

Atividade:

Nome: Luiza Taís Müller e Thaís Labely Marques Soares

1→ O que é versionamento de software e qual é a sua importância no desenvolvimento de projetos?

O versionamento é essencial na gestão e controle de mudanças, é o processo de controlar e registrar mudanças realizadas no código ao longo do tempo. Como por exemplo correção de bugs, novas funcionalidades... Facilita o desenvolvimento e possibilita o rastreamento das alterações, além de permitir que equipes trabalhem simultaneamente no mesmo objetivo, e caso alguma alteração comprometa o sistema, o versionamento permite restaurar uma versão anterior.

2→ Quais são as principais vantagens de utilizar sistemas de controle de versão, como o Git, em projetos de desenvolvimento de software?

R: O Git: os usuários não precisam estar conectados o tempo inteiro. Isso permite que eles façam commits, branches e outras operações localmente, resultando em um ganho de performance e maior flexibilidade.

Controle de versão:

- Centralização: O controle de versão oferece à sua equipe uma única fonte de verdade.
- Governança, risco e Compliance: O controle de versão mantém um histórico completo do arquivo e melhora a visibilidade.
- Desenvolvimento paralelo: O controle de versão permite o desenvolvimento simultâneo.
- Controle: O controle de versão permite uma melhor colaboração em equipe.
- Automação: O controle de versão oferece suporte à automação no desenvolvimento

3 → Quais são os principais sistemas de controle de versão disponíveis atualmente?

Git - Mais popular, é distribuído permitindo que cada desenvolvedor possua uma cópia completa do repositório e histórico.

Subversion (SVN) - Controle centralizado, o repositório central é a única fonte de verdade.

Mercurial - Distribuído e semelhante ao Git, porém com abordagens mais simples e intuitivas.

Perforce - Centralizado, e muito utilizado em projetos de grandes empresas, especialmente na

4→ Como funciona o processo de criação e gerenciamento de um repositório em um sistema de controle de versão?

R:•Repositório: Um repositório é um local centralizado onde todo o código-fonte e os arquivos relacionados de um projeto são armazenados.

1º Início “git init”

2º Commit - “git add”, registrar mudanças “git commit”

3º Branches: “git branch, git checkout”

4º Sincronização , “git push”

5 → O que são commits e como eles são úteis para controlar as mudanças em um projeto?

Um commit é uma operação em sistemas de controle de versão que registra o código em um determinado estado. Ele representa uma alteração no código-fonte de um projeto, podendo incluir desde a criação de novos arquivos e pastas até modificações em trechos de código ou exclusões de componentes.

Utilidade:

Cada commit forma um registro das mudanças feitas ao longo do tempo.

Caso uma modificação introduza um erro, é possível reverter para um estado anterior do projeto ao retornar a um commit específico. Isso dá segurança para experimentar novas ideias, sabendo que o histórico anterior está preservado.

Em equipes, cada desenvolvedor pode trabalhar em diferentes partes do projeto e fazer commits para registrar o progresso.

Commits com descrições detalhadas ajudam a rastrear onde um bug específico foi introduzido, facilitando a análise e correção de problemas.

6 → Como os branches são utilizados no versionamento de software? Quais são os benefícios de trabalhar com branches em projetos de desenvolvimento?

R: Branch é uma ramificação independente do código em um repositório. Quando um novo branch é criado, ele herda todo o histórico de commits do branch principal (geralmente chamado de "master" ou "main"). Branch auxilia na otimização do desenvolvimento de projetos e evita conflitos entre equipes ou desenvolvedores individuais.

7 → O que é uma mesclagem (merge) no contexto do versionamento de código? Como ela é realizada?

R: O merge é o processo de combinar as alterações feitas em um branch com outro. Geralmente, isso significa incorporar as alterações de um branch secundário de volta ao branch principal.

8 → Quais são os principais desafios que podem surgir ao trabalhar com vários desenvolvedores em um mesmo projeto e como o versionamento de software pode ajudar a superá-los?

O versionamento de software ajuda equipes a enfrentar os desafios do desenvolvimento colaborativo, como conflitos de código, rastreamento de mudanças, e manutenção de estabilidade. Ele permite que cada desenvolvedor faça suas alterações de forma isolada, utilizando branches para separar funcionalidades e commits para registrar mudanças. Isso facilita a resolução de conflitos, a reversão para versões anteriores, e o

histórico detalhado de cada contribuição. Ferramentas de integração contínua automatizam testes para garantir que o código integrado esteja sempre estável. Mensagens de commit claras e revisões de código também promovem transparência e qualidade, permitindo que a equipe colabore de maneira mais eficiente e organizada.

9→ Quais são as melhores práticas para escrever mensagens de commit significativas e úteis?

R: Clareza, usando verbos como “ADD ou FIX”> Padrões, seguir formatos como : tipo descrição > Especifique o Escopo> Evitar termos genéricos > Commits pequenos, pois commits focados e menores são mais fáceis a compreensão.

10 → Como resolver conflitos de mesclagem (merge) que ocorrem quando dois ou mais desenvolvedores modificam o mesmo trecho de código?

R:

Identificar o conflito: Ferramentas de controle de versão destacam automaticamente as linhas em conflito.

Analisar e escolher a versão correta: Revisar o código conflitante e decidir qual versão manter ou se uma combinação de ambas é a melhor solução.

Testar o código: Verificar se as mudanças não causam novos problemas no código principal.

Realizar o commit final: Após resolver o conflito, é feito um commit para concluir a mesclagem