

Questão 1

Parcialmente correto

Atingiu 0,2 de 1,0

Marcar questão

O Git pronunciado é um sistema de controle de versões, usado principalmente no desenvolvimento de software, mas pode ser usado para registrar o histórico de edições de qualquer tipo de arquivo. O Git foi inicialmente projetado e desenvolvido por Linus Torvalds para o desenvolvimento do kernel Linux, mas foi adotado por muitos outros projetos.

Quando o criou, Torvalds queria um sistema em que ele pudesse utilizar de forma similar ao BitKeeper, mas nenhum dos sistemas gratuitos disponíveis atendia suas necessidades, particularmente com relação à performance. fonte: (https://pt.wikipedia.org/wiki/Git)

A utilização do git pode ser realizada tanto por terminal (linha de comando) ou através de ferramentas visuais.

Se tratando dos comandos possíveis através do terminal, relacione a lista de comandos abaixo aos seus conceitos.

busca as atualizações do repositório remoto para o repositório local	<div><div>pull</div><div>✖</div></div>
troca de ramo	<div><div>checkout</div><div>✖</div></div>
envia as modificações realizadas para o repositório local	<div><div>commit</div><div>✖</div></div>
Inicializa o repositório	<div><div>init</div><div>✔</div></div>
Copia o repositório remoto para a workstation	<div><div>clone</div><div>✔</div></div>
apresenta as gravações realizadas no repositório	<div><div>log</div><div>✖</div></div>
Cria um novo ramo	<div><div>branch</div><div>✖</div></div>
envia as modificações para o status que permite que sejam controladas pelo git e que possam, posteriormente, serem armazenadas.	<div><div>add</div><div>✖</div></div>
verifica se há modificações no repositório que ainda não foram armazenadas	<div><div>status</div><div>✖</div></div>

Sua resposta está parcialmente correta.

Você selecionou corretamente 2.

A resposta correta é: busca as atualizações do repositório remoto para o repositório local - pull, troca de ramo - checkout, envia as modificações realizadas para o repositório local - commit, Inicializa o repositório - init, Copia o repositório remoto para a workstation - clone, apresenta as gravações realizadas no repositório - log, Cria um novo ramo - branch, envia as modificações para o status que permite que sejam controladas pelo git e que possam, posteriormente, serem armazenadas. - add, verifica se há modificações no repositório que ainda não foram armazenadas - status.

Questão 2

Correto

Atingiu 1,0 de 1,0

Marcar questão

Os sistemas de controle de versão que permitem que mais de uma pessoa manipule o mesmo arquivo, ao mesmo tempo, geralmente permitem a realização do processo de merge. No Git, o merge pode ocorrer automaticamente em muitos casos, mas terá que ser feito manualmente quando houver um conflito. Em qual situação o merge precisa ser manual?

Escolha uma:

☐ a. Quando o mesmo arquivo é alterado em linhas diferentes

☒ b. Quando o mesmo arquivo é alterado nas mesmas linhas ✔ Quando a mesma linha de um mesmo arquivo é alterada por dois desenvolvedores ao mesmo tempo, ocorre um conflito. Neste caso, é necessário que o desenvolvedor que for executar o merge decida como lidar com o conflito, seja escolhendo entre uma das alterações ou escolhendo a ordem em que as duas serão inseridas.

☐ c. Quando o mesmo arquivo é alterado em linhas próximas

☐ d. Quando arquivos diferentes são alterados, nas mesmas linhas

☐ e. Quando arquivos diferentes são alterados, em linhas diferentes

Sua resposta está correta.

A resposta correta é: Quando o mesmo arquivo é alterado nas mesmas linhas.

Próximo ►

Questão 3

Correto

Atingiu 1,0 de 1,0

Marcar questão

Mudanças durante o desenvolvimento de software são inevitáveis; o ambiente no qual o sistema opera muda, o entendimento dos usuários e desenvolvedores sobre o sistema muda, os requisitos mudam. Por isso a Engenharia de Software tem uma área que trata exatamente desse ponto, a qual é conhecida como Gerência de Configuração de Software.

A Gerência de Configuração de Software (GCS) é um conjunto de atividades que se complementam e que permite a absorção ordenada das mudanças inerentes ao desenvolvimento de software, que mantém a integridade e a estabilidade durante a evolução do projeto.

Portanto, a GCS atua (I) registrando a evolução do projeto com base no (II) controle e acompanhamento das alterações necessárias à uma versão e (III) estabelece um ambiente harmônico entre todas as partes necessárias para a disponibilização de uma versão do software.

Nesse sentido, os itens I, II e III dizem respeito, respectivamente à

Escolha uma:

☒ a. Controle de Versão, Controle de Mudança e Integração contínua ✔

☐ b. Controle de Versão, Integração contínua e Controle de Mudança

☐ c. Controle de Mudança, Controle de Versão e Integração contínua

☐ d. Controle de Mudança, Integração contínua e Controle de Versão

☐ e. Integração contínua, Controle de Mudança e Controle de Versão

Sua resposta está correta.

A resposta correta é: Controle de Versão, Controle de Mudança e Integração contínua.

Questão 4

Incorreto

Atingiu 0,0 de 1,0

Marcar questão

O processo de merge muitas vezes pode ser confuso, demorado e trabalhoso. Justamente por isso, muitos desenvolvedores não gostam de fazê-lo e alguns deles inclusive possuem técnicas para minimizar a quantidade de conflitos que se gera em um projeto ou até mesmo simplificá-los.

Quais técnicas é indicado que possam ser utilizadas para simplificar o processo de merge?

Escolha uma ou mais:

☐ a. Conflitos sempre ocorrerão independente da frequência da sincronização.

☒ b. Sincronização do repositório remoto somente ao fim da tarefa para que consigamos atualizar todos os dados que outros programadores já tenham enviado. ✖ Nesses casos, o desenvolvedor minimiza o número de merges, porém cada um deles pode ser complicado e longo, já que o número de alterações sincronizadas pode ser enorme.

☐ c. Commits com bastante conteúdo: assim evitamos que aconteçam conflitos a todo momento.

☐ d. Sincronização frequente do repositório remoto para que consigamos atualizar todos os dados que outros programadores já tenham enviado.

☐ e. Commits com pouco conteúdo

Sua resposta está incorreta.

A resposta correta é: Commits com pouco conteúdo, Sincronização frequente do repositório remoto para que consigamos atualizar todos os dados que outros programadores já tenham enviado..

Próximo ►

Questão 5
Incorreto
Atingiu 0,0 de 1,0
Marcar questão

Os repositórios locais consistem em três "árvores" mantidas pelo git. a **primeira** delas contém os arquivos vigentes. a **segunda** funciona como uma área temporária e finalmente a **terceira** que aponta para o último envio que foi realizado.

fonte: http://rogerdudler.github.io/git-guide/index.pt_BR.html

Com base nessas informações, respectivamente, quais palavras podem ser substituídas pelas palavras: **primeira, segunda e terceira**?

Escolha uma:

- ☐ a. working directory, local, remoto
- ☒ b. working directory, index e HEAD
- ☐ c. init, HEAD e commit ✗ Esses são alguns dos comandos que podemos utilizar no repositório, mas que não representam seus estados.
- ☐ d. commit, add e clone
- ☐ e. untracked, tracked e modified

Sua resposta está incorreta.

A resposta correta é: working directory, index e HEAD.

Próximo ►

Questão 6
Correto
Atingiu 1,0 de 1,0
Marcar questão

No processo de desenvolvimento de software mudanças ocorrem a todo momento. As mudanças ocorrem por inúmeras razões, tais como, novas condições de negócios e de mercado, novas necessidades de clientes, reorganização, crescimento ou declínio dos negócios, restrições de tempo e custo. Tudo isso pode demandar mudanças nas prioridades do software, na estrutura do time de desenvolvimento, nas informações produzidas pelo software, nos requisitos de produto e nas regras de negócio [Pressman].

Neste contexto, surge o gerenciamento de configuração de software (GCS) como uma disciplina da engenharia de software que é responsável pelo desenvolvimento da gerência da mudança durante todo o ciclo de vida do software [Pressman].

O GCS é uma das ramificações mais sucedidas da engenharia de software, tanto que é atualmente considerado uma ferramenta essencial para o sucesso de qualquer projeto de desenvolvimento de software, sendo necessário para o atendimento do segundo nível de maturidade do CMMI.

Seus principais conceitos são:

I - Item de configuração: Resultado da liberação de uma determinada configuração do produto para o ambiente externo ao desenvolvimento: 1) liberação para testes de sistema; 2) liberação para homologação; 3) liberação para entrega ao cliente;

II - baseline: um conjunto bem definido de itens de configuração que representam um estágio do desenvolvimento e que é passível de modificações apenas mediante um mecanismo formal de alterações;

III - Versão: Instâncias de um mesmo item de configuração que diferem entre si em algo; estado definido de um item em um dado momento; "fotografia" do item de configuração.

Está correto o que se afirma em:

Escolha uma:

- ☐ a. I e III, apenas
- ☐ b. Todas estão corretas
- ☐ c. II, apenas
- ☐ d. I e II, apenas
- ☒ e. II e III, apenas ✓

Sua resposta está correta.

A resposta correta é: II e III, apenas.

Questão 7
Incorreto
Atingiu 0,0 de 1,0
Marcar questão

Uma tarefa bem comum é a criação de uma nova branch seguida da troca para essa branch criada. Poderíamos fazer isso com a sequência de comandos, mas o Git nos fornece uma opção do comando git checkout que realiza este trabalho. Contudo, ambas as formas são válidas e podem ser utilizadas.

Assim, identifique entre as opções abaixo e selecione a opção que realize a ação desejada.

Escolha uma:

- ☐ a. git branch nomeDaBranch
e
git checkout nomeDaBranch
- ☐ b. git -add branch nomeDaBranch
e
git commit nomeDaBranch
- ☐ c. git add nomeDaBranch
e
git merge nomeDaBranch
- ☐ d. git branch -add nomeDaBranch
e
git commit nomeDaBranch
- ☒ e. git branch -add nomeDaBranch
e
git checkout nomeDaBranch ✗

Sua resposta está incorreta.

A resposta correta é: git branch nomeDaBranch
e
git checkout nomeDaBranch.

Questão 8

Correto

Atingiu 1,0 de 1,0

Marcar questão

Um SCV – Sistema de Controle de Versão é um software que tem a finalidade de gerenciar diferentes versões de um artefato qualquer. Esses sistemas são comumente utilizados no desenvolvimento de software para controlar as diferentes versões -- histórico e desenvolvimento -- dos códigos-fontes e também da documentação.

Entre as ferramentas mais comuns encontram-se as soluções livres: CVS, Mercurial, Git e SVN (Subversion); além das comerciais: SourceSafe, TFS, PVCS (Serena) e ClearCase. As quais se dividem, ainda, em dois grupos de acordo com a forma de armazenamento e compartilhamento do acesso aos dados.

A partir disso, pode-se inferir que um sistema de controle de versão

Escolha uma:

☐ a. 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.

☐ 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. 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.

☐ e. 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.

Sua resposta está correta.

A resposta correta é: 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..

Questão 9

Completo

Atingiu 0,4 de 2,0

Marcar questão

Você como artesão de software da empresa XYZ foi chamado para uma reunião onde foi apresentada a demanda para o desenvolvimento de um novo sistema. Esse sistema será criado e sua equipe foi incumbida de desenvolvê-lo.

Você como o responsável pelo controle de versão, recebeu a missão de criar o repositório para que os outros programadores começassem a utilizá-lo.

para tanto você deve:

1. Criar o repositório;

2. Adicionar o arquivos README.txt e .gitignore e adicionar a sua primeira versão no repositório;

3. Por fim, deve enviar os arquivos ao repositório remoto.

Com base nas responsabilidades atribuídas a você, escreva os comandos que devem ser realizados para atender a solicitação.

git init >> git add nome-do-arquivo >> git

1. Iniciar o repositório (init)

2. adicionar o arquivo (add)

3. commitar o arquivo (commit)

4. enviar ao remoto (push)

Comentário: não adicionou os arquivos solicitados
não comitou nem enviou so repositorio remoto