Mapa Mental Estruturado: Versionamento de Software (GQS - Lab05)

Este documento apresenta o resumo dos termos técnicos do material de aula, com suas definições e exemplos, em um formato estruturado para facilitar a criação de um mapa mental visual.

1. Versionamento de Software (Conceitos Fundamentais)

Termo	Definição	Exemplo Prático
Versionamento	Controle das mudanças no software, permitindo auditoria e retorno a versões anteriores.	Reverter da versão 2.3 (com bug) para a versão 2.2 (estável).
Repositório	Base de código versionada onde todo o histórico de alterações é armazenado.	O projeto da disciplina no GitHub.
Commit	Registro de uma alteração no repositório, contendo autor, data e mensagem descritiva.	"Ana – adicionou validação de senha – 01/09/2025".
Branch	Linha paralela de desenvolvimento, isolando novas funcionalidades ou correções.	Criar a `feature/pagamento-pix` para desenvolver a integração PIX.
Merge	Processo de união das alterações de uma branch em outra (ex: unir uma feature finalizada à branch principal).	Unir a branch `feature/pagamento-pix` à branch `main`.

2. Ferramentas e Plataformas

Termo	Definição	Exemplo Prático
Git	Sistema de controle de versão distribuído (DVCS), criado por Linus Torvalds.	Usar o Git localmente no notebook para controlar as versões do TCC.
GitHub	Plataforma de hospedagem de repositórios Git, com foco em colaboração (Issues, PRs, CI/CD).	Time open source usa *Pull Requests* (PRs) e *GitHub Actions* para testes automáticos.
GitLab	Plataforma alternativa ao GitHub, com foco em um ciclo de DevOps completo e integrado.	Startup usa o pipeline de CI/CD integrado do GitLab para publicar novas versões em produção.
Bitbucket	Serviço de hospedagem de repositórios com forte integração com ferramentas Atlassian (Jira, Confluence).	Rastrear commits diretamente ligados a tarefas do backlog no Jira.

3. Fluxos de Trabalho (Workflows)

Termo	Definição	Exemplo Prático
Git Flow	Fluxo de trabalho complexo com branches dedicadas para *features*, *releases* e *hotfixes*.	Criar `release/2.0` para estabilização e `hotfix/login-bug` para correção crítica em produção.
GitHub Flow	Fluxo de trabalho simples, baseado na branch principal (`main`) e no uso intensivo de *Pull Requests*.	Criar `feature/comentarios` , abrir PR para revisão e fazer merge no `main` após aprovação.
Trunk-Based Development	Prática de integração contínua onde commits pequenos e frequentes são feitos diretamente na branch principal.	E-commerce faz commits no `main` várias vezes ao dia, garantindo estabilidade com testes automáticos.

4. Práticas Modernas e Qualidade

Termo	Definição	Exemplo Prático
Integração Contínua (CI)	Automatização da compilação e execução de testes a cada commit de integração.	O *GitHub Actions* compila o projeto e roda testes unitários a cada push.
Entrega Contínua (CD)	Automatização do processo de liberação do software para ambientes de homologação ou produção.	Publicação automática no ambiente de homologação se todos os testes da CI passarem.
Code Review	Revisão obrigatória do código por pares antes do merge, para garantir qualidade e evitar bugs.	Carlos revisa o *Pull Request* de Ana e sugere melhorias antes de aprovar o merge.
Pull Request (PR)	Mecanismo para propor e discutir alterações, centralizando commits, comentários e aprovações.	A equipe discute a implementação da `feature/chat` no PR antes de integrá-la.

5. Estratégias de Ramificação (Branching)

Termo	Definição	Exemplo Prático
Feature Branch	Branch para desenvolvimento de uma nova funcionalidade isolada.	`feature/pagamento-pix` .
Release Branch	Branch para preparação e estabilização de uma nova versão de lançamento.	`release/2.1` (aceita apenas correções de bugs).
Hotfix Branch	Branch para correção rápida de um bug crítico em produção.	`hotfix/erro-login` .

6. Versionamento Semântico (SemVer)

Componente	Significado	Exemplo Prático
MAJOR	Mudanças incompatíveis (quebra de API).	Versão 3.0.0 (incompatível com código antigo).
MINOR	Novas funcionalidades compatíveis.	Versão 3.2.0 (adiciona suporte a biometria, mantendo compatibilidade).
PATCH	Correções de bugs compatíveis.	Versão 3.2.5 (corrige falha no cálculo de frete).
Estrutura	`MAJOR.MINOR.PATCH`	Versão 2.5.3 (2ª versão principal, 5ª melhoria, 3ª correção).