MONAN
Criando Forks do repositório
Forks - Clones - Pull Request

Workshop Interno da DIMNT para início dos trabalhos com o MONAN-ATM/SFC de 2 a 3 de outubro de 2023.

Grupo de Computação Científica - GCC
Divisão de Modelagem Numérica do Sistema Terrestre - DIMNT
Coordenação-Geral de Ciências da Terra - CGCT

MONAN
Model for Ocean-laNd-Atmosphere predictioN

MONAN
Model for Ocean-laNd-Atmosphere predictioN
INPE
MINISTÉRIO DA
CIÊNCIA,TECNOLOGIA
E INOVAÇÃO
GOVERNO FEDERAL
BRASIL
UNIÃO E RECONSTRUÇÃO

Git e GitHub: o que é?

Git?

- É um sistema de controle de versão distribuído (DVCS).
- Monitora e gerencia o histórico de alterações.

GitHub?

- É um portal de serviço baseado na nuvem que integra o sistema de controle de versão Git e outras ferramentas.
- Permite a colaboração de desenvolvedores através de projetos compartilhados.

Repositórios

- Coleção de arquivos e pastas associados a um projeto.
- Unidades auto-confinadas, qualquer pessoa que tiver uma cópia do repositório pode acessar toda a base de código e seu histórico.

0 Git e GitHub: uso em centros de referência

Commits
- Nome dado às mudanças de estado que são salvas no Git.
- É um "instantâneo" no tempo do histórico de arquivos.
- Os commits podem ser organizados em branches.

Branches
- São as linhas de desenvolvimento.
- Um ponteiro móvel que leva um commit.
- O Branch sempre aponta para um commit.

Fork

- É uma cópia de um projeto na conta pessoal de um desenvolvedor.
- Cada desenvolvedor tem controle total de seu fork e é livre para implementar uma correção ou novo recurso.
- O trabalho concluído no fork é mantido separado ou é retornado para o projeto original por meio de um *pull request.*

Pull Request

- Uma solicitação que gera uma notificação sinalizando a conclusão do desenvolvimento de uma *feature* qualquer sugerindo um merge.
- Permite ver as diferenças entre o projeto original e o seu *branch* de recurso.

Merge
- Mesclagem de branches; Gera um novo commit.

Clone
- Gera uma cópia local de um projeto já existente.

Pull
- Atualiza a linha de desenvolvimento local com atualizações do equivalente remoto.
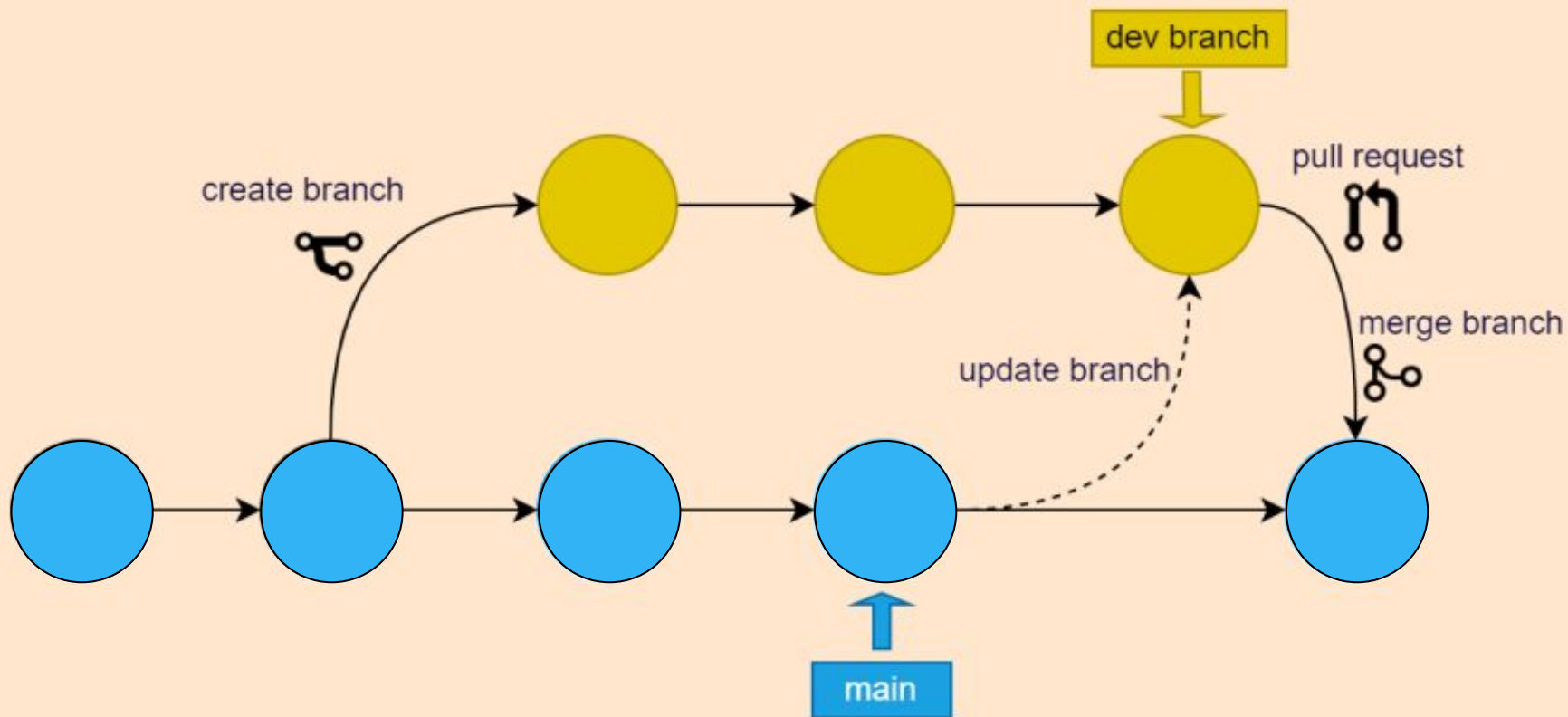
Push
- Atualiza o repositório remoto com todos os commits feitos localmente em um branch.

Checkout
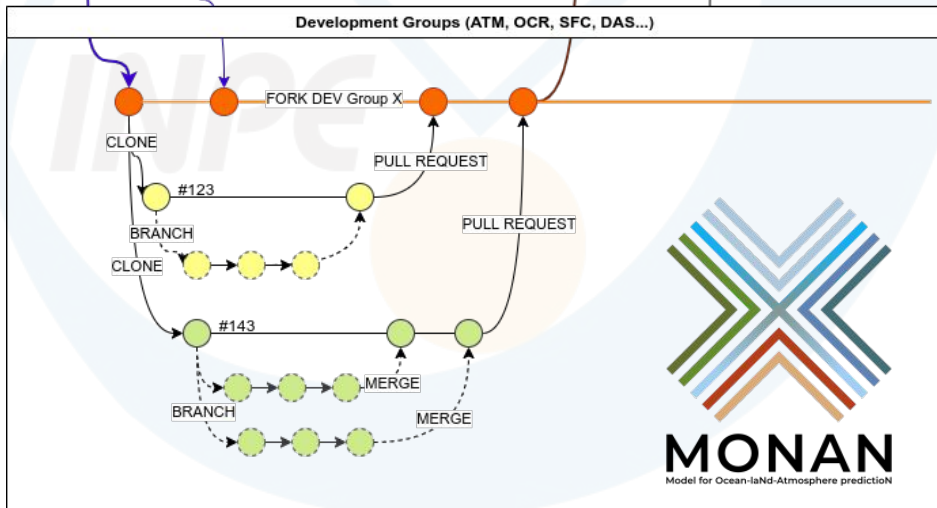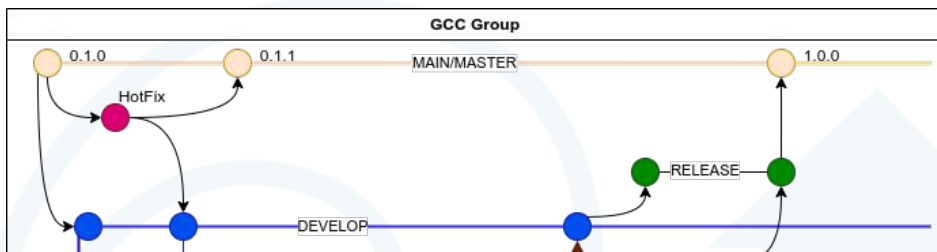- Faz a troca entre branches ou as cria e também restaura arquivos da árvore de trabalho.

- O versionamento usa gitflow, com duas linhas de controle:
  - **main** (ou master) [●]
  - **develop** [●]

- Os grupos devem fazer uma bifurcação **(fork) do develop** [●] para seu próprio desenvolvimento.

- De cada bifurcação criam-se **clones** [● ●] conforme a necessidade do desenvolvimento.

- Cada clone pode ter um ou mais **branchs** [● ●] para atender a suas tarefas específicas.

- O grupo GCC cuida do gerenciamento dos pedidos de inserção das versões [↑].

- Bugs do main geram **Hotfix** (●) para manutenção. Releases (●) para testes do GCC (qualidade e performance) e do GAM (funcionalidade) são geradas para aprovação ou não.

Endereço
https://github.com/monanadmin/MONAN-Model

Modelo Básico
MPAS-v8.0.1

```
MPAS-Model
├── src
│   ├── driver -- Main driver for MPAS in stand-alone mode (Shared)
│   ├── external -- External software for MPAS (Shared)
│   ├── framework -- MPAS Framework (Includes DDT Descriptions, and shared routines. Shared)
│   ├── operators -- MPAS Opeartors (Includes Operators for MPAS meshes. Shared)
│   ├── tools -- Empty directory for include files that Registry generates (Shared)
│   │   ├── registry -- Code for building Registry.xml parser (Shared)
│   │   └── input_gen -- Code for generating streams and namelist files (Shared)
│   └── core_* -- Individual model cores.
│       └── inc -- Empty directory for include files that Registry generates
├── testing_and_setup -- Tools for setting up configurations and test cases (Shared)
└── default_inputs -- Copies of default stream and namelists files (Shared)
```

MINISTÉRIO DA
CIÊNCIA,TECNOLOGIA
E INOVAÇÃO

GOVERNO FEDERAL
BRASIL
UNIÃO E RECONSTRUÇÃO

Página inicial do MONAN - Repositório
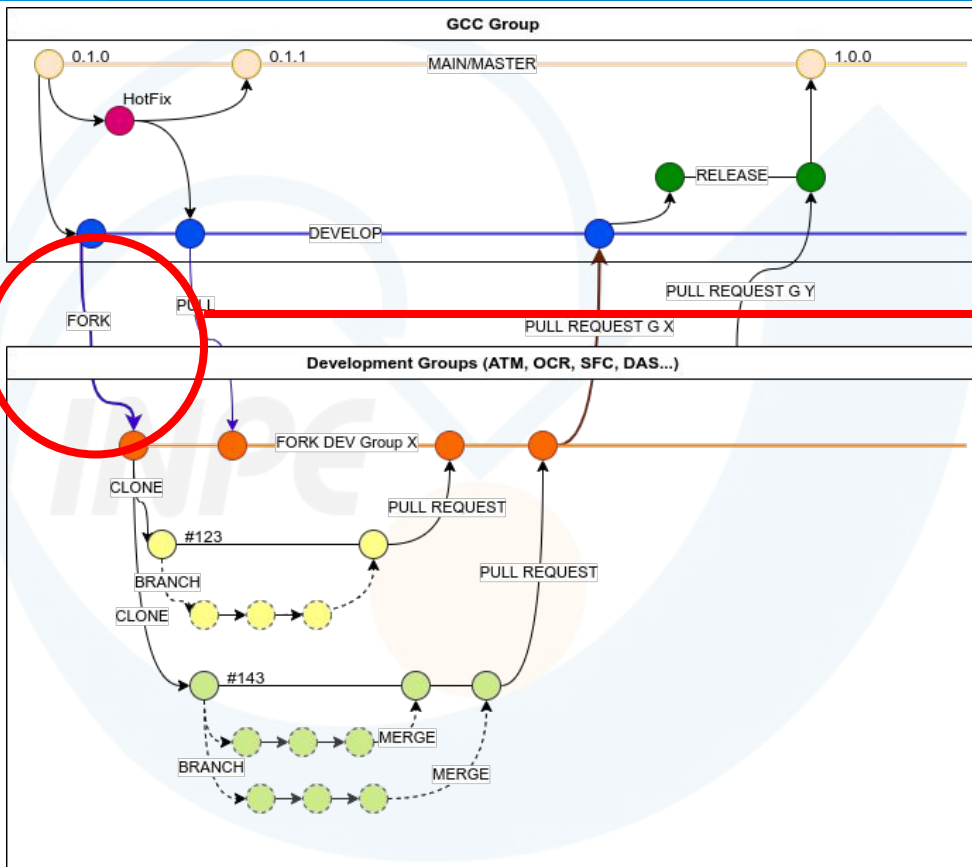
Clicar na chave Fork

Veja que um novo fork (from monanadmin) foi criado para o proprietário

Clique
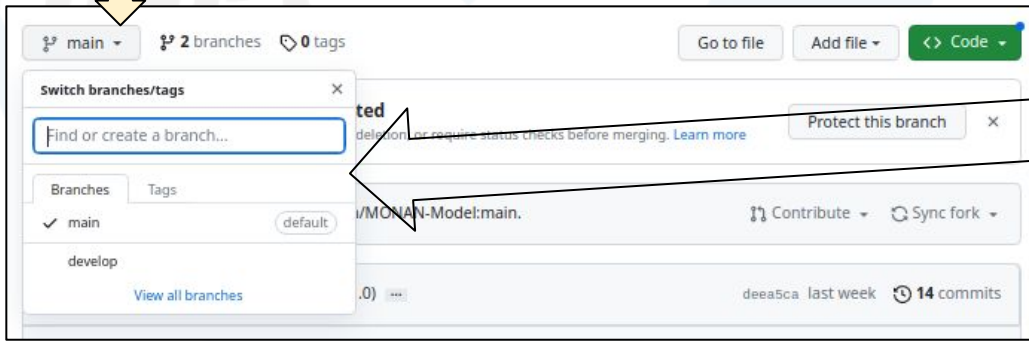
Observe que além do branch main (master) existe o branch de desenvolvimento "**develop**"

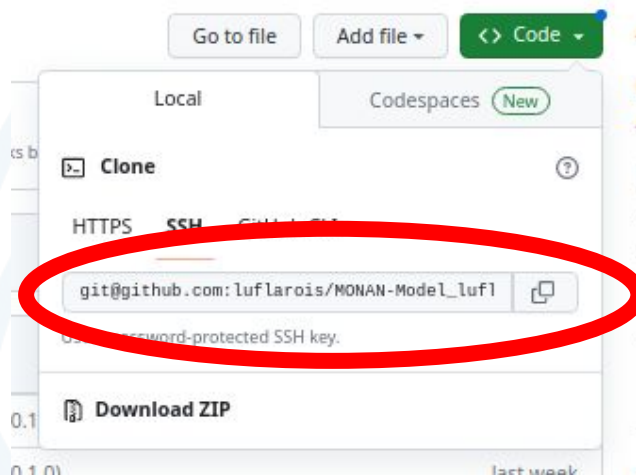**Obedece o git flow usado no padrão de versionamento!**

Workflow

Fork Criado!

Copie o endereço do git
Esse endereço será usado
para criar seu clone de trabalho!

Fazendo um clone do Fork criado


```
  lufla@petara in ~/monan_development took 1ms
  λ git clone git@github.com:luflarois/MONAN-Model_lufla.git        (base)
Cloning into 'MONAN-Model_lufla'...
remote: Enumerating objects: 1688, done.
remote: Counting objects: 100% (1688/1688), done.
remote: Compressing objects: 100% (985/985), done.
remote: Total 1688 (delta 656), reused 1671 (delta 650), pack-reused 0
Receiving objects: 100% (1688/1688), 3.57 MiB | 4.42 MiB/s, done.
Resolving deltas: 100% (656/656), done.

  lufla@petara in ~/monan_development took 2s
  λ □                                                                (base)
```

Abra um terminal e proceda com o comando de clonagem (use sua área de desenvolvimento)

Observe que os arquivos de origem foram copiados

Fazendo um clone do Fork criado
Indo para o branch de desenvolvimento


```
fish /home/lufla/monan_development/MONAN-Model_lufla

┌─lufla@petara in ~/monan_development
└─λ cd MONAN-Model_lufla/                                           (base)

┌─lufla@petara in repo: MONAN-Model_lufla on ⎇ main
└─λ git branch -a                                                  (base)
* main
  remotes/origin/HEAD -> origin/main
  remotes/origin/develop
  remotes/origin/main

┌─lufla@petara in repo: MONAN-Model_lufla on ⎇ main took 1ms
└─λ git checkout develop                                           (base)
Branch 'develop' set up to track remote branch 'develop' from 'origin'.
Switched to a new branch 'develop'

┌─lufla@petara in repo: MONAN-Model_lufla on ⎇ develop took 63ms
└─λ _                                                              (base)
```

vá para o diretório com o clone

Observe os branchs ("-a" é fundamental)

Mude para o branch de desenvolvimento

MONAN
Model for Ocean-land-Atmosphere predictioN

INPE

MINISTÉRIO DA
CIÊNCIA,TECNOLOGIA
E INOVAÇÃO

GOVERNO FEDERAL
BRASIL
UNIÃO E RECONSTRUÇÃO

Criando um branch de trabalho

Criando o branch
MONAN_Model_issue_256

Verificando os branchs
existentes

Nesse caso o branch parte
de "**develop**"

Workflow
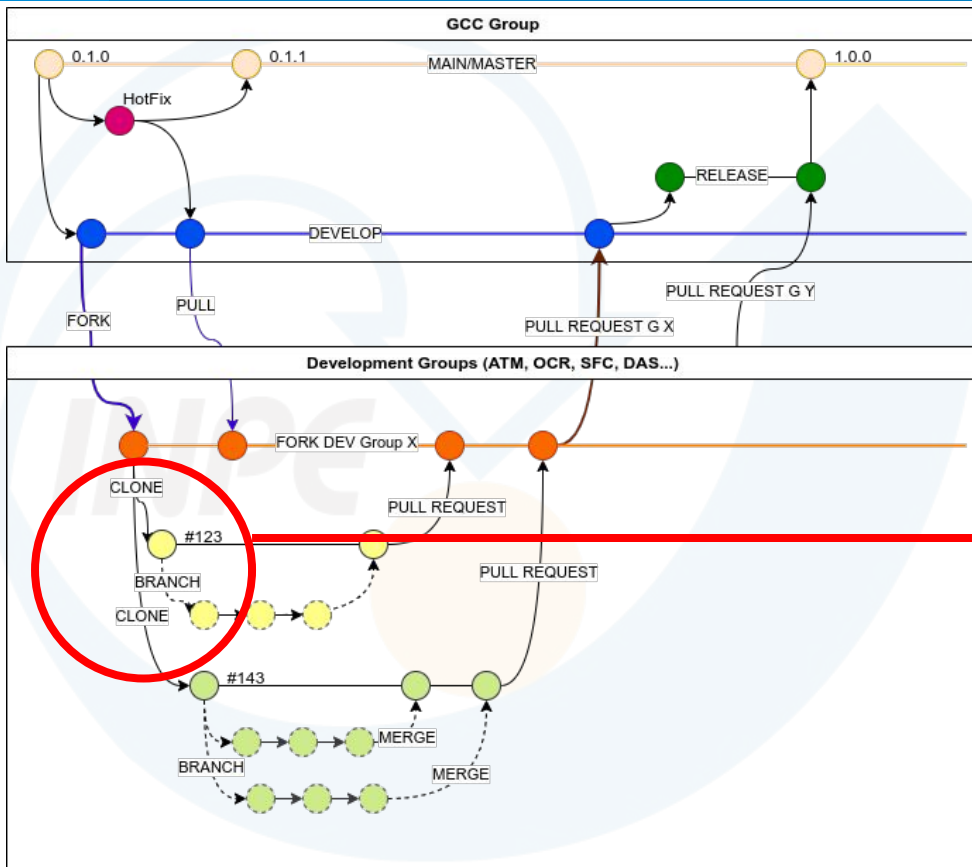
Clone realizado e Branch criada!

Alterando e "comitando" as alterações

- criando um arquivo novo
- Adicionando na lista de commit
- Verificando o status
- Comitando a atualização

Obs: Ainda não está no fork

Subindo alterações para o fork local

- Fazendo um "push" no repositório local

  Obs: veja que no primeiro "push" de um branch o git reclama. O comando precisa ser completo com o nome do branch

Subindo alterações para o fork local

- O **push** realizado **atualiza** o seu branch no repositório **fork**

- Ao abrir o repositório pelo **browser** (github) haverá uma possibilidade de um **PULL REQUEST** para o main (do monanadmin), develop (do monanadmin) ou para o fork criado.

- O grupo GCC deverá trabalhar como "**gatekeeper**" do repositório **principal**. **Nas etapas intermediárias** fica a **critério do grupo** atualizar ou não seu próprio fork via PULL REQUEST.

- PULL REQUESTs **para** o repositório **principal** só **devem** ser feitos quando o desenvolvimento for considerado **apto** pela equipe de desenvolvimento.

- O grupo GCC vai analisar os PULL REQUESTs **somente** para o repositório **principal** fazendo o trabalho de avaliação de qualidade, performance e funcionalidade básica.

15

Subindo alterações para o fork local
Pull Request Automáticos! Se for usar no momento

Obs: Observe que esse push gerou um pull request automático.

Veja que há um novo branch listado

Clicando em "branches" você pode ver a lista de branchs.

clique em "base repository" e **escolha o seu fork**

Na tela que abrir clique em "base" e escolha **"develop"**

Depois clique em **"Create Pull Request"**

Subindo alterações para o fork local
Mesclando as alterações na Develop

Nesse caso não há conflitos verificados

Clique então em "Merge Pull Request"

Depois, na tela que surgir, em "Confirm Merge"

Subindo alterações para o fork local
Mesclando as alterações na Develop

O sistema mostra que o Pull Request foi mesclado e fechado

Subindo alterações para o fork local
Somente o selecionado é atualizado

Workflow

"Gatekeeper" Somente no Repositório Principal

Pull Request para o fork realizado!

Material extra

- Vídeo: Sistema de Controle de Versões Git e GitHub
- Autor: Carlos F. Bastarz (Grupo de Assimilação de Dados - DIMNT)
- Link: https://www.youtube.com/watch?v=wZLr6ARrGIU

Sistema de Controle de Versões – Git e GitHub (INPE)

Referências

BASTARZ, C. F. Sistema de Controle de Versões - Git e GitHub (INPE). Disponível em: <https://www.youtube.com/watch?v=wZLr6ARrGIU>.

DEVOPSCOUNSEL. Disponível em: <https://devopscounsel.com/git-update-a-branch-with-remote-main-branch/>.

GitHub Docs. Sobre o Git. Disponível em: <https://docs.github.com/pt/get-started/using-git/about-git>.

HOSTINGER Tutoriais. O Que é GitHub, Para Que Serve e Como Usar. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-github#3_Entenda_Como_Funcionam_Commits_no_GitHub>.

Agradecimentos

Ao Carlos F. Bastarz e ao Grupo DAS

Obrigado

CGCT/DIMNT/GCC