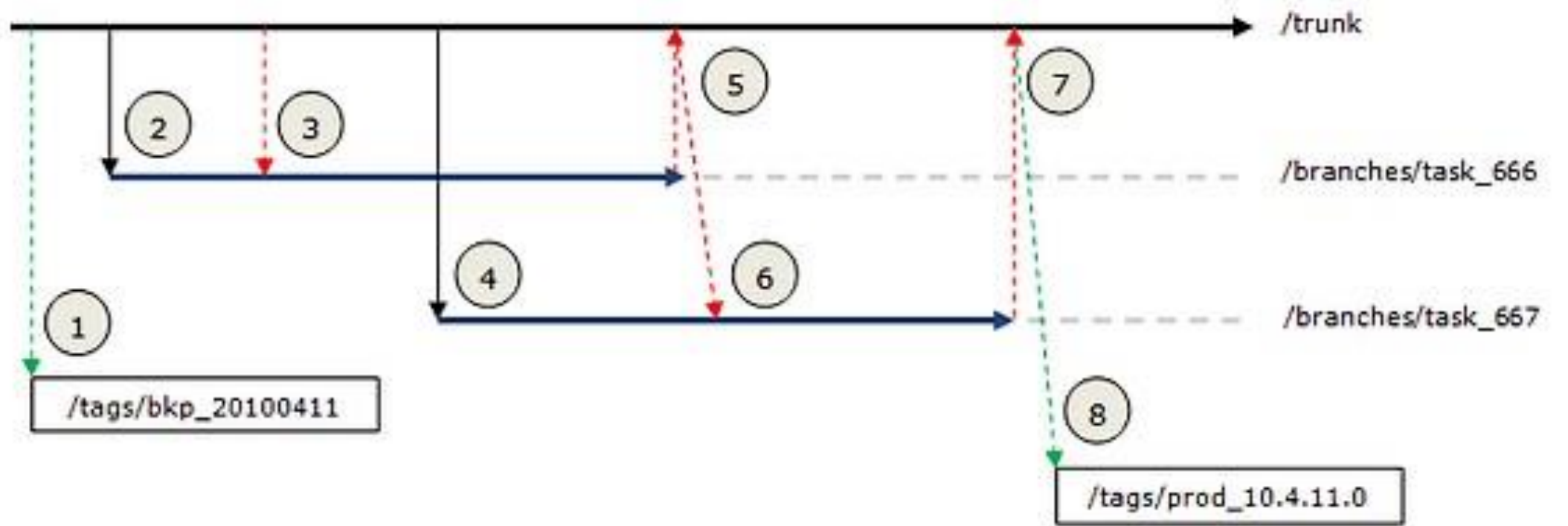
The logo for Subversion, featuring the word "SUBVERSION" in a bold, sans-serif font. The letters are blue with a slight gradient and a registered trademark symbol (®) at the end.

- Open Source.
- Sucessor do cvs, e ainda hoje é muito utilizado.
- Evoluiu em relação ao CVS:
 - **Comando rename:** Os repositórios agora podem ser renomeados diretamente de um cliente.
 - **Comando move:** Os repositórios também podem ser movidos para um outro diretório.

Estrutura

- **Trunk:** Corresponde a versão funcional mais recente do software.
- **Branch:** Implementação das alterações que serão futuramente incorporadas ao Trunk.
- **Tag:** Marcação de releases no projeto.





Comandos Básicos

SVN checkout	Atualiza o workspace com os dados do repositório, mantendo o versionamento.
SVN import	Importa uma árvore de arquivos não versionados para um repositório.
SVN export	Baixa arquivo do repositório
SVN add	Agenda arquivo, ou diretório para ser adicionado ao repositório.
SVN delete	Agenda arquivo para ser excluído do repositório. OBS: Se for um arquivo ou link, é imediatamente excluído, se for um diretório, só no momento do commit.
SVN copy	Replica determinado arquivo para outro diretório.
SVN move	Recorta determinado arquivo para outro diretório
SVN mkdir	Permite a criação de um novo diretório.

Outros comandos

SVN update	Atualiza o workspace com os dados do repositório, mantendo o versionamento.
SVN commit	Importa uma árvore de arquivos não versionados para um repositório.
SVN resolve	Comando utilizado para resolver conflitos em merge.
SVN log	Retorna informações sobre os últimos commits, como: data, autor, etc.
SVN diff	Especifica as diferenças de arquivos versionados.
SVN revert	Permite reverter alterações.
SVN list	Exibe os arquivos de um determinado repositório
SVN cleanup	Limpa os logs que o SVN cria para os arquivos.

Características

- SVN faz versionamento de todo o diretório, e não apenas arquivos
- Permite renomear e mover diretórios.
- Versionamento de metadados. Cada arquivo tem conjunto de propriedades.
- Manipulação consistente dos dados, sejam eles, arquivo texto ou binários.
- Commits atômicos: Operações realizadas em blocos, e não por arquivo individual.
- A URL de um repositório é composta: `svn checkout + [nomeRepositório]/[nomeDiretório]`. Porta pode ser especificado ou não.
 - Ex: `svn checkout http://svn.example.com.com:9834/repos.`

- Fusão contextual: atualizações baseadas em linhas.
 - Exceto, para os arquivos não textuais. Ex: **svn mime-type**.
 - Ao invés disso, o svn cria o arquivo .oldrev e .newrev.
- Locking no SVN: Utilizado na manipulação de arquivos binários.
 - SVN oferece um único algoritmo de diferenciação de três meios.
 - Suficiente para manipular dados em granularidade de uma linha de texto.
 - Permite complementar o processo de fusão de conteúdo.

Terceira Bateria de Questões



Q22 - [FCC DPE SP 2015] Um Analista deseja utilizar um sistema de controle de versão centralizado (Centralized Version Control System - CVCS) que possui um único servidor central contendo todos os arquivos versionados e vários clientes que podem resgatar os arquivos desse servidor. Ele deve utilizar, nesse caso, o

a) Subversion.

b) Darcs.

c) Bazaar.

d) Mercurial.

e) Git.

Q23 - [CESPE MEC 2015] A utilização do branching no SVN somente existe por motivo de becape.

Q22 - [FCC DPE SP 2015] Um Analista deseja utilizar um sistema de controle de versão centralizado (Centralized Version Control System - CVCS) que possui um único servidor central contendo todos os arquivos versionados e vários clientes que podem resgatar os arquivos desse servidor. Ele deve utilizar, nesse caso, o

a)Subversion.

b)Darcs.

c)Bazaar.

d)Mercurial.

e)Git.

Q23 - [CESPE MEC 2015] A utilização do branching no SVN somente existe por motivo de becape. ERRADO.

Ferramentas

- Tortoise SVN: Ferramenta gráfica para utilização do SVN em Windows.

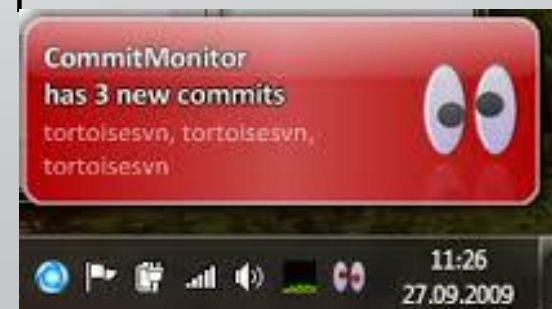


- RabbitCVS: Ferramenta gráfica para utilização do SVN em Linux.



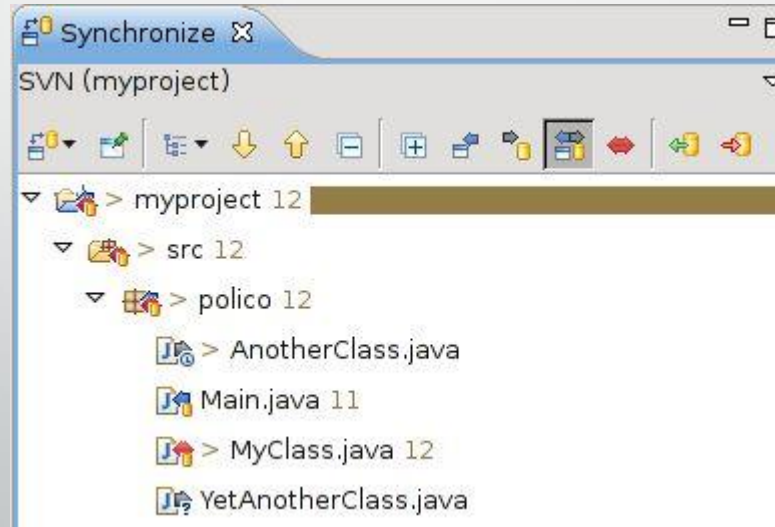
RabbitVCS

- CommitMonitor: Notifica os commits realizados no repositório SVN



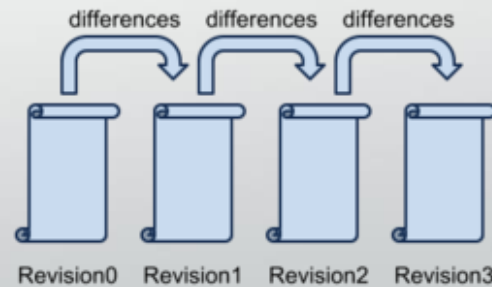
Plugin para IDEs

- Subversive
- SubEclipse



Boas Práticas

- Não sobrepor arquivos de outros desenvolvedores.
- Todo backup deve ser feito no repositório, preferencialmente numa tag
- Toda revisão deve ser comentada assim que cada commit é feito.
 - `svn commit -m "[Digite o comentário]"`.



Q24 - [CESPE MEC 2015] SVN working copy não é responsável por armazenar o histórico de mudanças completo do código.

Q25 - [CESPE MEC 2011] O Subversion é ferramenta capaz de controlar versões dos arquivos referentes ao código-fonte de um software.

Q26 - [CESPE TCU 2015] Caso um usuário, ao utilizar o software de controle de versões Subversion, tente sobrescrever uma mudança já realizada por outro e isso gere uma situação designada como conflito, o software não tentará resolver automaticamente essa situação

Q24 - [CESPE MEC 2015] SVN working copy não é responsável por armazenar o histórico de mudanças completo do código. CERTO.

Q25 - [CESPE MEC 2011] O Subversion é ferramenta capaz de controlar versões dos arquivos referentes ao código-fonte de um software. CERTO.

Q26 - [CESPE TCU 2015] Caso um usuário, ao utilizar o software de controle de versões Subversion, tente sobrescrever uma mudança já realizada por outro e isso gere uma situação designada como conflito, o software não tentará resolver automaticamente essa situação. CERTO.

Q27 - [CONSULPLAN TSE 2012] O Subversion ou simplesmente SVN é uma ferramenta de controle de versão de projeto muito poderosa que permite, além do desenvolvimento colaborativo a partir de um repositório único, merge de conteúdo, armazenamento de logs e geração de estatísticas diversas. Dentre as boas práticas, toda revisão deve ser comentada para facilitar o entendimento das alterações realizadas. Além disso, o código no diretório trunk deve sempre estar pronto para ser compilado e colocado em produção, se necessário. Nesse sentido, uma ferramenta de Integração Contínua deve ser utilizada para a geração de builds de teste a cada commit em todas as noites ao longo da semana. Uma dessas ferramentas é conhecida por

- a) TestControl.
- b) DataControl.
- c) CruiseControl.
- d) BranchControl.

Q28 - [CESPE 2014 ANTAQ] A ferramenta SVN, também denominada subversion, é mais eficiente na transmissão e no armazenamento de arquivos textos do que de arquivos binários.

Q27 - [CONSULPLAN TSE 2012] O Subversion ou simplesmente SVN é uma ferramenta de controle de versão de projeto muito poderosa que permite, além do desenvolvimento colaborativo a partir de um repositório único, merge de conteúdo, armazenamento de logs e geração de estatísticas diversas. Dentre as boas práticas, toda revisão deve ser comentada para facilitar o entendimento das alterações realizadas. Além disso, o código no diretório trunk deve sempre estar pronto para ser compilado e colocado em produção, se necessário. Nesse sentido, uma ferramenta de Integração Contínua deve ser utilizada para a geração de builds de teste a cada commit em todas as noites ao longo da semana. Uma dessas ferramentas é conhecida por

- a) TestControl.
- b) DataControl.
- c) CruiseControl.
- d) BranchControl.

Q28 - [CESPE 2014 ANTAQ] A ferramenta SVN, também denominada subversion, é mais eficiente na transmissão e no armazenamento de arquivos textos do que de arquivos binários. ERRADO.

Q29 - [CESGRANRIO BACEN 2010] Uma equipe de desenvolvimento de sistemas, que utiliza Eclipse como IDE Java, deseja utilizar o Subversion para controle de versão de seus programas. Que plugins podem ser instalados no Eclipse para que haja a integração da IDE com o repositório do Subversion?

- a)FileSync e Subclipse
- b)ProjectSet e Subclipse
- c)ProjectSet e FileSync
- d)Subversive e Subclipse
- e)Subversive e FileSync

Q30 - [CESPE STJ 2015] O Subversion, que oferece diversos algoritmos de diferenciação de três vias para manipulação de dados até uma granularidade de caracteres individuais, não permite complementar o processamento de fusão de conteúdo com utilitários de diferenciação externos.

Q29 - [CESGRANRIO BACEN 2010] Uma equipe de desenvolvimento de sistemas, que utiliza Eclipse como IDE Java, deseja utilizar o Subversion para controle de versão de seus programas. Que plugins podem ser instalados no Eclipse para que haja a integração da IDE com o repositório do Subversion?

- a)FileSync e Subclipse
- b)ProjectSet e Subclipse
- c)ProjectSet e FileSync
- d)Subversive e Subclipse
- e)Subversive e FileSync

Q30 - [CESPE STJ 2015] O Subversion, que oferece diversos algoritmos de diferenciação de três vias para manipulação de dados até uma granularidade de caracteres individuais, não permite complementar o processamento de fusão de conteúdo com utilitários de diferenciação externos. **ERRADO.**

Q31 - [CESPE MEC 2015] Os desenvolvedores devem implementar novas funcionalidades no SVN somente no código que recebeu uma tag.

Q32 - [CESPE TJDFT 2015] No Subversion 1.7, considerando-se os arquivos versionados cuja propriedade svn:mime-type seja definida com um tipo MIME não textual, suas fusões contextuais são realizadas durante as atualizações.

Q33 - [CESPE STF 2013] Para criar uma cópia de trabalho do Subversion, pode-se utilizar o comando abaixo, que carrega uma cópia de trabalho para o diretório local do projeto a ser trabalhado que esteja compartilhado no repositório. `svn checkout svn://repositorio/diretorio diretorio_local`

Q31 - [CESPE MEC 2015] Os desenvolvedores devem implementar novas funcionalidades no SVN somente no código que recebeu uma tag. ERRADO.

Q32 - [CESPE TJDFT 2015] No Subversion 1.7, considerando-se os arquivos versionados cuja propriedade svn:mime-type seja definida com um tipo MIME não textual, suas fusões contextuais são realizadas durante as atualizações. ERRADO.

Q33 - [CESPE STF 2013] Para criar uma cópia de trabalho do Subversion, pode-se utilizar o comando abaixo, que carrega uma cópia de trabalho para o diretório local do projeto a ser trabalhado que esteja compartilhado no repositório. `svn checkout svn://repositorio/diretorio diretorio_local`. CERTO.

GABARITO

Q22 - LETRA A.

Q23 - E(ERRADO).

Q24 - C(CERTO).

Q25 - C(CERTO).

Q26 - C(CERTO).

Q27 - LETRA C.

Q28 - C(CERTO).

Q29 - LETRA D.

Q30 - E(ERRADO).

Q31 - E(ERRADO).

Q32 - E(ERRADO).

Q33 - C(CERTO).

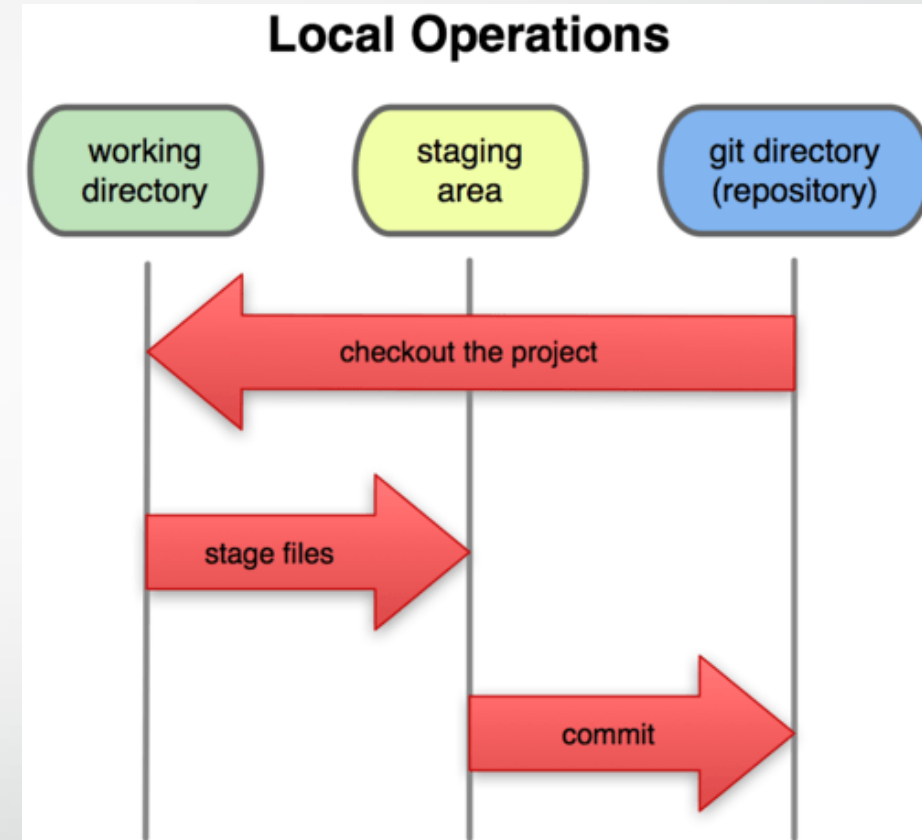
GIT



- Ferramenta de repositório distribuído.
- Utiliza o hash SHA-1 para prover integridade nas revisões.
- Quase todas operações são locais.
- Suporte robusto a desenvolvimento não linear (muitos branches paralelos).
- Originou como ferramenta de controle do Kernel do Linux.

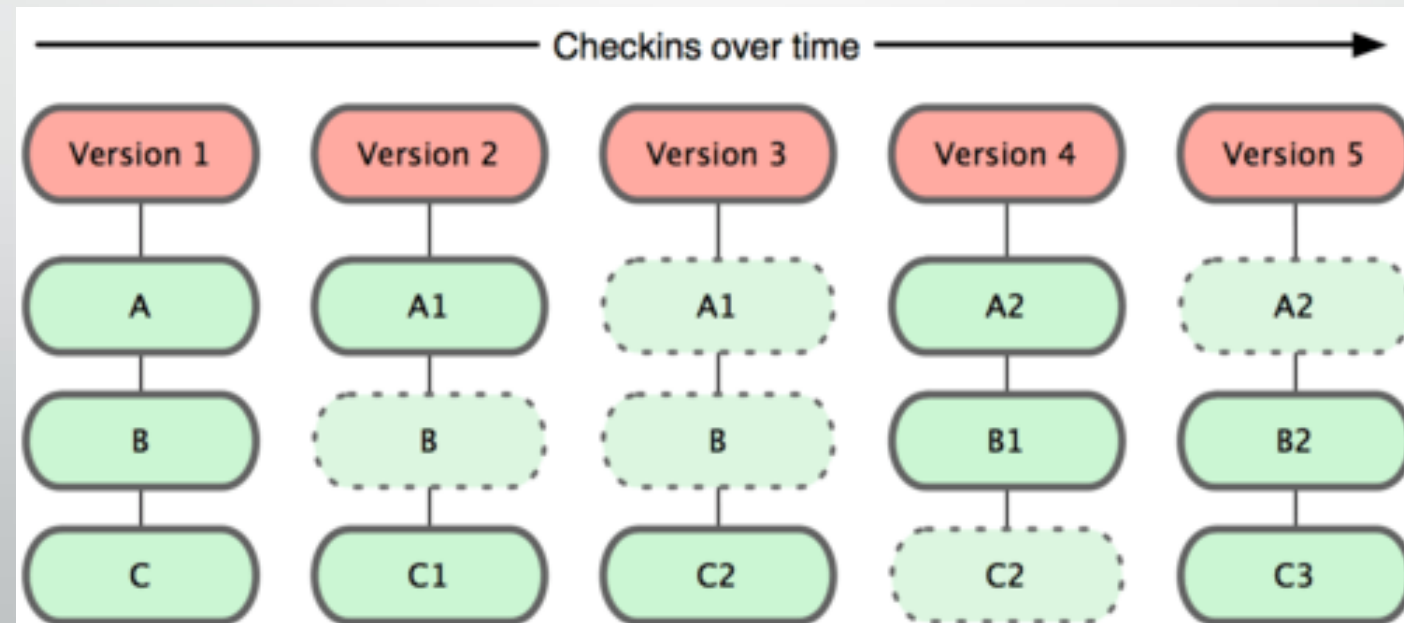
Estados

- **Consolidado**(committed) : Dados estão armazenados na base local.
- **Modificado**(modified) : Arquivo que sofreu mudanças e não foi armazenado na base local.
- **Preparado**(staged) : Quando marca um arquivo modificado para que faça parte do próximo commit.



Modo de Operação

- Diferente dos outros SCVS, o GIT não trabalha com lista de mudanças de um determinado arquivo ao longo do tempo.
- O GIT opera sempre realizando snapshots a cada vez que um commit é feito.
- Se um arquivo não foi alterado, o GIT cria apenas um link da revisão anterior.



Quarta Bateria de Questões



Q34 - [FCC MPE PB 2015] O GIT possibilita controlar versões de componentes de software com as facilidades de

- a) armazenar alterações de fontes de programas como uma lista de mudanças por arquivo, registrando portanto um conjunto de arquivos e as mudanças feitas a cada arquivo ao longo do tempo.
- b) armazenar alterações de fontes de programas em um único arquivo de registro de alterações subsequentes que funciona como um LOG de modificações de fontes que pode ser aplicado para desfazê-las a qualquer tempo.
- c) replicar cópias de cada componente quando da sua modificação, atualizando repositórios de arquivos locais nas estações de trabalho de cada desenvolvedor catalogado como membro do projeto de software em questão.
- d) navegar no histórico das modificações, consultando diretamente as cópias dos arquivos físicos de cada versão, a partir de um repositório localizado no servidor de programas fonte.
- e) navegar no histórico das versões, consultando snapshots referentes aos arquivos físicos do servidor de fontes que são guardados em um banco de dados local da máquina do desenvolvedor.

Q34 - [FCC MPE PB 2015] O GIT possibilita controlar versões de componentes de software com as facilidades de

- a) armazenar alterações de fontes de programas como uma lista de mudanças por arquivo, registrando portanto um conjunto de arquivos e as mudanças feitas a cada arquivo ao longo do tempo.
- b) armazenar alterações de fontes de programas em um único arquivo de registro de alterações subsequentes que funciona como um LOG de modificações de fontes que pode ser aplicado para desfazê-las a qualquer tempo.
- c) replicar cópias de cada componente quando da sua modificação, atualizando repositórios de arquivos locais nas estações de trabalho de cada desenvolvedor catalogado como membro do projeto de software em questão.
- d) navegar no histórico das modificações, consultando diretamente as cópias dos arquivos físicos de cada versão, a partir de um repositório localizado no servidor de programas fonte.
- e) navegar no histórico das versões, consultando snapshots referentes aos arquivos físicos do servidor de fontes que são guardados em um banco de dados local da máquina do desenvolvedor.

Q35 - [CESPE TRE RS 2015] Assinale a opção que apresenta a sequência correta de atividades do fluxo básico de um processo de controle de versões por meio do Git.

- a) 1. Modificar os arquivos no diretório de trabalho; 2. selecionar os arquivos e adicionar seus snapshots na área de preparação; 3. fazer um commit, que levará os arquivos como estão na área de preparação e os armazenará permanentemente no diretório Git.
- b) 1. Selecionar os arquivos e adicionar seus snapshots na área de preparação; 2. modificar os arquivos no diretório de trabalho; 3. fazer um commit, que levará os arquivos como estão na área de preparação e os armazenará permanentemente no diretório Git.
- c) 1. Modificar os arquivos no diretório de trabalho; 2. selecionar os arquivos e adicionar seus snapshots na área de preparação; 3. armazenar esses arquivos permanentemente no diretório Git; 4. fazer um commit, que levará os arquivos como estão na área de preparação.
- d) 1. Fazer um commit, que levará os arquivos como estão na versão inicial da área de preparação; 2. modificar os arquivos no diretório de trabalho; 3. selecionar os arquivos e adicionar seus snapshots na área de preparação; 4. armazenar esses arquivos permanentemente no diretório Git.
- e) 1. Modificar os arquivos no diretório de trabalho; 2. fazer um commit, que levará os arquivos como estão na área de preparação e os armazenará permanentemente no diretório Git; 3. selecionar os arquivos e adicionar seus snapshots na área de preparação

Q35 - [CESPE TRE RS 2015] Assinale a opção que apresenta a sequência correta de atividades do fluxo básico de um processo de controle de versões por meio do Git.

a) 1. Modificar os arquivos no diretório de trabalho; 2. selecionar os arquivos e adicionar seus snapshots na área de preparação; 3. fazer um commit, que levará os arquivos como estão na área de preparação e os armazenará permanentemente no diretório Git.

b) 1. Selecionar os arquivos e adicionar seus snapshots na área de preparação; 2. modificar os arquivos no diretório de trabalho; 3. fazer um commit, que levará os arquivos como estão na área de preparação e os armazenará permanentemente no diretório Git.

c) 1. Modificar os arquivos no diretório de trabalho; 2. selecionar os arquivos e adicionar seus snapshots na área de preparação; 3. armazenar esses arquivos permanentemente no diretório Git; 4. fazer um commit, que levará os arquivos como estão na área de preparação.

d) 1. Fazer um commit, que levará os arquivos como estão na versão inicial da área de preparação; 2. modificar os arquivos no diretório de trabalho; 3. selecionar os arquivos e adicionar seus snapshots na área de preparação; 4. armazenar esses arquivos permanentemente no diretório Git.

e) 1. Modificar os arquivos no diretório de trabalho; 2. fazer um commit, que levará os arquivos como estão na área de preparação e os armazenará permanentemente no diretório Git; 3. selecionar os arquivos e adicionar seus snapshots na área de preparação

Q36 - [FCC TRT3 2015] Um técnico de TI, utilizando o Git, marcou um arquivo modificado em sua versão corrente para que ele faça parte do snapshot do próximo commit.

Esta ação colocou o arquivo no estado

- a) preparado.
- b) modificado.
- c) consolidado.
- d) persistido.
- e) depreciado.

Q37 - [CESPE PF 2013] O SHA-1, comumente usado em protocolos de segurança, como TLS, SSH e IPSec, também é utilizado por alguns sistemas de controle de versão como Git e Mercurial para garantir a integridade das revisões.

Q36 - [FCC TRT3 2015] Um técnico de TI, utilizando o Git, marcou um arquivo modificado em sua versão corrente para que ele faça parte do snapshot do próximo commit.

Esta ação colocou o arquivo no estado

- a) preparado.
- b) modificado.
- c) consolidado.
- d) persistido.
- e) depreciado.

Q37 - [CESPE PF 2013] O SHA-1, comumente usado em protocolos de segurança, como TLS, SSH e IPSec, também é utilizado por alguns sistemas de controle de versão como Git e Mercurial para garantir a integridade das revisões. CERTO.

Q38 - [FCC TRT13 2014] Paulo está executando o Git no Linux. Ele tem um repositório Git e um checkout ou cópia funcional dos arquivos para o projeto atual. Cada arquivo, no diretório de trabalho de Paulo, pode estar em um de dois estados: monitorado ou não monitorado. Arquivos monitorados são arquivos que estavam no último snapshot; podendo estar inalterados, modificados ou selecionados. Arquivos não monitorados são os restantes.

Para Paulo verificar, em linha de comando, quais arquivos estão em quais estados ele utilizou o comando `git status`. Em seguida, ele adicionou um novo arquivo chamado `trt` ao projeto.

a) `git add trt`

b) `git -a --sx trt`

c) `git state trt`

d) `git fetch trt`

e) `git -mu trt`

Q38 - [FCC TRT13 2014] Paulo está executando o Git no Linux. Ele tem um repositório Git e um checkout ou cópia funcional dos arquivos para o projeto atual. Cada arquivo, no diretório de trabalho de Paulo, pode estar em um de dois estados: monitorado ou não monitorado. Arquivos monitorados são arquivos que estavam no último snapshot; podendo estar inalterados, modificados ou selecionados. Arquivos não monitorados são os restantes.

Para Paulo verificar, em linha de comando, quais arquivos estão em quais estados ele utilizou o comando `git status`. Em seguida, ele adicionou um novo arquivo chamado `trt` ao projeto.

a) `git add trt`

b) `git -a --sx trt`

c) `git state trt`

d) `git fetch trt`

e) `git -mu trt`

Configuração

- Especificar nome e email do usuário do git:
 - `Git config --global user.name [""]`
 - `Git config --global user.email [""]`
- Especificar ferramenta padrão para resolver conflitos:
 - `Git config --global merge.tool [nameFerramenta]`.
- Alterando editor padrão do git:
 - `Git config --global core.editor [name]`

Comandos Básicos

git status	Apresenta situação dos arquivos no repositório.
git log	Apresenta histórico detalhado de commits. git log -p : mostra o diff em cada commit git log -2 : mostra os 2 últimos commit. git log -stat : resume as informações dos commits.
git clone	Permite fazer o primeiro download de um repositório.
git add	Permite monitorar arquivos. E deixa-lo a ponto de ser comitado. Para adicionar todos arquivos: git add .
git rm	Remove arquivos.
git checkout	Desfaz commits, mudar arquivos de branches.
git fetch	Obtém download de arquivos do repositório sem o merge.
git pull	Fetch + Merge.

Outras informações

- Por padrão, as verificações de integridade, não são verificadas a cada Push.
 - Porém, pode ser definido através: `git config --system receive.fsckObjects [true/false]`.
- Problemas de espaços em Branco: Git traz 4 soluções como:
 - Trailing-space: procura por espaços no final da linha.
- Problemas de quebra de linha no Windows:
 - `Git config --global core.autocrlf [true/false]`.

Github



- Repositório remoto mais popular do git.
- **Git remote add [nomeServidor]:** Adiciona novos repositórios remotos.
- **Git remote:** Permite listar os servidores remotos configurados no git.
 - **Origin** é o nome padrão para se referir ao servidor remoto do git.
 - **Git remote – v:** Além de mostrar o nome, mostra também url do servidor.

Q39 - [FCC TRT2 2014] No sistema de controle de versões Git, para efetuar o download dos commits de um repositório remoto para o repositório local é utilizado o comando git

a)apply.

b)get.

c)push.

d)fetch.

e)update.

Q40 - [CESPE STJ 2015] O Git, sistema de controle de versões que mantém um histórico completo de todas as alterações, permite a recuperação das versões do projeto na busca de informações sobre o estado dos arquivos em versões anteriores.

Q39 - [FCC TRT2 2014] No sistema de controle de versões Git, para efetuar o download dos commits de um repositório remoto para o repositório local é utilizado o comando git

a)apply.

b)get.

c)push.

d)fetch.

e)update.

Q40- [CESPE STJ 2015] O Git, sistema de controle de versões que mantém um histórico completo de todas as alterações, permite a recuperação das versões do projeto na busca de informações sobre o estado dos arquivos em versões anteriores. CERTO.

Q41 - [CESPE TRE RS 2015] A respeito da customização do uso Git, assinale a opção correta.

- a) O Git, por padrão, verifica a consistência de todos os objetos que ele recebe durante um push.
- b) O Git, ao permitir configuração personalizada unicamente no lado cliente, por segurança, bloqueia o lado servidor para a execução de comandos remotos.
- c) O Git realiza a detecção e a correção de problemas relacionados a formatação e espaços em branco, tratando-se, entretanto, de um pacote adicional que deve ser ativado pelo usuário.
- d) O Git usa, por padrão, o editor de texto definido no shell, opção que pode ser alterada pelo comando `git config --global core.editor nomeedoeditor`.
- e) Não se pode configurar uma ferramenta externa gráfica de merge e diff para a resolução de conflitos.

Q41 - [CESPE TRE RS 2015] A respeito da customização do uso Git, assinale a opção correta.

a) O Git, por padrão, verifica a consistência de todos os objetos que ele recebe durante um push.

b) O Git, ao permitir configuração personalizada unicamente no lado cliente, por segurança, bloqueia o lado servidor para a execução de comandos remotos.

c) O Git realiza a detecção e a correção de problemas relacionados a formatação e espaços em branco, tratando-se, entretanto, de um pacote adicional que deve ser ativado pelo usuário.

d) O Git usa, por padrão, o editor de texto definido no shell, opção que pode ser alterada pelo comando `git config --global core.editor nomeedoeditor`.

e) Não se pode configurar uma ferramenta externa gráfica de merge e diff para a resolução de conflitos.

Q43 - [CESPE TRE RS 2015] Acerca do controle de versão com Git, é correto afirmar que

- a) o uso do protocolo HTTP para a transferência de dados é recomendado em projetos grandes com permissão de acesso público de leitura anônimo, por ser o mais rápido entre os disponíveis.
- b) o comando `git add` muda, automaticamente, o estado do arquivo para monitorado, não sendo válido para diretórios.
- c) a execução do comando `git log` com os argumentos `-p -2` permite obter o diff introduzido em cada commit, limitando a saída somente às duas últimas entradas.
- d) o comando `git fetch [nome-remoto]` traz os dados do repositório remoto para seu repositório local, realizando o merge automaticamente.
- e) o servidor central é um ponto único de falha.

Q43 - [CESPE TRE RS 2015] Acerca do controle de versão com Git, é correto afirmar que

- a) o uso do protocolo HTTP para a transferência de dados é recomendado em projetos grandes com permissão de acesso público de leitura anônimo, por ser o mais rápido entre os disponíveis.
- b) o comando `git add` muda, automaticamente, o estado do arquivo para monitorado, não sendo válido para diretórios.
- c) a execução do comando `git log` com os argumentos `-p -2` permite obter o diff introduzido em cada commit, limitando a saída somente às duas últimas entradas.
- d) o comando `git fetch [nome-remoto]` traz os dados do repositório remoto para seu repositório local, realizando o merge automaticamente.
- e) o servidor central é um ponto único de falha.

Q44 - [IFPE 2016] Modificações são inevitáveis quando um software é construído. Um objetivo primordial da Engenharia de Software é otimizar a forma como as modificações podem ser acomodadas e reduzir a quantidade de esforço despendido quando elas estiverem sendo feitas. Uma das ferramentas existentes para dar suporte ao processo de Gestão de Modificações é a ferramenta GIT. A cerca dessa ferramenta e considerando o comando abaixo, assinale a alternativa **CORRETA**.

`git push -u origin master`

- a) O comando push não existe na ferramenta GIT.
- b) O comando acima cria uma cópia de trabalho/repositório na máquina onde esse comando foi executado.
- c) O comando acima envia alterações para o repositório local, criando uma revisão.
- d) O comando acima envia cópia para o repositório local das alterações feitas em um repositório remoto.
- e) O comando acima envia revisões locais para outro repositório remoto.

Q44 - [IFPE 2016] Modificações são inevitáveis quando um software é construído. Um objetivo primordial da Engenharia de Software é otimizar a forma como as modificações podem ser acomodadas e reduzir a quantidade de esforço despendido quando elas estiverem sendo feitas. Uma das ferramentas existentes para dar suporte ao processo de Gestão de Modificações é a ferramenta GIT. A cerca dessa ferramenta e considerando o comando abaixo, assinale a alternativa **CORRETA**.

`git push -u origin master`

- a) O comando push não existe na ferramenta GIT.
- b) O comando acima cria uma cópia de trabalho/repositório na máquina onde esse comando foi executado.
- c) O comando acima envia alterações para o repositório local, criando uma revisão.
- d) O comando acima envia cópia para o repositório local das alterações feitas em um repositório remoto.
- e) O comando acima envia revisões locais para outro repositório remoto.

Q45 - [FCC TRT3 2015] Após o analista de TI alterar no Git um arquivo previamente monitorado chamado trt.py e após ter passado a monitorar um novo arquivo chamado Tribunal, executou-se o comando git status e foi exibido:

```
# On branch master
```

```
# Changes to be committed:
```

```
#   (use "git reset HEAD ..." to unstage)
```

```
#
```

```
#   new file:   Tribunal
```

```
#
```

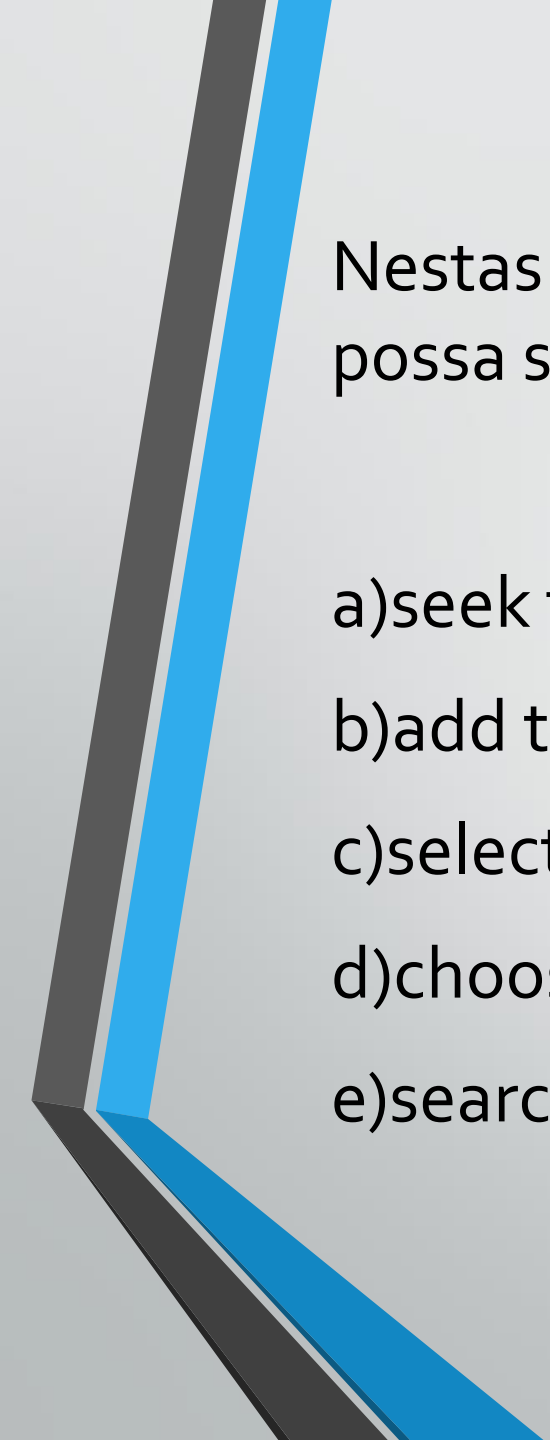
```
# Changes not staged for commit:
```

```
#   (use "git add ..." to update what will be committed)
```

```
#
```

```
#   modified:   teste.py
```

```
#
```



Nestas condições, para seleccionar o arquivo trt.py de forma que ele possa ser consolidado no próximo commit, utiliza-se o comando git

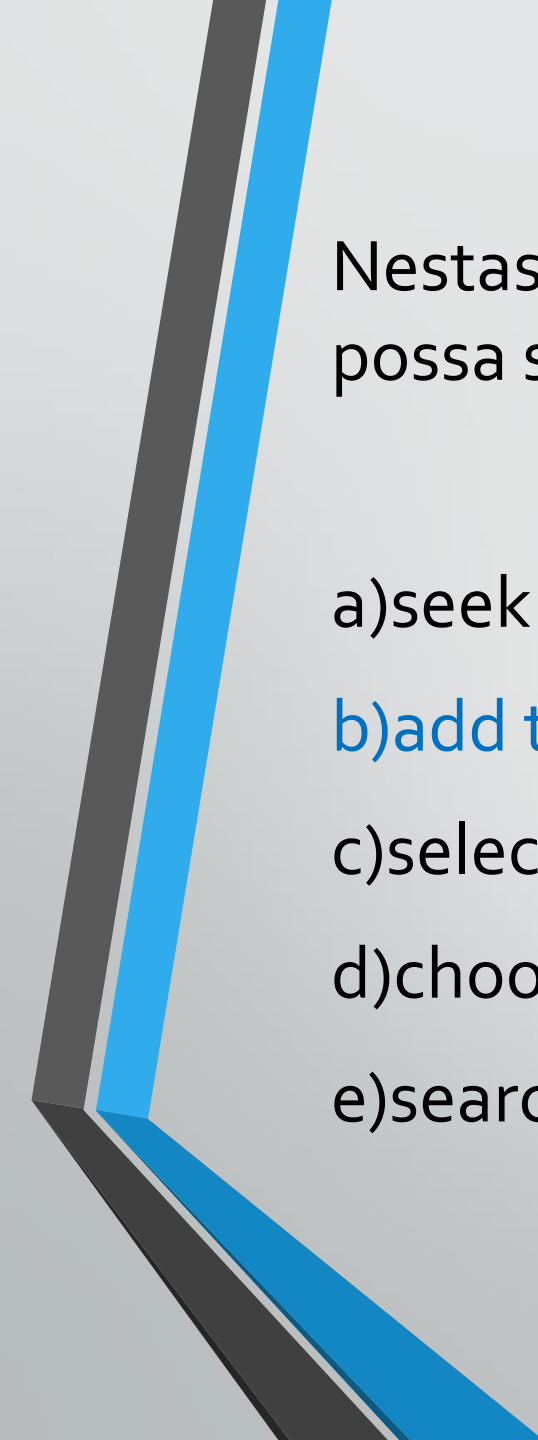
a)seek trt.py

b)add trt.py

c)select trt.py

d)choose trt.py

e)search trt.py



Nestas condições, para seleccionar o arquivo trt.py de forma que ele possa ser consolidado no próximo commit, utiliza-se o comando git

a)seek trt.py

b)add trt.py

c)select trt.py

d)choose trt.py

e)search trt.py

GABARITO

Q34 - LETRA E

Q35 - LETRA A

Q36 - LETRA A

Q37 - C(CERTO)

Q38 - LETRA A

Q39 - LETRA D.

Q40 - C(CERTO)

Q41 - LETRA D.

Q42 - REMOVIDA

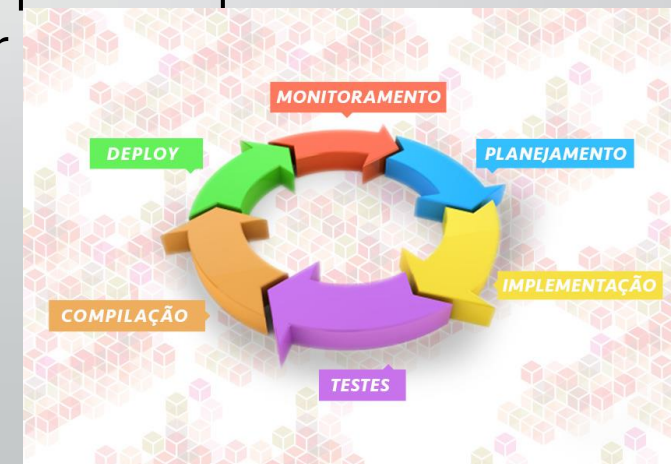
Q43 - LETRA C.

Q44 - LETRA E.

Q45 - LETRA B.

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

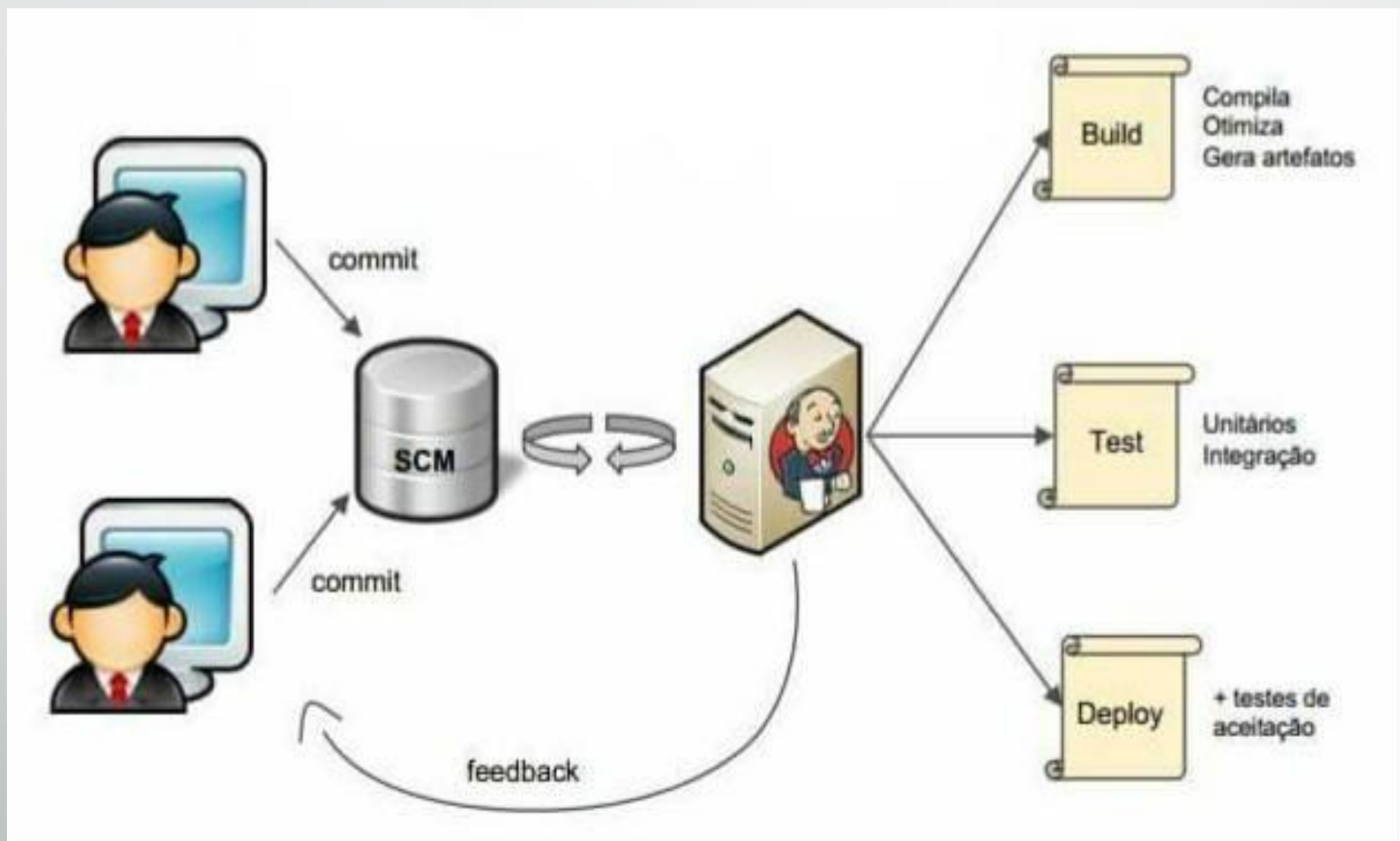
- Utilizar alguma ferramenta para controle de versão.
- Utilização de ferramentas para criação de testes automáticos(TDD).
- Utilização de ferramentas que venham prover o processo de build(compila, roda os testes, etc) automático



Vantagens

- Feedback instantâneo.
- Caso houve alguma 'quebra de build' a ferramenta irá notificar a equipe de desenvolvimento
- O projeto só entra em produção se passar pelos testes.





Quinta Bateria de Questões



Q46 - [CESPE ANAC 2012] A integração contínua é uma técnica de alto impacto no levantamento de requisitos.

Q47 - [FCC TJPI 2009] XP (eXtreme Programming) é uma metodologia ágil para equipes pequenas e médias que desenvolverão software com requisitos vagos e em constante mudança. Para isso, adota a estratégia de constante acompanhamento e realização de vários pequenos ajustes durante o desenvolvimento de software. Para aplicar os valores e princípios durante o desenvolvimento de software, a XP propõe uma série de práticas, sendo uma delas: sempre que produzir uma nova funcionalidade, nunca esperar uma semana para integrar à versão atual do sistema a fim de evitar o aumento da possibilidade de conflitos e da possibilidade de erros no código fonte. Tal prática é denominada

- a) Time Coeso.
- b) Refatoração.
- c) Integração Contínua.
- d) Desenvolvimento Orientado a Testes.
- e) Ritmo Sustentável.

Q46 - [CESPE ANAC 2012] A integração contínua é uma técnica de alto impacto no levantamento de requisitos. ERRADO.

Q47 - [FCC TJPI 2009] XP (eXtreme Programming) é uma metodologia ágil para equipes pequenas e médias que desenvolverão software com requisitos vagos e em constante mudança. Para isso, adota a estratégia de constante acompanhamento e realização de vários pequenos ajustes durante o desenvolvimento de software. Para aplicar os valores e princípios durante o desenvolvimento de software, a XP propõe uma série de práticas, sendo uma delas: sempre que produzir uma nova funcionalidade, nunca esperar uma semana para integrar à versão atual do sistema a fim de evitar o aumento da possibilidade de conflitos e da possibilidade de erros no código fonte. Tal prática é denominada

- a) Time Coeso.
- b) Refatoração.
- c) Integração Contínua.
- d) Desenvolvimento Orientado a Testes.
- e) Ritmo Sustentável.

Q48 - [CESPE ANATEL 2014] A integração contínua constitui um dos pilares das práticas ágeis: a constante integração das respectivas partes deve ocorrer diariamente e ser realizada concomitantemente à atividade de programação em pares.

Q49 - [CESPE STF 2013] Integração contínua, entrega contínua, teste contínuo, monitoramento contínuo e feedback são algumas práticas do DevOps.

Q50 - [CESPE STJ 2015] A finalidade das ferramentas de integração contínua é a criação de soluções integradas com foco em sistemas fortemente acoplados, com necessidade de criação de uma documentação contínua.

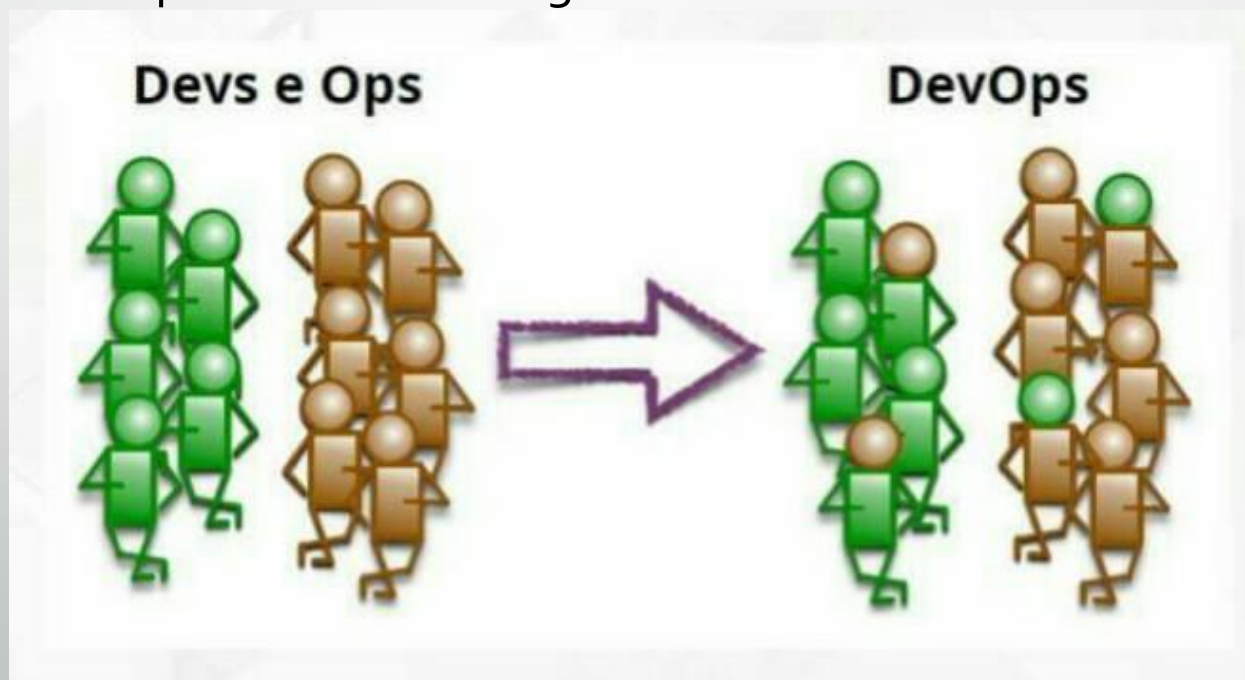
Q48 - [CESPE ANATEL 2014] A integração contínua constitui um dos pilares das práticas ágeis: a constante integração das respectivas partes deve ocorrer diariamente e ser realizada concomitantemente à atividade de programação em pares. **ERRADO.**

Q49 - [CESPE STF 2013] Integração contínua, entrega contínua, teste contínuo, monitoramento contínuo e feedback são algumas práticas do DevOps. **CERTO.**

Q50 - [CESPE STJ 2015] A finalidade das ferramentas de integração contínua é a criação de soluções integradas com foco em sistemas fortemente acoplados, com necessidade de criação de uma documentação contínua. **ERRADO.**

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.



Melhoria Contínua



Q51 - [CESPE TRE RS 2015] Quanto às técnicas de entrega contínua e de integração contínua, é correto afirmar que

- a) o objetivo da integração contínua é manter o software em um estado funcional ininterrupto.
- b) é evitada a compilação do código da aplicação a cada mudança realizada, na integração contínua.
- c) as duas técnicas são independentes, embora a técnica de integração contínua preceda a de entrega contínua.
- d) a integração contínua, caso o processo de compilação esteja quebrado, logo identifica a causa do problema e a corrige.
- e) é recomendável um sistema de controle de versão centralizado, quando se implementa a integração contínua.

Q51 - [CESPE TRE RS 2015] Quanto às técnicas de entrega contínua e de integração contínua, é correto afirmar que

- a) o objetivo da integração contínua é manter o software em um estado funcional ininterrupto.
- b) é evitada a compilação do código da aplicação a cada mudança realizada, na integração contínua.
- c) as duas técnicas são independentes, embora a técnica de integração contínua preceda a de entrega contínua.
- d) a integração contínua, caso o processo de compilação esteja quebrado, logo identifica a causa do problema e a corrige.
- e) é recomendável um sistema de controle de versão centralizado, quando se implementa a integração contínua.

Q52 - [CESPE TCU 2015] Para que a prática de integração contínua seja eficiente, é necessário parametrizar e automatizar várias atividades relativas à gerência da configuração, não somente do código-fonte produzido, mas também de bibliotecas e componentes externos.

Q53 - [CESPE STF 2013] São características de um servidor de integração contínua: compilação do código-fonte, realização de operações de checkout do gerenciador de configuração, execução de um conjunto de testes, auditoria do código-fonte e testes funcionais.

Q54 - [CESPE ANATEL 2014] Por não existir relação direta entre as práticas ágeis de integração contínua e de entrega contínua, pode-se garantir a agilidade do processo pela realização de tais atividades em paralelo, ainda que relacionadas à mesma versão.

Q52 - [CESPE TCU 2015] Para que a prática de integração contínua seja eficiente, é necessário parametrizar e automatizar várias atividades relativas à gerência da configuração, não somente do código-fonte produzido, mas também de bibliotecas e componentes externos. CERTO.

Q53 - [CESPE STF 2013] São características de um servidor de integração contínua: compilação do código-fonte, realização de operações de checkout do gerenciador de configuração, execução de um conjunto de testes, auditoria do código-fonte e testes funcionais. CERTO.

Q54 - [CESPE ANATEL 2014] Por não existir relação direta entre as práticas ágeis de integração contínua e de entrega contínua, pode-se garantir a agilidade do processo pela realização de tais atividades em paralelo, ainda que relacionadas à mesma versão. ERRADO.

Q55 - [ESAF 2015] Entre as melhores práticas da Integração Contínua citadas por Martin Fowler, no seu famoso artigo "Continuous Integration", não se encontra.

- a) automatize o processo de Build.
- b) cada modificação salva deve gerar um Build automaticamente.
- c) execute os testes finais no ambiente de produção.
- d) garanta que o Build seja rápido.
- e) mantenha um repositório de fontes unificado.

Q55 - [ESAF 2015] Entre as melhores práticas da Integração Contínua citadas por Martin Fowler, no seu famoso artigo "Continuous Integration", não se encontra.

- a) automatize o processo de Build.
- b) cada modificação salva deve gerar um Build automaticamente.
- c) execute os testes finais no ambiente de produção.
- d) garanta que o Build seja rápido.
- e) mantenha um repositório de fontes unificado.

Q56- [IADES EBSE RH 2013] Tomando por base a figura anterior, assinale a alternativa que define, corretamente, cada nível do Gerenciamento de Configuração de Software.

a) Controle de versão: identifica em qual versão está o software e quais as características de cada versão. Controle de mudanças: identifica quais foram as mudanças, efetuadas na versão. Integração contínua: tem como características testar as mudanças, assim que são realizadas.

b) Controle de versão: identifica em qual versão está o software e quais as características de cada versão. Controle de mudanças: tem como característica testar as mudanças, assim que são realizadas. Integração contínua: identifica quais foram as mudanças, efetuadas na versão.

c) Controle de versão: identifica quais foram as mudanças, efetuadas na versão. Controle de mudanças: tem como característica testar as mudanças, assim que são realizadas. Integração contínua: identifica em qual versão está o software e quais as características de cada versão.

d) Controle de versão: identifica os usuários dos sistemas. Controle de mudanças: tem como característica testar as mudanças, assim que são realizadas. Integração contínua: identifica em qual versão está o software e quais as características de cada versão.

e) Controle de versão: identifica os usuários dos sistemas. Controle de mudanças: tem como característica testar as mudanças, assim que são realizadas. Integração contínua: identifica as aplicações do software.

Q56- [IADES EBSE RH 2013] Tomando por base a figura anterior, assinale a alternativa que define, corretamente, cada nível do Gerenciamento de Configuração de Software.

a) Controle de versão: identifica em qual versão está o software e quais as características de cada versão. Controle de mudanças: identifica quais foram as mudanças, efetuadas na versão. Integração contínua: tem como características testar as mudanças, assim que são realizadas.

b) Controle de versão: identifica em qual versão está o software e quais as características de cada versão. Controle de mudanças: tem como característica testar as mudanças, assim que são realizadas. Integração contínua: identifica quais foram as mudanças, efetuadas na versão.

c) Controle de versão: identifica quais foram as mudanças, efetuadas na versão. Controle de mudanças: tem como característica testar as mudanças, assim que são realizadas. Integração contínua: identifica em qual versão está o software e quais as características de cada versão.

d) Controle de versão: identifica os usuários dos sistemas. Controle de mudanças: tem como característica testar as mudanças, assim que são realizadas. Integração contínua: identifica em qual versão está o software e quais as características de cada versão.

e) Controle de versão: identifica os usuários dos sistemas. Controle de mudanças: tem como característica testar as mudanças, assim que são realizadas. Integração contínua: identifica as aplicações do software.

GABARITO

Q46 - E(ERRADO).

Q47 - LETRA C.

Q48 - E(ERRADO).

Q49 - C(CERTO).

Q50 - E(ERRADO).

Q51 - LETRA A.

Q52 - C(CERTO).

Q53 - C(CERTO).

Q54 - E(ERRADO).

Q55 - LETRA C.

Q56 - LETRA A.

Obrigado!!

