Nome: Dilcilene Hettwer

**Versionamento de código e Git**

*Importância*

Versionamento é algo importantíssimo quando estamos falando de desenvolvimento de software. Uma ferramenta de versionamento bem organizada torna muito mais fácil a comparação de arquivos, realização de mesclagem de ramos (merge) ou até rollbacks de emergência no ambiente de produção.

*Boas práticas*

1. Utilize o modo imperativo.
2. Nunca use o git “commit-m/message”;
3. Considere incluir o link do card da sua história no commit;
4. Limite o assunto do commit a 50 caracteres;
5. Capitalize o assunto do seu commit;

## O que é Git

O Git é um software de linha de comando. Podemos criar versões do nosso sistema ou voltar para versões anterior facilmente, assim como analisar o histórico do que já foi feito.

### *Repositórios Git*

Um repositório nada mais é que uma pasta onde estão todos os arquivos do nosso projeto, incluindo nosso versionamento.

Isso é uma característica legal do Git: nós trabalhamos de modo descentralizado de desenvolvimento, onde cada pessoa possui uma versão inteira do repositório em sua máquina e só envia os pedaços que alterou para um local onde todo mundo pode baixar essas alterações, porém conseguimos trabalhar até mesmo sem internet e depois enviamos o que fizemos.

### *O que é um commit*

Commits são como fotos da última versão do nosso código. Eles carregam tudo o que foi alterado em nosso projeto para que, quando precisarmos, possamos voltar ao commit onde possuíamos a versão que gostaríamos de utilizar.

A cada nova alteração que fazemos, que seja uma versão do nosso software, podemos fechar um commit e guardar isso no histórico do nosso projeto.

As seguintes regras podem ser seguidas:

1. Separe as linhas de assunto e corpo do commit com uma linha em branco
2. Limite o tamanho da linha do assunto para 50 caractéres (colunas)
3. Letra maiúscula no início do assunto
4. Não termina o assunto com um ponto final
5. Utilize o modo imperativo na linha do assunto
6. Encapsule corpo do commit em 72 colunas
7. Utilize o corpo para explicar o o que e o porque vs. como.

*Benefícios*

O Git usa a concepção de repositório local, onde você tem uma cópia do “repositório completo” do seu projeto. Você pode commitar as modificações que fez nos arquivos do seu projeto para o repositório local. O Git também permite que você trabalhe completamente offline.

Depois, você pode sincronizar ou compartilhar as modificações que fez, quando tiver acesso online ao repositório remoto.

Alguns outros benefícios são: facilidade de trabalhar, os repositórios são menores, a capacidade de mover ou adicionar arquivos, você pode ignorar certos arquivos, staches banches, encontrar a versão que introduziu um bug no usando a busca binária.

*Branches*

O branch chamada master e nesta branch temos a versão oficial do nosso projeto. Quando vamos fazer uma alteração, podemos criar uma ramificação da master e trabalhar somente nela, sem afetar o código oficial. Ao finalizarmos nosso trabalho na nossa ramificação, então podemos juntar o nosso trabalho com a master.

Algumas simples regras podem ser seguidas ao originar seu novo branch:

* O nome deve ser simples conciso, que represente sua tarefa;
* Um único idioma deve ser utilizado;
* Não utilize códigos que não tem sentido quando sem contexto;
* Começar com letras minúsculas;
* Palavras separadas por hífens -.

*Referências:*

<https://woliveiras.com.br/posts/introdu%C3%A7%C3%A3o-a-versionamento-de-c%C3%B3digo-e-conhecendo-o-git/#Entendendosobreversionamento>

<http://ditech.com.br/git-beneficios-ferramenta-projetos/>

<http://blog.victorpre.com/git-melhores-praticas/>

<https://blog.locaweb.com.br/desenvolvedores/5-boas-praticas-de-git/>