**Passos para Criação , Desenvolvimento e Entrega de um Branch no Git**

**0- Criando um Branch para Angular e Scala.**

Para iniciar sempre é necessário atualizar a branch master. Logo siga os passos abaixo.

**git update**

**git fetch**

**git pull**

Realizados os passos acima podemos criar um branch para o seu **Chamado** ou **Task**

**1-Criar um branch**

**git co –b** **{tipo-de-atendimento}{número-do-chamado}-{seu-texto}-{seu-texto}**

Descrição dos campos:

**{tipo-de-atendimento}** : Poderá ser usado os tipos **bugFix** ou **fix**

**{número-do-chamado}** : Deve ser o número da SD ou ALM

**{seu-texto}** : Deve ser uma pequena descrição da correção

**2- Testes**

Após a realização das devidas alterações é necessário realizar os devidos teste para o front.

Acessar o terminal da VM é rodar o seguinte comando.

**./frontend-test.sh**

Caso ocorra algum erro podemos verificar os detalhes no seguinte arquivo.

Y:\workspace\credenciado-medico\report\cucumber\html-report

**3- Adicionar os arquivos**

Após a realização dos devidos testes , podemos incluir os arquivos alterados em nossa branch. Para isso siga os passos abaixo:

Verificar se está na branch correta: **git br**

Adicionar os arquivos alterados : **git add .**

Realizar o commit e incluir comentário: **git commit -m** **"Descrição de sua alteração"**

Realizar o push : **git push**

**4- Configuara PR no GitHub**

No site http://github.amil.com.br/multicanal-prd/credenciado-medico/pull/**{número-gerado}**

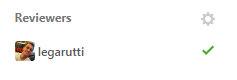
Devemos realizar algumas configurações. Nesta fase temos alguns passos para cada fase do processo.

**Fase -1** :

O processo deve ser iniciado com a tags 

É necessário a indicação do responsável pelo desenvolvimento no campo Assigness (Indicar seu usuário) 

É necessário indicar outro desenvolvedor para a realização de Review



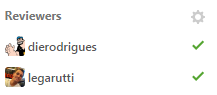
**Fase – 2**:

Após a realização do Review podemos seguir para o próximo passo.

É necessário indicar outro desenvolvedor para a realização do Core Commiter

Para isso altere a tag para 

É necessário indicar outro desenvolvedor para a realização de Review no Core Commiter.



**Fase – 3**:

Após a realização da fase anterior , podemos indicar a branch para a equipe de QA

**5- Deploy QA**

Download da branch : **git co qa2**

Verificar se realmente criou: **git br**

Mudar para a master : **git co master**

Atualizar a master

**git update**

**git fetch**

**git pull**

Rertorno a minha branch (a branch do pull request): **git co minha\_branch**

**Obs: caso exista mais de uma branch para deploy realizar o passo 3 novamente na branch em questão.**

Realizar merge com a master para evitar conflitos : **git merge master**

Mudo para o branch qa2 : **git co qa2**

Atualizar o qa2:

**git pull**

**git fetch**

**git update**

Realizar merge do qa2 com seu branch: **git merge meu branch**

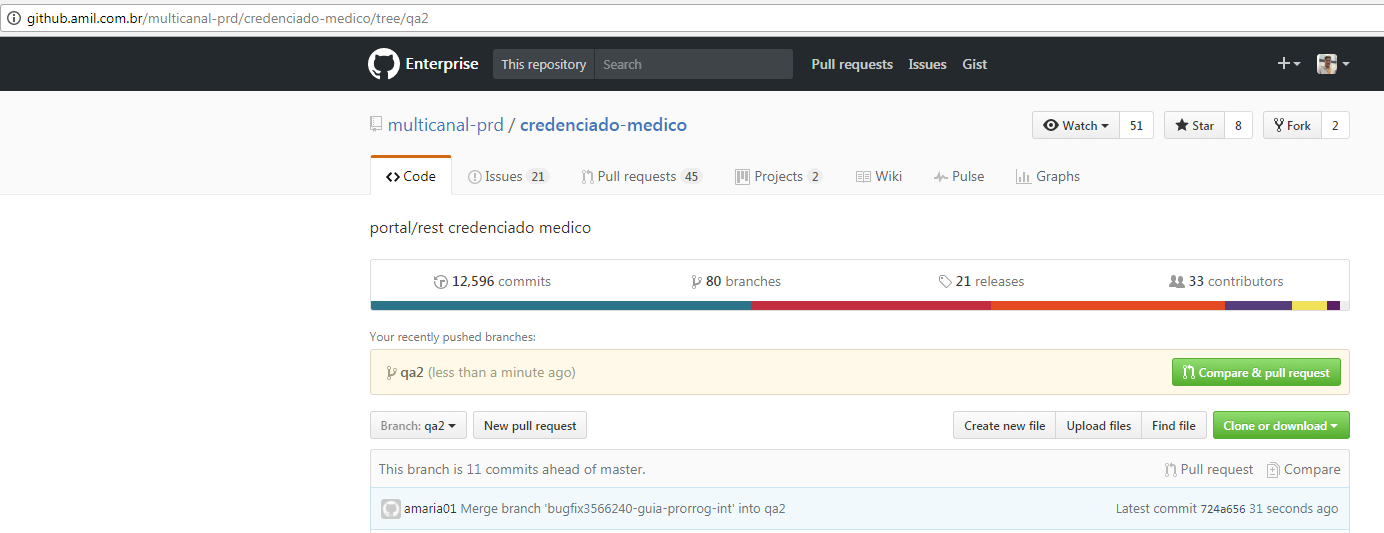
**Obs: caso exista mais de uma branch para deploy realizar o passo 7 novamente na branch em questão.**

Realizar o Commit: **git push**

**6- Deploy-QA[Jenkins]**

Após o passo anterior podemos verificar a inclusão dos arquivos no seguinte endereço:

<http://github.amil.com.br/multicanal-prd/credenciado-medico/tree/qa2>



Visto que o mesmo está no site , podemos realizar deplay no site <http://ci.amil.com.br/view/All/job/deploy-qa/build?delay=0sec>



Preencha os campos conforme imagem.

Logo após podemos visualizar as alterações o seguinte endereço: https://credenciado2.qa.amil.com.br/login

**7- QA(Abrir chamado ALM)**

Seguir passo do doc Submeter Correções para QA

**8- Deploy-QA[Jenkins] para C#**

<http://ci.amil.com.br/view/.Net/job/deploy-qa-dotnet/build?delay=0sec>

Removendo Arquivo

git rm --cached seu\_arquivo

git st

git clean –fd

git add .

git commit -m "Removendo arquivo .iml"

git push

Deploy no API.

<http://ci.amil.com.br/view/.Net/job/deploy-qa-dotnet/build?delay=0sec>