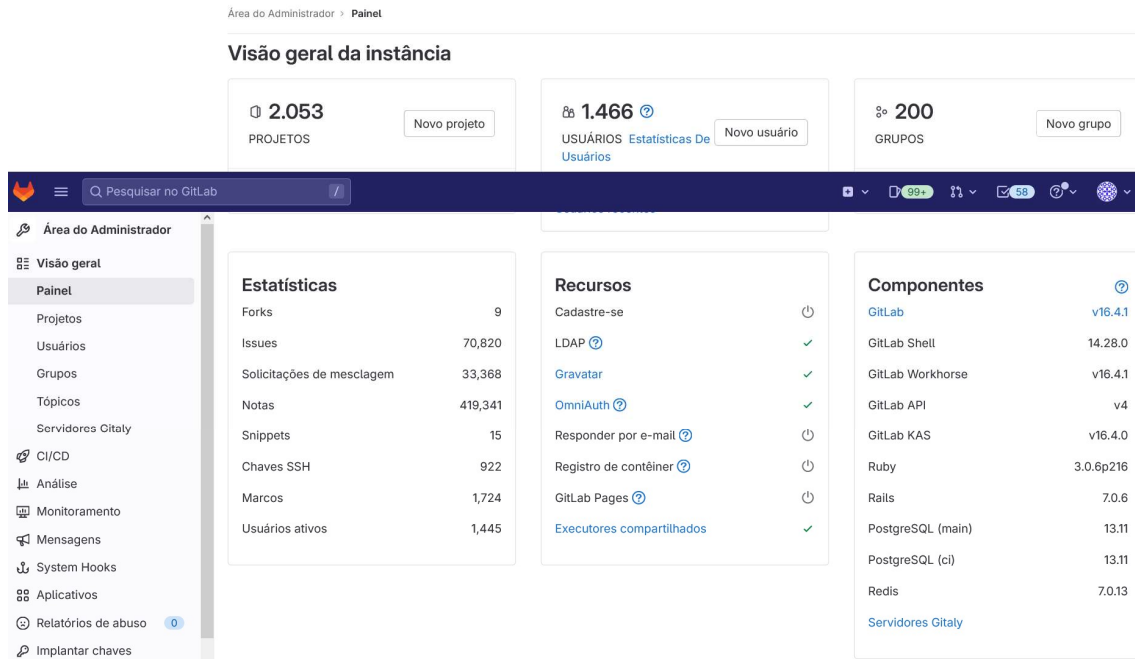


Configurando pipelines de banco de dados

Primeiro passo deve ser criar o projeto no Git:



Após acionar novo projeto deve se optar pela opção criar projeto em branco, depois preencher o projeto com o nome do repositório “código do sistema – bd” e em url do projeto o grupo de default é prodemge. Seguir como a imagem exemplo:



Criar projeto em branco

Crie um projeto em branco para armazenar seus arquivos, planejar seu trabalho e colaborar no código, entre outras coisas.

Nome do projeto

URL do projeto

Slug do projeto

https://git.prodeng.gov.br/

Escolha um grupo ou espaço de nome

/

my-awesome-project

Quer organizar vários projetos dependentes no mesmo espaço de nome? [Crie um grupo](#).

Nível de visibilidade ?

☒ Privado
O acesso ao projeto deve ser concedido explicitamente a cada usuário. Se este projeto fizer parte de um grupo, o acesso será concedido aos membros do grupo.

☐ Interno
O projeto pode ser acessado por qualquer usuário entrado, exceto usuários externos.

☐ Público
O projeto pode ser acessado sem nenhuma autenticação.

Configuração do projeto

☒ Inicializar repositório com um README
Permite que você clone imediatamente o repositório deste projeto. Pule esta etapa se você planeja fazer o push de um repositório existente.

☐ Ativar teste de segurança de aplicativo estático (SAST)
Analise seu código-fonte para vulnerabilidades de segurança conhecidas. [Saiba mais](#).

Criar projeto

Cancelar

Na próxima página criar o diretório Migrations e adicionar o arquivo dbchangelog-master.xml

T

treinamento-bd

ID do Projeto: 3069

☆ Marcar como favorito

0

Forks

0

Update dbchangelog-master.xml

JOSE SERGIO PEREIRA NETO Autor 9 minutos atrás

f5d869c5

main

treinamento-bd

+

Histórico

Localizar arquivo

Alterar

Clonar

README

Auto DevOps ativo

Adicionar LICENSE

Adicionar CHANGELOG

Adicionar CONTRIBUTING

Adicionar cluster Kubernetes

Adicionar wiki

Configurar integrações

Nome	Último commit	Última atualização
Migrations	Adicionar novo diretório	12 minutos atrás
README.md	Initial commit	13 minutos atrás
dbchangelog-master.xml	Update dbchangelog-master.xml	9 minutos atrás

Próximo passo será adicionar os usuários ao projeto criado:

Verificação do repositório

This repository has never been checked. ?

Ativar a verificação do repositório

Acesso concedido o meses atrás

jenkinsdeploy	@jenkinsdeploy · prodemge	Maintainer
Acesso concedido 5 anos atrás		
Jenkins Redmine	@jenkinsredmine · prodemge	Maintainer
Acesso concedido 4 anos atrás		
RODRIGO HERINGER MENDONCA	@p059866 · prodemge	Maintainer
Acesso concedido 3 anos atrás		
userdeploy	@userdeploy · prodemge	Developer
Acesso concedido 3 anos atrás		

< Prev 1 2 Próximo >

treinamento-bd membros do projeto 0 Gerenciar acesso

Acionar o botão

prodemge > treinamento-bd > Membros

Membros do projeto

Você pode convidar um novo membro para **treinamento-bd** ou convidar outro grupo.

Importar de um projeto

Convidar um grupo

Convidar membros

Membros 38

🕒 Filtar membros

Conta ▾

⌵

Conta	Origem	Cargo max	Expiração	Atividade
-------	--------	-----------	-----------	-----------

Acionar o botão

Convidar membros

Você está convidando membros para o projeto **treinamento-bd**.

Nome de usuário ou endereço de e-mail

Selecione membros ou digite o endereço de e-mail

Selecione um cargo

Guest ▾

[Leia mais](#) sobre as permissões de cargos

Data de expiração do acesso (opcional)

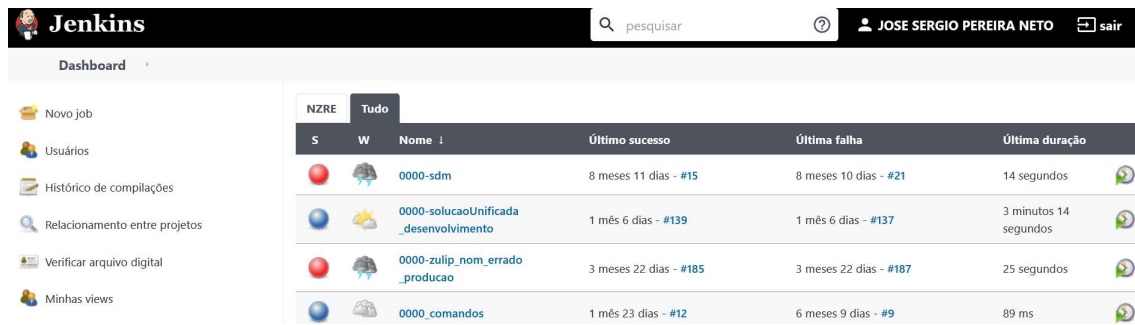
YYYY-MM-DD 📅

Cancelar

Convidar

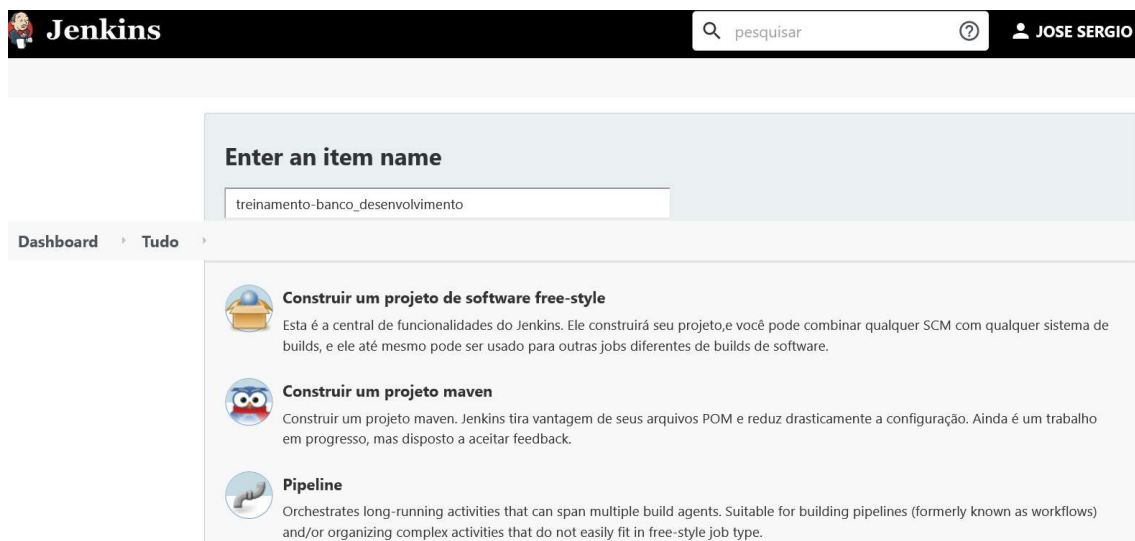
Preencher os dados nos membros na tela acima informando se são mantenedores ou desenvolvedores. Assim finalizando a etapa de configuração do projeto no Git.

Continuando a etapa da criação da pipeline iremos trabalhar no Jenkins. Na tela inicial do Jenkins após autenticar iremos na opção: “Novo Job”



NZRE		Tudo				
S	W	Nome ↓	Último sucesso	Última falha	Última duração	
		0000-sdm	8 meses 11 dias - #15	8 meses 10 dias - #21	14 segundos	
		0000-solucaoUnificada_desenvolvimento	1 mês 6 dias - #139	1 mês 6 dias - #137	3 minutos 14 segundos	
		0000-zulip_nom_errado_producao	3 meses 22 dias - #185	3 meses 22 dias - #187	25 segundos	
		0000_comandos	1 mês 23 dias - #12	6 meses 9 dias - #9	89 ms	

Na próxima tela usaremos a opção de Pipeline e iremos informar o nome do job que dever seguir o padrão “código do projeto – banco_ambiente”. Conforme imagem:



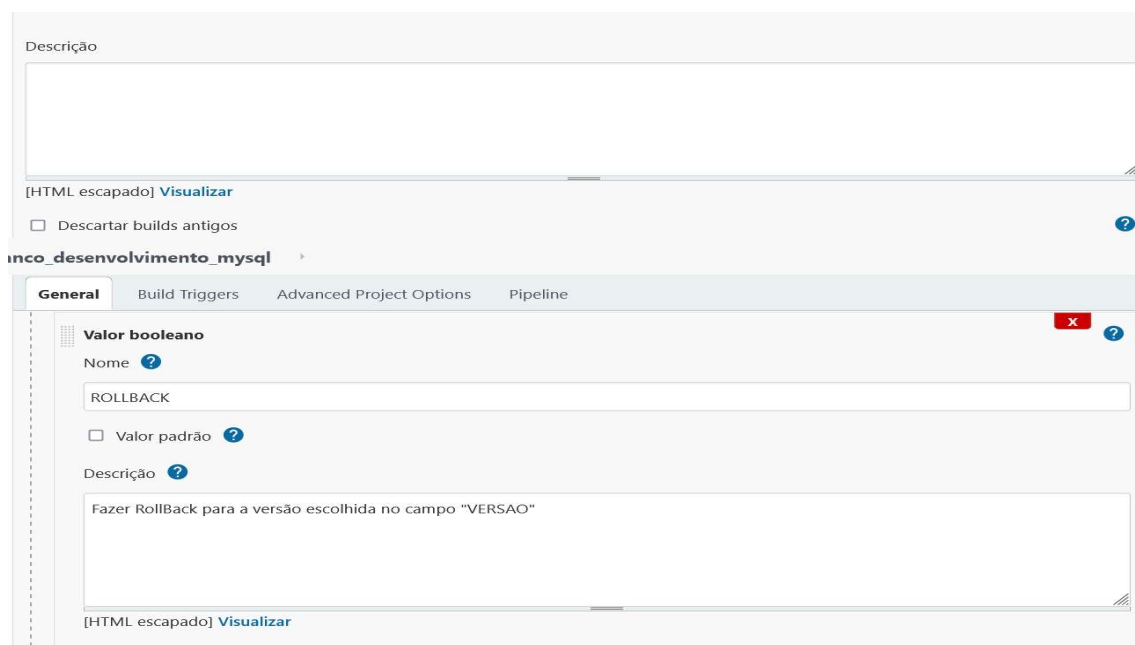
Enter an item name

treinamento-banco_desenvolvimento

Dashboard **Tudo**

- Construir um projeto de software free-style**
Esta é a central de funcionalidades do Jenkins. Ele construirá seu projeto, e você pode combinar qualquer SCM com qualquer sistema de builds, e ele até mesmo pode ser usado para outras jobs diferentes de builds de software.
- Construir um projeto maven**
Construir um projeto maven. Jenkins tira vantagem de seus arquivos POM e reduz drasticamente a configuração. Ainda é um trabalho em progresso, mas disposto a aceitar feedback.
- Pipeline**
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Na próxima tela iremos preencher as informações da pipeline iremos montar o job, para isso iremos marcar que “Este build é parametrizado”. O primeiro que iremos adicionar é o de rollback como sendo um valor booleano, conforme abaixo:



Descrição

[HTML escapado] Visualizar

☐ Descartar builds antigos

inco_desenvolvimento_mysql

General Build Triggers Advanced Project Options Pipeline

Valor booleano

Nome ?

ROLLBACK

☒ Valor padrão ?

Descrição ?

Fazer RollBack para a versão escolhida no campo "VERSAO"

[HTML escapado] Visualizar

O próximo parâmetro é do tipo string e o mesmo será denominado “Arquivo” e o valor padrão do mesmo é o dbchangelog-master.xml, veja na imagem:



The image shows a configuration window titled "Parâmetro string". It contains three input fields: "Nome" with the value "ARQUIVO", "Valor padrão" with the value "dbchangelog-master.xml", and "Descrição" with the value "Arquivo de ChangeLog". Below these fields is a link "[HTML escapado] Visualizar" and a checkbox "Trim the string" which is currently unchecked.

O próximo parâmetro é um hider parameter que irá conter a informação do repositório do projeto:



The image shows a configuration window titled "Hidden Parameter". It contains three input fields: "Name" with the value "GIT_PROJECT", "Default Value" with the value "https://git.prodemge.gov.br/prodemge/treinamento-bd.git", and "Description" which is empty.

Agora iremos configurar o parâmetro versão, marcando a opção scrip groovy e marcando a opção, use Groovy sandbox:



The image shows a configuration window for a parameter named "VERSAO". Under the "Script" section, the "Groovy Script" option is selected. The script content is "evaluate(new File("/produtos/scripts_groovy/gettags.groovy"))". The "Use Groovy Sandbox" checkbox is checked. There is also an "Additional classpath" section with an "Add entry" button.

Choice Type

Single Select ▾

Referenced parameters ?

GIT_PROJECT

☒ Enable filters ?

Filter starts at ?

1

Continuando no próximo hidden parameter iremos informar o parâmetro liquibase_url:

Hidden Parameter X

Name ?

liquibase_url

Default Value ?

jdbc:mysql://mysqlidescomp:3306/gpen

Description ?

Continuando com a parametrização iremos agora informar o campo de credencial :

Hidden Parameter X

Name ?

liquibase_credentialsId

Default Value ?

liquibase-jenkins-mysql

Description ?

O próximo passo será informar os valores do classpath e driver na pipeline>

Hidden Parameter

Name ?

liquibase_driverClassname

Default Value ?

com.mysql.jdbc.Driver

Description ?

Hidden Parameter

Name ?

liquibase_arquivo

Default Value ?

dbchangelog-master.xml

Description ?

Hidden Parameter

Name ?

liquibase_classpath

Default Value ?

mysql-connector-java-5.1.6.jar

Description ?

Adicionar parâmetro ▾

☐ GitHub project

GitLab Connection

Vamos agora preencher informações fundamentais para o job, classpath, class name, liquibase arquivo os drivers.

Os próximos parâmetros são informações essenciais para o funcionamento do job e devem ser preenchidas de imediato. Que se tratam do apontamento do Git e do Jenkins file arquivo de configuração utilizado em todos os jobs:

The screenshot shows the Jenkins configuration interface for a job named "o_desenvolvimento_mysql". The "Pipeline" tab is selected. The configuration includes:

- Git** dropdown menu.
- Repositories** section:
 - Repository URL:** `http://git.prodemge.gov.br/prodemge/jenkinsfile.git`
 - Credentials:** `jenkinsdeploy` (with an "Add" button)
 - Buttons:** "Avançado...", "Add Repository"
- Branches to build** section (with a red "X" icon).
- o_desenvolvimento_mysql** job name.
- General**, **Build Triggers**, **Advanced Project Options**, and **Pipeline** tabs.
- Add Branch** button.
- Navegar no repositório** dropdown menu (set to "(Auto)").
- Additional Behaviours** section with an "Adicionar" button.
- Script Path** text field containing `liquibase/Mysql/Jenkinsfile`.
- Lightweight checkout** checkbox (checked).

Conteúdo do arquivo Jenkins file:

```
pipeline {
    agent any
    parameters {
        booleanParam(name: 'ROLLBACK', defaultValue: false,
description: 'Fazer RollBack para a versão escolhida no campo
"VERSAO"')
        //booleanParam(name: 'CRIAR_SINONIMOS', defaultValue: false,
description: 'Criar sinônimos no ORACLE')
        string(name: 'ARQUIVO', defaultValue: 'dbchangelog-
master.xml', description: 'Arquivo de ChangeLog')
        // string(name: 'SCHEMA', defaultValue: 'client', description:
'Esquema que vai ser executado o script')
    }
    options {
```

```

        ansiColor('xterm')
    }

    tools {
        maven 'Maven 3.5.4';
        jdk 'JAVA-11.0.5';
    }

    stages {

        stage('Checkout'){
            steps{
                script{

                    sh "curl -L --write-out '%{http_code}'
http://desenvolvedor:senha2020@artifactory.prodemge.gov.br/artifactory
/repo-prodigio/mysql/mysql-connector-java/5.1.48/mysql-connector-java-
5.1.48.jar -o mysql-connector-java-5.1.48.jar"

                    // Limpando dados desnecessários da versao
                    VERSAO_SIMPLIFICADA =
params.VERSAO.replace('tags/', '')

                    checkout([$class: 'GitSCM', branches: [[name:
VERSAO_SIMPLIFICADA]],
                        userRemoteConfigs: [[url:
"${params.GIT_PROJECT}", credentialsId: 'jenkinsdeploy']]])
                }
            }

        }

        stage('Liquibase') {

            steps{
                script {

                    def sistema = env.JOB_BASE_NAME.split('_')[0]
                    def ambiente = env.JOB_BASE_NAME.split('_')[1]

                    def url_liquibase = params.liquibase_url
                    echo "Liquibase URL"

                    if ( ambiente == "producao"){

                        url_liquibase =
params.liquibase_url_producao

                    }

                    echo url_liquibase
                }
            }
        }
    }

```


liquibase-codigo_esquema+adm

Para Oracle em Exadata:

liquibase- codigo_esquema-EXADATA

liquibase-codigo_esquema+adm-EXADATA

Observação no caso do Oracle, temos que criar as credenciais por esquema de banco, então se faz necessário o apoio e atuação da GBD, na configuração e criação das credenciais para os SGBDs.

Para todas as situações relacionadas a criação e configuração, abrimos solicitações para GBD a qual é a única que possui acesso ao gerenciamento dessas credenciais, isso é feito por meio de OCR como está: OCR733742

Após toda configuração realizada nos fazemos os testes e então solicitamos a liberação dos jobs para os usuários por meio de OCR como está: OCR736897