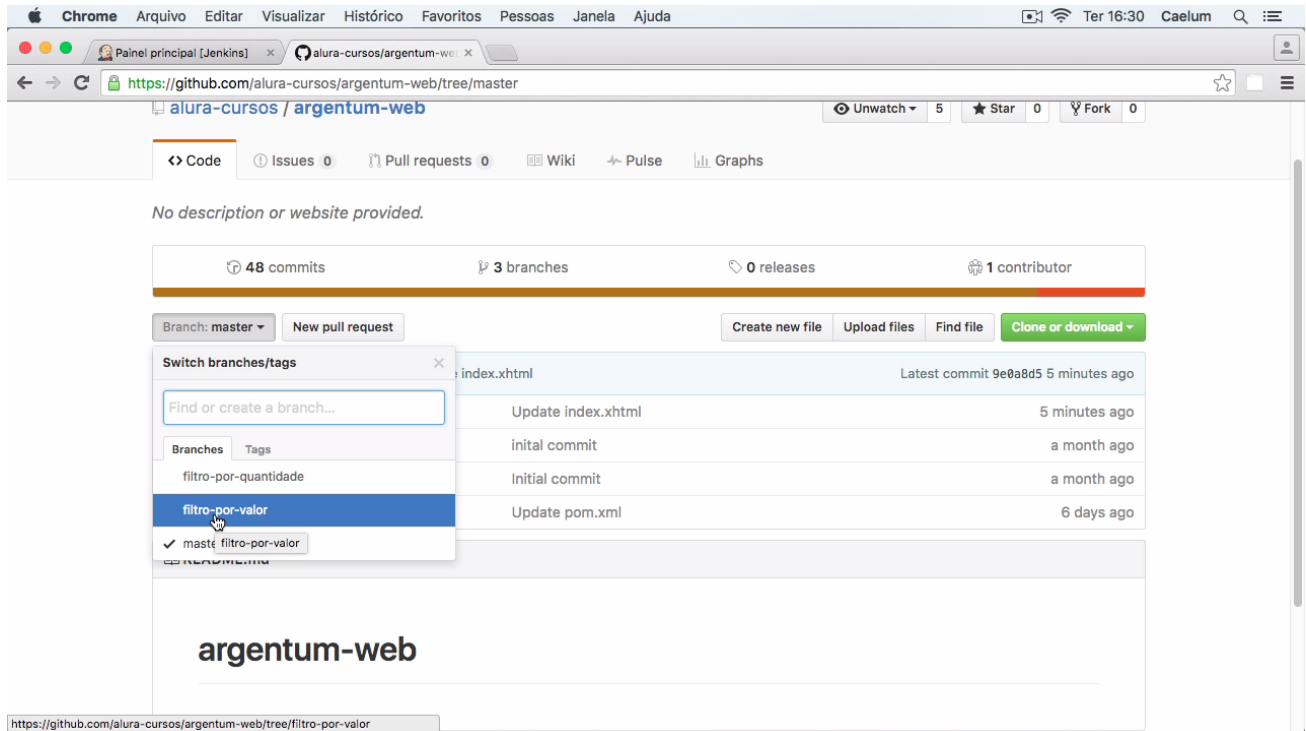


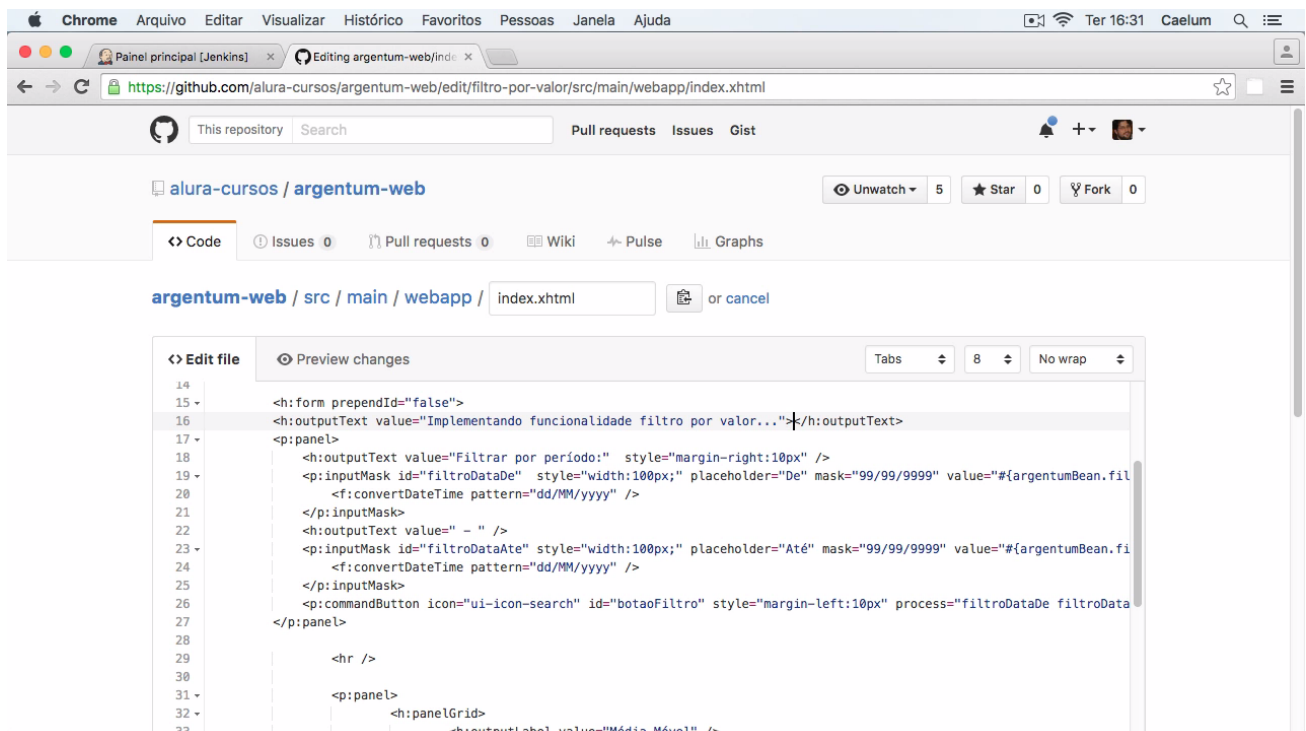
Merge com Jenkins

Transcrição

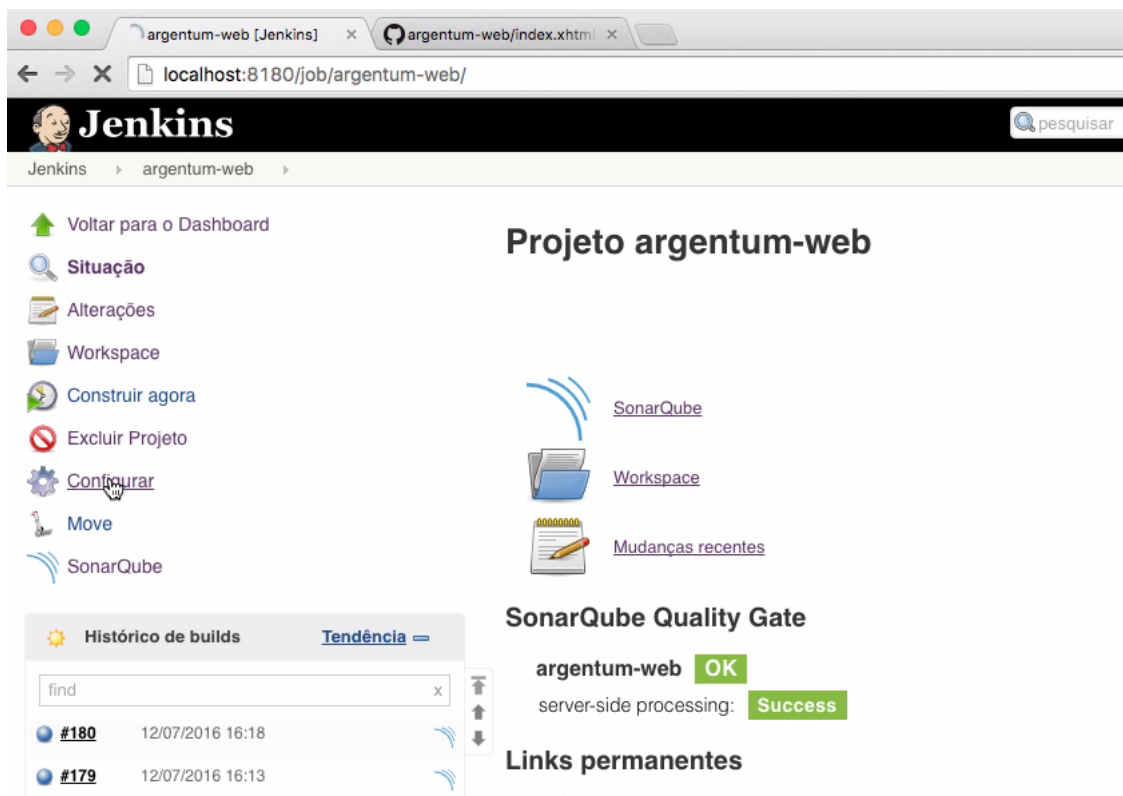
Para passarmos por esse tipo de cenário criamos duas branches no repositório do argentum-web: filtro-por-quantidade e filtro-por-valor. Basicamente duas branches para duas funcionalidades completamente distintas.



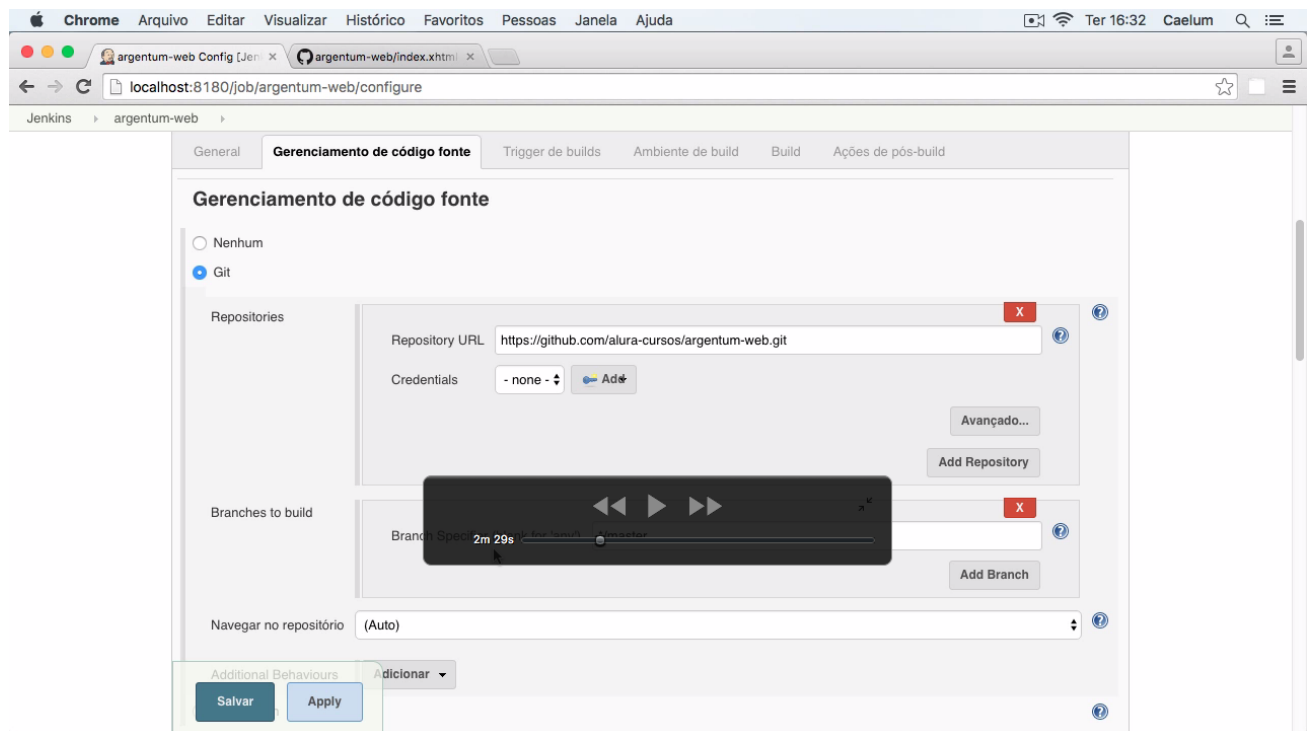
Na funcionalidade filtro-por-valor, iremos alterar o arquivo index.xhtml, colocando apenas um outputText de exemplo e aplicando um commit.



Agora queremos construir nosso projeto testando se essa alteração continua deixando o projeto sem erros tanto na compilação quanto nos testes. Portanto vamos agora voltar para o nosso job e fazer as configurações necessárias. Para isso basta clicar no job e depois em configurar:



Na aba de Gerenciamento de código fonte podemos notar que o repositório utiliza-se do controlador de versão GIT com um endereço do github. E o mais importante nessa tela para o assunto que estamos trabalhando é que o branch principal do nosso build já está configurado para ser o MASTER, embora o campo também deixe claro que caso não seja preenchido podemos fazer o build de qualquer branch ("blank for any").

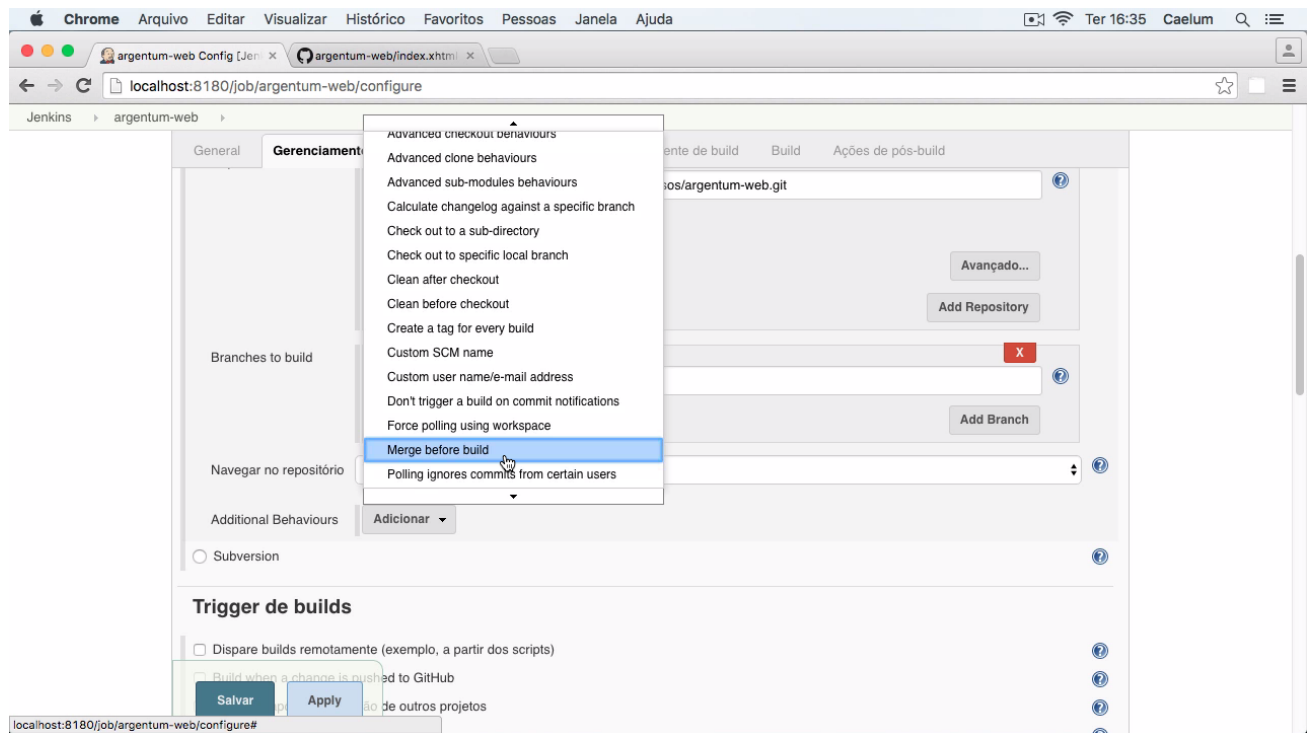


Vamos deixar esse campo em branco para que ele construa para a gente em todas as branches. Ou seja, com essa simples alteração o Jenkins vai executar todas as fases do nosso build(compilação, execução de testes, ...) para cada um

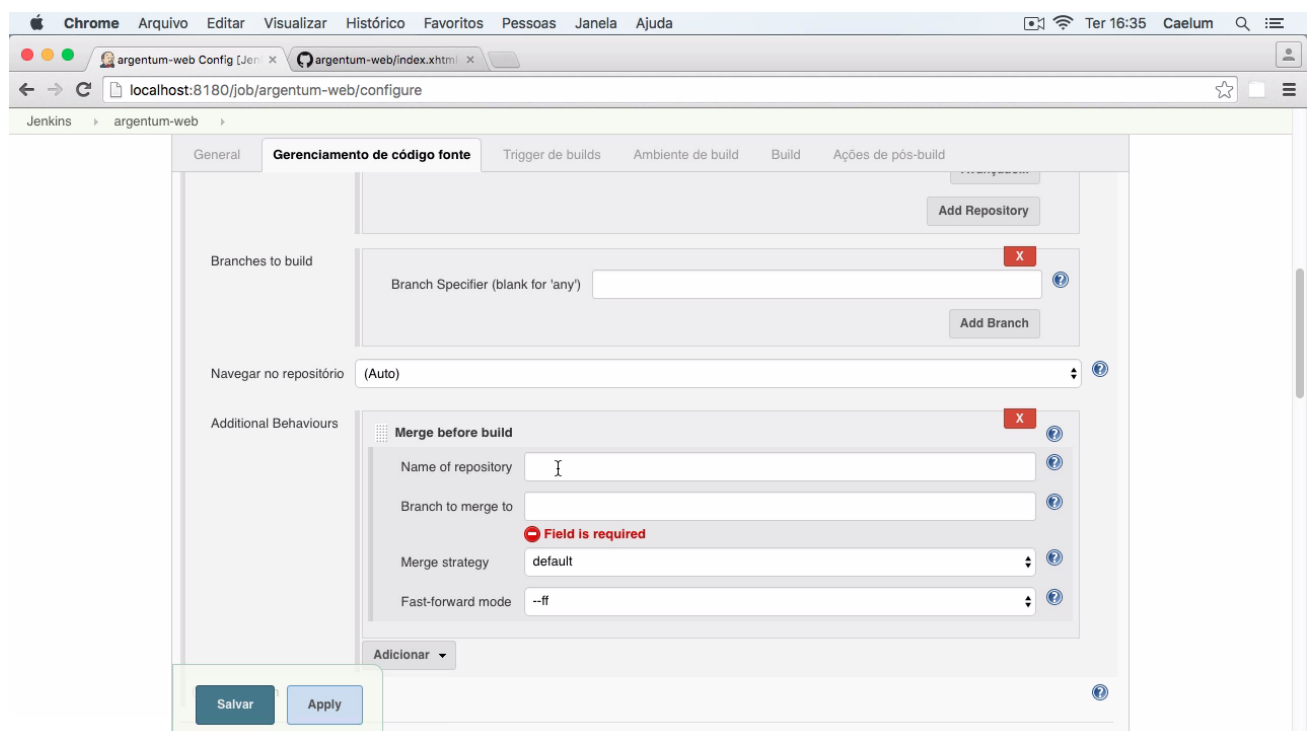
dos branches que o nosso repositório tiver. Parece bastante interessante e mais efetivo do que criar um job para cada branch não é mesmo? E ela pode nos auxiliar ainda mais.

O que foi feito, e é preciso ficarmos muito atento, é que da forma como está configurado estamos validando que cada branch funciona isoladamente muito bem. Mas seria isso integrar continuamente? Nosso código não deveria funcionar efetivamente junto? Em outras palavras gostaríamos de avisar ao Jenkins para que ele faça um merge desses branches e verifique se todo esse código continua buildando corretamente de forma única.

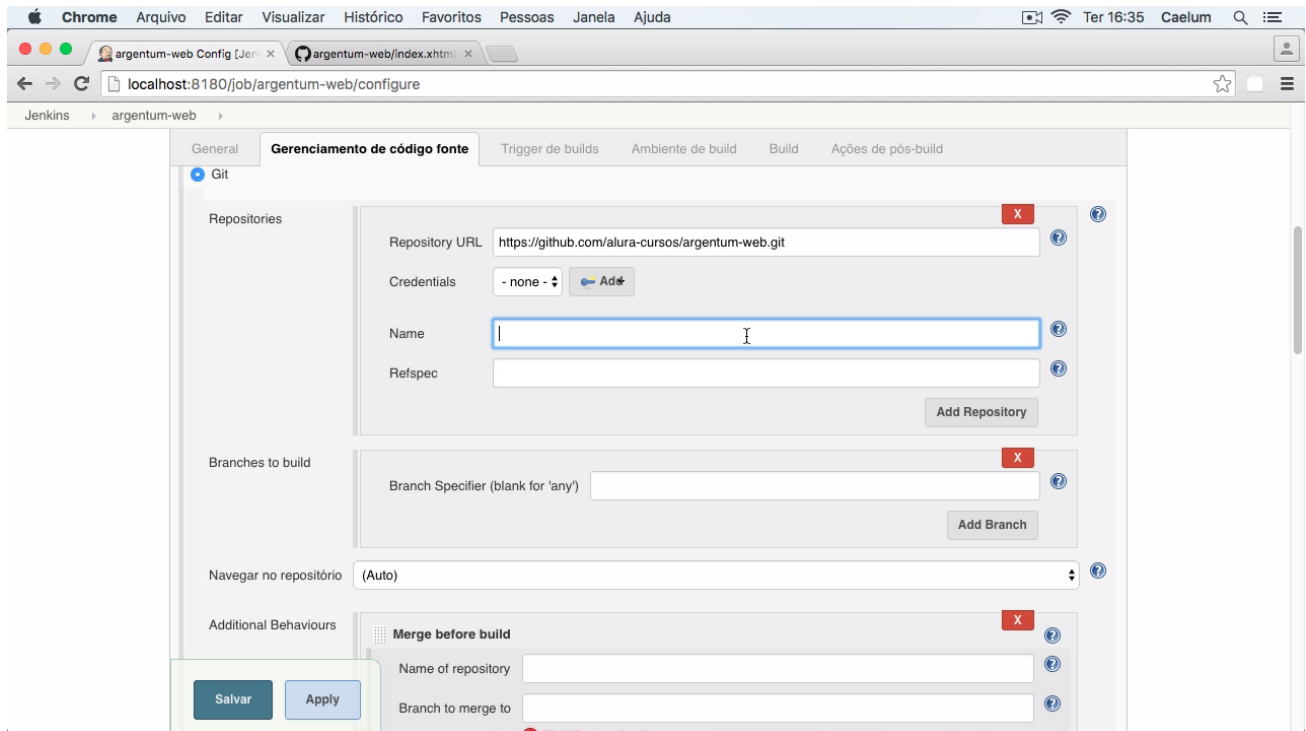
Vamos então configurar o Jenkins para que ele faça o merge antes do build pra gente. No campo 'Additional Behaviours' vamos adicionar um comportamento que vai ser o 'Merge before build'.



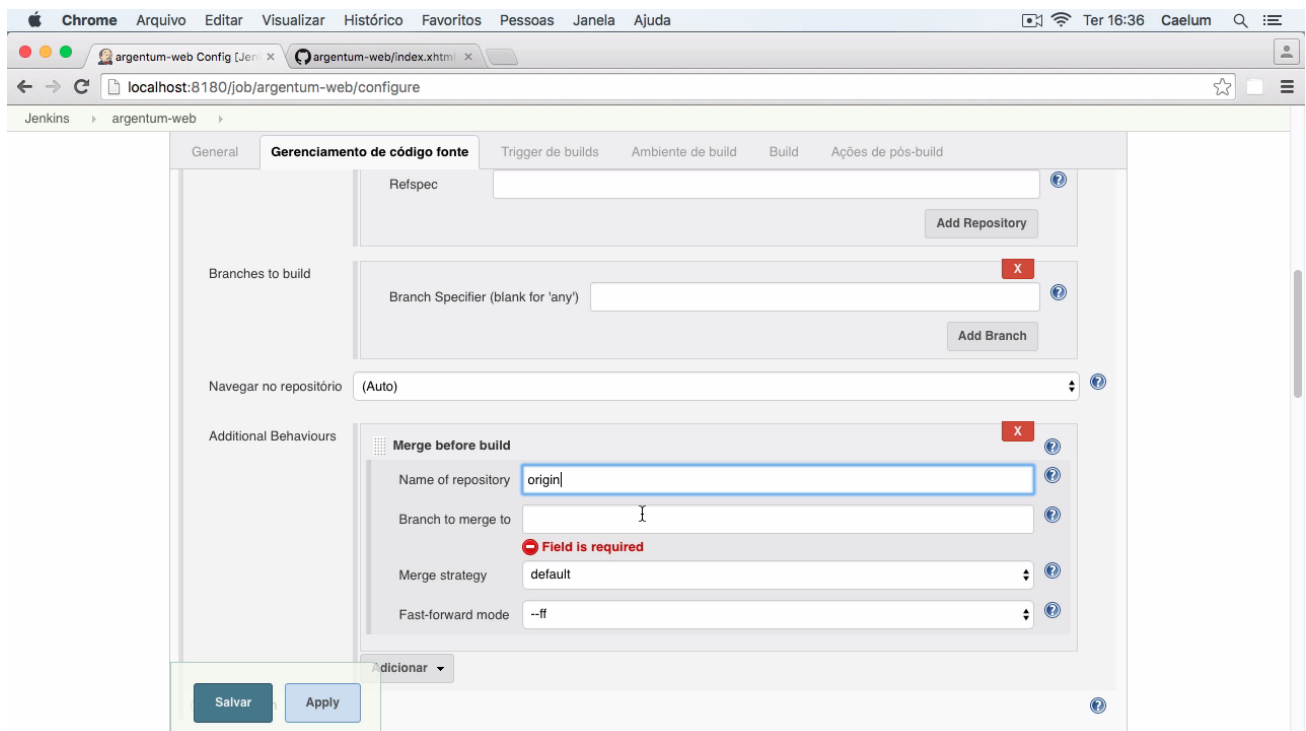
Logo após essa adição, ele pede mais algumas configurações:



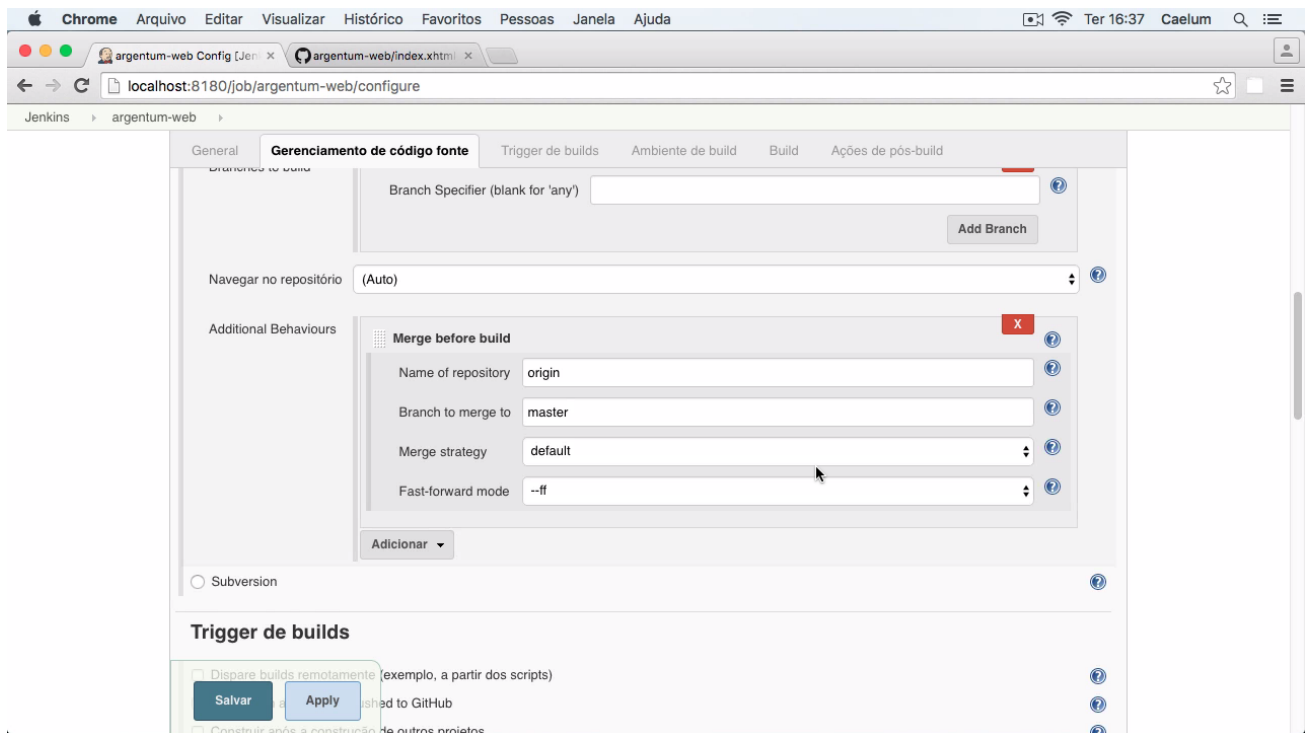
O nome do repositório é o que tínhamos feito lá em cima. Para isso vamos nomear o repositório na seção 'Repositories' ao clicar em 'Avançado' abre-se essa opção:



Observe que agora foi nomeado para 'origin'. Portanto agora podemos voltar as configurações do comportamento e preencher o nome do repositório como 'origin'.



No campo 'Branch to merge' vamos preencher com Master. É bom saber que não necessariamente o branch de integração vai ser sempre o master, poderíamos usar outro. A regra fundamental é que o branch escolhido não seja usado pelos membros da equipe para alterações.



Agora eu quero que após a realização do merge e o build sendo feito com sucesso, um push seja disparado para o repositório remoto(github). Isso conseguimos fazer lá na aba Ações pós build e na seção 'Git publisher', marcando a opção Push Only if Builds Succeeds.



Além disso vamos configurar para que um merge de resultado seja feito no branch de integração, no nosso caso o MASTER, de forma automática quando o build tiver sido feito com sucesso.



Agora que o Jenkins está configurado para fazer um push para o nosso repositório é necessário cadastrarmos as credenciais. Até então isso não foi necessário pois ele não alterava nada no repositório apenas baixava o fonte. Para isso devemos ir na configuração do repositório para adicionar credenciais:

The screenshot shows the Jenkins configuration page for a Git repository. The 'Repositories' section is active, showing the 'Repository URL' as `https://github.com/alura-cursos/argentum-web.git`. The 'Credentials' dropdown is set to '- none -', and a blue 'Add' button is visible. The 'Name' field is set to 'origin'. The 'Refspec' field is empty. Below this, the 'Branches to build' section is visible, with a 'Branch Specifier (blank for 'any')' field. At the bottom, there is a 'Navegar no repositório' dropdown set to '(Auto)'.

Agora basta dizermos o usuário e senha lá do Github:

The screenshot shows the 'Jenkins Credentials Provider: Jenkins' dialog box. The 'Add Credentials' section is active. The 'Domain' is set to 'Global credentials (unrestricted)'. The 'Kind' is set to 'Username with password'. The 'Username' field is filled with 'soeiosantos'. The 'Password' field is filled with a masked password '*****'. The 'ID' and 'Description' fields are empty. At the bottom, there are 'Add' and 'Cancel' buttons.

Depois de adicionar não podemos esquecer de setar essa credencial:

The screenshot shows the Jenkins configuration page for a Git repository, similar to the first one. The 'Credentials' dropdown is now set to 'soeiosantos/*****', indicating that the previously created credential has been selected. The 'Repository URL' remains `https://github.com/alura-cursos/argentum-web.git` and the 'Name' is still 'origin'.

Feito tudo isso, basta salvarmos e acionarmos uma construção. Acompanhando o build pelo console podemos notar que o Jenkins está olhando por todos os branches.



Saída do console

Progresso:

```

Iniciado pelo usuário Administrador Jenkins
Construindo no workspace /Users/Caelum/.jenkins/workspace/argentum-web
> git rev-parse --is-inside-work-tree # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/alura-cursos/argentum-web.git # timeout=10
Fetching upstream changes from https://github.com/alura-cursos/argentum-web.git
> git --version # timeout=10
using .gitcredentials to set credentials
> git config --local credential.username soeirosantos # timeout=10
> git config --local credential.helper store --
file=/var/folders/pk/scy262d55zz1d15zk12sz8qc0000gn/T/git8112277299753414082.credentials # timeout=10
> git -c core.askpass=true fetch --tags --progress https://github.com/alura-cursos/argentum-web.git
+refs/heads/*:refs/remotes/origin/*
> git config --local --remove-section credential # timeout=10
Seen branch in repository origin/filtro-por-quantidade
Seen branch in repository origin/filtro-por-valor
Seen branch in repository origin/master
Seen 3 remote branches

```

E o merging está sendo feito também:

```

Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/alura-cursos/argentum-web.git # timeout=10
Fetching upstream changes from https://github.com/alura-cursos/argentum-web.git
> git --version # timeout=10
using .gitcredentials to set credentials
> git config --local credential.username soeirosantos # timeout=10
> git config --local credential.helper store --
file=/var/folders/pk/scy262d55zz1d15zk12sz8qc0000gn/T/git8112277299753414082.credentials # timeout=10
> git -c core.askpass=true fetch --tags --progress https://github.com/alura-cursos/argentum-web.git
+refs/heads/*:refs/remotes/origin/*
> git config --local --remove-section credential # timeout=10
Seen branch in repository origin/filtro-por-quantidade
Seen branch in repository origin/filtro-por-valor
Seen branch in repository origin/master
Seen 3 remote branches
Merging Revision elc09b7b6dc948b66c633bf950b6eebe7c630136 (origin/filtro-por-valor) to origin/master,
UserMergeOptions{mergeRemote='origin', mergeTarget='master', mergeStrategy='default', fastForwardMode='--ff'}
> git rev-parse origin/master^{commit} # timeout=10
> git config core.sparsecheckout # timeout=10
> git checkout -f origin/master

```

Além disso os testes também estão sendo executados:

```

-----
T E S T S
-----
objc[2709]: Class JavaLaunchHelper is implemented in both
/Library/Java/JavaVirtualMachines/jdk1.8.0_51.jdk/Contents/Home/jre/bin/java and
/Library/Java/JavaVirtualMachines/jdk1.8.0_51.jdk/Contents/Home/jre/lib/libinstrument.dylib. One of the two will
be used. Which one is undefined.
Running br.com.caelum.argentum.indicadores.MediaMove1PonderadaTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.256 sec
Running br.com.caelum.argentum.indicadores.MediaMove1SimplesTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.01 sec
Running br.com.caelum.argentum.modelo.CandlestickFactoryTest
Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.024 sec
Running br.com.caelum.argentum.modelo.NegociacaoTest
Tests run: 6, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.004 sec
Running br.com.caelum.argentum.reader.LeitorXMLTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.943 sec
Results :

Tests run: 13, Failures: 0, Errors: 0, Skipped: 0

```

E também está subindo o Tomcat para executar os testes de integração:

```

objc[2711]: Class JavaLaunchHelper is implemented in both
/Library/Java/JavaVirtualMachines/jdk1.8.0_51.jdk/Contents/Home/jre/bin/java and
/Library/Java/JavaVirtualMachines/jdk1.8.0_51.jdk/Contents/Home/jre/lib/libinstrument.dylib. One of the two will
be used. Which one is undefined.
Running br.com.caelum.argentum.bean.FiltroNegociacoesIT
jul 12, 2016 4:39:54 PM org.apache.coyote.AbstractProtocol init
INFORMAÇÕES: Initializing ProtocolHandler ["http-bio-8888"]
jul 12, 2016 4:39:54 PM org.apache.catalina.core.StandardService startInternal
INFORMAÇÕES: Starting service arquillian-tomcat-embedded-7
jul 12, 2016 4:39:54 PM org.apache.catalina.core.StandardEngine startInternal
INFORMAÇÕES: Starting Servlet Engine: Apache Tomcat/7.0.64
jul 12, 2016 4:39:54 PM org.apache.coyote.AbstractProtocol start
INFORMAÇÕES: Starting ProtocolHandler ["http-bio-8888"]

```

Em mais alguns instantes ele vai publicar os dados do Sonar:

```

[INFO] --- maven-pmd-plugin:3.6:pmd (default) @ argentum-web ---
[WARNING] Unable to locate Source XRef to link to - DISABLED
[INFO]
[INFO] --- jacoco-maven-plugin:0.7.6.201602180812:report (default) @ argentum-web ---
[INFO] Analyzed bundle 'argentum-web' with 12 classes
[INFO]
[INFO] --- maven-failsafe-plugin:2.19.1:verify (default) @ argentum-web ---
[INFO]
[INFO] -----
[INFO] Building argentum-web 1.0.0-SNAPSHOT
[INFO] -----
[INFO]
[INFO] --- sonar-maven-plugin:3.0.2:sonar (default-cli) @ argentum-web ---
[INFO] User cache: /Users/Caelum/.sonar/cache
[INFO] Load global repositories
[INFO] Load global repositories (done) | time=256ms
[INFO] User cache: /Users/Caelum/.sonar/cache
[INFO] Load plugins index
[INFO] Load plugins index (done) | time=3ms
[INFO] SonarQube version: 5.5

```

Além disso envia os dados para o push no Github:

```

[INFO] -----
Injecting SonarQube environment variables using the configuration: SonarQube
> git tag -l jenkins-argentum-web-181 # timeout=10
> git tag -a -f -m Jenkins Build #181 jenkins-argentum-web-181-SUCCESS # timeout=10
Pushing HEAD to branch master of origin repository
> git --version # timeout=10
using .gitcredentials to set credentials
> git config --local credential.username soeirosantos # timeout=10
> git config --local credential.helper store --
file=/var/folders/pk/scy262d5zz1d15zk12sz8qc0000gn/T/git1852582023301227745.credentials # timeout=10
> git -c core.askpass=true push https://github.com/alura-cursos/argentum-web.git HEAD:master

```

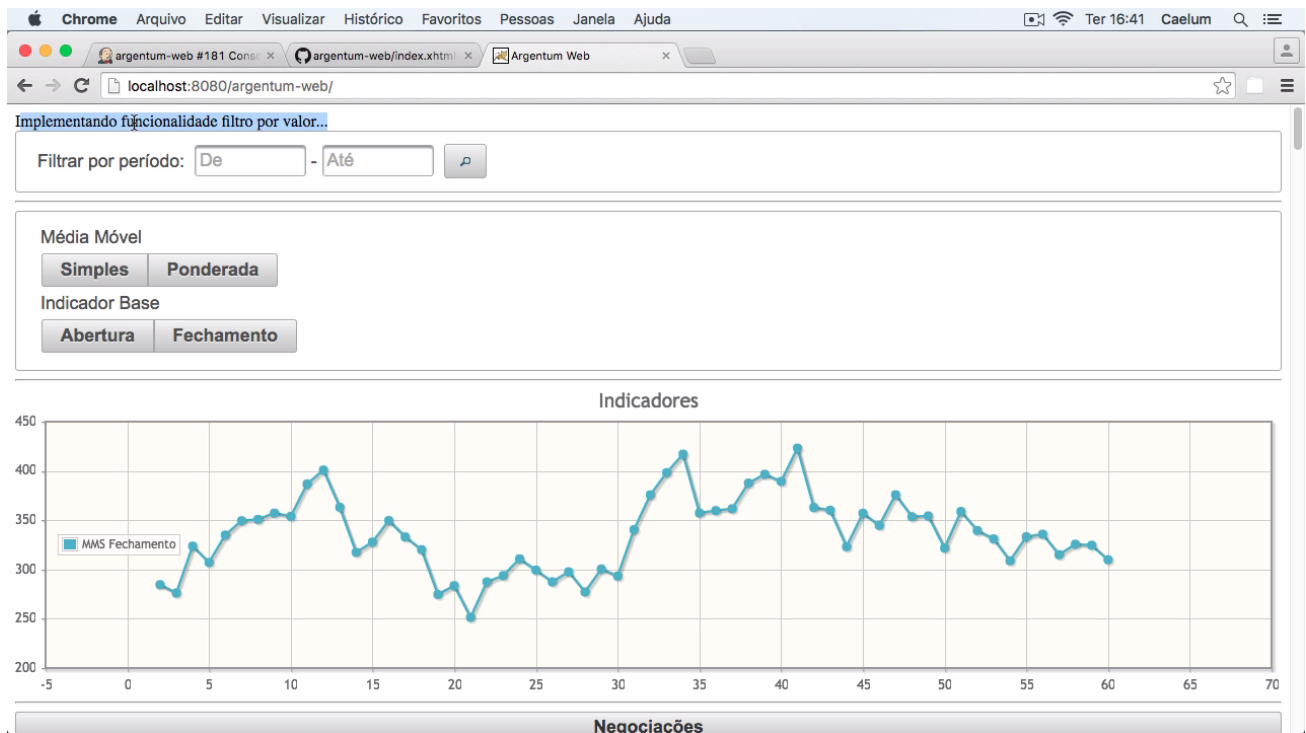
E faz o deploy para o Tomcat:

```

> git -c core.askpass=true push https://github.com/alura-cursos/argentum-web.git HEAD:master
> git config --local --remove-section credential # timeout=10
Injecting SonarQube environment variables using the configuration: SonarQube
Deploying [/Users/Caelum/.jenkins/workspace/argentum-web/target/argentum-web.war to container Tomcat 7.x Remote
Redeploying [/Users/Caelum/.jenkins/workspace/argentum-web/target/argentum-web.war]
Undeploying [/Users/Caelum/.jenkins/workspace/argentum-web/target/argentum-web.war]
Deploying [/Users/Caelum/.jenkins/workspace/argentum-web/target/argentum-web.war]

```

Pronto! A primeira coisa que podemos testar é se o nosso deploy carregou as alterações que fizemos no index.xhtml. Se você não lembra, a gente tinha adicionar um outputText via branch filtro-por-valor. Vamos testar se tudo foi feito corretamente acessando <http://localhost:8080/argentum-web> (<http://localhost:8080/argentum-web>) e verificando aquela mensagem:



Podemos verificar inclusive que a nossa alteração via branch filtro-por-valor foi efetivamente integrada ao MASTER:

<https://github.com/alura-cursos/argentum-web/blob/master/src/main/webapp/index.xhtml>

```
83 lines (69 sloc) | 3.09 KB
Raw Blame History
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3
4 <html xmlns="http://www.w3.org/1999/xhtml"
5       xmlns:h="http://java.sun.com/jsf/html"
6       xmlns:f="http://java.sun.com/jsf/core"
7       xmlns:p="http://primefaces.org/ui">
8
9 <h:head>
10 <title>Argentum Web</title>
11 </h:head>
12
13 <h:body>
14
15 <h:form prependId="false">
16 <h:outputText value="Implementando funcionalidade filtro por valor..."></h:outputText>
17
```

Exatamente o que queríamos! Um commit feito em uma branch diferente da MASTER foi integrado com toda a segurança de compilação e testes do nosso build.