

Programa de formação em gestão de
mudanças de banco de dados com
Liquibase.

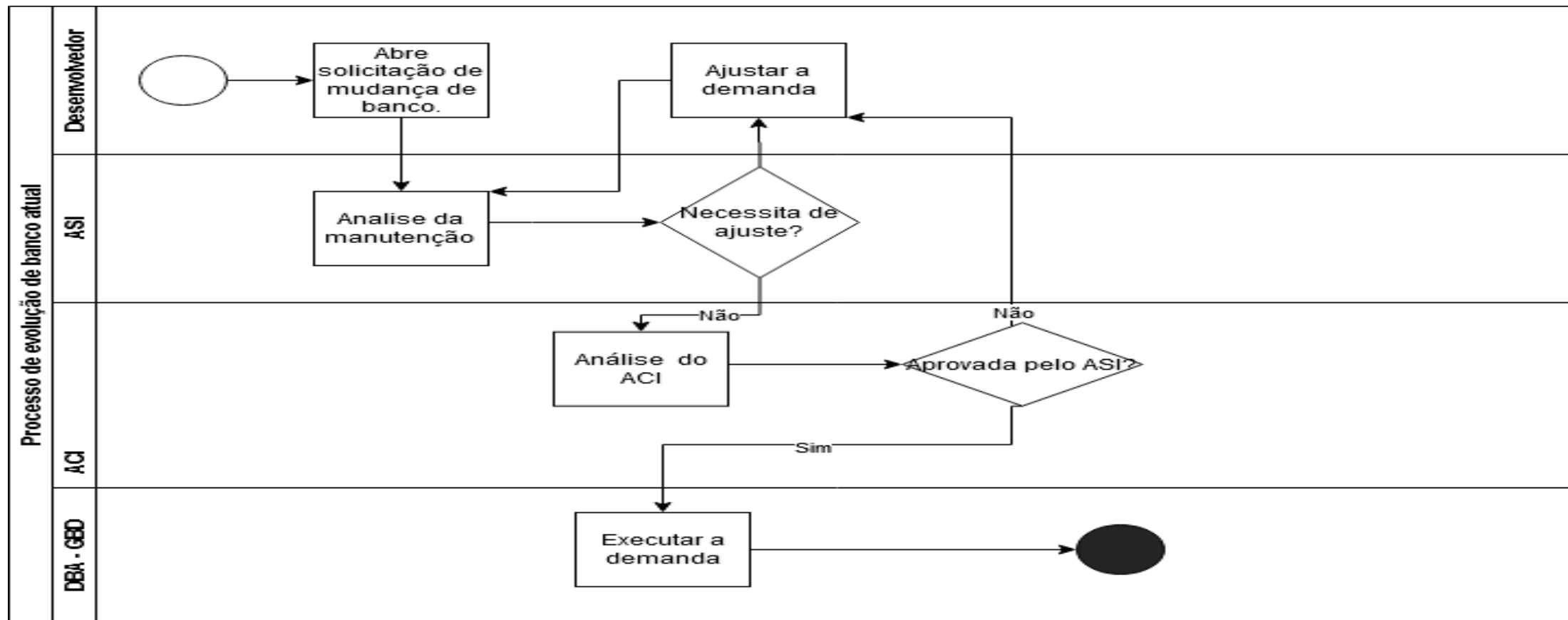


- Me chamo José Sérgio Pereira Neto, sou Analista de Sistemas desde 2008.
- Como colaborador da Prodemge atuei nas gerências: sistemas diversos, meio ambiente, processo e qualidade de software e atualmente na arquitetura corporativa.
- Curiosidades sobre mim: fui um dos três alunos de Minas Gerais a representar o estado, no segundo fórum nacional de Escola para Escola do ensino médio. No ano de 2014 participei do treinamento ministrado por Rahul Bhargava, pesquisador do MIT Media Lab sobre Storytelling.
- Hobbies: ler, escrever, dança de salão e jogos de mesa.

- **Módulo 1 - Fundamentos do Liquibase e configuração Inicial.**
- **Modulo 2 - Implementando Alterações de Banco de Dados, estudo de Caso.**
- **Modulo 3 - Integração com Ferramentas de DevOps e Versionamento de Alterações de Banco de Dados**
- **Modulo 4 - Colaboração, Controle de Acesso e Melhores Práticas.**
- **Provas teóricas a cada módulo**

- **Gestão de mudanças em banco de dados**
 - Controle das mudanças.
 - Auditoria das mudanças.
 - Capacidade de retorno.
 - Facilitar a execução de mudanças.

- Processo atual

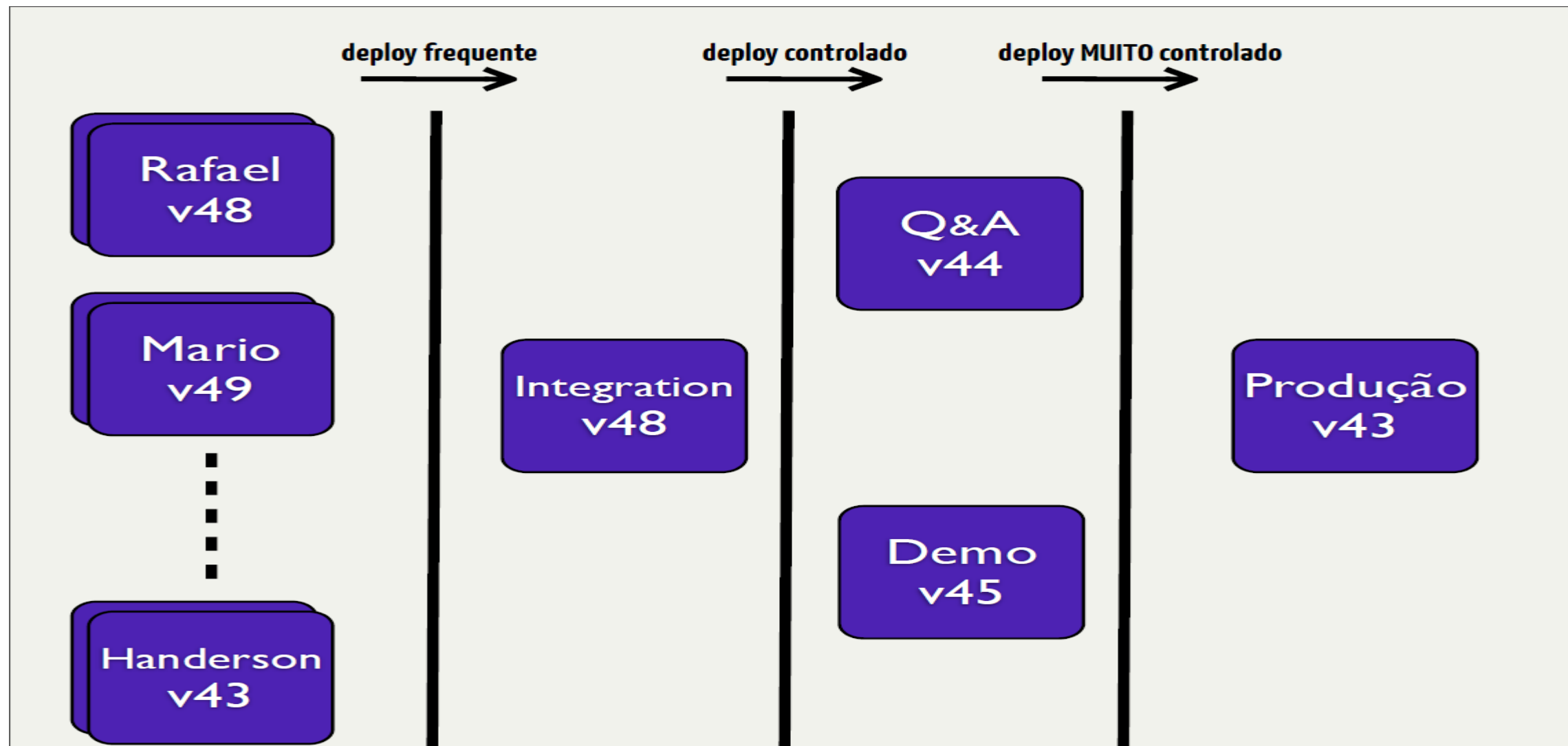


- **Migrations**

- Consiste em uma estratégia de aplicar mudanças em banco de dados, criando scripts versionados que contem o código a ser aplicado no banco a cada implantação de mudanças.
- Seguindo a ideia de pequenas migrações, geram grandes bases.

- **Exemplo**

Versão 1	Versão 2	Versão 3	Versão 4
• baseline.xml • V1.xml • V2.xml • V3.xml	• V4.xml • V5.xml	• V6.xml	• V7.xml • V8.xml • V9.xml



“

WARNING: We've seen Hibernate users trying to use SchemaUpdate to update the schema of a **production database** automatically. This can quickly **end in disaster** and won't be allowed by your DBA.

-- Java Persistence with Hibernate

O que é?

- Ferramenta de controle de versão de banco de dados open source que implementa o conceito **Migrations**.
- Ele permite a organização **rastrear, versionar e fazer deploy** de alterações em esquemas de banco de dados.

O que é?

- Ferramenta de controle de versão de banco de dados open source que implementa o conceito **Migrations**.
- Ele permite a organização **rastrear, versionar e fazer deploy** de alterações em esquemas de banco de dados.

Características

- Ferramenta Open source com versão comunitária e versão pro paga. Trabalhamos com a versão community.
- Possui mais de 1 milhão de downloads da ferramenta.
- Atende mais de 60 SGBDs, possuindo interoperabilidade.
- Demonstrativo das versões: <https://www.liquibase.com/pricing>

- DB2
- H2
- HyperSQL
- Firebird
- MariaDB
- MS SqlServer
- Oracle
- PostgreSQL
- Snowflake
- SQLite

- **E possível fazer com ele**
 - Habilidade de merge de mudanças
 - Trabalhar com tags
 - Múltiplos databases
 - Trabalhar com bases já existentes
 - Gerar documentação de mudanças
 - Updates automáticos ou geração de SQL
 - Independente de linguagem de programação.
 - Pipeline no Jenkins

- **O que ele não faz**

- Não gera modelos visuais
- Rollback automatizados sem usar xml.
- Não gera código sem uma base já existente para extrair
- Não gera e nem compara functions, triggers, sinônimos, procedures e etc.
- Não permite o uso de tags de functions, sinônimos e triggers na versão community
- Não compara SGBD's distintos.
- Requer conhecimentos dos SGBD's.



Changesets: Define database changes



Changelog: Add database changes to a file



Update: Run the command to deploy database changes

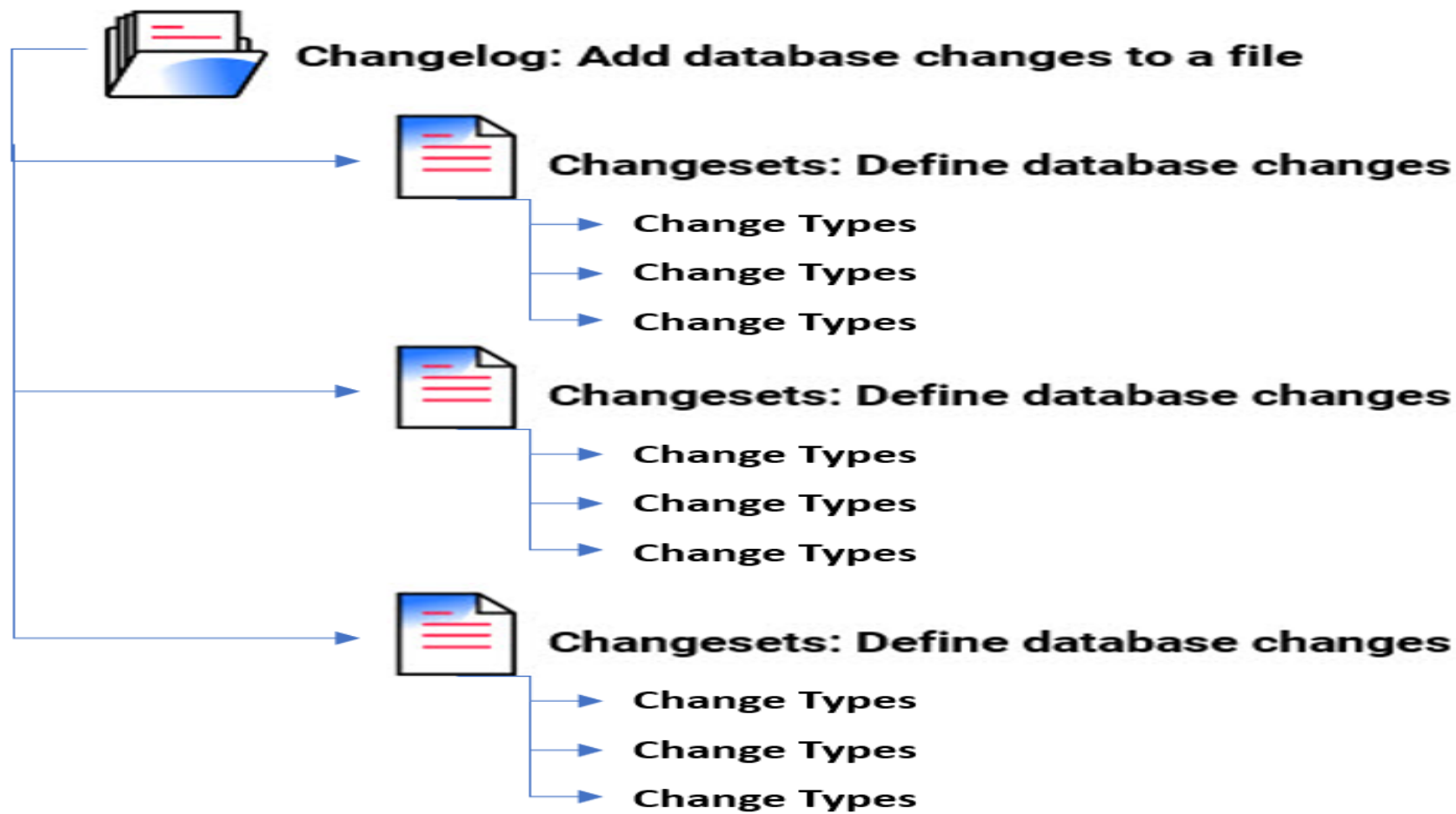


DATABASECHANGELOG and DATABASECHANGELOGLOCK:
Track and version database changes



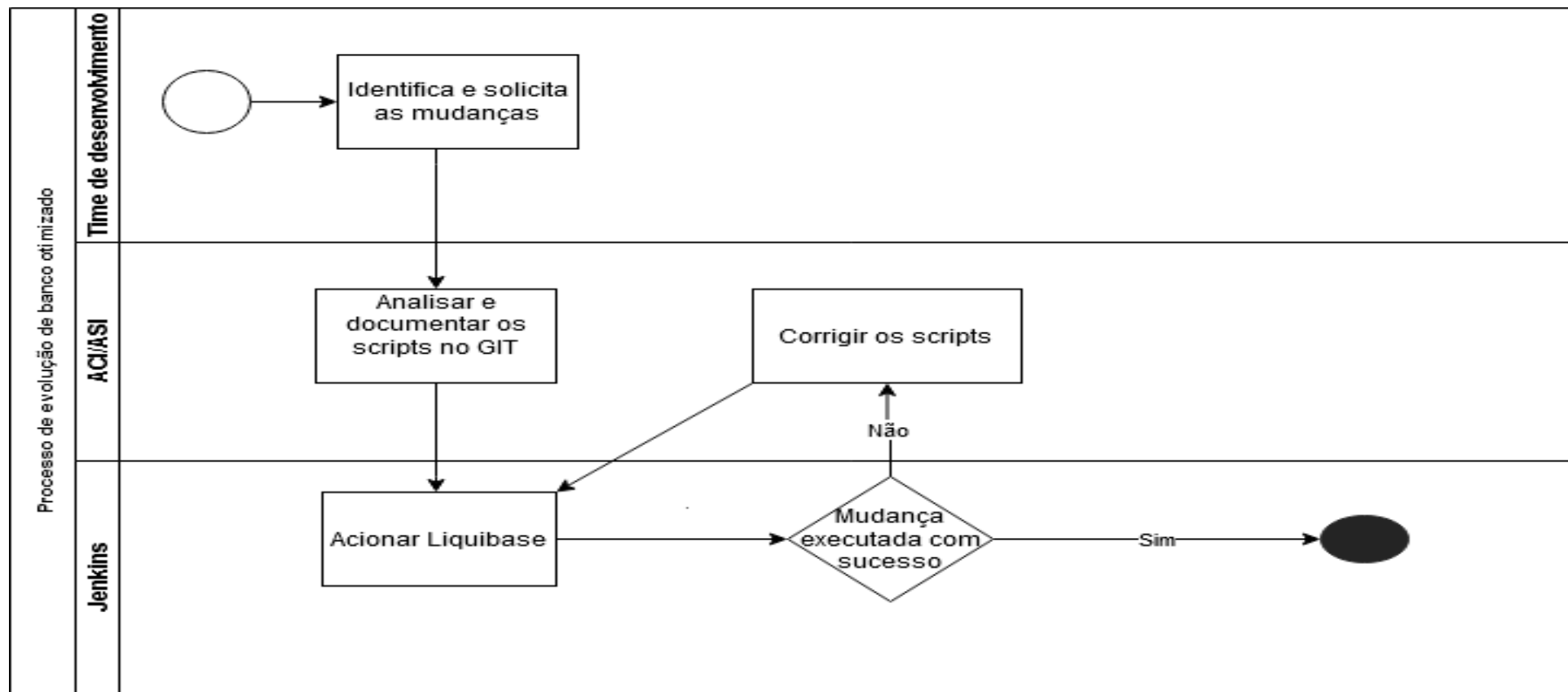
Manage your database changes with other Liquibase commands

- Possui dois conceitos fundamentais:
 - Schema Changes que consiste em changelogs, changesets e change types
 - Tracking tables que registram as mudanças



- **DATABASECHANGELOGS:** Nessa tabela é armazenado as alterações por changelog inserindo informações como, id, autor e nome do arquivo. Essa tabela cria a rastreabilidade do que foi alterado.
- **DATABASECHANGELOGSLOCK:** Essa tabela garante que apenas uma instância do Liquibase esteja em execução por vez. Ele bloqueia outros para evitar que vários comandos do Liquibase sejam executados ao mesmo tempo, o que poderia causar conflitos no banco de dados. Isso ajuda a evitar que outros desenvolvedores, dbas ou equipes substituam suas alterações acidentalmente.

Objetivo



Obs.: Nesse modelo a GBD atua adicionando dados dos ambientes no Jenkins e pode focar mais no trabalho de consultoria técnica as equipes de desenvolvimentos e tarefas do DBA.

- **Instalação do Liquibase**

- <https://docs.liquibase.com/start/install/liquibase-windows.html>
- Após download
 - Clicar no arquivo e prosseguir
 - Configurar variáveis de ambiente Java Home e Path
 - Executar o comando liquibase -version no prompt
 - OCR exemplo para instalação: **OCR738819**

- **Visão do liquibase.properties**
 - changeLogFile – Parâmetro destinado ao arquivo orquestrador de mudanças.
 - driver – O nome do class driver utilizado pelo database.
 - url – Parâmetro de conexão com o banco de desenvolvimento.
 - username – usuário do banco.
 - password – senha do banco.
 - referenceUrl – Host de conexão com o banco de homologação.
 - referenceUsername – usuário do banco.
 - referencePassword – senha do usuário.
 - classpath – jdbc driver utilizado para conexão.
 - liquibaseSchemaName – nome do esquema da aplicação não é obrigatório

- Workflow de 5 passos
 - Criar as alterações (changesets) nos arquivos (changelog).
 - Validar o SQL
 - Salvar na ferramenta de controle
 - Executar update no banco de dados
 - Verificar execução (Validar Rollback)
- Observar as melhores práticas
 - Organização dos changelogs
 - Adicionar comentários
 - Refinar os changelogs

- **Projeto IBGE**
- Liquibase changelog e changesets
 - Criação de tabelas
 - Alteração em tabelas
 - Criação de índices
 - Aplicação de rollback
- O projeto trata-se de um software para captar dados de estados e municípios, buscando detalhar a situação das diversas regiões do país. Teremos informações demográficas e sociais.

- **Comandos principais**

- liquibase status
- liquibase updateSQL
- liquibase update
- liquibase history
- liquibase rollback <tag>
- liquibase rollbackSQL <tag>
- liquibase diff
- liquibase update --url=host --username=usuário --password=senha
- liquibase updateSQL --url=host --username=usuário --password=senha

- **Comandos disponíveis**
 - liquibae rollbackCountSQL <n>
 - liquibase rollbackCount <n>
 - liquibase rollbackToDate <yyyy-mm-dd'T'hh:mm:ss.ms>
 - liquibase updateToTag <tag>
 - liquibase diffChangeLog
 - liquibase changelogSyncSQL
 - liquibase changelogSync

- Passos para automatizar
 - Criar repositório no git para armazenar dchangelog-master.xml e migrations
 - Liberar acesso dos usuários ao git
 - Criar e configurar job no jenkins
 - Solicitar criação das credências pela GBD
 - Testar a esteira
 - Solicitar a liberação de acesso ao job para os usuários

- **Automatização Liquibase e Jenkins**



Jenkins

Jenkins
 01_liquibase_pipeline

[Back to Dashboard](#)
[Status](#)
[Changes](#)
[Construir com parâmetros](#)
[Configurar](#)
[Full Stage View](#)
[Rename](#)
[Pipeline Syntax](#)

Pipeline 01_liquibase_pipeline


[Recent Changes](#)

Stage View

Histórico de builds

Tendência

find x

#28	10/09/2020 11:13
#27	10/09/2020 11:12
#26	10/09/2020 11:11

Average stage times:

(Average full run time: ~16s)

#28

Sep 10 11:13

No Changes

Declarative: Checkout SCM	Declarative: Tool Install	Checkout	Liquibase
748ms	179ms	1s	4d 9h
453ms	222ms	1s	6s

- **Automatização Liquibase e Jenkins**

 **Jenkins**

Jenkins > 01_liquibase_pipeline >

 Back to Dashboard

 Status

 Changes

 Construir com parâmetros

 Configurar

 Full Stage View

 Rename

 Pipeline Syntax

Pipeline 01_liquibase_pipeline

Esta builds requer parâmetros:

☐ ROLLBACK

Fazer RollBack para a versão escolhida no campo "VERSAO"

ARQUIVO

Arquivo de ChangeLog

VERSAO

Versão da TAG ou Branch do GIT

- **Automatização Liquibase e Jenkins**



Iniciado pelo(a) usuário(a) [JOSE SERGIO PEREIRA NETO](#)



Revision: 07cfc1a735b998a0238e33d2f577620c32afdb5c

- refs/remotes/origin/master



Revision: 8df6c6868861e8c69aa5a41564c2cae4b3749ceb

- 1.0.6



Rolled back 8 changeSets:

- 1007
- 1006
- 1005
- 1004
- 1003
- 1002
- 1001
- tagDatabase_1.6

- Como funciona a segurança? O acesso as informações dos scripts no git e a execução da pipeline no Jenkins são de acesso restrito por meio de perfis de autenticação.
- A equipe terá acesso ao sysadmin? Não, as credenciais cadastradas no Jenkins são criptografadas. Somente os DBA's tem acesso as credenciais.
- Posso alterar um changeset já aplicado e executar de novo? O Liquibase acusará erro, ele valida o conteúdo antes de executar.
- Posso rodar scripts de alteração de dados no Liquibase? Sim, mas requer muita atenção para não comprometer os dados.
- Consigo realizar drop de qualquer objeto no banco via script com o Liquibase? Sim, esse tipo de comando deve ser executado sempre com a ciência do DBA.
- E possível alterar esquemas de terceiros com o Liquibase? A pipeline impede que haja conexão em esquemas distintos.

Dúvidas ???

Equipe de atuação:

José Sérgio Pereira Neto (GAC)

Bruno Gonçalves Joaquim (GAC)

Carlos Henrique de Moro (GAC)

Gestão e apoio:

Alexandre Sidnei (Supervisor)

Rodrigo Santos (Gerente)

Marcelo Scheffer (GAC)