

**ENGENHARIA DE SOFTWARE**

6º semestre - Noturno

MARIA GABRIELA COSTA LAZARETTI

1846008-2

**GERENCIAMENTO DE CONFIGURAÇÃO EM PROJETOS OPEN SOURCE  
ECLIPSE**

MARINGÁ  
2020

## **GERENCIAMENTO DE CONFIGURAÇÃO EM PROJETOS OPEN SOURCE**

### **INTRODUÇÃO**

Atualmente, existe no mercado muitos softwares open source, ou seja, que qualquer pessoa pode ter acesso ao código fonte, sugerir alterações e até mesmo trabalhar com seu código. Um desses software é o Eclipse IDE, que falaremos neste trabalho.

### **OBJETIVO**

O objetivo deste trabalho é apresentar as características do projeto Eclipse enquanto estratégia de branching, merging, controle de solicitações, aprovação de solicitações, core review, aprovação de merge request, rastreabilidade de requisitos, testes, critérios de aceitação do merge, controle de versões e de releases.

### **DESENVOLVIMENTO**

#### **1. Eclipse IDE**

Eclipse é uma IDE para desenvolvimento de código em Java, e seu desenvolvimento é open source. Ele também dá suporte à outras linguagens como C/C++, PHP, Python entre outros.

#### **2. Open Source**

Por ser um projeto open source, o Eclipse permite que os usuários: reportem bugs e possíveis melhorias, alterar o código arrumando os bugs encontrados, ou implementando melhorias. Podendo também iniciar um novo projeto ou commitar em um projeto já existente. Para isso, eles possuem em sua página um manual de como funciona o git<sup>1</sup>.

#### **3. Como adicionar um projeto ao controle de versão**

---

<sup>1</sup> Como funciona o git: [https://wiki.eclipse.org/EGit/Git\\_For\\_Eclipse\\_Users](https://wiki.eclipse.org/EGit/Git_For_Eclipse_Users)

Para iniciar o versionamento de um novo projeto ao controle de versão, é necessário primeiramente realizar as configurações do usuário como: identificação (user name e user e-mail), posteriormente é necessário configurar o seu ambiente no windows, adicionando as variáveis do ambiente.

#### **4. Criando um repositório**

Todos os objetos na database são identificados através de uma chave de um hash de 20 bytes SHA-1, o que traz inúmeros benefícios:

- A comparação de dois objetos se dá através de dois hashes;
- Um objeto terá a mesma identificação em todos os repositórios que o contenham;
- Um objeto nunca muda após alterado;
- Se um repositório for corrompido, ele é facilmente detectado verificando se os nomes dos objetos ainda são os hashes seguros.

Ao criar um novo repositório é necessário, primeiramente, se atentar a duas recomendações:

- Não criar um repositório dentro da área de trabalho do Eclipse: é importante sempre tomar cuidado ao criar ou clonar um repositório, e sempre verificar se está usando o Git compartilhado de forma correta;
- Não criar um repositório com um projeto Eclipse como "root".

A página do Eclipse ensina, através de um passo a passo completo, como iniciar um novo repositório, como clonar um repositório existente, como extrair novas informações de uma branch, entre outros, é importante garantir todos os passos para garantir que está sendo executado da forma correta, e ter o melhor aproveitamento do projeto.

#### **5. Branch**

As branches são criadas com seus nomes referenciando um commit. Existem dois tipos de branches:

- Branches locais: são utilizadas para commitar alterações do repository. Ela ajuda com tasks informando um nome local pelo qual o commit pode ser encontrado.

- Branches remotas de acompanhamento: são criadas automaticamente ao clonar e obter repositórios remotos. Sua nomeação segue algumas convenções. A branch remota aponta para o mesmo commit correspondente ao repositório remoto.

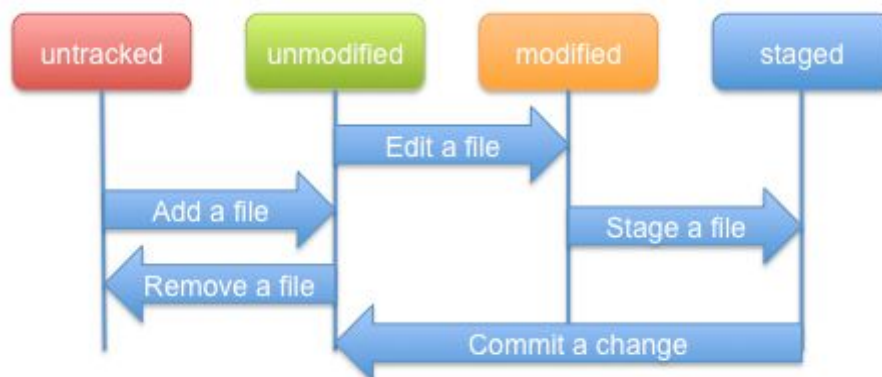
## 6. Gravando alterações no Repositório

Cada arquivo no diretório de trabalho pode ser rastreado ou não:

- Os arquivos rastreáveis são aqueles que foram colocados recentemente no índice.
- Os arquivos não rastreáveis são todos os outros arquivos.

Quando você clona um repositório, todos os arquivos do diretório serão rastreados e não serão modificados até que você inicie a editá-lo.

Enquanto você editar os arquivos, o git irá reconhecer que eles foram modificados, respeitando o último arquivo commitado.



## 7. Gerrit Code Review

Ela é uma ferramenta de revisão de código. Quando é utilizado, o repositório clonado é configurado de forma:

- Retorna revisão do código;
- Envia os commits para uma revisão; e
- As mensagens de commit automaticamente recebem um "Gerrit Change-ID".

O manual também ensina como utilizar o Gerrit Code Review Server, para garantir revisão.

## 8. Gitflow

O Eclipse trabalha com o modelo de Gerenciamento de Código Gitflow, que é baseado em feature, release e hotfix.

## 9. Quickdiff

É utilizado para comparar as diferenças entre os commits. Ele pode ser configurado em **General > Editors > Text Editors > Quick Diff**, definindo as cores que o texto ficará destacado para: mudanças, adições e deletados.

## 10. Commitando alterações

Para modificar um projeto que já foi compartilhado no git, é necessário excluir ou modificar os arquivos dentro do Eclipse ou se preferir pode ser feito diretamente no sistema de arquivos. Segundo a documentação, não é necessário informar com antecedência pelo Git.

Novos arquivos que devem ter um controle de versões, devem ser explicitamente submetidos ao versionamento do Git:

- Commitando: tem duas formas de commitar alterações com o EGit, utilizando o “Staging View” ou verificando diretamente a caixa de diálogo de commit.
- Commitando mensagem: deve especificar uma mensagem que esclareça quais alterações foram realizadas. Uma boa prática é indicar em uma linha curta, pular uma linha e a seguir o corpo de mensagem.

## 11. Merging

O *merging* realiza a mesclagem entre as alterações realizadas e o *branch* ou *tag*. Ela pode ser realizada a partir de visualização de histórico, pelo menu equipe ou diretamente no Git.

Na página, eles dão opções e explicam como trabalhar com: Rebasing, Cherry Picking, Tagging, Patches, Working with Tasks, Working with Submodules, GIT LFS Support.

Managing Repositories: apresentam como Adding Repositories to the Git Repositories View, Removing Repositories, Structure of the Git Repositories View, Repository Groups.

## REFERÊNCIAS

Eclipse Foundation. **Eclipse IDE**. Disponível em:  
<<https://www.eclipse.org/eclipseide/>>. Acesso em 15 set. 2020.

Eclipse Foundation. **Get Involved!** Disponível em:  
<<https://www.eclipse.org/contribute/>> Acesso em 15 set. 2020.

Eclipse Foundation. **EGit/User Guide**. Disponível em:  
<[https://wiki.eclipse.org/EGit/User\\_Guide#Possible\\_merge\\_results](https://wiki.eclipse.org/EGit/User_Guide#Possible_merge_results)> Acesso em 15 set. 2020.