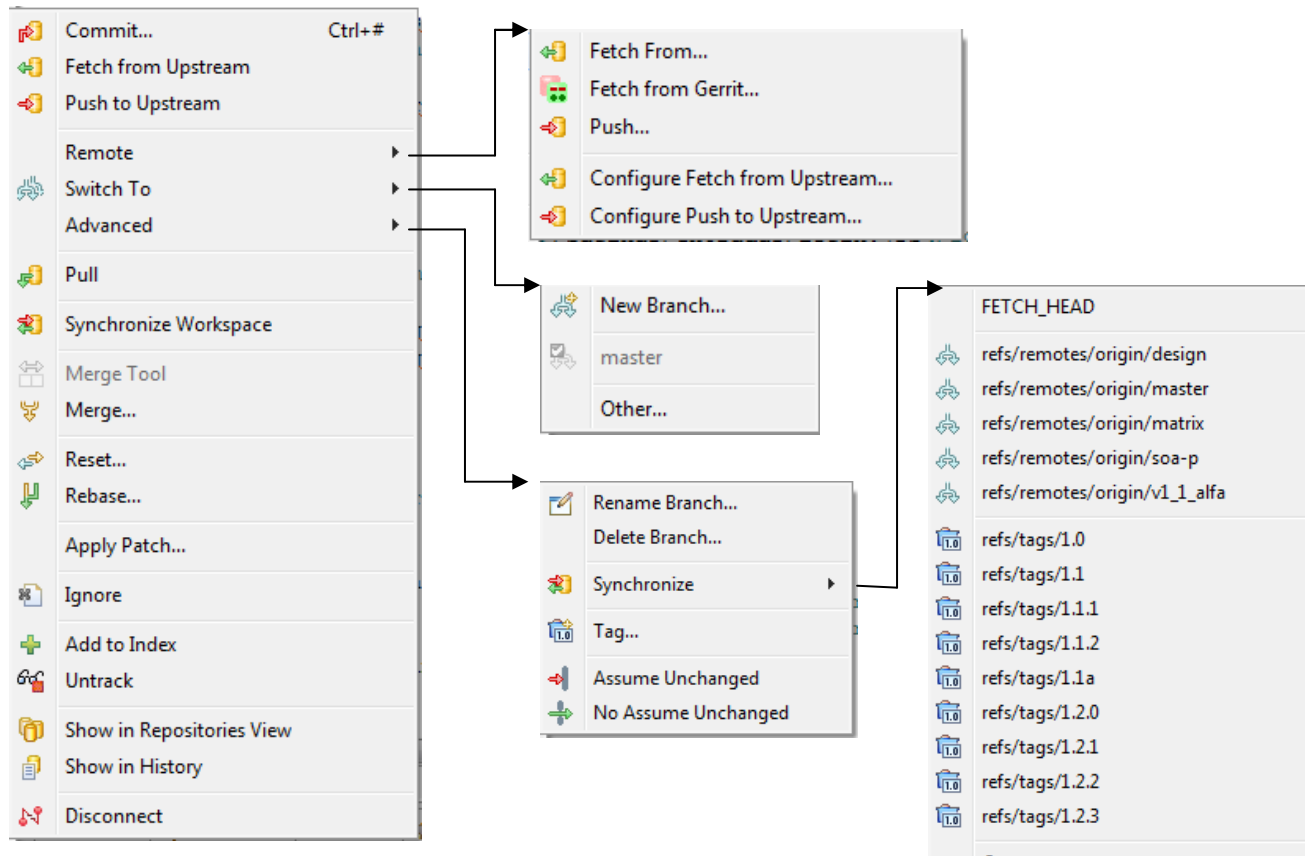


Anexo 8 - GIT no Eclipse/JBDS - EGit

8.1 - Visão geral pelo menu do produto

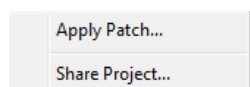
Quando se clica em qualquer projeto gerenciado pelo GIT, a seguinte árvore de menus é aberta.



8.2 - Projetos gerenciados e não gerenciados pelo GIT

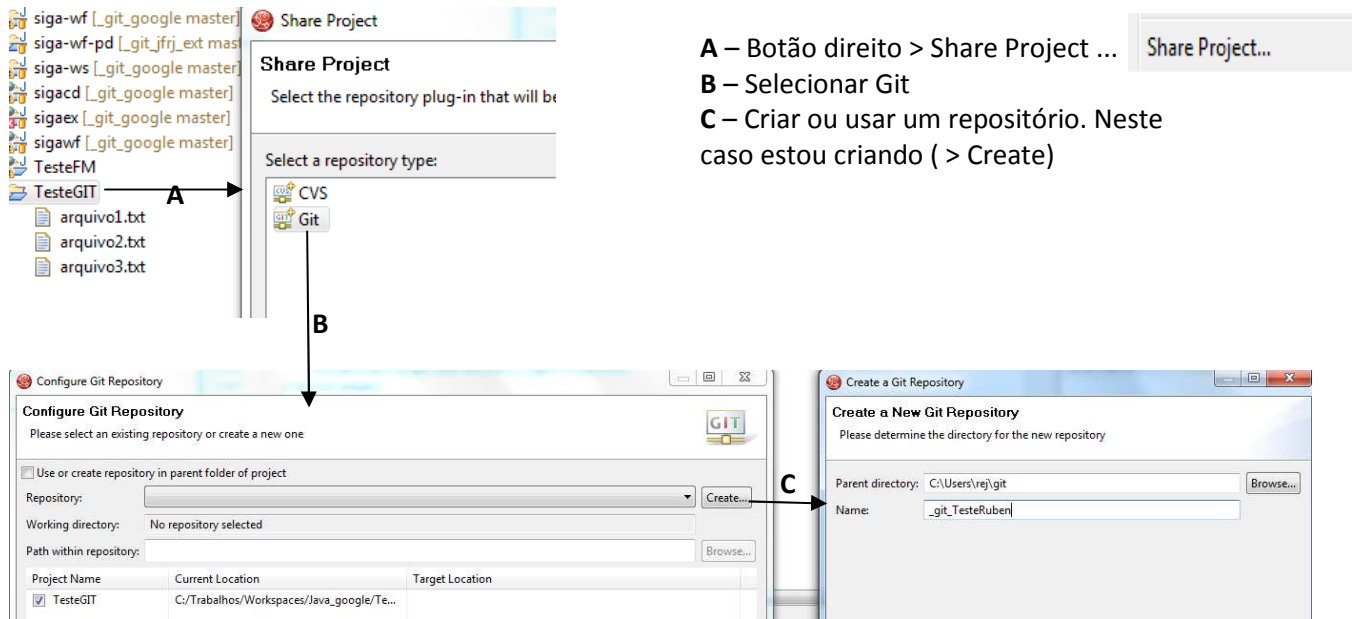
▷ siga-wt [git_google master]	}	Gerenciado, informando o repositório e o branch entre [], ou seja, [repositório branch]. O head do Git está apontando para o branch master.
▷ siga-wf-pd [git_jfrj_ext master]		
▷ siga-ws [git_google master]		
▷ sigacd [git_google master]		
▷ sigaex [git_google master]		
▷ sigawf [git_google master]		
▷ TesteFM	}	Não gerenciado, sem repostório
▷ TesteGIT		

Se colocarmos o cursor em TesteGIT, as seguintes opções serão fornecidas, visto que o projeto não está ainda gerenciado.

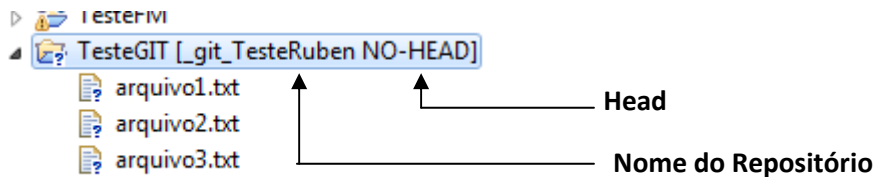


8.3 - SHARE PROJECT (Criando um repositório)

Vamos colocar o projeto testeGIT gerenciado pelo GIT.

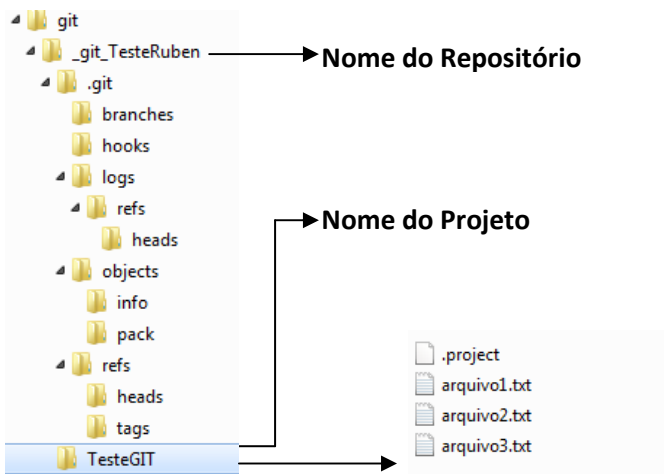


Após isso, o projeto passa a ser gerenciado:



Observar que:

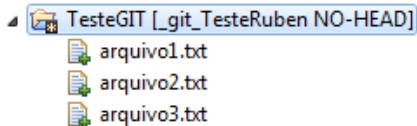
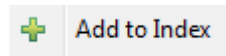
- a - não temos ainda um head porque não fizemos ainda nenhum commit e não temos um branch;
- b - ? nos arquivos, informando que os mesmos não estão ainda sob controle de versão;
- c - como está tudo fisicamente no disco (em `c:\users\rej\git`)





8.4 - ADICIONANDO O PROJETO / ARQUIVOS PARA CONTROLE DE VERSÃO

Como vimos anteriormente, os arquivos do projeto TesteGIT não estão sob controle de versão, e desta forma, devemos adicioná-lo.

Colocando o cursor no projeto TesteGIT > Team > Add to Index

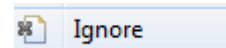


Observar que o ícone mudou de  para  informando que o projeto está sob controle de versão.

8.5 - RETIRANDO ARQUIVOS DO CONTROLE DE VERSÃO

E se não quisermos que um arquivo / diretório (como o /bin por exemplo) fique sob o controle de versão do GIT. É possível? Sim, utilizando-se a opção Ignore ...

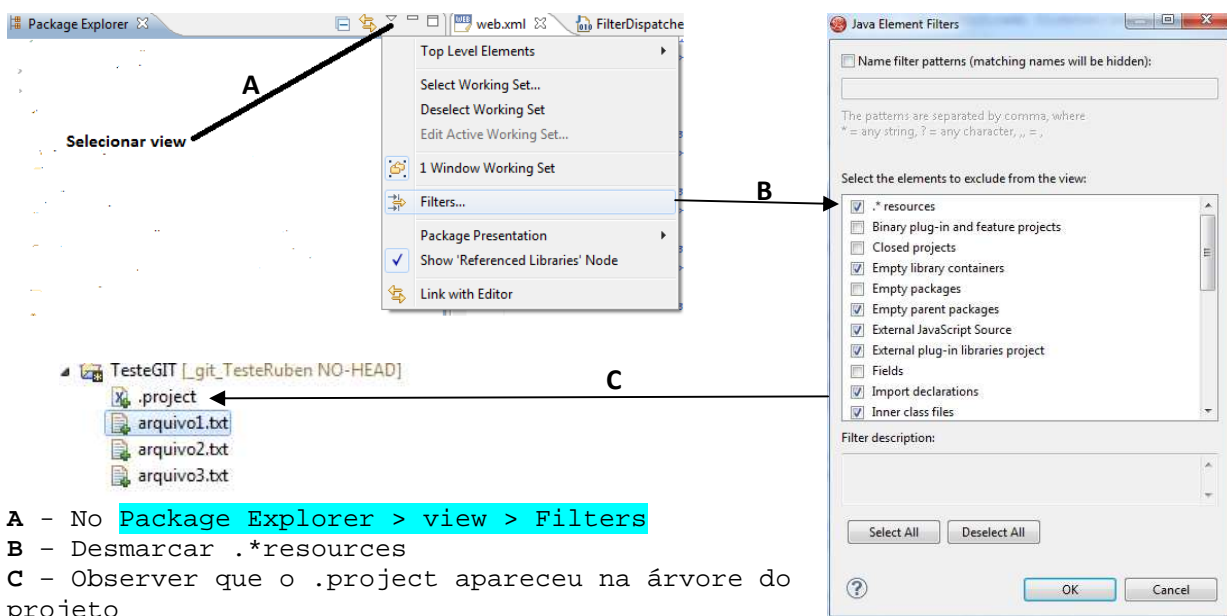
Coloque o cursor no arquivo / diretório > Team > Ignore



Ou, crie um arquivo chamado **.gitignore** na raiz do projeto e coloque o nome do diretório (ex: /bin) ou arquivo (ex: /bin/xpto.class) a ser ignorado



OK, porém não consigo ver o .gitignore ... o que fazer? O problema é que o eclipse / JBDS deve estar configurado para não exibir arquivos começando com ., e nesta caso proceda conforme ilustrado abaixo:



A - No Package Explorer > view > Filters

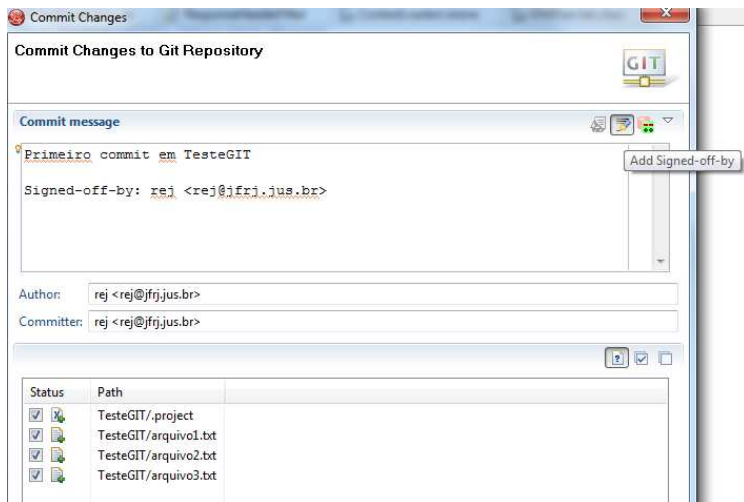
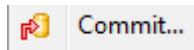
B - Desmarcar .*resources

C - Observer que o .project apareceu na árvore do projeto

8.6 - COMMITANDO O PROJETO

Vamos agora criar o primeiro commit do projeto.

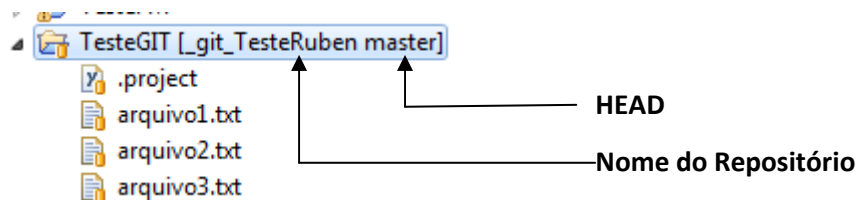
Cursor no projeto > team > Commit





Observar o Add Signed-off-by, que clicado inclui na mensagem do commit.

Autor: pode-se mudar o nome do autor, informando que o committer está fazendo o commit em nome de terceiro

Como resultado, teremos:



Observar que:

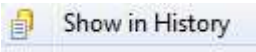
- a - após o commit, o ícone do projeto mudou de  para  informando que o projeto já foi commitado. O commit representa incluindo seu conteúdo e sua história;
- b - já temos um HEAD apontando para o branch default master.

Transição de estado dos recursos do projeto conforme o GIT:



8.7 - HISTÓRIA DO PROJETO

Cursor no projeto > Team > Show in History



commit a7d1f842b2a3d996df5c6ab5eb6435283189a8c4
Author: rej <rej@jfrj.jus.br> 2012-10-10 14:02:46
Committer: rej <rej@jfrj.jus.br> 2012-10-10 14:02:46
Branches: master

Primeiro commit em TesteGIT
Signed-off-by: rej <rej@jfrj.jus.br>

History
Mensagem do commit
Conteúdo do commit

TesteGIT/.project
TesteGIT/arquivo1.txt
TesteGIT/arquivo2.txt
TesteGIT/arquivo3.txt

Farei uma modificação no arquivo1.txt, adicionando uma quarta linha (Linha 4) e commitarei as mudanças.

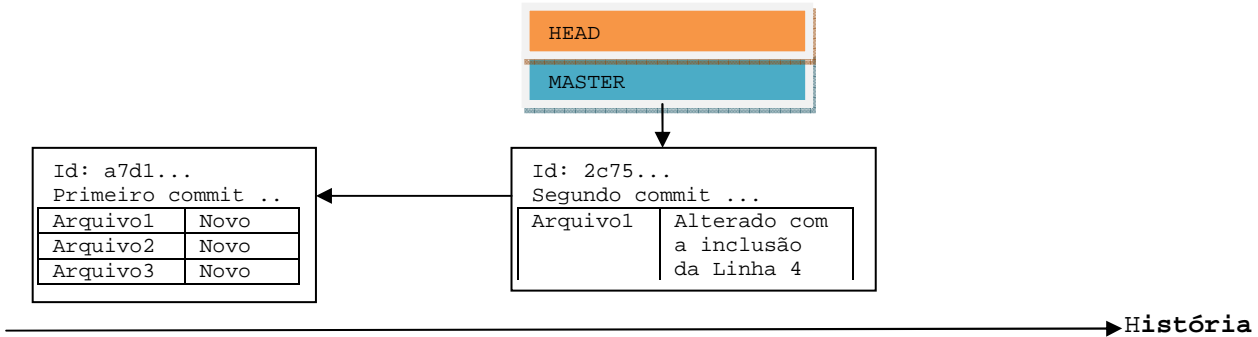
commit a7d1f842b2a3d996df5c6ab5eb6435283189a8c4
Author: rej <rej@jfrj.jus.br> 2012-10-10 14:02:46
Committer: rej <rej@jfrj.jus.br> 2012-10-10 14:02:46
Branches: master

Segundo commit. Alterei o arquivo1 adicionando uma quarta linha (Linha 4)
Signed-off-by: rej <rej@jfrj.jus.br>

Primeiro commit em TesteGIT
Signed-off-by: rej <rej@jfrj.jus.br>

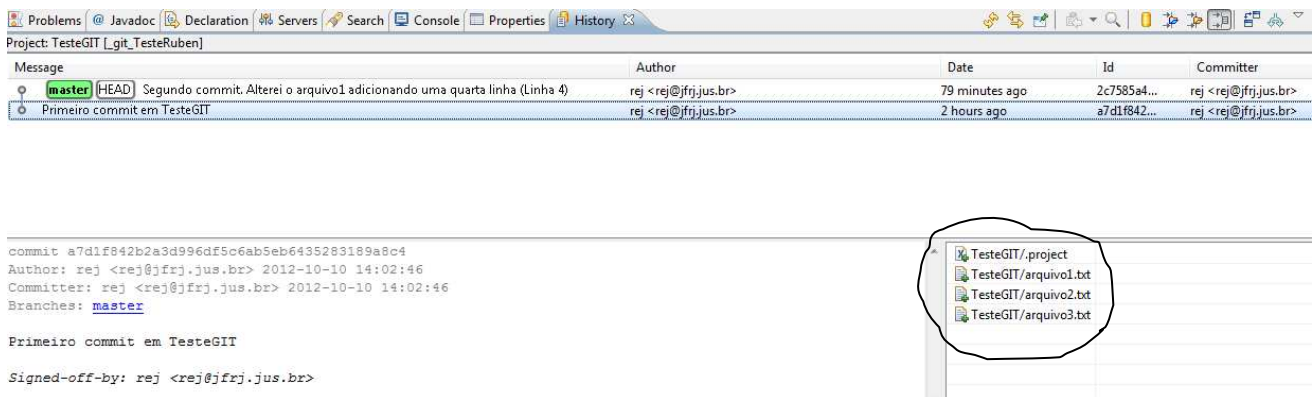
TesteGIT/.project
TesteGIT/arquivo1.txt
TesteGIT/arquivo2.txt
TesteGIT/arquivo3.txt

Assim está, diagramaticamente, o projeto para o GIT:



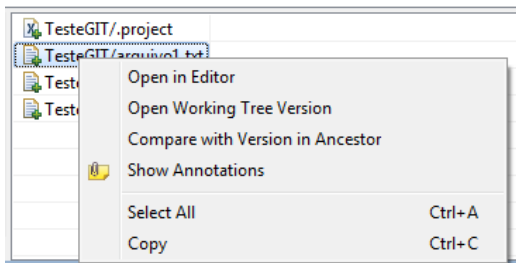
O que podemos fazer agora que temos 2 versões do projeto?

8.7.1 - Clicando no primeiro commit

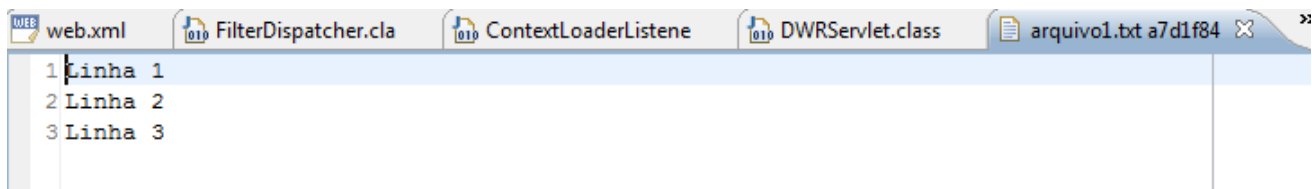


Observamos todos os arquivos que estão nesta versão.

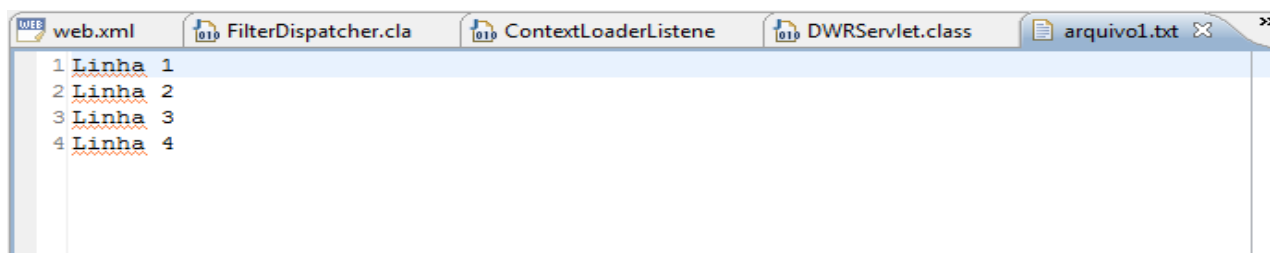
Ao clicarmos no arquivo1.txt teremos as seguintes opções:



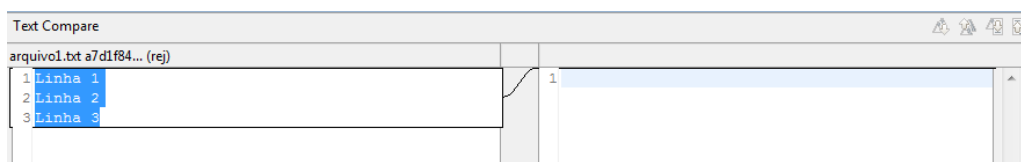
Open in Editor: aparecerá a versão antiga e no editor aparecerá arquivo1.txt at a7d1... (onde a7d1... é o hash sha-1 do primeiro commit).



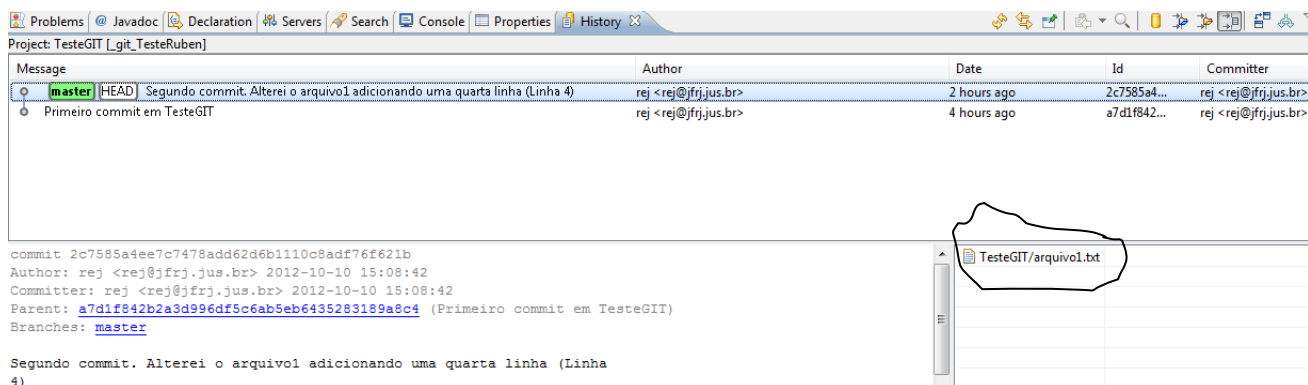
Open Working Tree Version: aparecerá a versão mais nova e no editor aparecerá arquivo1.tx.



Compare with Version in Ancestor: compara com a versão do ancestral, porém como não tínhamos nenhum commit antes, não temos o ancestral e a janela da versão anterior aparecerá vazia.

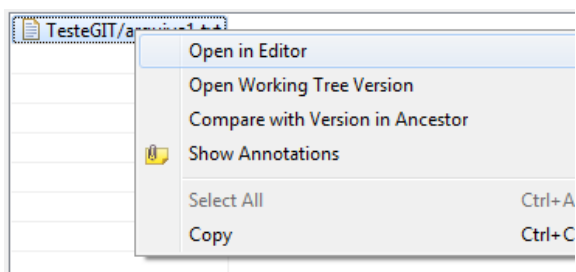


8.7.2 - Clicando no segundo commit

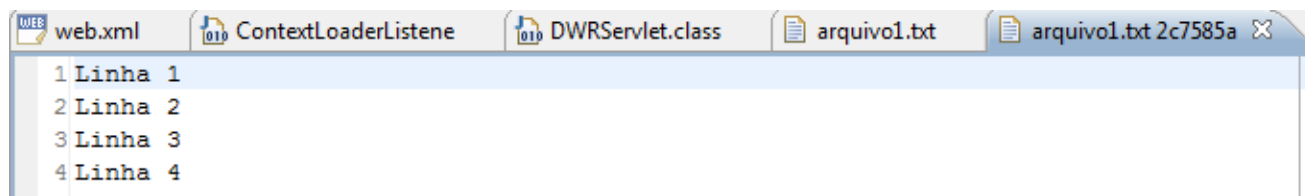


Observamos todos os arquivos que estão nesta versão.

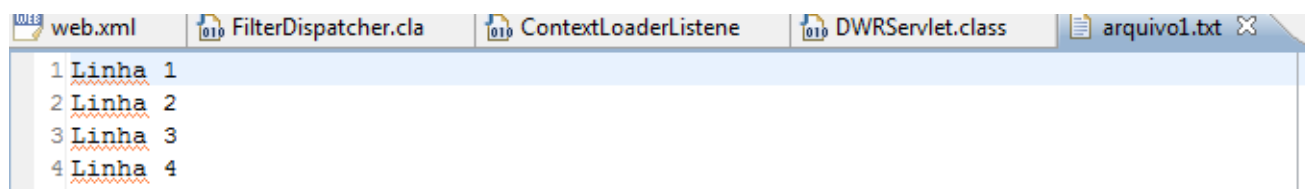
Ao clicarmos no arquivo1.txt teremos as seguintes opções:



Open in Editor: aparecerá a versão nova e no editor aparecerá arquivo1.txt at 2c75... (onde 2c75... é o hash sha-1 do segundo commit).



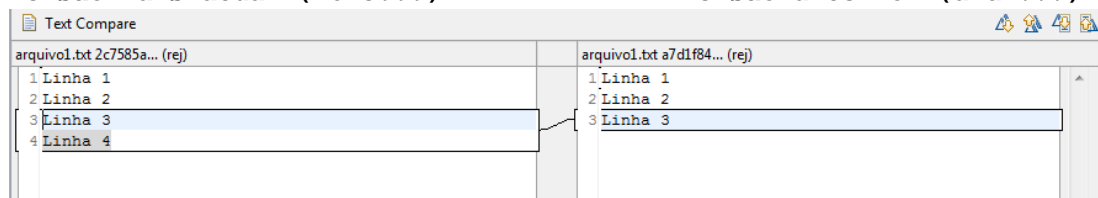
Open Working Tree Version: aparecerá a versão mais nova e no editor aparecerá arquivo1.tx.



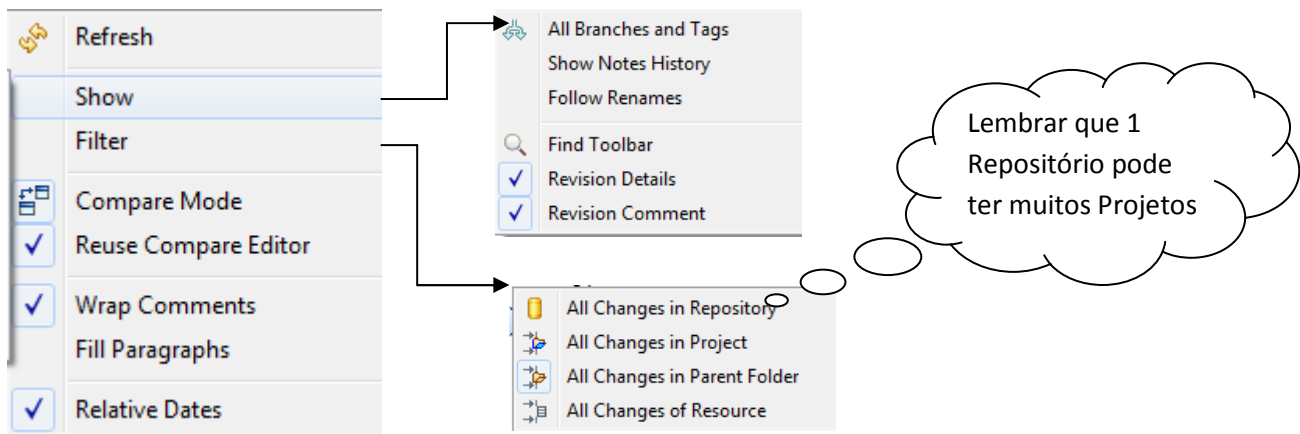
Compare with Version in Ancestor: compara com a versão do ancestral.

Versão mais atual (2c75...)

Versão anterior (a7d1...)



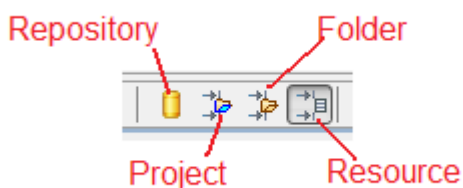
8.7.3 - Outras Opções quando analisando o History de um projeto



Para estes exemplos utilizarei os projetos do SIGA como referência, visto que eles possuem uma história mais complexa, e consequentemente mais interessante, para fins de consulta do que o nosso minúsculo projeto testeGIT.

FILTER:

Vamos supor que queremos analisar a classe FuncoesEL.java. Quantas vezes foi alterada? Quando? Por quem?



Selecionando um recurso (neste caso, FuncoesEL.java):

Com o cursor no recurso (neste caso FuncoesEL.java) > Team > Show in History > View Menu >

All Changes in Repository

8.7.3.1 - Show All Changes in Repository containing the selected resource

If the "Repository" button is down, the commit log is not filtered and shows all commits reachable from the currently checked out branch (or all commits, see below about the "All Branches" action)

All Changes in Project

8.7.3.2 - Show All Changes in Project containing the selected resource

If the "Project" button is down, the commit log is filtered to show all commits which **affected any of the resources in the project containing the current input.**

All Changes in Parent Folder

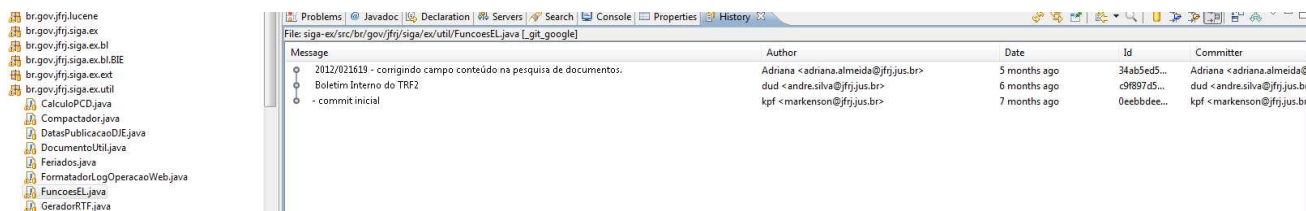
8.7.3.3 - Show All Changes in parent Folder of the selected resource

If the "Folder" toggle is down, the commit log is filtered to show all commits which **affected any of the resources in the parent folder of the current input**. Quem é o parent folder do FuncoesEL.java? É siga-ex/src/BR/gov/jfrj/siga/ex/util. Neste caso, serão listadas todas as mudanças que ocorreram em qualquer objeto deste folder, seja em FuncoesEL.java, seja em Notificador.java e etc.

All Changes of Resource

8.7.3.4 - Show All Changes of selected resource and its children

If the "Resource" button is down, the commit log is filtered to show only commits which **affected the current input**; the view menu item Show > Follow Renames allows to toggle whether renames of the selected resource should be followed by this filter.



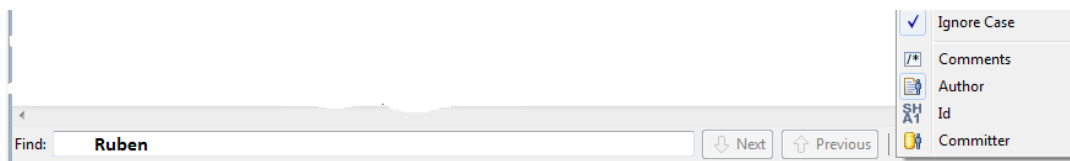
Observamos que:

- O recurso, FuncoesEL.java, foi alterado 3 vezes, com os autores e data acima exibidos;
- Caso o recurso existisse em mais de um projeto no repositório, seriam exibidas todas as ocorrências.

SHOW:

Show Find Toolbar

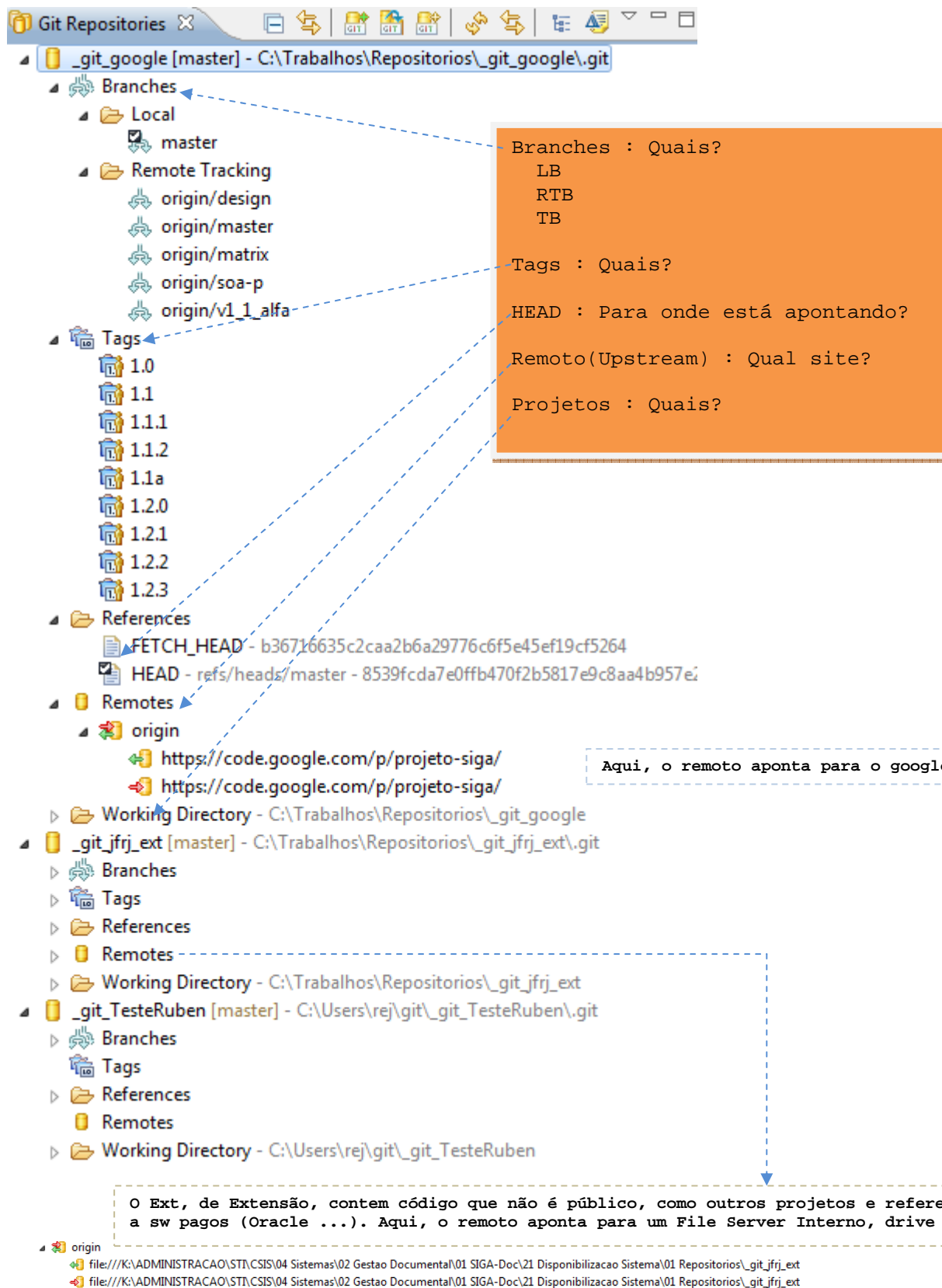
Permite pesquisar por autor, commiter, mensagem do commit e etc.



8.8 - Analisando Repositórios Git

Temos uma View do Git por Repositórios. Esta View abre outro menu referente aos Repositórios. É interessante porque temos acesso a todos dados como branches, tags e endereços dos sites remotos.

Cursor no Projeto > Team > Show In Repositories View  Show in Repositories View



The screenshot displays the 'Git Repositories' view in Visual Studio, showing a tree of repositories and their branches, tags, and remotes. The tree is organized as follows:

- _git_google [master] - C:\Trabalhos\Repositorios_git_google\.git**
 - Branches**
 - Local
 - master
 - Remote Tracking
 - origin/design
 - origin/master
 - origin/matrix
 - origin/soa-p
 - origin/v1_1_alfa
 - Tags**
 - 1.0
 - 1.1
 - 1.1.1
 - 1.1.2
 - 1.1a
 - 1.2.0
 - 1.2.1
 - 1.2.2
 - 1.2.3
 - References**
 - FETCH_HEAD - b36716635c2caa2b6a29776c6f5e45ef19cf5264
 - HEAD - refs/heads/master - 8539fcd7e0ffb470f2b5817e9c8aa4b957e2
 - Remotes**
 - origin
 - https://code.google.com/p/projeto-siga/
 - https://code.google.com/p/projeto-siga/
 - Working Directory - C:\Trabalhos\Repositorios_git_google**
- _git_jfrj_ext [master] - C:\Trabalhos\Repositorios_git_jfrj_ext\.git**
 - Branches
 - Tags
 - References
 - Remotes
 - Working Directory - C:\Trabalhos\Repositorios_git_jfrj_ext
- _git_TesteRuben [master] - C:\Users\rej\git_git_TesteRuben\.git**
 - Branches
 - Tags
 - References
 - Remotes
 - Working Directory - C:\Users\rej\git_git_TesteRuben

Annotations and Explanations:

- Branches : Quais?**
 - LB
 - RTB
 - TB
- Tags : Quais?**
- HEAD : Para onde está apontando?**
- Remoto(Upstream) : Qual site?**
- Projetos : Quais?**

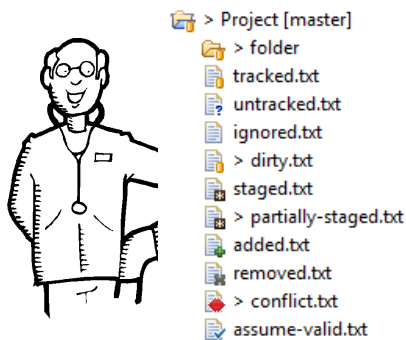
Aqui, o remoto aponta para o google

O Ext, de Extensão, contem código que não é público, como outros projetos e referentes a sw pagos (Oracle ...). Aqui, o remoto aponta para um File Server Interno, drive K

origin

- file:///K:/ADMINISTRACAO/STI/CSIS/04 Sistemas/02 Gestao Documental/01 SIGA-Doc/21 Disponibilizacao Sistema/01 Repositorios_git_jfrj_ext
- file:///K:/ADMINISTRACAO/STI/CSIS/04 Sistemas/02 Gestao Documental/01 SIGA-Doc/21 Disponibilizacao Sistema/01 Repositorios_git_jfrj_ext

Inspencionando o estado do repositório:
O que significam todos aqueles ícones?














Observação:

Não é só o ícone que acusa problema.

> É sinal de problema



Mesmos ícones, porém o sinal de > indica problema.

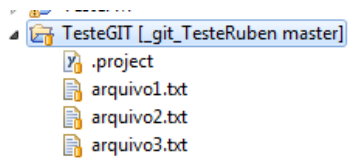
dirty (folder)  > folder	At least one file below the folder is dirty; that means that it has changes in the working tree that are neither in the index nor in the repository.
tracked  tracked.txt	The resource is known to the Git repository and hence under version control.
untracked  untracked.txt	The resource is not known to the Git repository and will not be version controlled until it is explicitly added.
ignored  ignored.txt	The resource is ignored by the Git team provider. The preference settings under Team > Ignored Resources , "derived" flag and settings from .gitignore files are taken into account.
dirtyv  > dirty.txt	The resource has changes in the working tree that are neither in the index nor in the repository.
staged  staged.txt	The resource has changes which have been added to the index. Note that adding changes to the index is currently possible only in the commit dialog via the context menu of a resource.
Partially-staged  > partially-staged.txt	The resource has changes which are added to the index and additional changes in the working tree that neither reached the index nor have been committed to the repository.
added  added.txt	The resource has not yet reached any commit in the repository but has been freshly added to the Git repository in order to be tracked in future.
removed  removed.txt	The resource is staged for removal from the Git repository.
conflict  > conflict.txt	A merge conflict exists for the file.
Assume-valid  assume-valid.txt	The resource has the "assume unchanged" flag. This means that Git stops checking the working tree files for possible modifications, so you need to manually unset the bit to tell Git when you change the working tree file. Also see Assume unchanged action

Lembrando que:

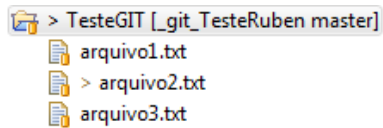
History	Contém a história (todas as versões) do repositório (projetos / arquivos). FILES COMMITED.
Stage (Index)	Conterá todos os arquivos modificados no diretório de trabalho e que estão prontos para ser commitados. É uma área intermediária. FILES TO GO IN NEXT COMMIT.
Working Directory	Diretório de Trabalho. Conterá os projetos / arquivos que o desenvolvedor está trabalhando. São estes projetos / arquivos que serão expostos a IDE Eclipse ou JEDS. FILES THAT YOU SEE. Também referenciado como WorkSpace e Working Tree

Exemplo:

No nosso projeto de teste, TesteGIT



Altere o arquivo2.txt, adicionando uma quarta linha



A imagem informa que o arquivo2.txt está sujo, ou seja, foi alterado no Workspace (Working Tree ou Working Directory), e está diferente da versão do Stage (ou Index) e/ou do repositório (História)

8.9 - Sincronização

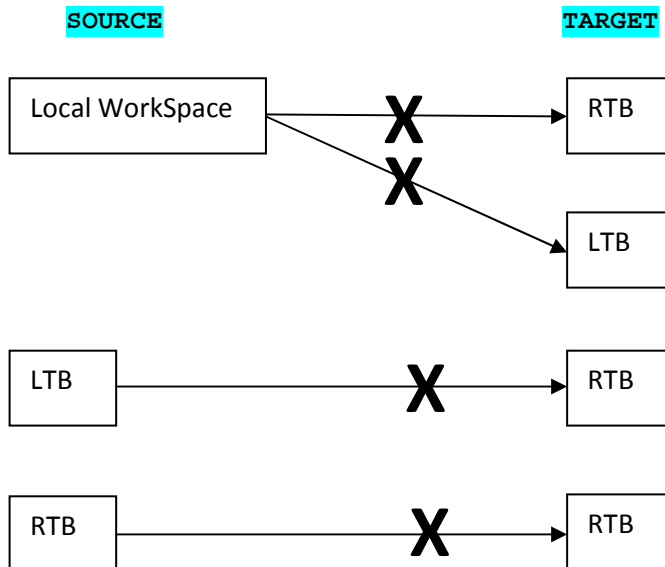
Permite Comparar os recursos entre:

Local WorkSpace

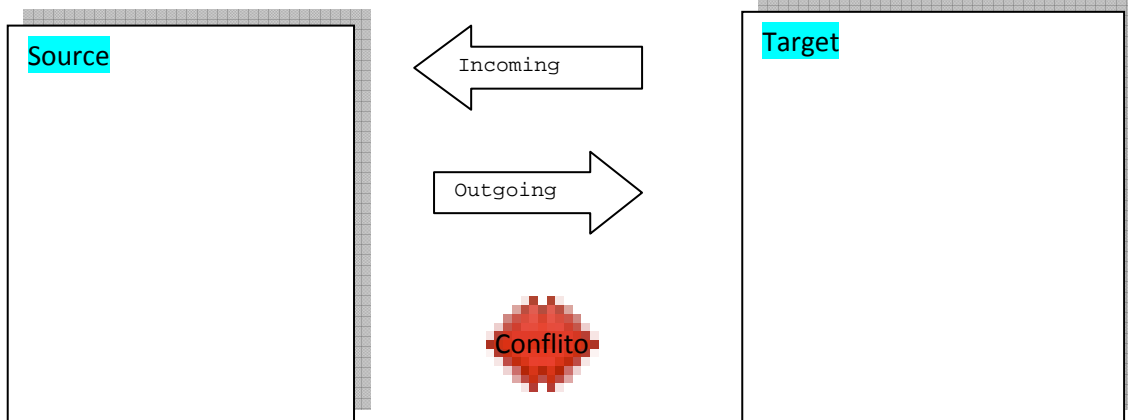
RTB: Remote Tracking Branch

LTB: Local Tracking Branch ou TB, Tracking Branch

Quais combinações são possíveis?



Que tipo de informações são geradas?



Em que granularidade?

+ Adição do recurso

- Exclusão de recurso

Alteração de recurso











Exemplo:  Significa que um recurso existe (foi adicionado) no Target e Não existe no Source

Tabela completa de inconsistências

Ícone	Descrição
INCOMING: Existe no target e não no source	
	An incoming addition means that a resource has been added to the target branch.
	An incoming change means that the file has changed in the target branch.
	An incoming deletion means that a resource was deleted from the target branch.
OUTGOING: Existe no source e não no target	
	An outgoing addition means that the file was added to your workspace or source branch and is not yet in the target branch.
	An outgoing change means that the file was changed in your workspace or source branch.
	An outgoing deletion is a resource that has been deleted in your workspace or source branch.
CONFLITO: Existe no target e no source	
	A conflicting addition means that the resource has been added in your workspace or source branch and in the target branch.
	A conflicting change means that the file has been changed in your workspace or local branch and in the target branch. A manual or automatic merge will be required. Also, any entries in the view that contain children that are conflicts will also be decorated with the conflict icon. This is done to make conflicts easy to find.
	A conflicting deletion means that the resource was deleted in your workspace or source branch and in the target branch.

Exemplo:

No final do item anexo 8.8 fizemos um exemplo. Agora vamos realizar o synchronize:

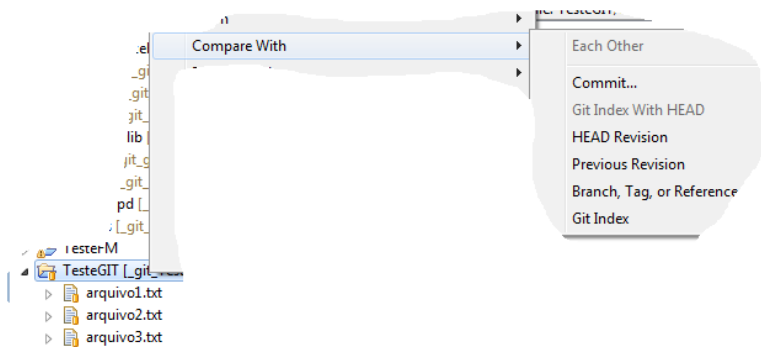


A figura indica que o arquivo2.txt foi mudado no Workspace e não foi mudado no target (branch master).

8.10 - Comparar Conteúdos - Compare with

Comparar versões (conteúdo) de recursos, tais como: arquivos, pastas ou projetos.

Com o cursor em um projeto:



Como o cursor em um arquivo:



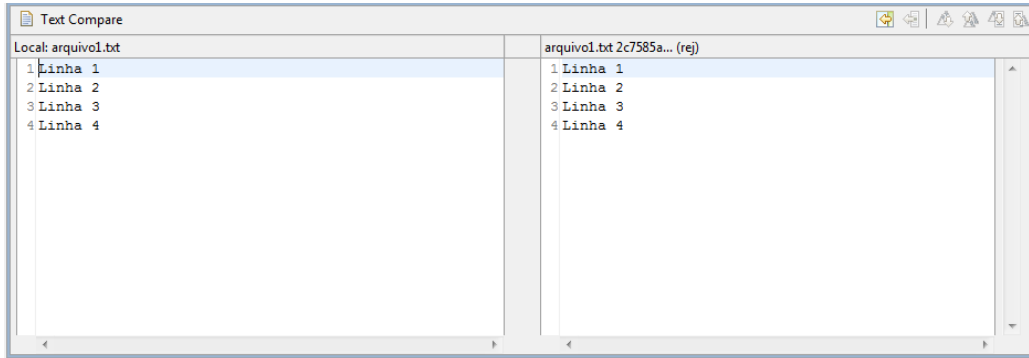
Permite comparar recursos pelas três camadas do Git:

History	Contém a história (todas as versões) do repositório (projetos / arquivos). FILES COMMITTED.
Stage (Index)	Conterá todos os arquivos modificados no diretório de trabalho e que estão prontos para ser commitados. É uma área intermediária. FILES TO GO IN NEXT COMMIT.
Working Directory	Diretório de Trabalho. Conterá os projetos / arquivos que o desenvolvedor está trabalhando. São estes projetos / arquivos que serão expostos a IDE Eclipse ou JBDS. FILES THAT YOU SEE. Também referenciado como Workspace e Working Tree

8.10.1 - Comparando a Working Tree com o último commit (HEAD Revision)

The difference between a resource in the current working directory and in the last commit in the current branch can be viewed from the context menu

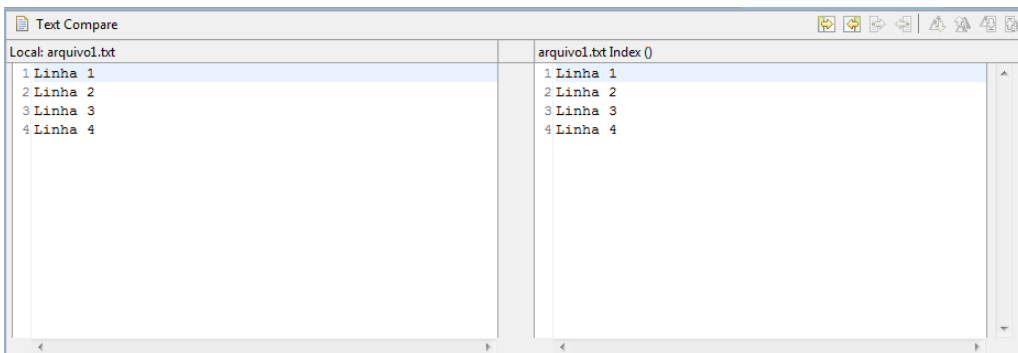
Compare With > HEAD revision. This feature is also available in the Commit dialog. Double clicking on an entry in the Commit dialog opens a compare dialog.



8.10.2 - Comparando Working Tree com Git Index

The differences between the current working tree and the index (based on the currently selected resource) can be viewed from the context menu

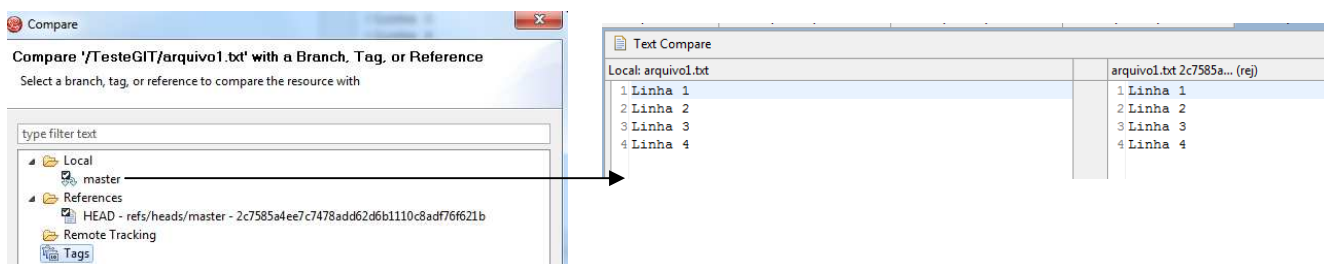
Compare With > Git Index.



8.10.3 - Comparando Working Tree com um branch, uma tag ou uma reference

- Select a resource;
- right-click **Compare With > Branch, Tag, or Reference...**;
- select a branch, tag or reference.

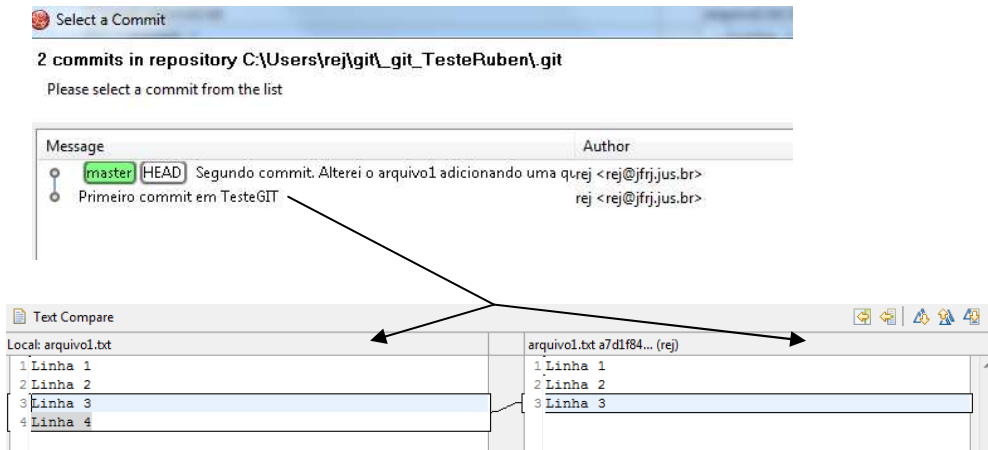
Será apresentada uma janela para você selecionar o Branch, Tag ou Reference



8.10.4 - Comparando Working Tree com qualquer Commit

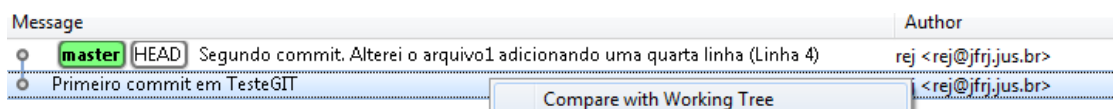
A partir do *project explorer*:

- Select a resource;
- right-click **Compare With > Commit...**;
- select a commit from the commit graph.



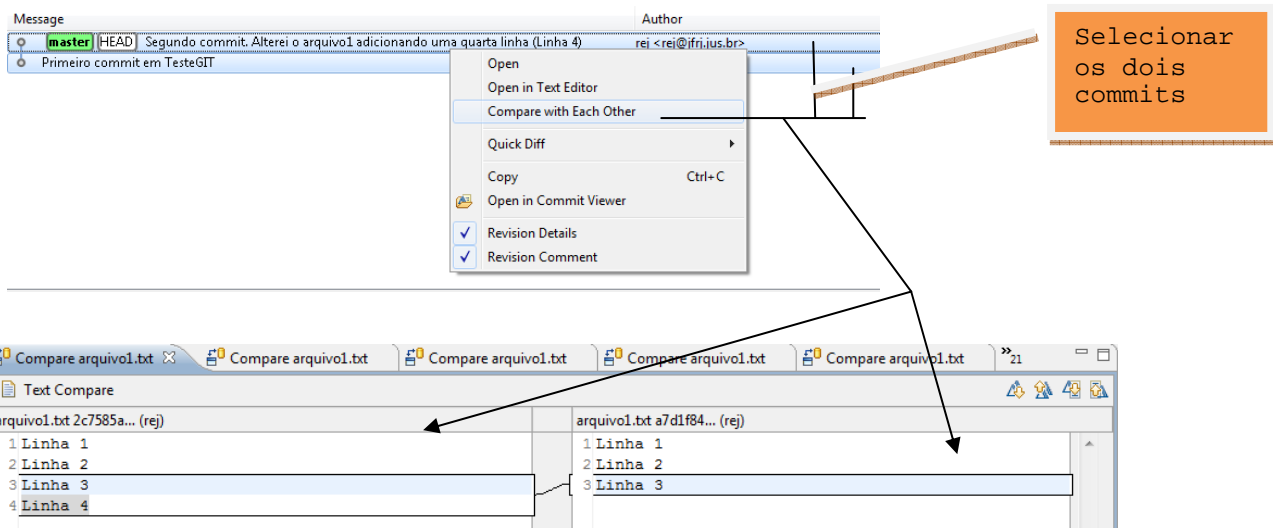
A partir da *history view* (arquivos somente):

- Select a file in the package explorer;
- right-click **Team > Show in History** or **Compare With > History...**;
- in the commit graph select a commit;
- from the context menu select **Compare with working tree**;
- this will open a compare dialog showing the changes between the selected commit and the current working tree.



8.10.5 - Comparando Dois Commits

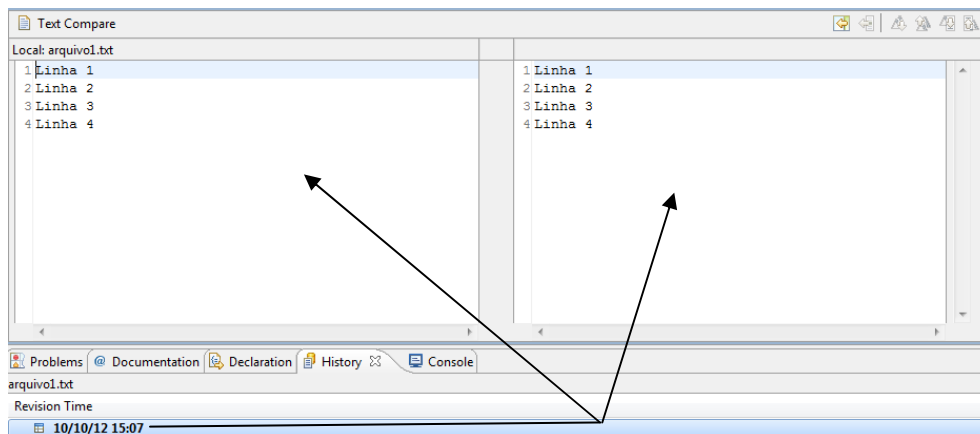
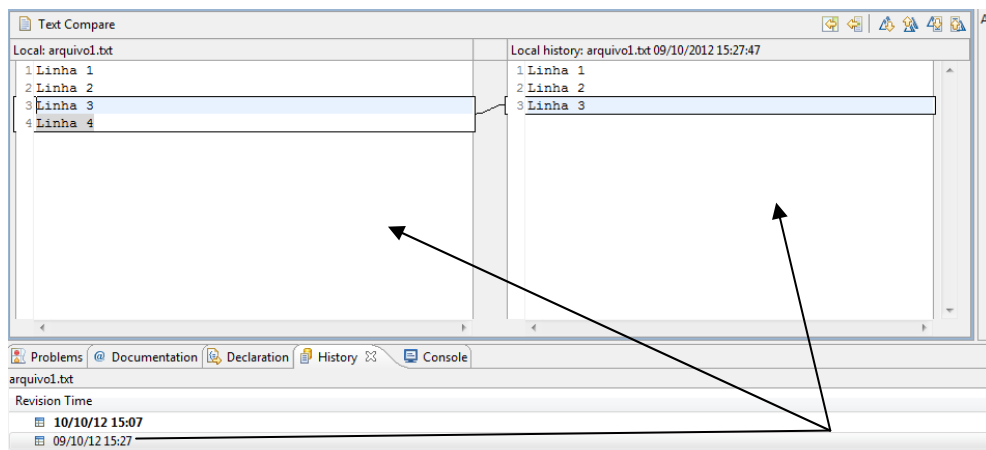
- Select a resource in the Package Explorer;
- click **Team > Show in History** or **Compare With > History...** (the latter for files only);
- in the commit graph select two commits;
- right-click **Compare with each other**;
- this will open a compare dialog showing the changes between the two selected commits;
- you can also open a Git Tree Compare view by right-clicking **Compare with each other in Tree**.



8.10.6 - Comparando Working Tree com Local History

- Select a file in the package explorer;
- **Compare With > Local History...**;

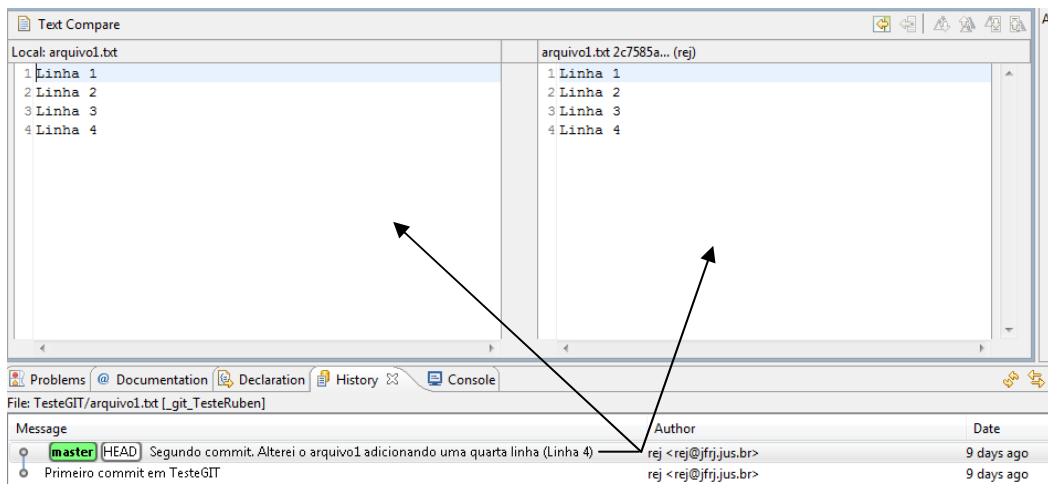
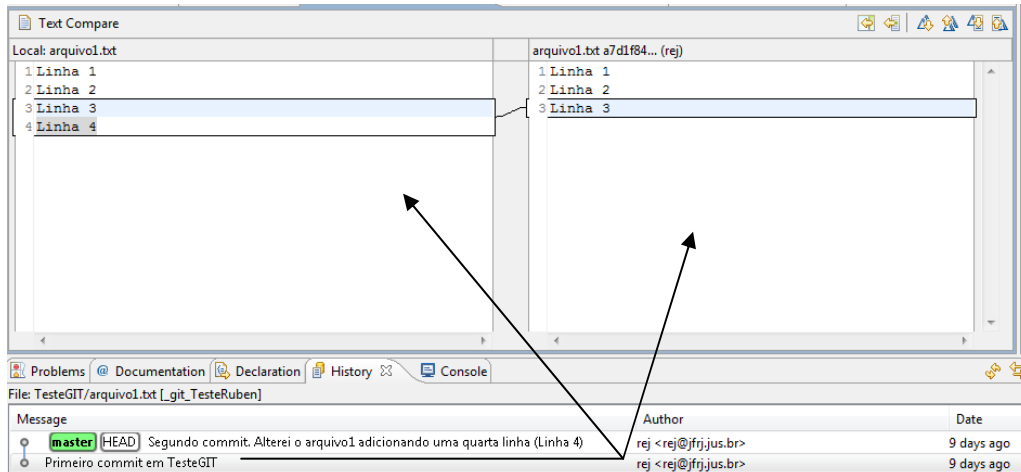
Será apresentado um Revision Time para que o usuário selecione uma revisão.



8.10.7 - Comparando Working Tree com History

- Select a file in the package explorer;
- **Compare With > History...**;

Será apresentado um Commit History para que o usuário selecione um commit.



8.10.8 - Comparando Index com HEAD ou qualquer outro Commit

Esta feature não foi ainda implementada.