

Introdução ao Git

1. O que é Git?

É um sistema de controlo de versões, distribuído para rastrear mudanças num qualquer conjunto de arquivos, originalmente projetado para coordenar o trabalho entre programadores que cooperam no desenvolvimento de código de software

2. Existem outros sistemas de controle de versão?

Existem outros sistemas de controle de versão como por exemplo: concurrent version system, subversion, GNU SSC..

3. Quais as vantagens de se utilizar um sistema de controle de versão como o Git?

Um sistema de controle de versões, como o Git, registra as alterações realizadas em um arquivo ou conjunto de arquivos ao longo do tempo e ajuda a recuperar versões específicas do código posteriormente, quando necessário. Um sistema de controle de versões permite: reverter os arquivos de código de volta ao estado anterior, recuperar e reverter todo o projeto de volta ao estado anterior, compara as alterações de código em durações de tempo específicas, permite saber quem modificou pela última vez o código que pode estar a causar um problema,

4. Qual o comando para indicar que uma pasta deve ser gerenciada pelo Git?

Git init

5. Ao adicionar/editar arquivos dentro de uma pasta gerenciada pelo Git, quais os comandos necessários para adicionar estas mudanças, de fato, ao Git?

Git add e git commit - m "texto"

6. Qual a função do comando "checkout"?

O git checkout permite a mudança do "branch" atual para outro escolhido e git checkout <nº do commit> restaura o ficheiro para a versão pretendida

7. Pesquise e explique o que é uma "branch"?

Um "branch" ou ramificação no git é um ponteiro para as alterações feitas nos arquivos do projeto. É útil em situações nas quais pretendemos adicionar um novo recurso ou corrigir um erro. Ao gerar uma nova ramificação, vamos garantir que o código "teste" não fique junto nos

arquivos do projeto principal. No final de concluir a atualização dos códigos, é possível juntar a ramificação que criámos com o “branch” principal, geralmente denominado de master.

8. Diversas plataformas existem no mercado para suportar o uso de Git via Web. Procure algumas destas plataformas. Procure informações como preço, número de pessoas, se oferece o serviço gratuitamente, se possui repositórios privados.

Obs: No mínimo 3 plataformas.

GitHub: armazena repositórios públicos ou privados. O github oferece versões livres e existem versões que são pagas. Estas versões vão se diferenciar na quantidade de espaço disponível no github, assim como na quantidade de minutos por mês para acesso à ferramenta de automatização de workflows GitHub Actions, entre outras funcionalidades mais avançadas.

O CVS é uma ferramenta de *open source* para controle de versões que segue um modelo cliente-servidor. Existe um repositório centralizado dos dados (ficheiros) que estão sobre controle de versões. Esse repositório está no servidor ao qual vários clientes podem ligar para obter cópias locais. Essas cópias locais podem ser modificadas e essas modificações propagadas de volta para o repositório; ou seja, nunca há interações directas entre clientes. Não há custos associados.

O RCS funciona apenas em arquivos individuais. Não há maneira de trabalhar com um projeto inteiro. Embora ele forneça ramificação para arquivos individuais, a sintaxe da versão é complicada. Em vez de usar ramos, muitas equipes usam apenas o mecanismo interno de bloqueio e trabalham em um único ramo *cabeça (head)*.