

## **Centro Universitário do Norte - UniNorte**

Aluna: Sanayane Costa

Matrícula: 03091039

Aluno: Arthur Anveres

Matrícula: 03337530

Aluno: Bruno Santos de Moraes

Matrícula: 03333503

Aluno: Vinicius Mihael Noronha Crispim

Matrícula: 03333882

Aluna: Izabelly Hayden Ramires

Matrícula: 03334753

Aluna: Giovana Vitória dos Santos Lima

Matrícula: 03334145

**Relatório de Ferramentas de Controle de Versão**

**Manaus – AM**

**2024**

## **Sistemas de Controle de Versão Centralizados**

### **1. IBM® Rational® ClearCase®**

O IBM Rational ClearCase é uma solução robusta de controle de versão e gerenciamento de configuração que oferece suporte para o desenvolvimento paralelo e a gestão de múltiplas versões de ativos de software. Suas principais características incluem:

- **Acesso Controlado:** Garante que apenas usuários autorizados possam acessar e modificar os ativos.
- **Desenvolvimento Paralelo:** Facilita a colaboração entre equipes que trabalham simultaneamente em diferentes partes de um projeto.
- **Gerenciamento Automatizado do Espaço de Trabalho:** Automatiza a configuração dos ambientes de desenvolvimento para melhorar a eficiência.
- **Gerenciamento de Linha de Base e Versões:** Permite a criação e gerenciamento de pontos de referência no desenvolvimento do software.

- **Auditoria de Compilação:** Fornece um registro detalhado das compilações e alterações.

**Uso:** Ideal para grandes equipes e projetos complexos onde a segurança e a integridade dos dados são críticas.

## 2. Visual SourceSafe (VSS)

O Visual SourceSafe (VSS) é um sistema de controle de versão que foi amplamente utilizado para gerenciar versões de código-fonte e documentos.

Suas principais características incluem:

- **Versionamento de Arquivos Diversos:** Capacidade de versionar não apenas código-fonte, mas também documentos como Word e Excel.
- **Banco de Dados em Pastas:** Armazena versões e metadados em pastas no local de instalação, facilitando a recuperação e gerenciamento dos dados.
- **Gerenciamento de Versões:** Permite a manutenção e recuperação de versões anteriores de arquivos e documentos.

**Uso:** Embora menos comum hoje em dia, ainda é útil para gerenciamento de documentos e código em ambientes com necessidades básicas de versionamento.

## 3. Perforce Helix Core

O Perforce Helix Core é um sistema centralizado conhecido por sua alta performance e escalabilidade. Suas características incluem:

- **Armazenamento Centralizado:** Mantém todos os arquivos e suas versões em um servidor central, garantindo que todos os desenvolvedores tenham acesso à versão mais recente.
- **Gerenciamento de Arquivos Grandes:** Capacidade de lidar eficientemente com grandes arquivos binários e código.

- **Suporte a Sites Remotos:** Usa servidores de réplica e proxy para melhorar o desempenho e o acesso remoto.

**Uso:** Ideal para grandes equipes e projetos com arquivos de grande tamanho, como desenvolvimento de jogos e semicondutores.

#### 4. TeamWare

O TeamWare foi uma ferramenta de controle de versão distribuída desenvolvida pela SUN. Suas características principais eram:

- **Controle de Versão Distribuído:** Permitia que diferentes equipes trabalhassem em cópias independentes de um projeto.
- **Histórico e Gerenciamento de Versões:** Facilitava o gerenciamento de alterações e histórico de versões.

**Uso:** Foi substituído por ferramentas mais modernas e não é amplamente utilizado atualmente.

#### 5. BitKeeper

O BitKeeper foi desenvolvido para lidar com o crescimento do Linux e ofereceu uma versão comunitária com certas restrições. Suas características incluem:

- **Versão Comunitária:** Disponível para projetos de código aberto com restrições sobre o desenvolvimento de ferramentas concorrentes.
- **Gerenciamento Centralizado:** Manteve uma cópia centralizada do código e das mudanças.

**Uso:** Usado inicialmente para projetos de código aberto como o Linux, mas posteriormente substituído por alternativas com licenças mais abertas.

#### 6. Microsoft Team Foundation Server (TFS)

Lançado em 2005, o Microsoft TFS é uma solução completa de gerenciamento de projeto e controle de versão. Suas características incluem:

- **Integração com IDEs:** Funciona bem com o Microsoft Visual Studio e Eclipse.
- **Gerenciamento de Projetos e Código:** Suporta itens de trabalho, histórias, tarefas e muito mais.
- **Suporte a Diversas Linguagens:** Oferece suporte a linguagens como C#, Python e Java.

**Uso:** Adequado para equipes que utilizam o Visual Studio e precisam de uma solução integrada para gerenciamento de projetos e código.

## 7. Azure DevOps Server

O Azure DevOps Server é a versão local do Azure DevOps, oferecendo uma solução robusta para gerenciamento de projetos e controle de versão. Suas características incluem:

- **Ambiente Local:** Permite que as equipes utilizem uma solução de DevOps em servidores dedicados.
- **Funcionalidades Avançadas:** Inclui suporte para gerenciamento de código, projetos e automação de build.

**Uso:** Ideal para empresas que preferem manter suas ferramentas de DevOps em um ambiente local.

## 8. Azure DevOps Services

O Azure DevOps Services é a versão em nuvem do Azure DevOps, proporcionando uma solução de DevOps sem necessidade de configuração local. Suas características incluem:

- **Serviço de Nuvem:** Executado na plataforma Azure, facilitando a configuração e o acesso.
- **Integração com Outras Ferramentas:** Oferece integração com ferramentas de gerenciamento de código e projetos.

**Uso:** Adequado para equipes que buscam uma solução de DevOps baseada em nuvem, com configuração mínima.

## **Sistemas de Controle de Versão Distribuídos**

### **9. Git**

O Git é um sistema de controle de versão distribuído criado por Linus Torvalds.

Suas características incluem:

- **Controle Distribuído:** Cada desenvolvedor tem uma cópia completa do repositório e do histórico.
- **Desempenho e Flexibilidade:** Oferece alto desempenho e flexibilidade para diversos fluxos de trabalho.
- **Segurança:** Inclui mecanismos robustos de segurança e integridade dos dados.

**Uso:** Ideal para projetos de todos os tamanhos e equipes distribuídas, especialmente com foco em colaboração e flexibilidade.

### **10. Mercurial**

O Mercurial é um sistema de controle de versão distribuído eficiente e fácil de usar. Suas características incluem:

- **Gerenciamento Eficiente:** Oferece um gerenciamento eficiente do histórico e dos arquivos.
- **Flexibilidade de Fluxo de Trabalho:** Suporta diferentes tipos de fluxos de trabalho distribuídos.

**Uso:** Adequado para projetos de diversos tamanhos e equipes que necessitam de uma solução distribuída confiável.

## 11. Bazaar

O Bazaar é um sistema de controle de versão distribuído que oferece flexibilidade e suporte a diversos fluxos de trabalho. Suas características incluem:

- **Trabalho Offline:** Permite que os desenvolvedores trabalhem sem conexão com a rede.
- **Gerenciamento de Versões:** Oferece um gerenciamento eficiente de alterações e versões.

**Uso:** Ideal para equipes que precisam de flexibilidade no gerenciamento de versões e trabalho offline.

## 12. GNU Arch

O GNU Arch é uma ferramenta de controle de versão distribuída com foco em rastreamento e gerenciamento de mudanças. Suas características incluem:

- **Rastreamento Detalhado:** Oferece um rastreamento detalhado das alterações em projetos de software.
- **Controle Distribuído:** Suporte para gerenciamento distribuído de código.

**Uso:** Embora mais antiga, ainda é usada para projetos que necessitam de um controle distribuído detalhado.

## 13. ClearCase

O ClearCase, além de ser um sistema centralizado, também possui características de controle distribuído. Suas principais características incluem:

- **Controle Detalhado de Versões:** Oferece controle e gerenciamento detalhado de versões.
- **Integração com Processos de Desenvolvimento:** Suporte para processos de desenvolvimento complexos.

**Uso:** Utilizado em ambientes onde o controle detalhado e a integração com processos de desenvolvimento são essenciais.

## 14. Code Co-op

O Code Co-op é um sistema de controle de versão peer-to-peer que se destaca por sua confiabilidade. Suas características incluem:

- **Replicação Completa:** Cada membro tem uma cópia completa do projeto, proporcionando segurança adicional.
- **Velocidade e Facilidade de Uso:** Oferece acesso rápido e uma interface intuitiva.

**Uso:** Ideal para equipes distribuídas que precisam de uma solução confiável e rápida para gerenciamento de versão.

## 15. Code Co-op Encrypt

O Code Co-op Encrypt é um utilitário que adiciona criptografia ao Code Co-op. Suas características incluem:

- **Criptografia Robusta:** Usa criptografia Blowfish e SHA-256 para proteger scripts e dados.
- **Criptografia por Projeto:** Permite criptografia individual para cada projeto.

**Uso:** Recomendado para ambientes onde a segurança dos dados é uma prioridade, e a criptografia adicional é necessária.

## 16. Plastic SCM

O Plastic SCM é uma solução de controle de versão distribuída voltada para o desenvolvimento de jogos. Suas características incluem:

- **Desempenho e Escalabilidade:** Oferece alta performance e escalabilidade, ideal para grandes bases de código.
- **Recursos Específicos para Jogos:** Inclui funcionalidades voltadas para o desenvolvimento de jogos e gerenciamento de ativos não codificados.

**Uso:** Ideal para equipes de desenvolvimento de jogos que precisam de uma solução de controle de versão especializada e escalável.



## 17. Perforce Federated Architecture

A Perforce Federated Architecture é uma solução distribuída que melhora o desempenho em grandes operações. Suas características incluem:

- **Desempenho Local:** Permite que equipes remotas experimentem desempenho local em grandes operações de build e clone.
- **Escalabilidade:** Suporte para grandes equipes e projetos distribuídos.

**Uso:** Adequado para grandes equipes que precisam de desempenho local em operações distribuídas.

## 18. Code Co-op Encrypt

O Code Co-op Encrypt é um utilitário adicional ao Code Co-op, que oferece criptografia avançada para comunicação de dados. Suas características incluem:

- **Criptografia Blowfish:** Usa chaves de 256 bits e criptografia robusta para proteger dados.
- **Integridade do Arquivo:** Verificação da integridade do arquivo com SHA256.

**Uso:** Ideal para ambientes onde a criptografia avançada e a segurança dos dados são cruciais.

## 19. BitKeeper

O BitKeeper, após sua introdução, foi utilizado principalmente para gerenciar o crescimento do Linux e outros projetos de código aberto. Suas características incluem:

- **Versão Comunitária com Restrições:** Disponível para projetos de código aberto, com restrições sobre desenvolvimento de ferramentas concorrentes.

- **Gerenciamento Centralizado:** Oferece uma abordagem centralizada para controle de versão.

**Uso:** Usado para projetos de código aberto até ser substituído por ferramentas com licenças mais abertas.

## Referência

**AIX 7.3 - Conceitos do sistema de controle de código-fonte.** IBM.

Disponível em: <https://www.ibm.com/docs/pt-br/aix/7.3?topic=concepts-source-codecontrol-system>. Acesso em: 15 ago. 2024.

**Análise Comparativa entre Sistemas de Controle de Versões.** TANURE, Daniel; DE FREITAS, Menandro. UFJF, mar. 2013. Disponível em: <https://www2.ufjf.br/getcomp/wp->

content/uploads/sites/645/2013/03/An%C3%A1lise-Comparativa-entreSistemas-de-Controle-de-Vers%C3%B5es-Daniel-Tannure-Menandro-deFreitas.pdf. Acesso em: 15 ago. 2024.

**BitKeeper.** Wikiwand. Disponível em: <https://www.wikiwand.com/pt/BitKeeper>. Acesso em: 15 ago. 2024.

**Code Co-op Encrypt.** JHC Software. Disponível em: <http://www.jhcsoftware.com/codecoopencrypt.shtml>. Acesso em: 15 ago. 2024.

**Conheça 5 ferramentas de controle de versão de software.** Gaea. Disponível em: <https://gaea.com.br/conheca-5-ferramentas-de-controle-de-versao-desoftware/#:~:text=Mercurial,como%20o%20Facebook%20e%20Google>. Acesso em: 15 ago. 2024.

**Controle de versões - Parte III: DVCS.** Embarcados. Disponível em: <https://embarcados.com.br/controle-de-versoes-parte-iii-dvcs/>. Acesso em: 15 ago. 2024.

**CVS: Mais controle no desenvolvimento em equipe.** DevMedia. Disponível em: <https://www.devmedia.com.br/conheca-o-cvs-mais-controle-nodesenvolvimento-em-equipe/9627>. Acesso em: 15 ago. 2024.

**Descrição do Bazaar.** Inf.ufsc. Disponível em: [https://www.inf.ufsc.br/~jean.hauck/guias/29110/Norma%20ISO%2029110%20Perfil%20de%20Entrada%20B%C3%A1sico/guidances/examples/descricao\\_bazaar\\_9187F4DC.html](https://www.inf.ufsc.br/~jean.hauck/guias/29110/Norma%20ISO%2029110%20Perfil%20de%20Entrada%20B%C3%A1sico/guidances/examples/descricao_bazaar_9187F4DC.html). Acesso em: 15 ago. 2024.

**DevOps do Microsoft Azure.** Visure Solutions. Disponível em: <https://visuresolutions.com/pt/integra%C3%A7%C3%B5es/DevOps-doMicrosoft-Azure/#:~:text=O%20Microsoft%20TFS%20%C3%A9%20um,colaborativo%20mais%20f%C3%A1cil%20e%20eficiente>. Acesso em: 15 ago. 2024.

**Gerenciar Tempo com Harvest.** Remessa Online. Disponível em: <https://www.remessaonline.com.br/blog/gerenciar-tempo-harvest/>. Acesso em: 15 ago. 2024.

**Git vs Perforce: Como escolher e quando usar ambos.** Perforce. Disponível em: <https://www.perforce.com/blog/vcs/git-vs-perforce-how-choose-and-whenuse-both>. Acesso em: 15 ago. 2024.

**GNU Arch.** Solvusoft. Disponível em: <https://www.solvusoft.com/pt-br/fileextensions/software/andy-tai/gnu-arch/>. Acesso em: 15 ago. 2024.

**IBM® Rational® ClearCase®.** IBM. Disponível em: <https://www.ibm.com/brpt/products/rational-clearcase>. Acesso em: 15 ago. 2024.

**Plastic SCM: Source Control for Game Development.** Kahn Code. Disponível em: <https://kahncode.com/2020/02/19/plastic-scm-source-control-for-gamedevelopment/>. Acesso em: 15 ago. 2024.

**RCS: Revision Control System by Walter F. Tichy.** RUPAREL, Viral. Medium, 19 mar. 2018. Disponível em: <https://medium.com/@viralruparel/rcs-revisioncontrol-system-by-walter-f-tichy-2786b0cb4691>. Acesso em: 15 ago. 2024.

**Visual SourceSafe - Parte 1.** Linha de Código. Disponível em: <http://www.linhadecodigo.com.br/artigo/1645/visual-source-safe-parte-1.aspx>. Acesso em: 15 ago. 2024.

**What is Git?** Atlassian. Disponível em: <https://www.atlassian.com/br/git/tutorials/what-is-git>. Acesso em: 15 ago. 2024.