**Porque é importante usar gerência de configuração?**

R: para que você possa gerenciar e evitar problemas futuros, assim garantindo a qualidade e a rastreabilidade de todas as mudanças e versões envolvidas no desenvolvimento do projeto.

**O que é integração contínua em um projeto de software? Quais ferramentas podem ser usadas para realizar a integração contínua?**

R: A integração contínua costuma ser usada com o fluxo de trabalho de desenvolvimento de software ágil,  Uma organização vai compilar uma lista de tarefas que contém um roteiro do produto. Essas tarefas são então distribuídas entre os membros da equipe de engenharia de software para a entrega.  
Ferramentas usadas para realizar a integração continua são :   
git, cvs, Subversion e Clear Case.

**O que é débito técnico e quais ferramentas podem ser usadas para verificar débito técnico?**

R: débito técnico diz respeito ao resultado de ações tomadas pelas equipes de desenvolvimento para agilizar a entrega de uma funcionalidade ou um projeto que precisa ser refatorado posteriormente.  
Ferramentas que podem ser utilizada para verificar debito técnico são:

SonarQube,nESLint, OMD, JUnit

**O que é DevOps? Qual o relacionamento da integração contínua com o conceito de DevOps?**

R: Devops, é a integração entre desenvolvimento e operações para automação e colaboração.  
Integração continua envolve mesclar código frequentemente, testado automaticamente. Essa pratica previne erros e facilita entrega continua. Integração continua é vital no DevOps, promovendo ciclos de desenvolvimento mais rápido e confiáveis. Juntas, elas impulsionam eficiência e qualidade no desenvolvimento de software.

**Quando usar o controle de versão centralizado e quando usar o distribuído? Pontos positivos e negativos de cada um.**

Centralizado, é utilizado para projetos pequenos ou equipes novas, onde a simplicidade é importante,   
Positivo: Fácil de entender e um ponto único de controle.  
Negativos: Dependência do servidor, possíveis conflitos, colaboração limitada.

Distribuído: Projetos maiores, equipes experientes, colaboração remota, flexibilidade.  
Positivos: Autonomia, trabalho offline, menos conflitos sérios.   
Negativos: Complexidade, mais espaço em disco, curva de aprendizado.