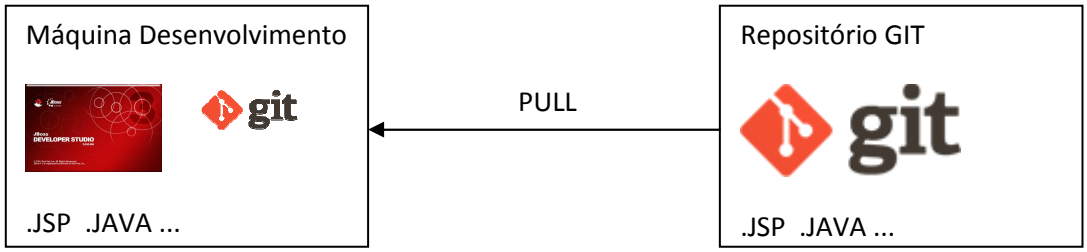
Anexo 11 – Procedimento para atualizar o repositório Git com fontes JSPs

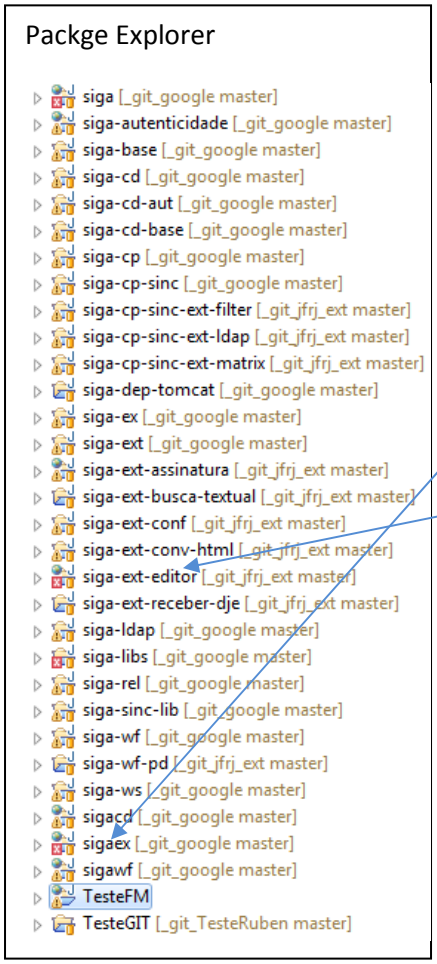
Estes procedimentos servem também para atualizar aplicações Java.

11.1 – Obter os fontes JSP (ou JAVA) mais recentes via PULL



O objetivo é pegar os fontes mais novos possíveis do JSP que serão alterados, visto que os fontes da máquina de desenvolvimento podem estar obsoletos.

Se observarmos o Package Explorer, nós temos o nome do projeto, e ao lado, o repositório onde ele se econtra. Um repositório pode, e deve, conter N projetos. O Pull é por repositório e não por projeto, ou seja, ao pegarmos qualquer projeto na árvore e clicarmos em **Team > Pull** estaremos atualizando também TODOS os projetos que temos naquele repositório.



Atualmente temos no SIGA dois repositórios, um remoto hospedado no Google (**_git_google**) e outro local **(_git_jfrj_ext**) hospedado no drive K.

Clicar em um projeto que contenha o repositório remoto, sigaex, por exemplo, e selecionar **team > pull**

Clicar em um projeto que contenha o repositório local, siga-ext-editor, por exemplo, e selecionar **team > pull**

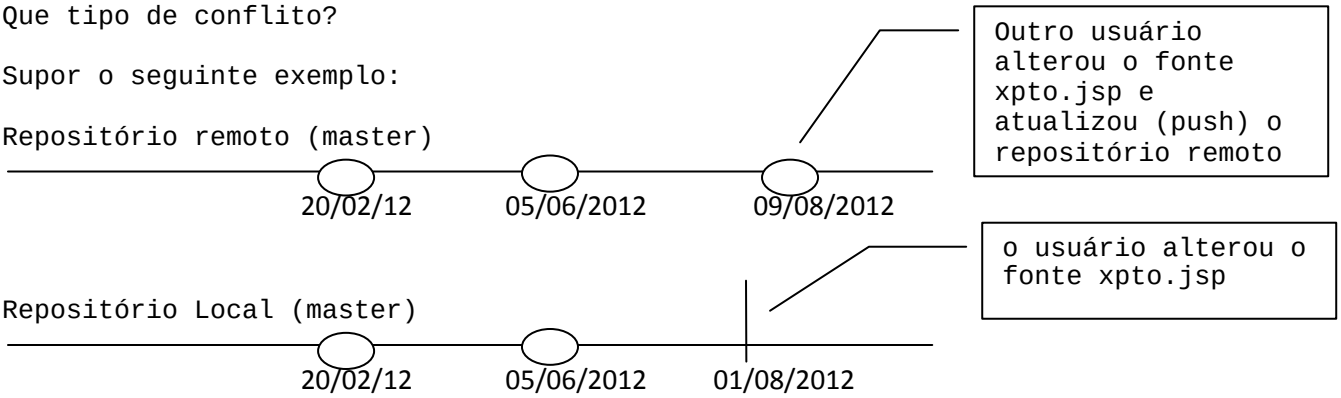
Caso o PULL não funcione pode ser porque: o endereço do repositório mudou, as suas credenciais para acessar o repositório remoto estão erradas e/ou você não tem acesso ao repositório. Ver observações, item B.

PULL = FETCH + MERGE

O MERGE está sujeito a conflitos como descrito em Anexos, item 7.3.1, que está copiado abaixo.

Que tipo de conflito?

Supor o seguinte exemplo:



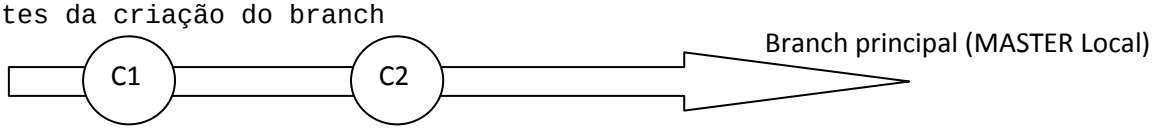
Se o usuário na máquina local realizar um PULL (fetch + merge), ou um MERGE (após o fetch no repositório remoto), o Git acusará um conflito no fonte xpto.jsp, conflito este que deverá ser administrado pelo usuário da máquina local, ou seja, receber o fonte mais novo do repositório remoto e aplicar as mudanças que ele realizou em 01/08/2012.

Observação: O PULL só afeta o repositório local.

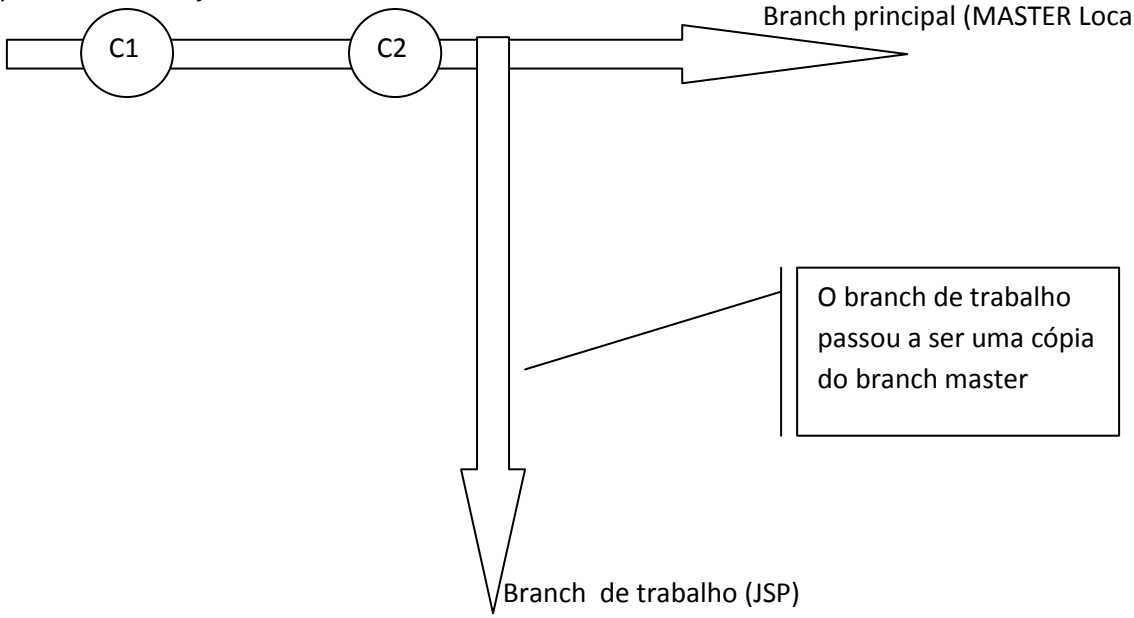
11.2 – Criar um branch (ramificação, desvio ...) para atualizar os fontes JSP

No exemplo abaixo, C1 e C2 são pontos de commit.

Antes da criação do branch

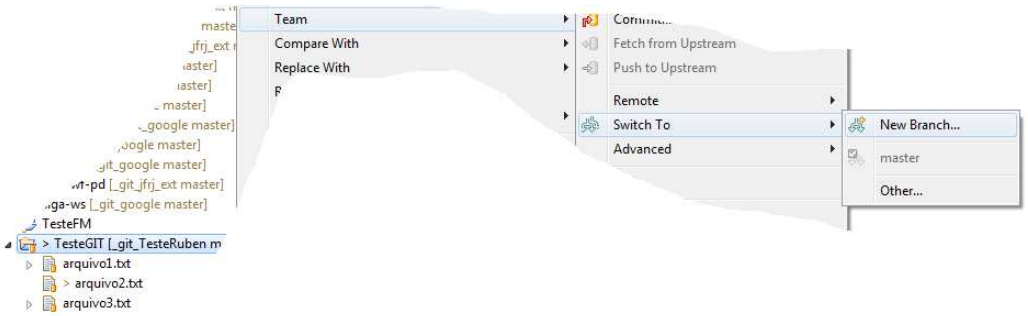


Depois da criação do branch

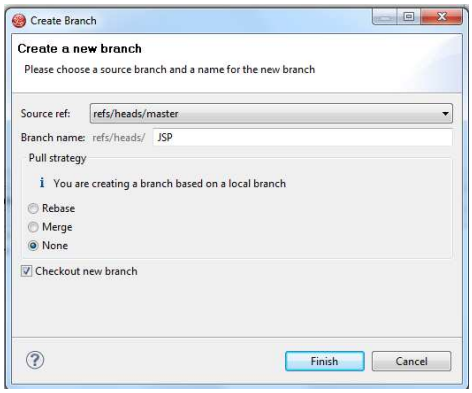


O objetivo é realizar as modificações dos fontes JSPs em uma área de trabalho (branch de trabalho), preservando os fontes originais na área principal (branch master)

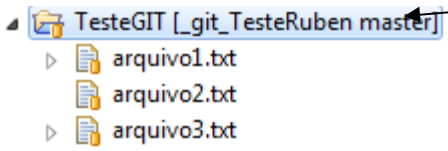
Utilizar o comando **Team > Switch to > New Branch** no mesmo projeto utilizado no item anterior (11.1). Observar que o New Branch afetará todos os projetos que estão no mesmo repositório.



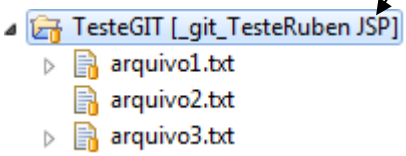
Criar o **branch de trabalho com nome semântico** (número do chamado, alteração principal ...). Neste exemplo chamaremos o novo branch simplesmente de JSP.



Observar que antes você estava trabalhando no branch master



e agora está no branch JSP



Observação: O BRANCH só afeta o repositório local.

11.3 – Alterar o(s) fonte(s) JSP

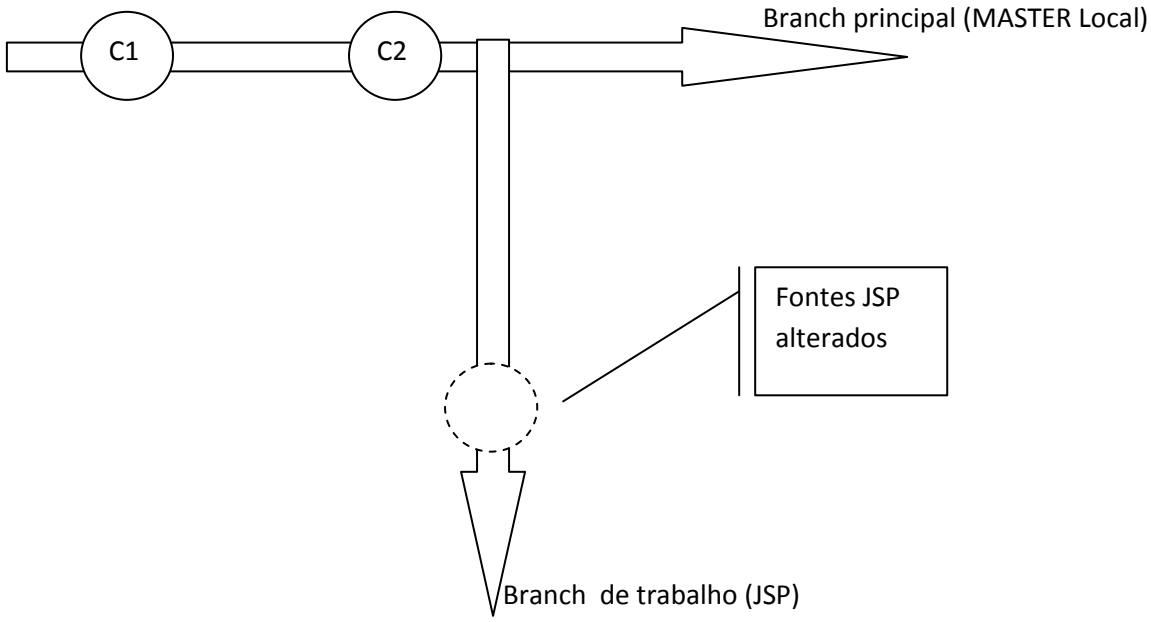
Agora que temos uma nova área de trabalho (Branch JSP) podemos alterar os nossos fontes sem preocupação.

A cada alteração, a aplicação JSP pode ser testada no ambiente local de desenvolvimento, ou seja, na IDE JBDS rodando o JBOSS (endereço: http://localhost:8080/siga)

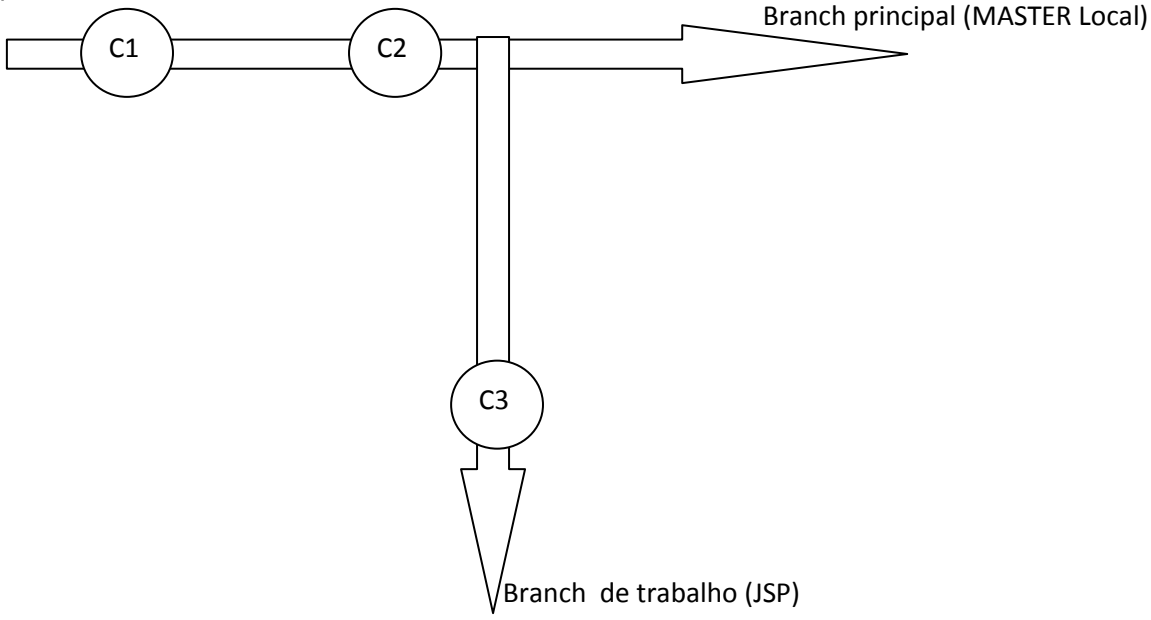
11.4 – Salvar (Commitar) as mudanças

O Commit "salva" (tecnicamente é criar um ponto de mudanças) as mudanças no branch onde o comando é emitido, no nosso caso, branch de trabalho (JSP no caso).

Antes do Commit

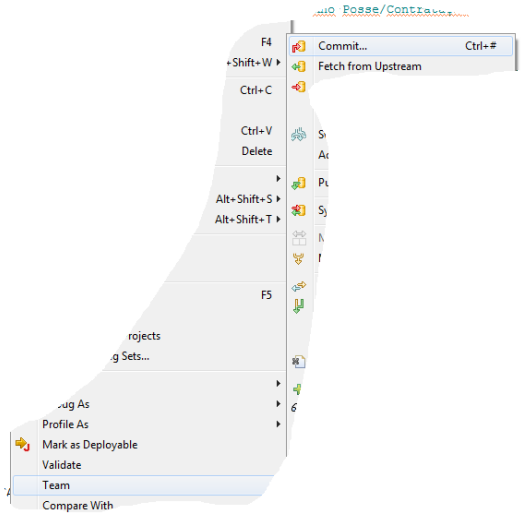


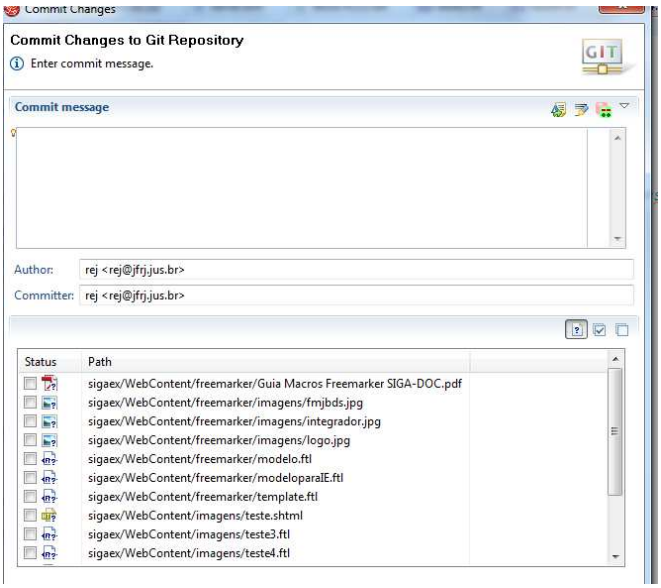
Depois do Commit



Observar que este novo ponto de commit (C3) não foi ainda replicado para o branch master local.

Utilizar o comando **Team > Commit** no mesmo projeto utilizado no item 11.1. Observar que o commit afetará todos os projetos que estão no mesmo repositório.



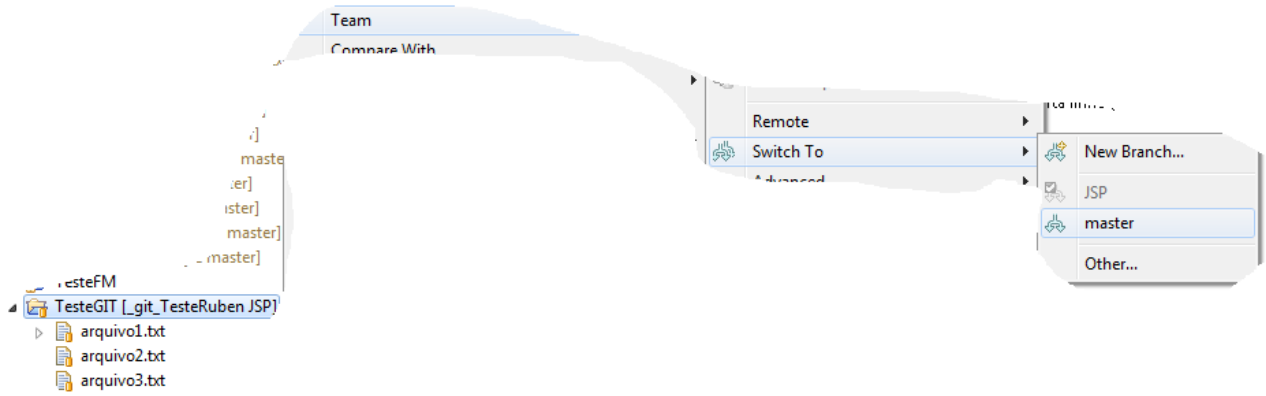


Colocar na mensagem do commit: uma breve descrição semântica, números do chamado e etc. Selecionar na parte inferior os arquivos que serão commitados.

Observação: O COMMIT só afeta o repositório local. Para que as mudanças se reflitam no repositório remoto elas devem ser publicadas via PUSH, como veremos a seguir

11.5 – Retornar (SWITCH) ao branch master

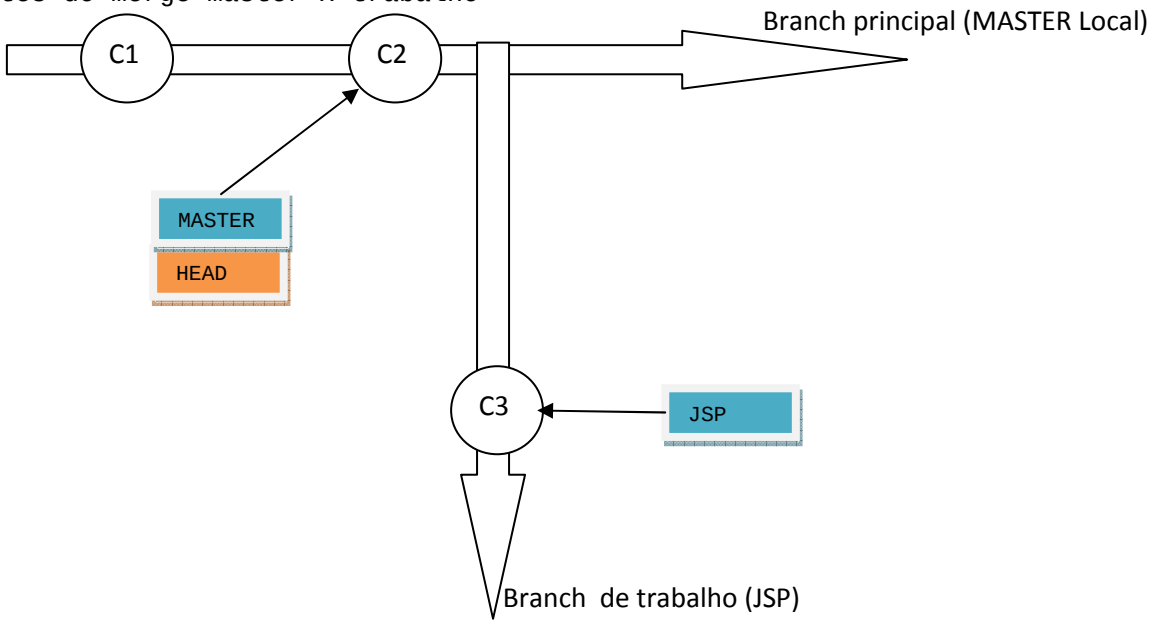
Team > Switch To > master



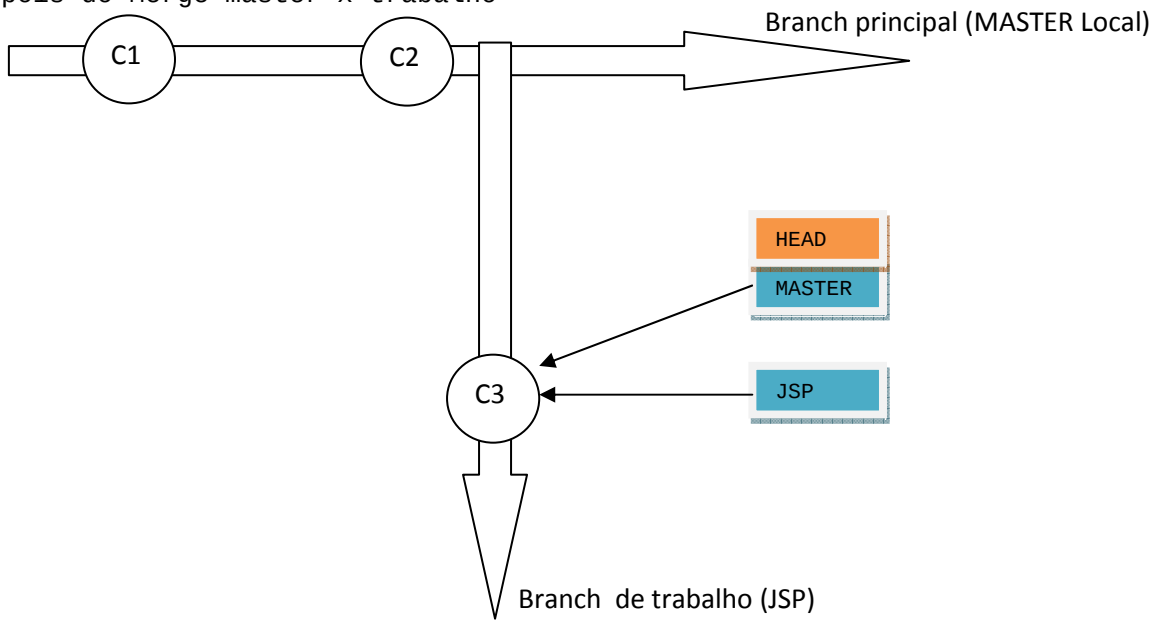
11.6 – Aplicar as mudanças do branch de trabalho no branch master via MERGE

O Merge cria um novo commit que incorpora as mudanças de outro commits. O exemplo abaixo é um exemplo de Fast Forward Merge.

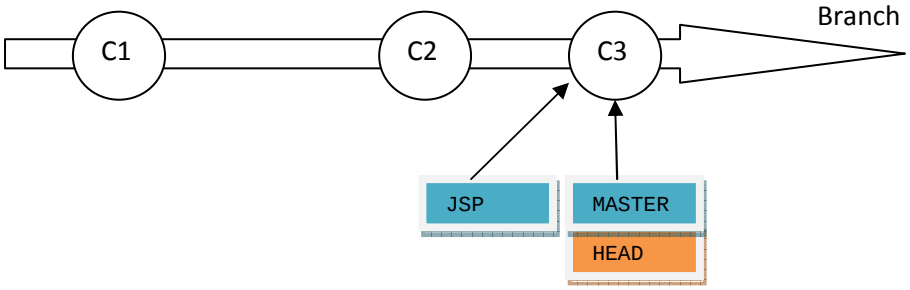
Antes do merge master x trabalho



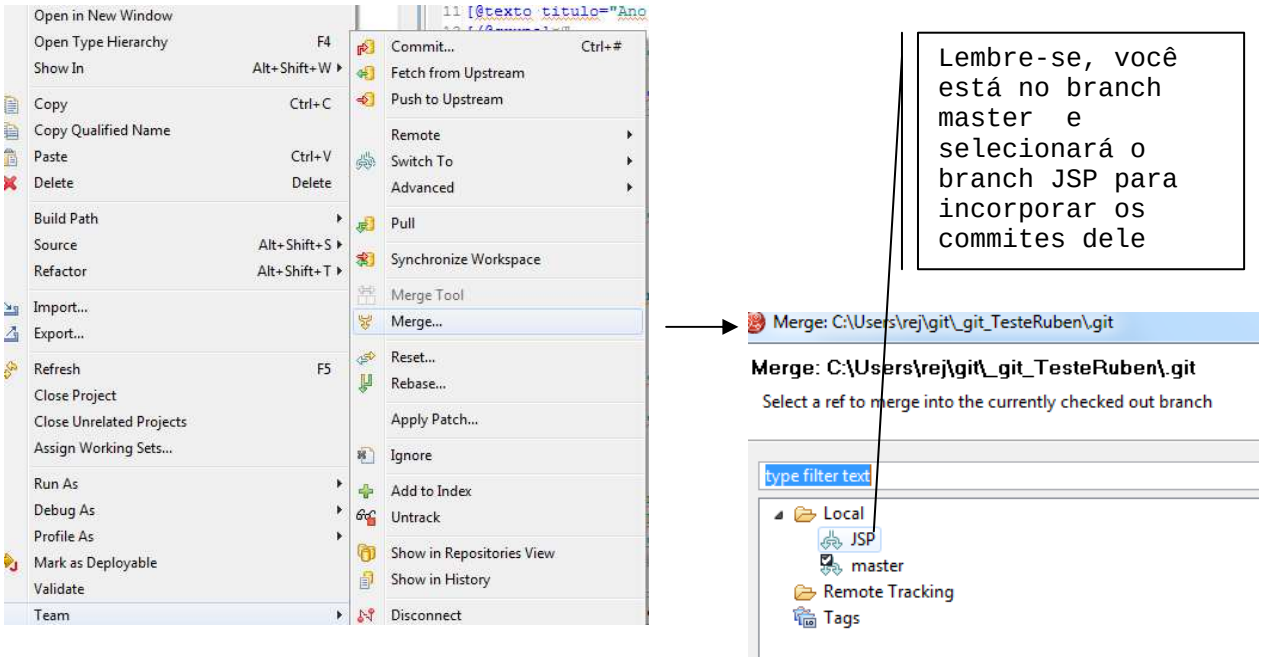
Depois do Merge master x trabalho



Ou simplificando,



Team > Merge

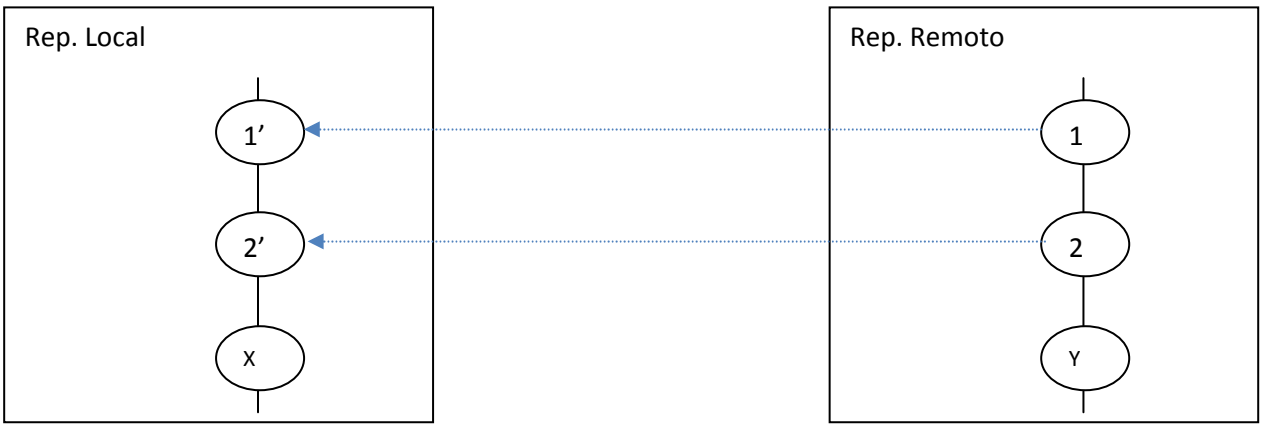


Observação: O MERGE só afeta o repositório local.

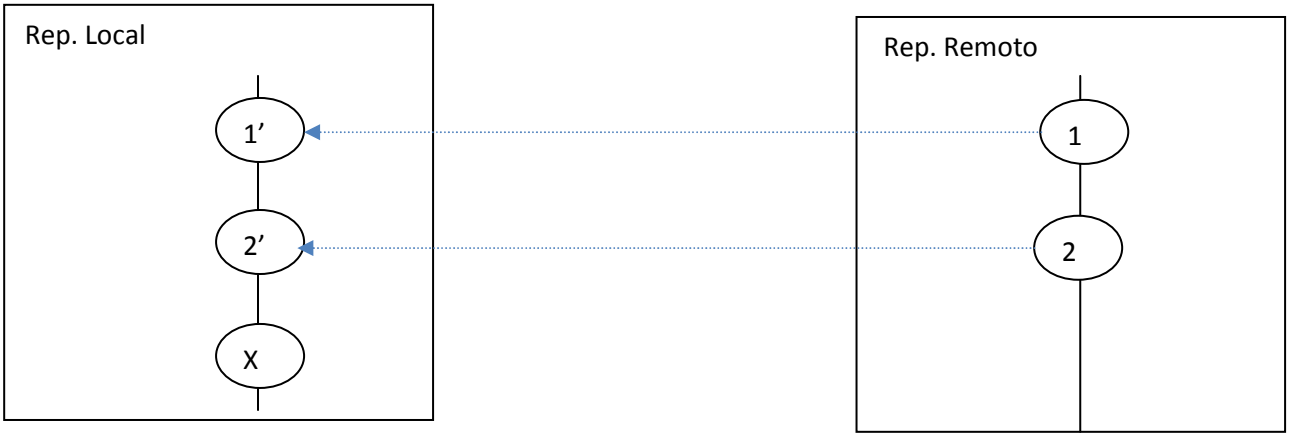
11.7 – Realizar o PULL novamente

De novo? Isto já foi feito no item 11.1!
O problema é que antes de realizarmos o PUSH necessitamos verificar se os níveis de commit do repositório local (master) e o remoto (msater)estão equalizados.

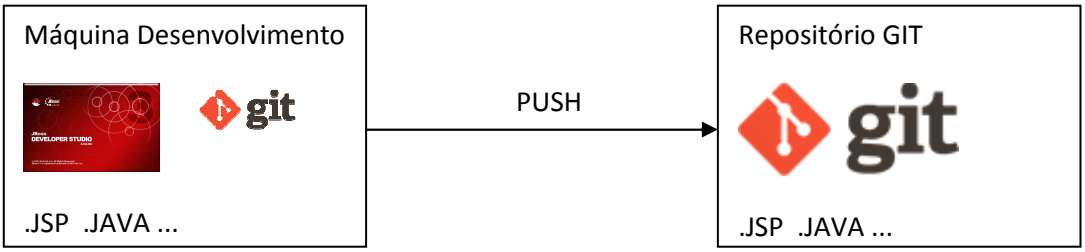
Neste exemplo, o repositório remoto possui o commit Y que não foi ainda aplicado ao local. O commit Y pode conter atualizações em alguns fontes que o usuário local tenha também feito. Aqui temos um conflito. O usuário local deve então realizar um PULL (fetch + merge), administrar os conflitos, commitar e aí sim, realizar o PUSH.



Neste exemplo, os níveis de commit estão OK, e o usuário local pode realizar um PUSH. Nem é necessário realizar um PULL antes.

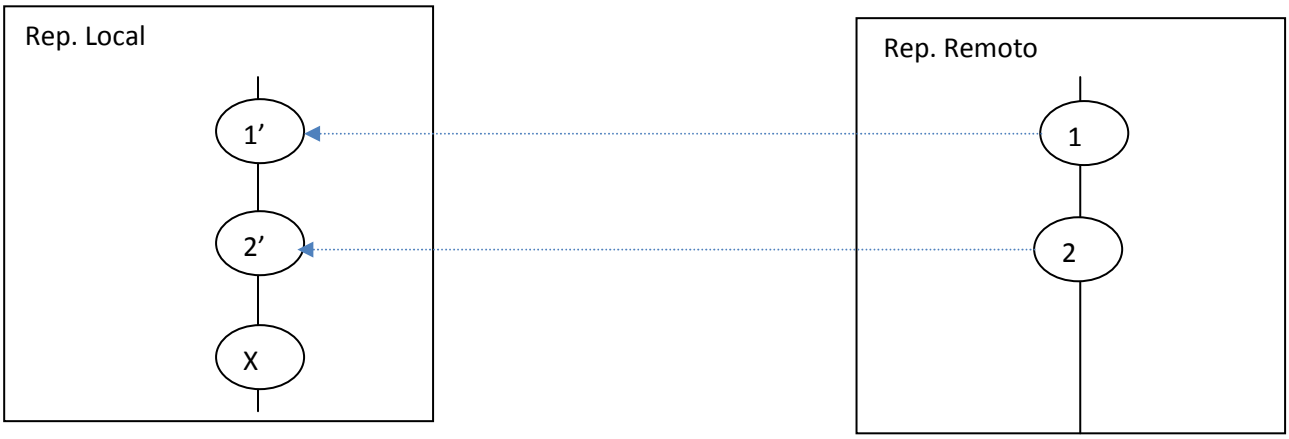


11.8 – Realizar a mudanças no repositório remoto (upstream) com o PUSH

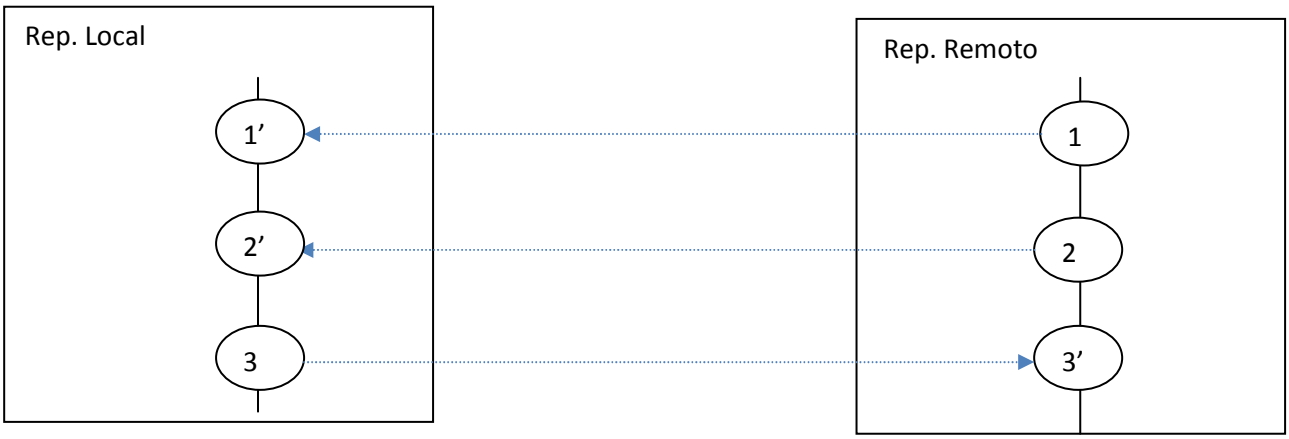


O PUSH atualiza o repositório remoto a partir do repositório local.

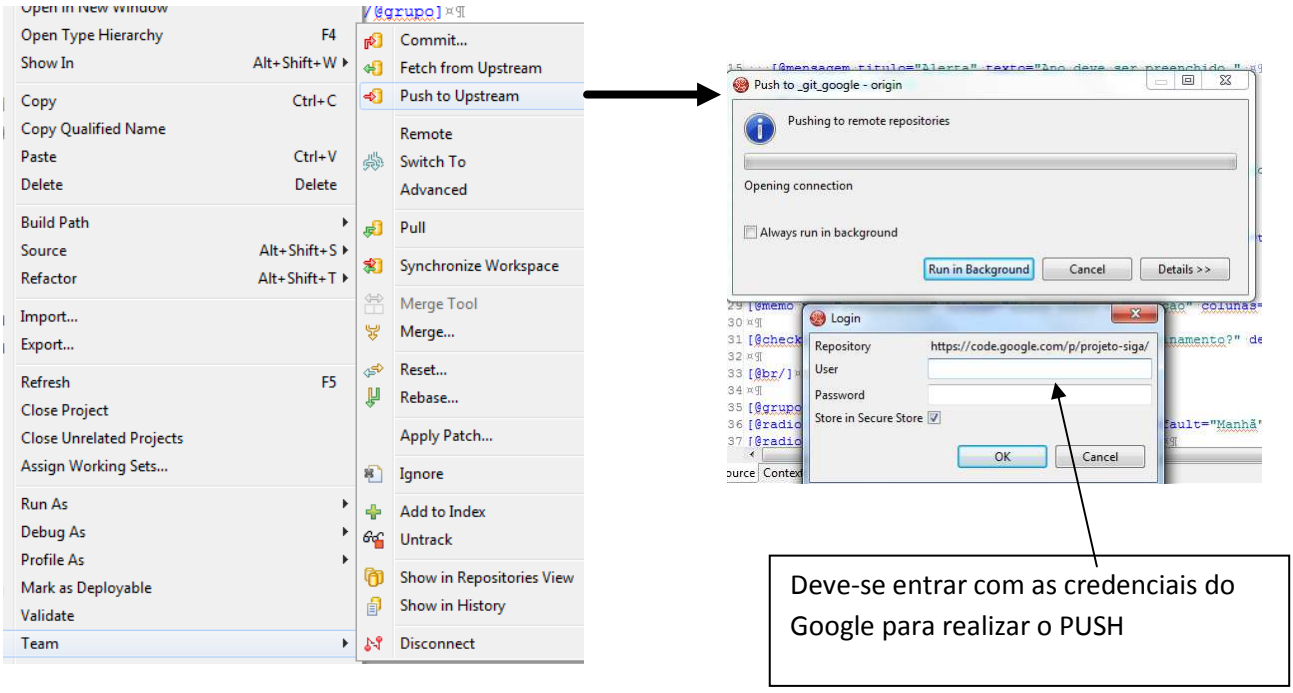
Antes do PUSH



Depois do PUSH



Team > Push to Upstream



Observação: O PUSH afeta o repositório remoto. Somente usuários autorizados podem atualizar o repositório remoto com a conta do Google.

Caso o PUSH não funcione pode ser porque: o endereço do repositório mudou, as suas credenciais para acessar o repositório remoto estão erradas e/ou você não tem acesso ao repositório. Ver observações, item B.

11.9 – Quando as mudanças terão efeito?

O fato de ter atualizado o repositório remoto não que dizer que as alterações já estão em produção e que o usuário possa utilizá-las. Só depois que o pessoal da infraestrutura do SIGA atualizar o servidor de aplicação é que as mudanças tomarão efeito.

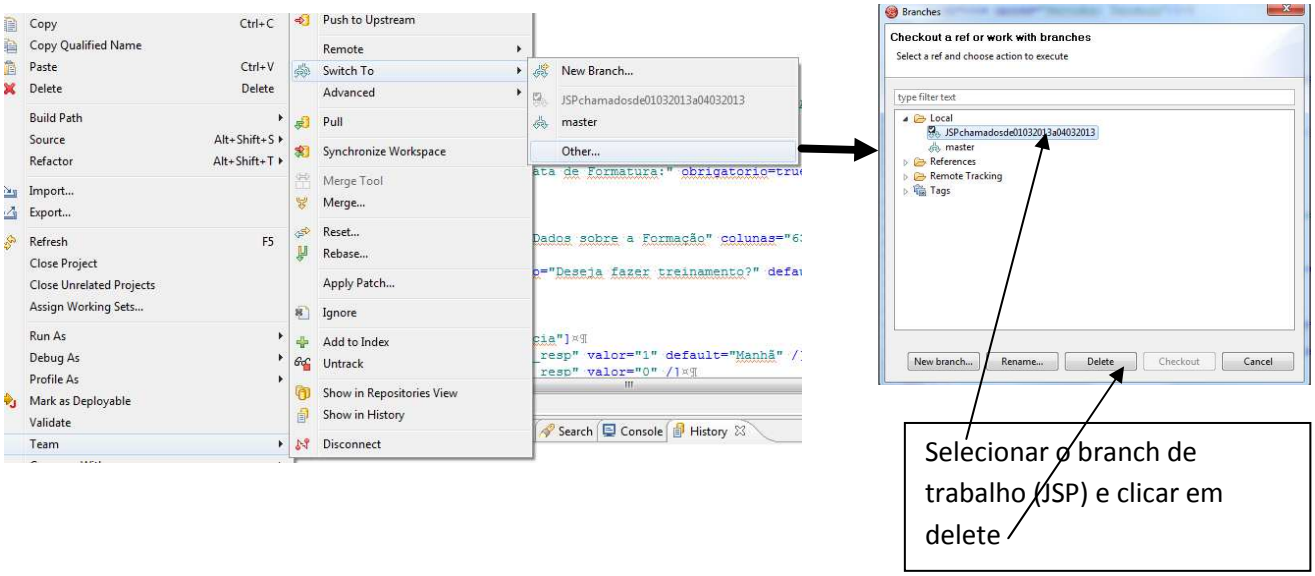
E se precisarmos colocar um JSP em produção COM URGÊNCIA?

Existe a hipótese de mover diretamente o arquivo .jsp para um diretório específico (.../paginas/expediente/modelos) do servidor de aplicação (JBOSS). É importante que esta operação seja apoiada pelo pessoal de infra do SIGA.

11.10 – Posso excluir o branch de trabalho (JSP)?

Depois que as alterações já tenham sido aplicadas ao servidor de aplicação e o usuário testado, aí sim podemos excluir o branch de trabalho.

Team > Switch to > Other …



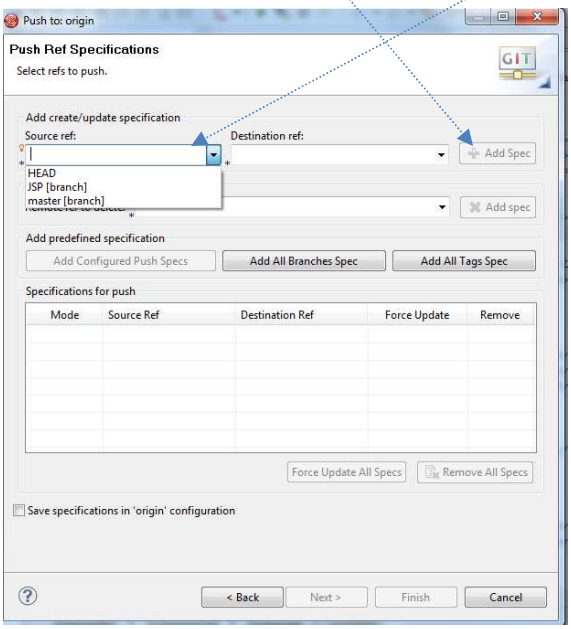
Observações:

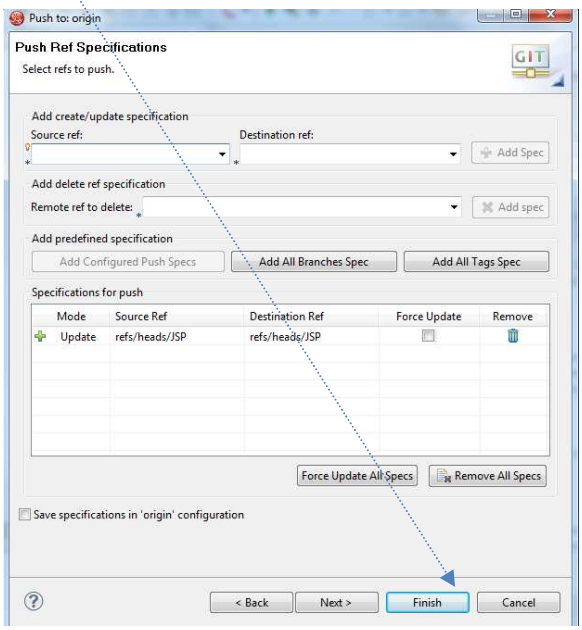
A - O procedimento acima é para funcionários da SJRJ

Para estagiários e terceiros temos outra política. Estes funcionários não podem atualizar o branch master remoto diretamente, e sim, salvar nele um branch contendo as modificações. Depois, um funcionário habilitado aplica as mudanças no branch master.

Estagiário / Terceiro:

- Realizar os passos de 1(11.1) a 4(11.4) exatamente como descrito acima;
- Realizar o pull do passo 7 (11.7), embora neste caso não seja necessário;
- Realizar o push, passo 8(11.8).
  No Push Ref Specifications, selecionar o branch criado contendo as modificações.Clicar em Add Spec depois Finish.





Funcionário:

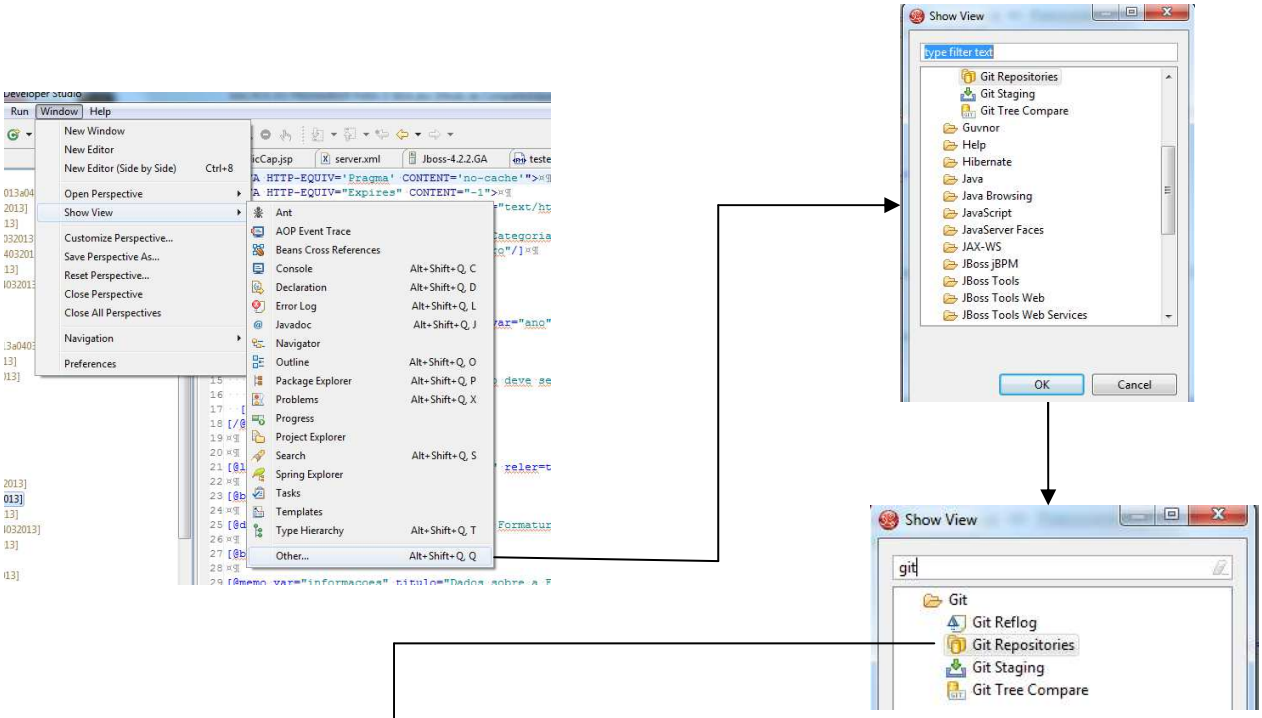
- Realizar o pull, passo 7 (11.7) para buscar o branch remoto com as modificações;
- Estar ciente que está no branch master local. Caso contrário, realizar um switch, como no passo 5 (11.5);
- Realizar um merge, passo 6 (11.6) entre o branch atual (master) e o branch contendo as modoficações (que deve ser selecionado);
- Realizar o push, passo 8 (11.8). Neste caso, no Push Ref Specifications selecionar o master [branch].

B – Configurando as credenciais e os endereços do repositório GIT

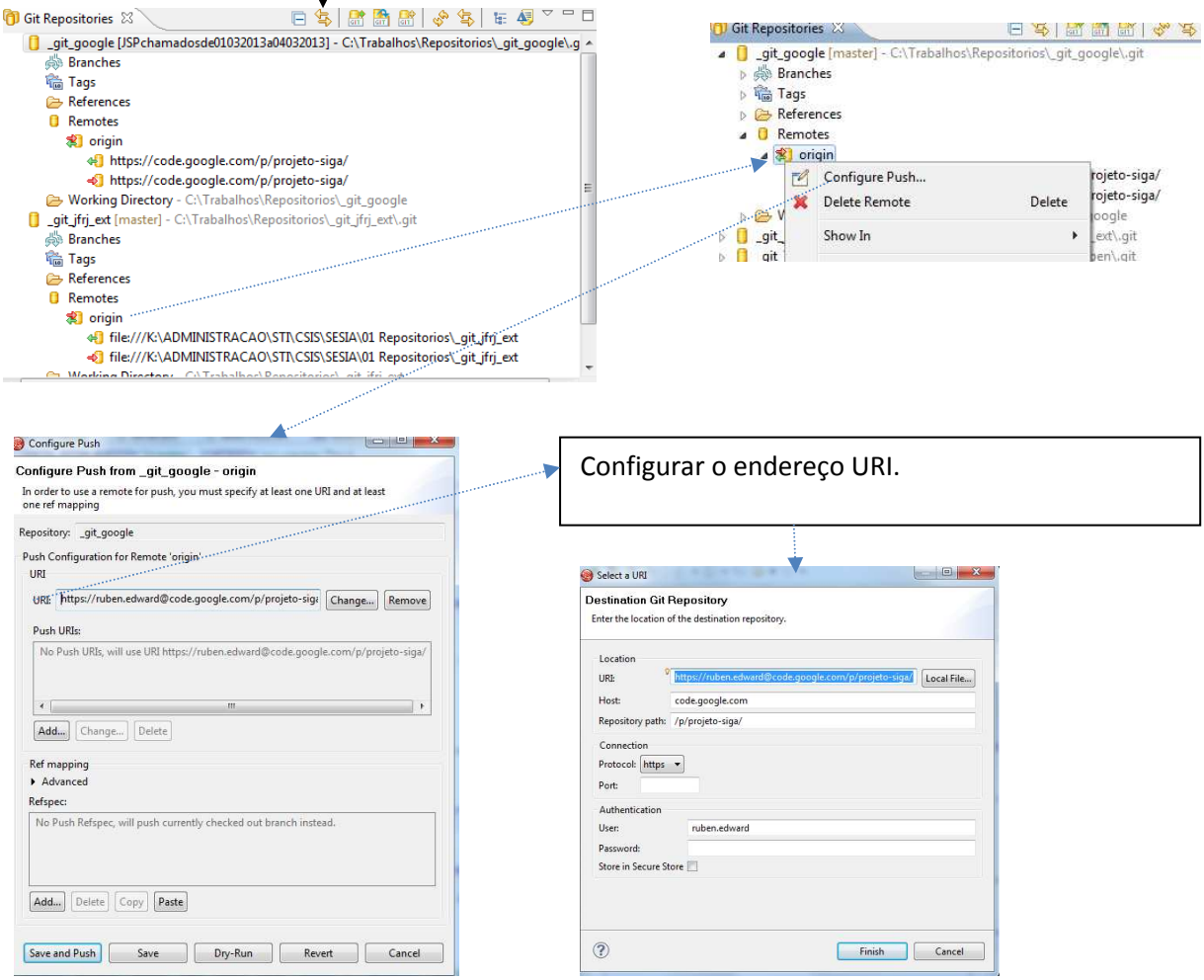
B.1 - Utilize o procedimento abaixo para verificar os endereços dos repositórios

Acessar a View dos repositórios Git

View > Show View > Other ... Git Repositories



Cursor em origin > Configure Push



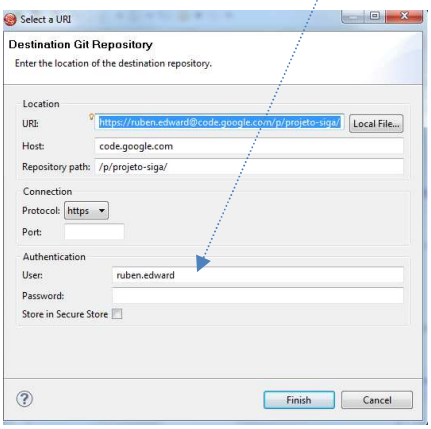
Atualmente são os endereços abaixo que estão valendo:
Repositório Local: K:\ADMINISTRACAO\STI\CSIS\SESIA\01 Repositorios_git_jfrj_ext
Repositório Remoto: https://code.google.com/p/projeto-siga/

Observação: No caso do Google o endereço que fica registrado é:
https://usuario@code.google.com/p/projeto-siga/, sendo que no meu caso, https://ruben.edward@code.google.com/p/projeto-siga/

O mesmo procedimento serve para configurar o FETCH (PULL)

B.2 – Verificando as credenciais para acessar o repositório remoto

No item anterior vimos como configura a URI, e nesta mesma tela, temos como configurar as credenciais, user e password.

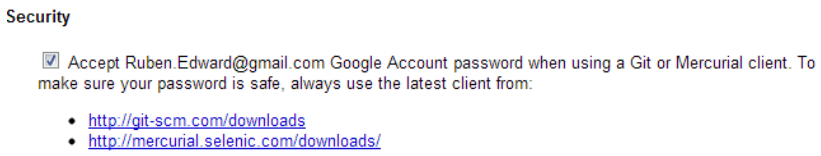


B.3 – Autorização para acessar o projeto e Conta no GoogleCode

Para que um usuário faça atualizações no repositório remoto (googlecode) via PUSH é necessário que: ele possua uma conta no google e que o administrador do repositório o habilite.

Após a habilitação o usuário ganhará uma password para o googlecode:

Caso o usuário deseje utilizar a password do Google / Gmail ele deve marcar a opção abaixo:



C – Revertendo um código ao original

Suponha que você já tenha atualizado o arquivo concessaoAposentadoria.jsp, porém deseja que ele volte ao original.

Selecionar o jsp > Replace With > HEAD Revision