**CENTRO PAULA SOUZA**

**ESCOLA TÉCNICA DE PERUIBE**

**TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS**

**Beatris Cecilia de Sousa**

**Giulia de Almeida Oliveira**

**Maria Priscila Souza Bueno**

**Controle de Versionamento no Desenvolvimento de Projetos**

**Trabalho de PW 1**

**Peruíbe**

**2024**

**Giulia de Almeida Oliveira**

**Controle de Versionamento no Desenvolvimento de Projetos**

**Trabalho de PW 1**

Trabalho entregue para menção parcial do componente PW1, para obtenção do certificado de Auxiliar de Desenvolvimento de Sistemas Integrado ao Ensino Médio da Etec de Peruíbe, orientados pelo Profª William.

Peruíbe

2024

1. **Introdução**

Nesse trabalho falaremos sobre o controle de versionamento, uma prática fundamental no desenvolvimento de software, e é essencial para garantir a colaboração eficiente entre os desenvolvedores, a manutenção do histórico de alterações e a recuperação rápida em caso de erros. Com o grande aumento da complexidade dos projetos, o uso de ferramentas adequadas se torna cada vez mais importante.

1. **O que é e sua importância.**

O controle de versão, também conhecido como controle de fonte, é a prática de rastrear e gerenciar as alterações em um código de software. Os sistemas de controle de versão são ferramentas de software que ajudam as equipes de software a gerenciar as alterações ao código-fonte ao longo do tempo. Como os ambientes de desenvolvimento aceleraram, os sistemas de controle de versão ajudam as equipes de software a trabalhar de forma mais rápida e inteligente. Eles são ainda mais úteis para as equipes de DevOps, pois as auxiliam a reduzir o tempo de desenvolvimento e aumentar as implementações bem-sucedidas.

O software de controle de versão mantém registro de todas as modificações no código em um tipo especial de banco de dados. Se um erro for cometido, os desenvolvedores podem voltar no tempo e comparar versões anteriores do código para ajudar a corrigir o erro enquanto diminuem interrupções para todos os membros da equipe.

Sua importância vem do seu uso essencial para gerenciar o histórico de alterações em um projeto, protegendo o código-fonte contra desastres e erros humanos. Ele permite o desenvolvimento simultâneo, automação de tarefas, rastreamento de mudanças e recuperação de desastres, garantindo alta disponibilidade e segurança. Além disso, facilita o acompanhamento de upgrades e correções de bugs.

Existem diversos benefícios, como a Colaboração Eficiente, que permite que múltiplos desenvolvedores trabalhem simultaneamente em um projeto, minimizando conflitos e facilitando a integração das contribuições. O Histórico de Alterações, que mantém um registro detalhado das modificações, possibilitando revisitar versões anteriores do código quando necessário, e a Recuperação de Erros, que em caso de falhas ou bugs, é possível reverter o código para um estado anterior, garantindo a continuidade do projeto sem perda significativa.

1. **Principais Ferramentas e Repositórios**

As ferramentas de controle de versão gerenciam todo o ciclo de vida do trabalho, guardando o histórico de cada documento do projeto. Permitindo desde rastrear alterações desde a versão original, a visualizar o histórico de mudanças e saber quem fez cada alteração. Suas principais ferramentas são:

**1. Git**

Uma ferramenta gratuita e de código aberto, que é uma das mais reconhecidas no DevOps. O Git é muito utilizado no desenvolvimento de software livre, o que a diferencia é o seu sistema distribuído que permite trabalho offline, ideal para projetos open source e gerenciamento eficiente do histórico.

**2. Subversion (SVN)**

Uma ferramenta de controle de versão que usa a arquitetura de rede cliente-servidor e a resolução de conflitos por mesclagem ou bloqueio, seus diferenciais são suas ferramentas centralizadas que oferecem simplicidade na execução de comandos, adequada para equipes menores.

**3. Mercurial**

Uma ferramenta de controle de versão que usa a arquitetura de rede distribuída e a resolução de conflitos por fusão, ela se diferencia com sua Interface amigável e eficiência em ambientes distribuídos, ideal para grandes equipes e projetos complexos.

Já os repositórios, são fundamentais para o gerenciamento e a colaboração no desenvolvimento de software, permitindo que os desenvolvedores armazenem, compartilhem e gerenciem alterações no código-fonte de maneira organizada e eficiente. Eles são divididos em tipos. Nos repositórios centralizados, existe um único servidor central que contém o repositório principal. Todos os desenvolvedores têm suas cópias locais, mas as interações com o código necessitam passar pelo servidor, o que simplifica na gestão do código, sendo ideal para equipes pequenas. Nos repositórios distribuídos, cada desenvolvedor possui uma cópia completa do repositório em sua máquina local, o que permite que funcionários trabalhem offline e realizem alterações sem depender de um servidor central, aumentando a flexibilidade e autonomia dos desenvolvedores e facilita a colaboração em grandes equipes. Seus principais repositórios são:

**1. GitHub**

GitHub é uma plataforma de hospedagem para repositórios Git, amplamente utilizada por desenvolvedores em todo o mundo. Tem como qualidade ser uma plataforma robusta para hospedagem de projetos Git, com funcionalidades como pull requests e integração contínua.

**2. Bitbucket**

Bitbucket é uma plataforma que suporta tanto Git quanto mercurial, oferecendo flexibilidade às equipes que utilizam diferentes sistemas de controle de versão, sua maior qualidade é integrar bem com ferramentas Atlassian e oferecer repositórios privados gratuitos para pequenas equipes.

**3. GitLab**

GitLab é uma plataforma completa que combina controle de versão com recursos de DevOps, incluindo CI/CD. Se diferencia por conta que fornece uma plataforma completa, facilitando a automação dos processos de desenvolvimento, oferecendo gestão de projetos, monitoramento e segurança em um único lugar.

1. **Comandos para Gerenciar um Projeto no GitHub**

Os melhores comandos utilizados para configurar, criar, clonar, alterar, adicionar e subir as alterações do projeto hospedado no GitHub são:

“

* **Configurar seu nome e email**

git config --global user.name "Seu Nome"

git config --global user.email ‘[seuemail@example.com’](mailto:seuemail@example.com) “

* **Criar um novo repositório local**

mkdir meu-projeto

cd meu-projeto

git init

* **Clonar um repositório existente**

git clone https://github.com/usuario/nome-do-repositorio.git

**- Fazer alterações em arquivos**

nano arquivo.txt # ou qualquer editor que você preferir

**- Adicionar as alterações ao staging area**

git add arquivo.txt

**- Realizar um commit das alterações**

git commit -m "Descrição das alterações realizadas"

**- Subir as alterações para o repositório remoto**

git push origin main # ou o nome da branch que você está utilizando

```

1. **Conclusão**

Entende-se que o controle de versionamento é uma prática indispensável no desenvolvimento moderno de software, se compreendermos suas ferramentas e técnicas, entendemos que ele é essencial para garantir a eficiência e a qualidade dos projetos. À medida que as demandas por colaboração aumentam, se manter atualizado sobre as melhores práticas se torna fundamental para o sucesso das equipes de desenvolvimento.

1. **Referências Bibliográficas**

Andreyna Carvalho. Conheça diferentes formas de versionar projetos.Coodesh. Disponível em: <https://coodesh.com/blog/candidates/conheca-diferentes-formas-de-versionar-projetos/>. Acesso em: 13 nov. 2024.

Isabele Reusing. Controle de versão: ferramentas e dicas para versionar seus projetos. Black Friday3. Disponível em: <https://www.cursospm3.com.br/blog/controle-de-versao/#:~:text=O%20controle%20de%20vers%C3%A3o%20%C3%A9,longo%20do%20processo%20de%20desenvolvimento>. Acesso em: 13 nov. 2024.

Malu. A importância do versionamento de software. UDS blog. Disponível em: <https://uds.com.br/blog/a-importancia-do-versionamento-de-software/#:~:text=O%20versionamento%20de%20software%20%C3%A9,garantindo%20alta%20disponibilidade%20e%20seguran%C3%A7a>. Acesso em: 13 nov. 2024.

O que é o controle de versão? Atlassian. Disponível em: <https://www.atlassian.com/br/git/tutorials/what-is-version-control#:~:text=O%20controle%20de%20vers%C3%A3o%2C%20tamb%C3%A9m,fonte%20ao%20longo%20do%20tempo>. Acesso em: 13 nov. 2024.