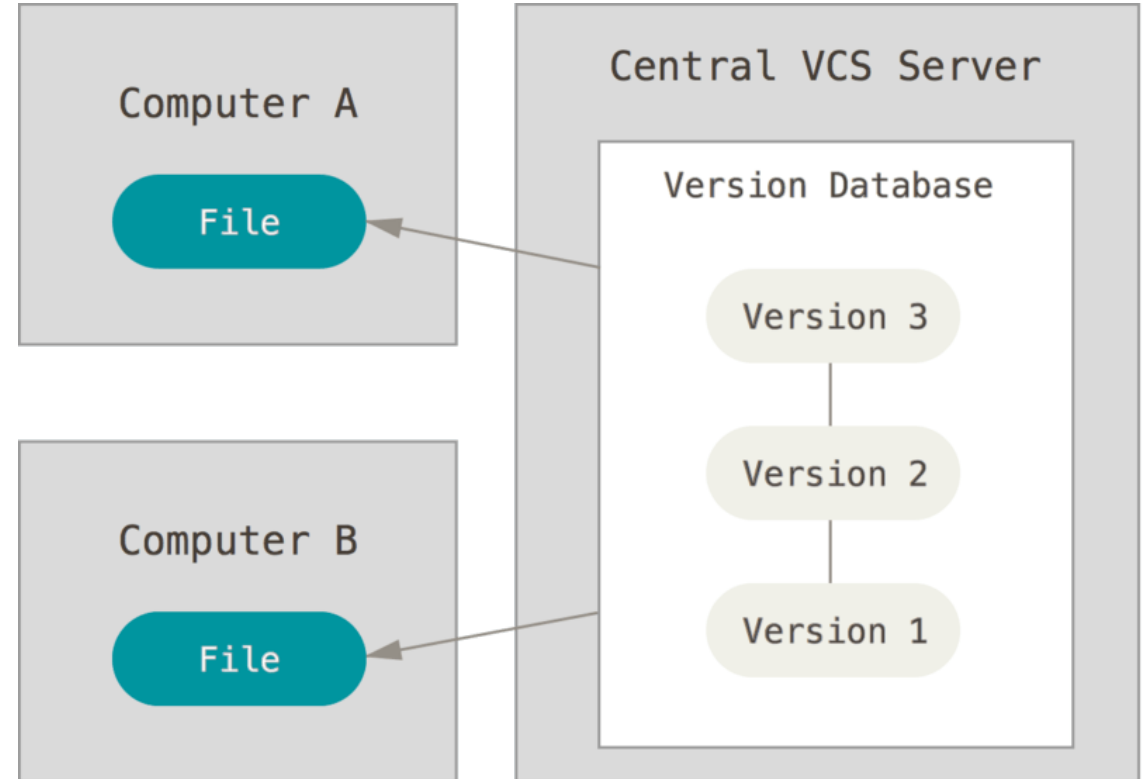


Treinamento Git

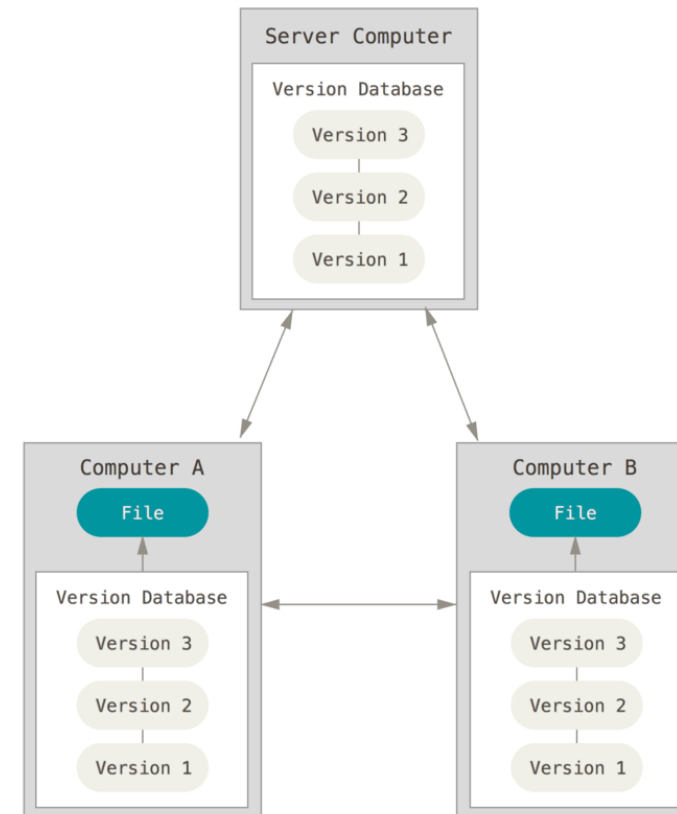
Luis Alexandre Deschamps Brandão

- O que é?
 - Sistema que registra alterações em um conjunto de arquivos ao longo do tempo.
- Porque é preciso?
 - Coordenar mudanças entre vários desenvolvedores
 - Registrar o que mudou, quando mudou e como mudou

- Sistema de versionamento **centralizado**
- Commits são realizados diretamente no server
- Adequado para fluxo de desenvolvimento **linear e serial**.



- Sistema de versionamento **Distribuído**
- Cada copia do repositório é completa e autônoma
- Desenvolvimento assíncrono e distribuído como pilar



O git foi desenvolvido originalmente para se trabalhar como o modelo de desenvolvimento do Kernel:

Desenvolvedores autônomos, geograficamente distribuídos e sem uma gerencia síncrona.

/ Git - instalação



- Site oficial do git:
 - Git-bash - <https://git-scm.com/download/win>
- Interfaces graficas
 - Tortoise - <https://tortoisegit.org/download/>



/ Git – Preparando sua máquina



- Listar a sua configuração
 - `git config --list`
- Configuração obrigatória
 - `git config --global user.name "Luis Alexandre Deschamps Brandão"`
 - `git config --global user.email luis.deschamps.ext@sascar.com.br`
- Configuração extra
 - `git config --global core.editor vim`
 - `git config --global core.editor "'C:\Program Files\Notepad++\notepad++.exe' -multiInst -notabbar -nosession -noPlugin"`
 - `git config --global merge.tool meld`
`git mergetool`



/ Git – Dump dos Comandos básicos



- `Git init`
cria um novo repositório
- `Git clone https://gitlab.com/group/repo`
Clona um repositório existente
- `git status`
Exibe o status do seu repositório local
- `git fetch -pt`
Atualiza localmente as referencias do repositório remoto
- `git branch`
Lista todas as suas branches locais
- `git branch abc-123`
Cria uma nova branch chamada abc-123 com o histórico baseado na árvore atual.
- `git add some/file.txt`
Adiciona a área de staging o arquivo especificado no path
- `git commit`
Commita as alterações presentes na area de staging
- `git checkout some/file.txt`
Descarta as alterações realizadas em file.txt
- `git checkout -b def-5678`
Cria localmente uma nova branch chamada def-5678 e muda a branch ativa para ela
- `git checkout homologacao`
muda para a branch homologacao
- `git merge master`
mergeia a branch master na branch atual
- `git stash`
guarda na área de stash todas as modificações não commitadas
- `git stash pop`
reaplica as modificações salvas no stash na branch atual



- Git init
 - Inicializa o controle de versão, para um novo projeto, no diretório corrente
- Git clone
 - Clona um repositório remoto para a sua máquina
 - A branch “inicial” é a master

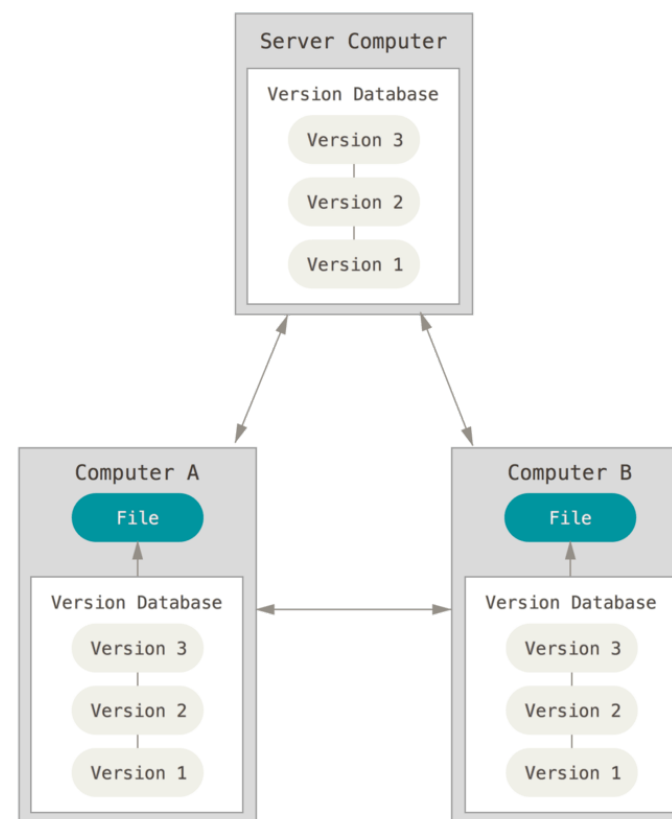
```
luis.deschamps.ext@LuisThinkpad:~/git-treino
luis.deschamps.ext@LuisThinkpad:~/git-treino 79x11
luis.deschamps.ext@LuisThinkpad ~/git-treino $ git init
Initialized empty Git repository in /home/luis.deschamps.ext/git-treino/.git/
luis.deschamps.ext@LuisThinkpad ~/git-treino $
```

```
luis.deschamps.ext@LuisThinkpad:~
luis.deschamps.ext@LuisThinkpad:~ 96x10
luis.deschamps.ext@LuisThinkpad ~ $ git clone git@git.sascar.com.br:bestdriver/java.git
Cloning into 'java'...
remote: Counting objects: 1061, done.
remote: Compressing objects: 100% (898/898), done.
remote: Total 1061 (delta 149), reused 1031 (delta 133)
Receiving objects: 100% (1061/1061), 62.40 MiB | 4.47 MiB/s, done.
Resolving deltas: 100% (149/149), done.
luis.deschamps.ext@LuisThinkpad ~ $
```

/ Local X Remoto



- Em geral, os comando do git executam as operações **localmente** no repositório
- Quando pronta, as operações executadas localmente deve ser sincronizadas com o servidor (git fetch, pull, push...)
- O gitlab é apenas uma interface para o git. Qualquer repositório git é compatível com qualquer sistema git. (github, gogs, bitbucket)



/ Git - trabalhando localmente



A primeira delas é o diretório de trabalho (também chamado de **working tree**), que é onde ficam os arquivos do seu projeto.



Quando você adiciona algum arquivo alterado (**git add**), você envia as suas alterações para o **Index**, que é como uma área de preparação (**staging**) das suas alterações.



/ Git - trabalhando localmente



E por fim, quando você confirma uma nova alteração (**git commit**), você envia ela para a área de **HEAD**, que é onde está a última confirmação que você fez.



```
luis.deschamps.ext@LuisThinkpad:~/tutorial
luis.deschamps.ext@LuisThinkpad:~/tutorial 75x25
(use "git add <file>..." to include in what will be committed)

    produto.php

luis.deschamps.ext@LuisThinkpad ~/tutorial (master) $ vim cliente.php
luis.deschamps.ext@LuisThinkpad ~/tutorial (master) $ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   cliente.php

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   cliente.php
        modified:   usuarios.php

Untracked files:
  (use "git add <file>..." to include in what will be committed)

    produto.php

luis.deschamps.ext@LuisThinkpad ~/tutorial (master) $
```



/ Git - Fluxo básico



- git status
Mostra a sua branch atual, e o que foi alterado.
- git diff
Mostra o que foi modificado em um arquivo
- git add main.c
Adiciona para a área de staging o arquivo main.c
- git add *
Move todos os arquivos do diretório atual para a área de staging
- git commit -m "Adiciona a função cliente"
Executa (localmente) o commit dos arquivos da área de staging com o comentário especificado.

```
luis.deschamps.ext@LuisThinkpad:~/tutorial
luis.deschamps.ext@LuisThinkpad:~/tutorial 75x34
luis.deschamps.ext@LuisThinkpad ~/tutorial (master) $ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   main.c

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        cliente.c

no changes added to commit (use "git add" and/or "git commit -a")
luis.deschamps.ext@LuisThinkpad ~/tutorial (master) $ git diff main.c
diff --git a/main.c b/main.c
index 53e8644..0578916 100644
--- a/main.c
+++ b/main.c
@@ -1,3 +1,6 @@
+#include <cliente.c>
+
+main() {
+    cliente();
+}

luis.deschamps.ext@LuisThinkpad ~/tutorial (master) $ git add main.c
luis.deschamps.ext@LuisThinkpad ~/tutorial (master) $ git add cliente.c
luis.deschamps.ext@LuisThinkpad ~/tutorial (master) $ git commit -m "Adiciona a função cliente"
[master 97ed865] Adiciona a função cliente
 2 files changed, 6 insertions(+)
 create mode 100644 cliente.c
luis.deschamps.ext@LuisThinkpad ~/tutorial (master) $
```



- Origin
nome padrão do repositório remoto principal
- Git remote add origin <endereço>
Adiciona uma origin ao repositório atual
- git fetch -pt
O git fetch atualiza as referências do repositório remoto, mas não altera o código local
- git pull
Atualiza histórico da branch atual com o que está no remoto
- git push
- git push origin branch-123
Envia os commits da branch atual para o repositório remoto

- **Mas o que é uma branch?**

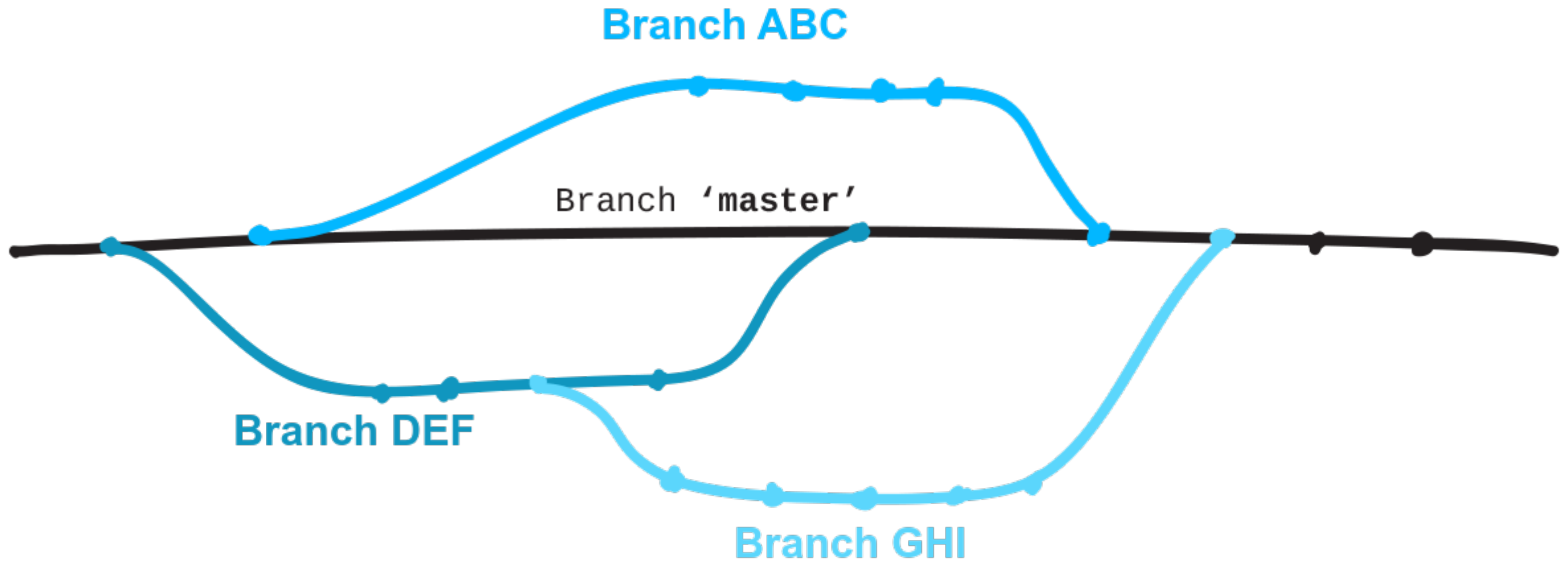
Uma branch é uma ramificação do utilizada para se isolar o desenvolvimento de uma nova funcionalidade.

- **Quando devo criar uma nova branch?**

Sempre! Convencionamos na sascar, que as branches devem ter como nome, a task do jira relacionada.

- **Subi minha branch e ela não passou no teste, posso commitar a correção direto na branch de homologação?**

Não. Em nenhum momento deve-se fazer commits diretamente em uma branch de controle. As branches de controle apenas recebem merges.



GIT
Branches

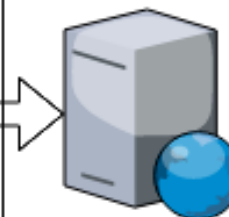


Fluxo git

Ambientes nos
servidores

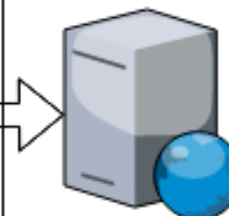


Desenvolvimento



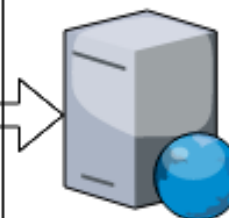
Desenvolvimento

Testes_integrados



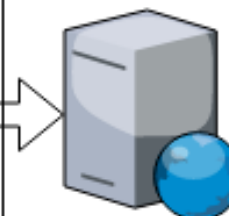
Testes Integrados

Homologacao



Homologacao

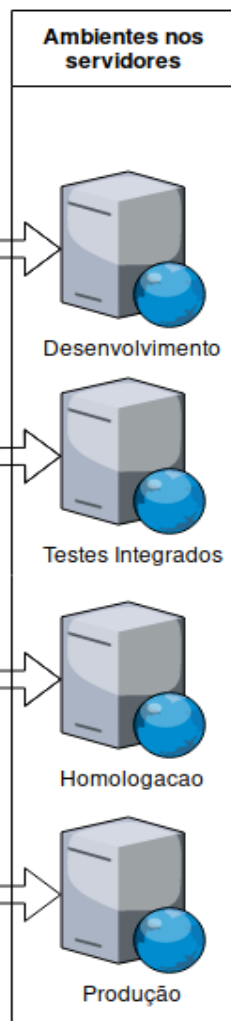
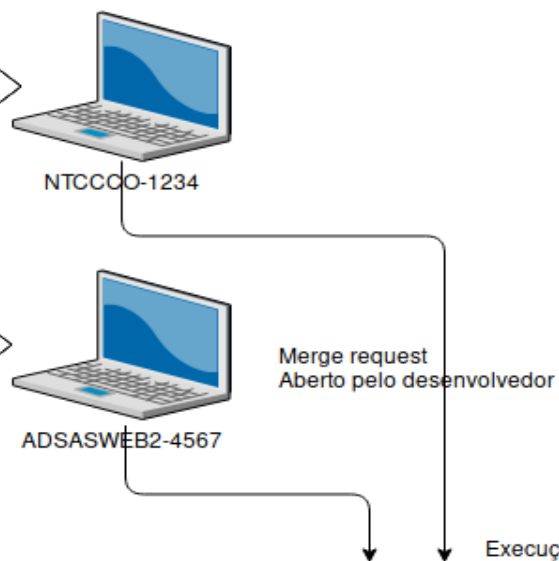
Master



Produção



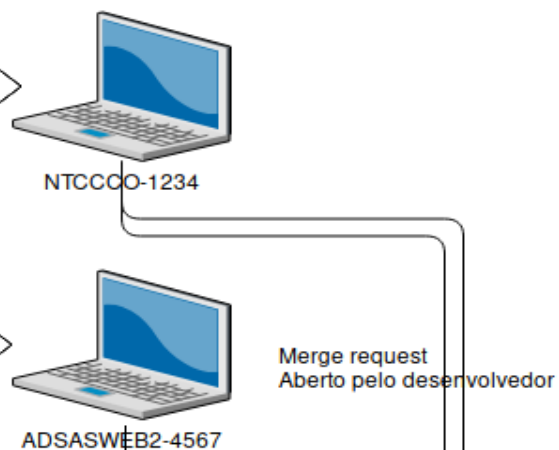
GIT Branches	
NTCCCO-1234	
ADSASWEB2-4567	
Desenvolvimento	
Testes_integrados	
Homologacao	
Master	



sascar



GIT Branches	
NTCCCO-1234	
ADSASWEB2-4567	
Desenvolvimento	
Testes_integrados	
Homologacao	
Master	



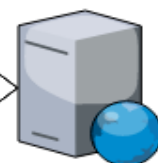
Merge request
Aberto pelo desenvolvedor

Execução do merge pelo Desenvolvedor

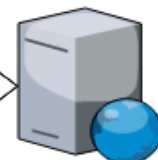
Label: Reviewed

Execução do merge pelo QA

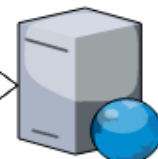
Ambientes nos
servidores



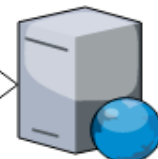
Desenvolvimento



Testes Integrados



Homologacao



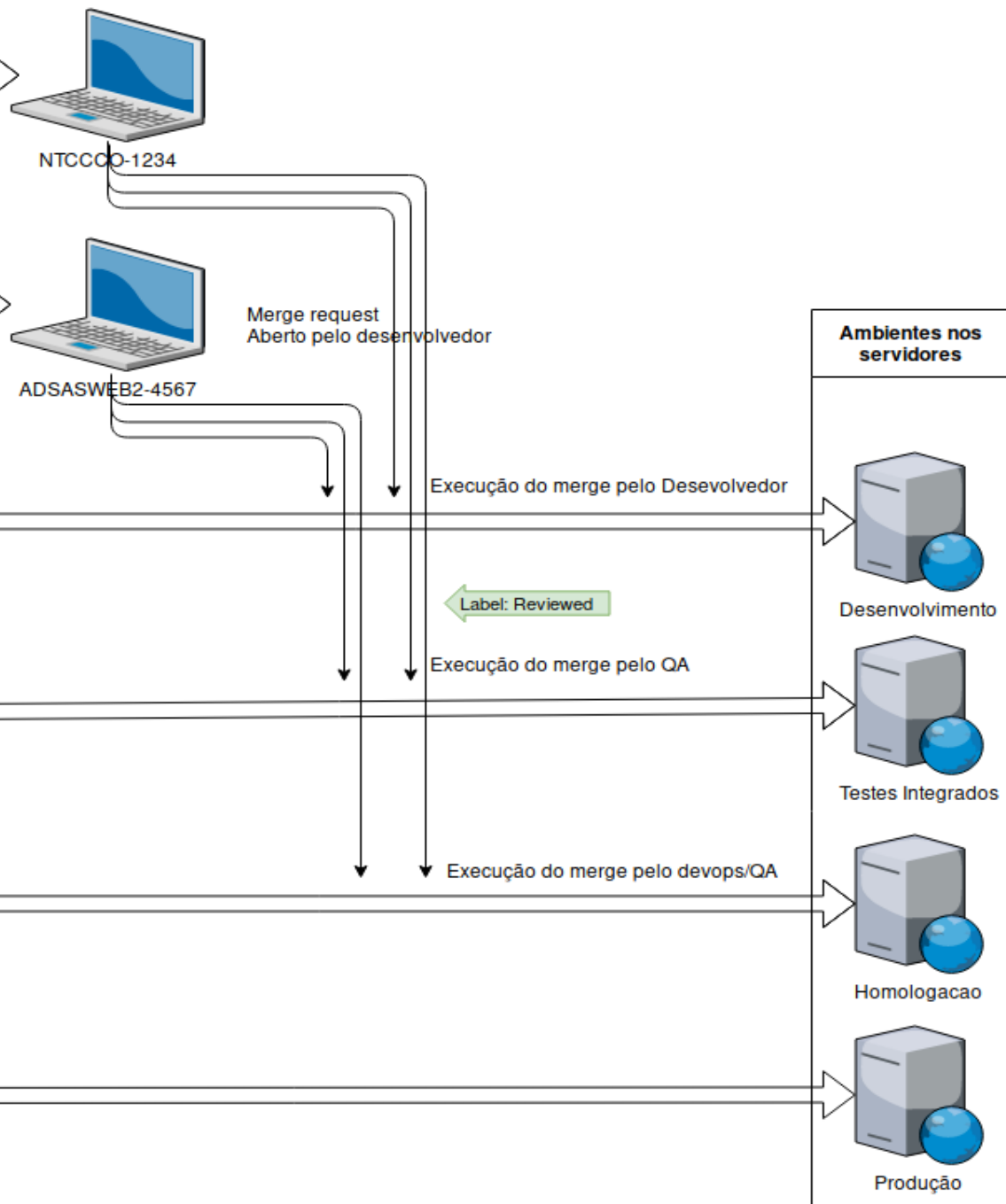
Produção



sascar



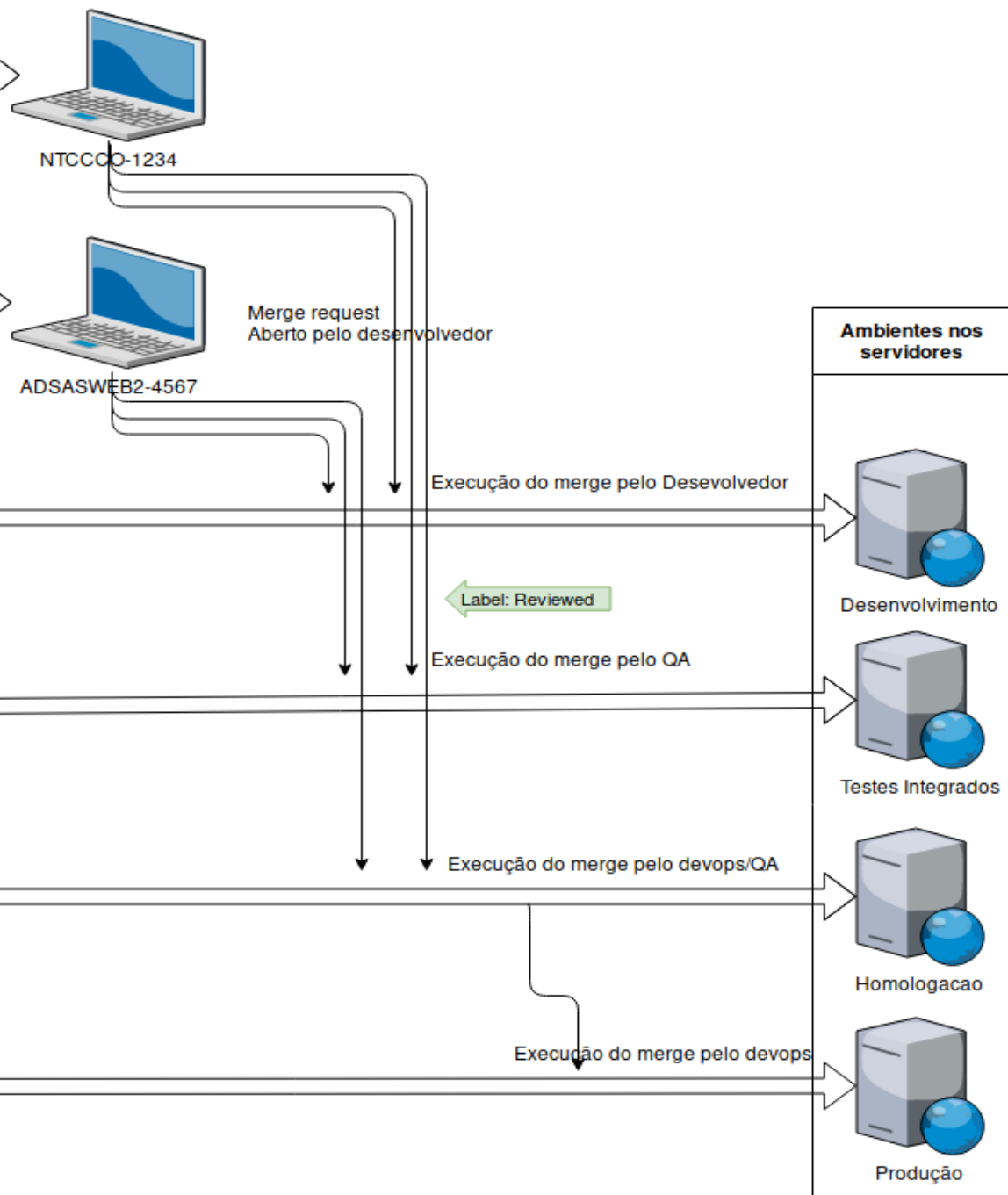
GIT Branches	
NTCCCO-1234	
ADSASWEB2-4567	
Desenvolvimento	
Testes_integrados	
Homologacao	
Master	



sascar



GIT Branches	
NTCCCO-1234	
ADSASWEB2-4567	
Desenvolvimento	
Testes_integrados	
Homologacao	
Master	



sascar



/ Fluxo de trabalho – criação de branch



luis.deschamps.ext@LuisThinkpad:~/git/front-gestao-frotas

luis.deschamps.ext@LuisThinkpad:~/git/front-gestao-frotas 115x23

```
luis.deschamps.ext@LuisThinkpad ~/git/front-gestao-frotas (desenvolvimento) $ git checkout master
```

Switched to branch 'master'

Your branch is up-to-date with 'origin/master'.

```
luis.deschamps.ext@LuisThinkpad ~/git/front-gestao-frotas (master) $ git fetch -pt
```

remote: Counting objects: 78, done.

remote: Compressing objects: 100% (72/72), done.

remote: Total 78 (delta 41), reused 11 (delta 6)

Unpacking objects: 100% (78/78), done.

From git.sascar.com.br:gestao-frotas/front-gestao-frotas

* [new branch] NTCCCO-1083 -> origin/NTCCCO-1083

52b323468..71b1c49cc NTCCCO-1235 -> origin/NTCCCO-1235

eda2bb21b..4c017634d NTCCCO-1698 -> origin/NTCCCO-1698

* [new branch] NTCCCO-1698-DEV -> origin/NTCCCO-1698-DEV

d5ea45838..d9729c554 desenvolvimento -> origin/desenvolvimento

9c0d1b306..9486c2bab testes_integrados -> origin/testes_integrados

```
luis.deschamps.ext@LuisThinkpad ~/git/front-gestao-frotas (master) $ git pull
```

Already up-to-date.

```
luis.deschamps.ext@LuisThinkpad ~/git/front-gestao-frotas (master) $ git checkout -b ORGTECDVOP-123
```

Switched to a new branch 'ORGTECDVOP-123'

```
luis.deschamps.ext@LuisThinkpad ~/git/front-gestao-frotas (ORGTECDVOP-123) $ git status
```

On branch ORGTECDVOP-123

nothing to commit, working tree clean

```
luis.deschamps.ext@LuisThinkpad ~/git/front-gestao-frotas (ORGTECDVOP-123) $
```

Fluxo de trabalho - edição/commit










```
luis.deschamps.ext@LuisThinkpad:~/git/front-gestao-frotas
luis.deschamps.ext@LuisThinkpad:~/git/front-gestao-frotas 115x28
luis.deschamps.ext@LuisThinkpad ~/git/front-gestao-frotas (ORGTECDVOP-123) $ git status
On branch ORGTECDVOP-123
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)


        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
luis.deschamps.ext@LuisThinkpad ~/git/front-gestao-frotas (ORGTECDVOP-123) $ git add README.md
luis.deschamps.ext@LuisThinkpad ~/git/front-gestao-frotas (ORGTECDVOP-123) $ git commit -m "Adicionado descrição ao README.md"
[ORGTECDVOP-123 6873402b2] Adicionado descrição ao README.md
 1 file changed, 4 insertions(+)
luis.deschamps.ext@LuisThinkpad ~/git/front-gestao-frotas (ORGTECDVOP-123) $ git push origin ORGTECDVOP-123
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 398 bytes | 398.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote:
remote: To create a merge request for ORGTECDVOP-123, visit:
remote:   https://git.sascar.com.br/gestao-frotas/front-gestao-frotas/merge_requests/new?merge_request%5Bsource_branch%5D=ORGTECDVOP-123
remote:
To git.sascar.com.br:gestao-frotas/front-gestao-frotas.git
 * [new branch]      ORGTECDVOP-123 -> ORGTECDVOP-123
luis.deschamps.ext@LuisThinkpad ~/git/front-gestao-frotas (ORGTECDVOP-123) $
```

/ Gitlab – merge request



Groups Activity Milestones Snippets   This project Search     1 



gestao-frotas >  front-gestao-frotas > Merge Requests

New Merge Request

Source branch

gestao-frotas/front-gestao-frotas ▾

ORGTECDVOP-123 ▾



 **Adicionado descrição ao README.md**
Luis Alexandre Deschamps Brandão committed 13 minutes ago
6873402b 

Compare branches and continue

Target branch

gestao-frotas/front-gestao-frotas ▾

testes_integrados ▾

 **Merge branch 'NTCCCO-1083' into 'testes_integrados'** ...
Guztavo de Sousa Franco Grande committed 45 minutes ago
9486c2ba 



front-gestao-frotas

- Overview
- Repository
- Issues 0
- Merge Requests 0
- CI / CD
- Wiki
- Snippets
- Settings

New Merge Request

From **ORGTECDVOP-123** into **testes_integrados**

[Change branches](#)

Title

Start the title with **WIP:** to prevent a **Work In Progress** merge request from being merged before it's ready.

Add [description templates](#) to help your contributors communicate effectively!

Description

Write Preview

B **I** **”** **</>**

Write a comment or drag your files here...

Markdown and [quick actions](#) are supported

[Attach a file](#)

Assignee

[Assign to me](#)

Milestone

Labels

Source branch

Target branch

[Change branches](#)

☐ Remove source branch when merge request is accepted.

Submit merge request

Cancel

- Git diff
 - `git diff a68u2o1 78231ihe`
Compara a diferença entre dois commits
 - `git diff master homologacao`
Compara a branch master com a branch homologacao
 - `git blame arquivo.php`
Lista quem alterou cada linha de um arquivo
 - `git log`
Exibe a historia da branch atual

/ Git - Conflitos



```
luis.deschamps.ext@LuisThinkpad:~/git/tutorial
luis.deschamps.ext@LuisThinkpad:~/git/tutorial 109x22
luis.deschamps.ext@LuisThinkpad ~/git/tutorial (ORGTECDVOP-123) $ git merge master
Auto-merging main.c
CONFLICT (content): Merge conflict in main.c
Automatic merge failed; fix conflicts and then commit the result.
luis.deschamps.ext@LuisThinkpad ~/git/tutorial (ORGTECDVOP-123) $ git status
On branch ORGTECDVOP-123
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)

        both modified:   main.c

no changes added to commit (use "git add" and/or "git commit -a")
luis.deschamps.ext@LuisThinkpad ~/git/tutorial (ORGTECDVOP-123) $ vim main.c
luis.deschamps.ext@LuisThinkpad ~/git/tutorial (ORGTECDVOP-123) $ git add main.c
luis.deschamps.ext@LuisThinkpad ~/git/tutorial (ORGTECDVOP-123) $ git commit
[ORGTECDVOP-123 40a90f4] Merge branch 'master' into ORGTECDVOP-123
luis.deschamps.ext@LuisThinkpad ~/git/tutorial (ORGTECDVOP-123) $
```

/ Git - conflitos



luis.deschamps.ext@LuisThinkpad:~/git/tutorial

luis.deschamps.ext@LuisThinkpad:~/git/tutorial 109x22

```
/* add.c
 * a simple C program
 */

#include <stdio.h>
#define LAST 10
<<<<<<< HEAD

int subtract(int a, int b){
    return (a-b);
}

int sum(int a, int b) {
    return a + b;
}

>>>>>>> master
}

int main()
{
    int i, sum = 0;
```

"main.c" 28L, 355C

14,1

Topo

Para manter a consistencia, algumas branches podem ser recriadas a partir da master.

Neste caso, qualquer copia local da branch deve ser descartada. O não descarte da branch resultará na reinserção do código removido na branch.

/ Recriação/Limpeza de branch



```
luis.deschamps.ext@LuisThinkpad ~/git/gitlab/gestao-frotas/front-gestao-frotas (testes_integrados) $ git status
On branch testes_integrados
Your branch and 'origin/testes_integrados' have diverged,
and have 2 and 24 different commits each, respectively.
  (use "git pull" to merge the remote branch into yours)

nothing to commit, working tree clean
luis.deschamps.ext@LuisThinkpad ~/git/gitlab/gestao-frotas/front-gestao-frotas (testes_integrados) $
```





SASCAR. TECNOLOGIA QUE TE LEVA MAIS LONGE.