1. O que é o gerenciamento de configuração de software?

Este termo refere-se à disciplina de avaliação, coordenação, aprovação ou rejeição e implementação de mudanças em artefatos que são usados para construir e manter sistemas de software

1. Cite 6 exemplos de ITENS DE CONFIGURAÇÃO de um projeto de desenvolvimento de software.

Documentos de requisitos, especificação e projeto, código fonte, plano de testes, manuais,...

1. Cite 3 exemplos de BASELINES em um projeto de desenvolvimento de software.

Modelagem de negócios, Requisitos, Análise e Design

1. Cite 4 exemplos de ferramentas de controle de versão de código-fonte.

Microsoft Source Control, IBM Rational Clear Case, GIT

1. Referente a ferramentas de controle de versão, como elas podem ser classificadas de acordo com sua arquitetura? Descreva as diferenças entre estas arquiteturas.

**Arquitetura Centralizada:** Trabalha com um servidor

**Arquitetura descentralizada (ou distribuída):** Trabalha no próprio computador

1. Cite uma vantagem de uma ferramenta de controle de versão de arquitetura descentralizada sobre as ferramentas de arquitetura centralizada.

A principal vantagem da ferramenta de arquitetura descentralizada é que vários colaboradores podem trabalhar fazendo alterações em seus repositórios locais e posteriormente fazer um merge em uma única branch no repositório remoto, otimizando assim a produtividade e o tempo de trabalho do projeto.

1. Quando você utiliza o GitHub através da interface web da aplicação, está sendo criado algum repositório local no seu computador? Explique sua resposta.

Não, quando se utiliza o GitHub é criado apenas um repositório remoto. Para gerar um repositório local é necessário clonar o repositório remoto e integra-lo a alguma IDE (Eclipse, NetBeans, Visual Studio) ou fazer esse procedimento através da instalação de alguma aplicação que permite realizar o gerenciamento de repositórios locais e remotos (Git, TortoiseGit, SourceTree, GitHub Desktop).

8) Cite 5 operações que são comuns às ferramentas de controle de versão de código-fonte.

* Gerenciamento de usuário
* Gerenciamento de acesso
* Commit/ Update
* Merge
* Lock/unlock

1. No cenário abaixo, qual será o conteúdo do arquivo após o 2o. merge? Há algum conflito que deverá ser resolvido manualmente? Explique a sua resposta.

*Sim, há conflito, pois houve remoção de conteúdo de dentro do arquivo, após o tratamento o arquivo final será:*

*#file quadrado.py*

*print('inicio')*

*n1 = int(input('n1:'))*

*quadrado = n1\*n1*

*print(quadrado)*

*#calculo do quadrado*

*print('fim')*

1. No cenário abaixo, qual será o conteúdo do arquivo após o 2o. merge? Há algum conflito que deverá ser resolvido manualmente? Explique a sua resposta.

*a Não ocorrem conflitos após 2º merge.*

*#file dobro.py*

*n = int(input('n:'))*

*dobro = 2\*n*

*print(dobro)*

*print('fim')*