Trabalho versionamento

Prof.ª: Rafex

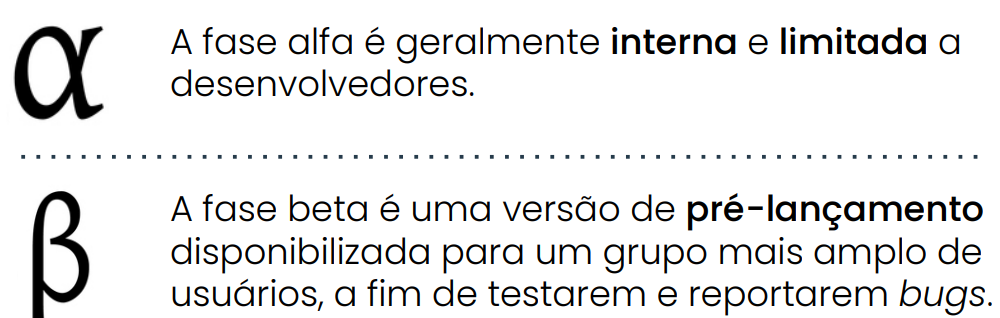
17/05/2024

Participantes: Gabriel Martins Cordeiro Rodrigues, Kevyn Cardoso dos Santos e Vitor dos Santos Fabiano

Resumo de todas as aulas

S1a1: O versionamento de código refere-se ao processo de manter um registro de alterações feitas em um arquivo ou conjunto de arquivos ao longo do tempo

S1a2: Versão beta/alfa: Antes de um software ser oficialmente lançado, ele pode passar por fases de testes chamadas "alfa" e "beta".



S1a3: 1. Rastreabilidade de alterações: O versionamento permite que os desenvolvedores rastreiem cada alteração feita no código-fonte

2. histórico completo: Com o versionamento, cada mudança no código é registrada

3.prevenção de conflitos: O versionamento de código ajuda a identificar e resolver conflitos de código antes que eles cheguem à produção.

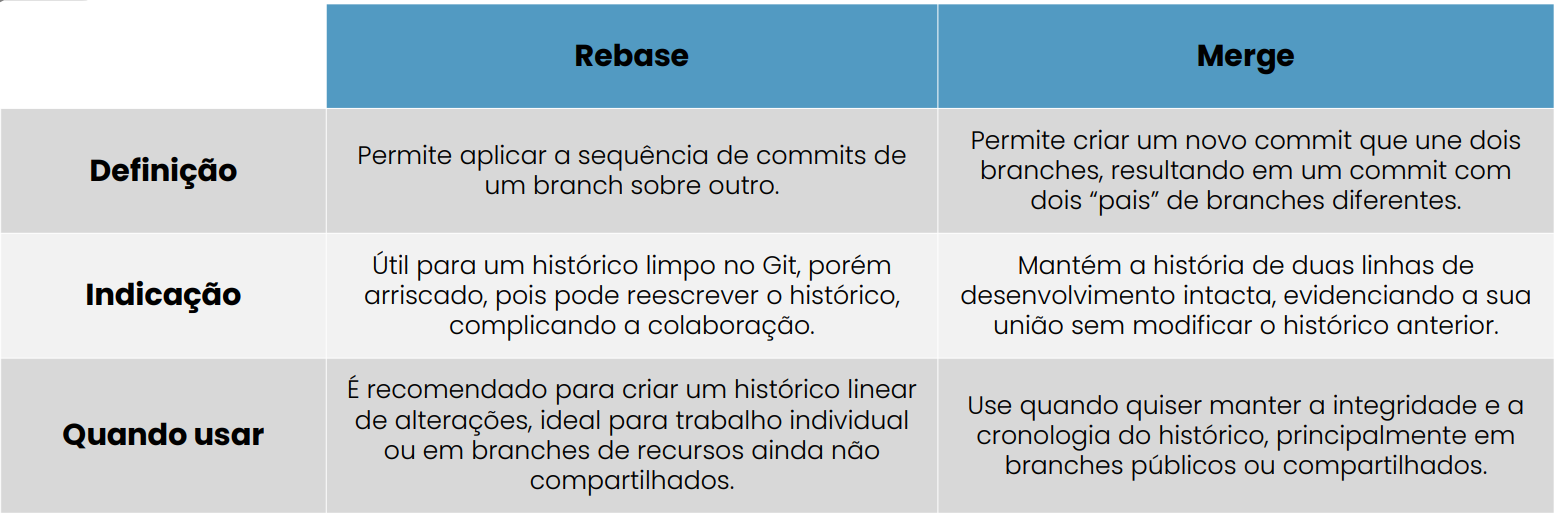
S2a1: controle de versão de código: É um sistema que grava alterações em arquivos, possibilitando ao usuário acessar versões anteriores quando for necessário.

Ele é essencial para gerenciar versões do código-fonte no desenvolvimento de software

Permite voltar arquivos ou o projeto todo a estados anteriores, comparar alterações e melhorar a colaboração entre desenvolvedores.

S2a2: Branch: Um Branch no Git é uma ramificação do projeto, que permite trabalhar em características ou correções de forma independente, sem impactar a linha principal ou outros branches

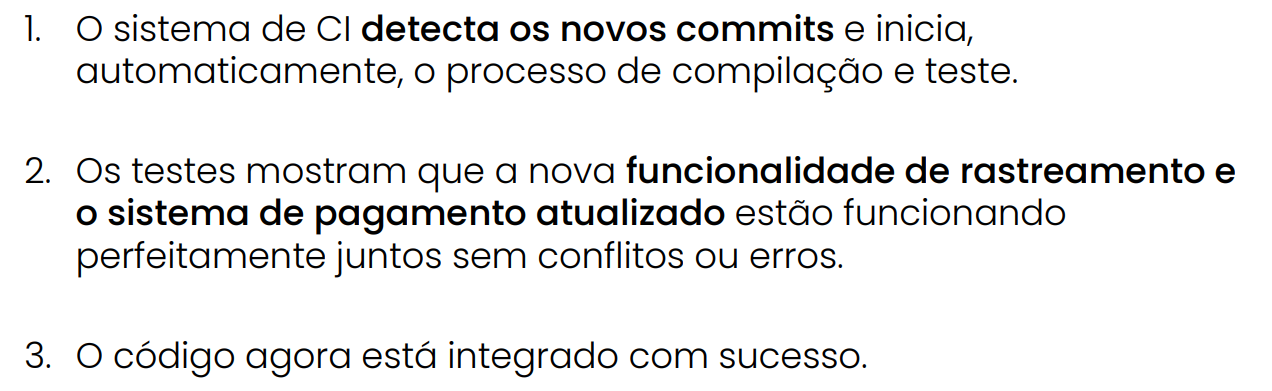
S2a3:



S3a1: CI: Em um ambiente de CI, desenvolvedores frequentemente fazem commit de seu código em um repositório centralizado. Após cada commit, os sistemas de CI compilam e testam o projeto inteiro automaticamente, para garantir, assim, que a nova alteração não gere falhas.

S3a2: CD: Entrega contínua, frequentemente abreviada como CD (do inglês continuous delivery), é uma abordagem de desenvolvimento de software em que o código é construído, testado e preparado para ser lançado em produção de forma contínua.

S3a3:

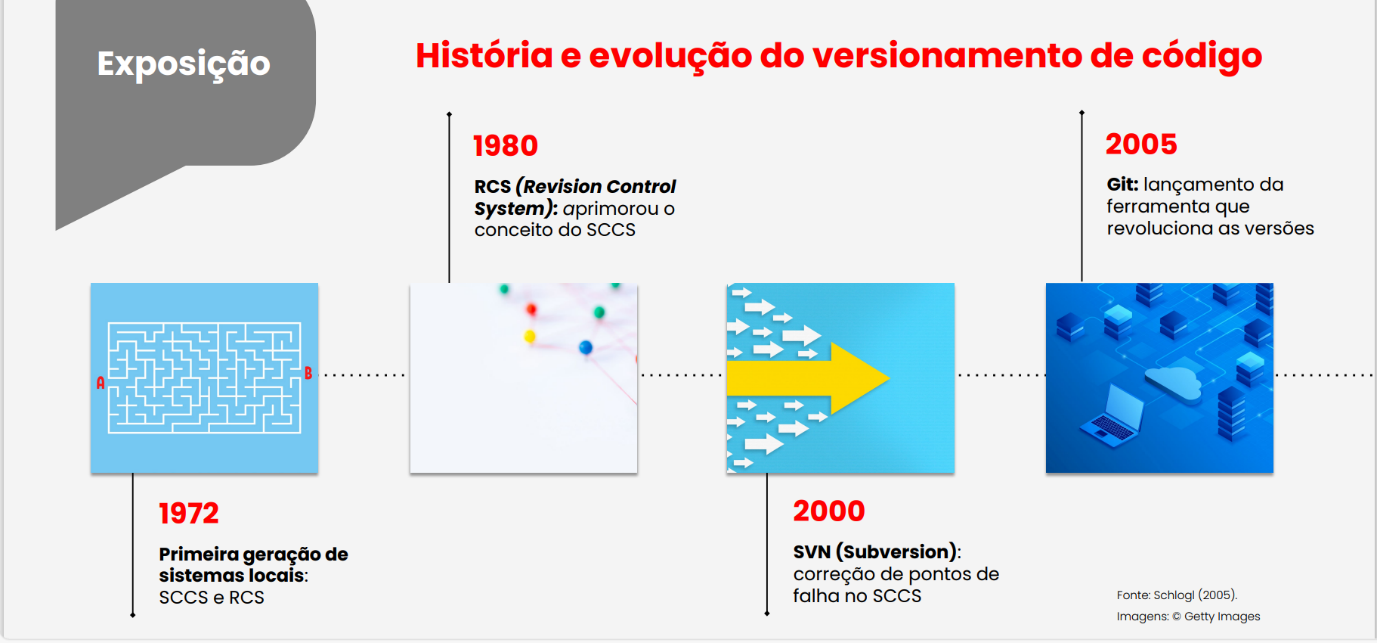


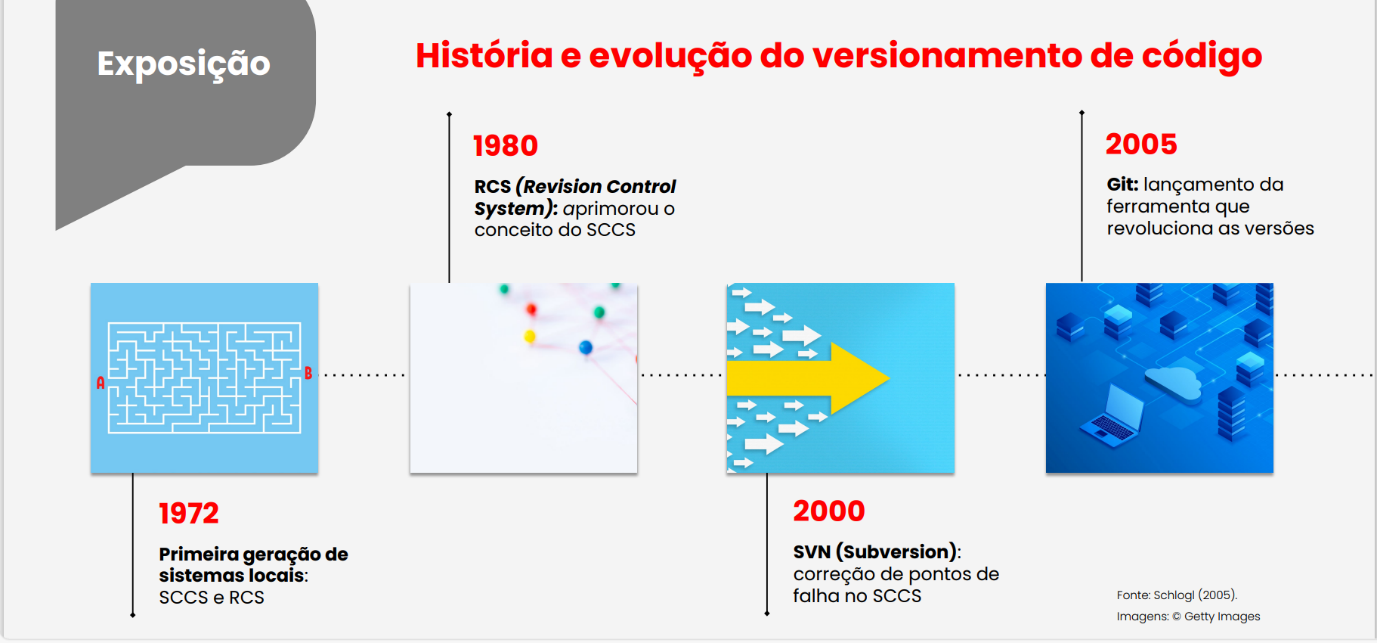
S4a1: commits atômicos: Utilizar commits atômicos facilita o entendimento e a revisão do código. Se surgir um problema, é mais seguro e prático reverter uma pequena mudança específica do que um grande lote de alterações

S4a2: rebase vs. Merge: O merge e o rebase são dois comandos que atingem um objetivo comum no Git: integrar alterações de um Branch em outro. No entanto, eles fazem isso de maneiras muito diferentes e têm implicações diferentes para o histórico do projeto

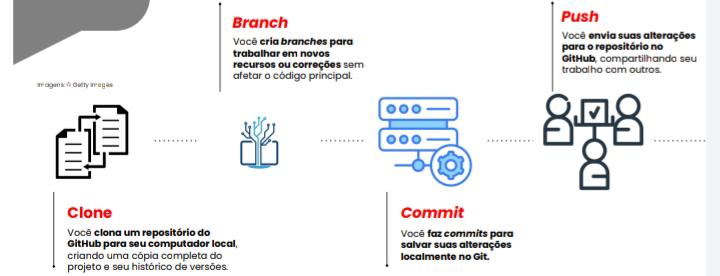
S4a3: Git Flow: O Git Flow é ideal para lançamentos que seguem um cronograma e para a manutenção de versões anteriores. Ele se alinha bem com ciclos de desenvolvimento que ocorrem semanal ou mensalmente, como os sprints

S5a1:

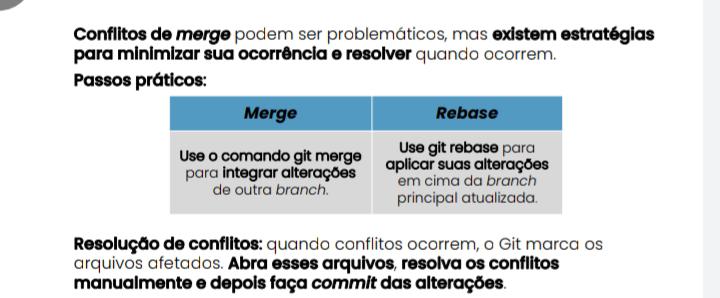


S5a2: 

S5a3: repositório: Um repositório é um local centralizado onde dados são armazenados e gerenciados. No contexto do controle de versão e desenvolvimento de software, um repositório refere-se especificamente a uma estrutura de armazenamento que contém todos os arquivos de código do projeto, bem como o histórica de todas as mudanças feitas nesses arquivos ao longo do tempo

S6a1:

S6a2:nova branch: Dentro de reposkoro no oftHub, ellique em "main" (oυ nome da sua branch principal) para abrir o menu dropdown de branches
No campo de texto digite o nome da sua nova branch (por exemplo, feature calculadora) pressione "Enter".

S6a3:

S7a1:Uma alternativa popular para integração continua, especialmente para projetos hospedados no GitHub, é o GitHub Actions.
Clin sub Actions é uma ferramenta /Co integrado diretamente co citHub, permitindo que você automatize seus fluxos de trabalho de integração e entrega continuas diretamente a partir do seu repositorio GitHub.

S7a2:Criação de branch para nova funcionalidade: quando você inicia o desenvolvimento de uma nova funcionalidade, você cria uma nova branch a partir da brarich principal Desenvolvimento: você faz todos os seus com/mits nessa branch. Pull request/merge request: quando a funcionalidade esto pronta, você cria um pull request para mesclar sua branch de funcionalidade de volta à branch principal Revisão de código: a equipe revisa o código antes de ser mesciado Mesclagem: uma vez aprovado, o código è mesclado na branch principal.

S7a3: uso de runners em ambientes virtuais: Codo GitHub-hosted runner vom com um conjunto especifico de softwares e ferromeritas pre-instalados dependendo do sistema operacional Você pode escolher o ambiente mais adequando as necessidades do seυ Projeto. ubuntu-latest, Windows-latest, macOs-latest são exemplos de ambiantes virtuais disponíveis.

S8a1: O Git Flow é uma estratégia de ramificação que define um conjunto estrito de branches e suas respectivas funções: • Master; • Develop;

• Feature;

• Release;

• Hotfix.

S8a2: Escolher entre merge e rebase depende do contexto especifico das preferências da equipe. O merge é mais seguro e preserva o histórico completo, mas pode resultar em um histórico de projeto bagunçado; O rebase cria um histórico mais limpo, mas pode ser mais complexo e arriscado de usar, especialmente para usuários menos experientes ou em branches compartilhadas.

S8a3:

Participantes: Gabriel Martins Cordeiro Rodrigues, Kevyn Cardoso dos Santos e Vitor dos Santos Fabiano