1. O que é o Git?

Git é um sistema de controle de versionamento de código aberto, criado pelo Linus Torvalds (o criador do Linux), onde podemos hospedar o código fonte de projetos e o atualizando conforme sobem de versão, arquivando o seu histórico de de alterações, além de podermos voltar há algum ponto anterior e recomeçar sem precisar partir do zero, não perdemos nada do projeto, utilizando assim repositórios remotos. É um dos maiores depósitos de trabalho colaborativo do mundo, desenvolvedores e outros usuários podem contribuir com projetos privados ou de código aberto, além de facilitar a comunicação de relatos de erros e combinar repositórios remotos. Funciona como uma rede social para developers, muito comum para divulgar seus trabalhos.

2. O que é versionamento de software?

Em um projeto grande devemos armazenar cada etapa do projeto antes de publicar aos usuários, o processo de *build and deploy*; se por acaso a versão mais recente tiver algum erro ou algum problema não identificado durante o teste ou em outra etapa, podemos voltar rapidamente para a versão anterior. O versionamento de um software também auxilia a registrar alterações realizadas, correção de bugs, adição de ferramentas e ajustes estéticos, além de ser muito mais seguro, pois podemos identificar em qual momento do sistema foi ao ar determinada função ou alterado algo do código, o ideal ainda é utilizar uma ferramenta de controle de versão.

3. Por que utilizar o Git como controle de versionamento?

Para uma segurança maior de um projeto, mesmo que pequeno, devemos usar uma ferramenta de controle de versão, assim caso haja algum imprevisto ou erro não detectado na etapa de teste, podemos voltar um passo e corrigir o *bug*, além de que o Git facilita muito esse processo, criando uma ramificação do código fonte onde podemos trabalhar livremente sem correr o risco de afetar a versão oficial do software, depois juntar as duas linhas em uma versão oficial; ainda temos o espaço colaborativo com outros desenvolvedores o redor do mundo, onde o Git nos ajuda a saber em que parte do projeto cada um está trabalhando, facilitando a comunicação entre os usuários e tornando possível o trabalho em equipe.

4. Quais as vantagens do Git?

Um dos diferenciais do Git são suas ramificações ou *branchs,* uma maneira de organizar o nosso versionamento, quando vamos trabalhar em cima do código fonte para adicionar uma ferramenta ou corrigir um erro o Git cria uma *branch* que podemos modificar sem alterar a versão oficial do software, assim podemos desenvolver sem preocupação de afetar o sistema, realizar testes e depois subir uma nova versão, onde o próprio Git nos auxilia em juntar a ramificação com a última versão em uma nova atualização. Essa funcionalidade nos auxilia muito na hora de trabalhar com o código, além de não corrermos o risco de perder tudo, podemos ajudar alguma versão de emergência e depois voltar a desenvolver normalmente.

Um dos diferenciais do Git são suas ramificações ou *branchs,* uma maneira de organizar o nosso versionamento, quando vamos trabalhar em cima do código fonte para adicionar uma ferramenta ou corrigir um erro o Git cria uma *branch* que podemos modificar sem alterar a versão oficial do software, assim podemos desenvolver sem preocupação de afetar o sistema, realizar testes e depois subir uma nova versão, onde o próprio Git nos auxilia em juntar a ramificação com a última versão em uma nova atualização. Essa funcionalidade nos auxilia muito na hora de trabalhar com o código, além de não corrermos o risco de perder tudo, podemos ajudar alguma versão de emergência e depois voltar a desenvolver normalmente.

5. Qual a importância da utilização do controle de versionamento no desenvolvimento de um software?

Controlar a versão do sistema nos permite saber qual foi o desenvolvedor que trabalhou em cada parte do projeto, ou se for uma única pessoa, esta consegue identificar mais fácil algum erro no código e trabalhar na  correção sem precisar disponibilizar o software neste meio tempo, além de armazenar com uma maior segurança o código fonte, caso haja algum imprevisto com o desenvolvedor por exemplo, onde outro usuário pode intervir e dar continuidade na tarefa; é mais fácil também voltar de versão caso seja identificado algum *bug* do sistema; mais simples também de encontrar em qual versão foi adicionada alguma ferramenta ou extensão.