



PGCONF.BRASIL 2024  
BELO HORIZONTE, MG  
7 E 8 DE NOVEMBRO



# Implementando um Cluster Master-Master no PostgreSQL com Bucardo

## PGCONF Brasil 2024

Marcio Junior Vieira  
CEO & Data Scientist, Ambiente Livre

## Mini-CV

- 22 anos de experiência em TI, vivência em desenvolvimento e análise de sistemas de gestão empresarial e ciência de dados.
- CEO da Ambiente Livre atuando como Cientista de Dados, Engenheiro de Dados e Arquiteto de Software.
- Professor dos MBAs em Big Data & Data Science, Inteligência Artificial e Business Intelligence e Analytics da Universidade Positivo.
- Professor do MBA Artificial Intelligence e Machine Learning da FIAP.
- Pesquisador do Laboratório de Tecnologias para Tomada de Decisão da Universidade de Brasília (Unb/Latitude).
- Trabalhando com Free Software e Open Source desde 2000 com serviços de consultoria e treinamento.
- Graduado em Tecnologia em Informática(2004) e pós-graduado em Software Livre(2005) ambos pela UFPR.
- Palestrante FLOSS em: FISL, TDC, Latinoware, Campus Party, Pentaho Day, Ticonova, PgDay e FTSL.
- Organizador Geral: Pentaho Day 2017, 2015, 2019 e apoio nas ed. 2013 e 2014.
- Data Scientist, instrutor e consultor de Big Data e Data Science com tecnologias abertas.
- Ajudou a capacitar equipes de Big Data na IBM, Accenture, Tivit, Serpro, Natura, MP, Netshoes, Embraer entre outras.
- Especialista em implantação e customização de Big Data com Hadoop, Spark, Pentaho, Cassandra e MongoDB.
- Contribuidor de projetos internacionais, tais como Pentaho, LimeSurvey, SuiteCRM e Camunda.
- Especialista em implantação e customização de ECM com Alfresco e BPM com Activiti, Flowable e Camunda.
- Certificado (Certified Pentaho Solutions) pela Hitachi Vantara (Pentaho).
- Membro da The Order Of de Bee (comunidade Alfresco para desenvolver o ecossistema Alfresco independente)
- Trabalha profissionalmente com PostgreSQL desde 2006.



# Sobre a Ambiente Livre

## Open Software for Business

- Fundada em 2004 com foco em consultoria com FLOSS.
- Experts em 34 soluções para geração de negócios com Software Livre/Código Aberto.
- Atualmente estamos sediados no Hub de Inovação Mindhub em Curitiba (FAE).

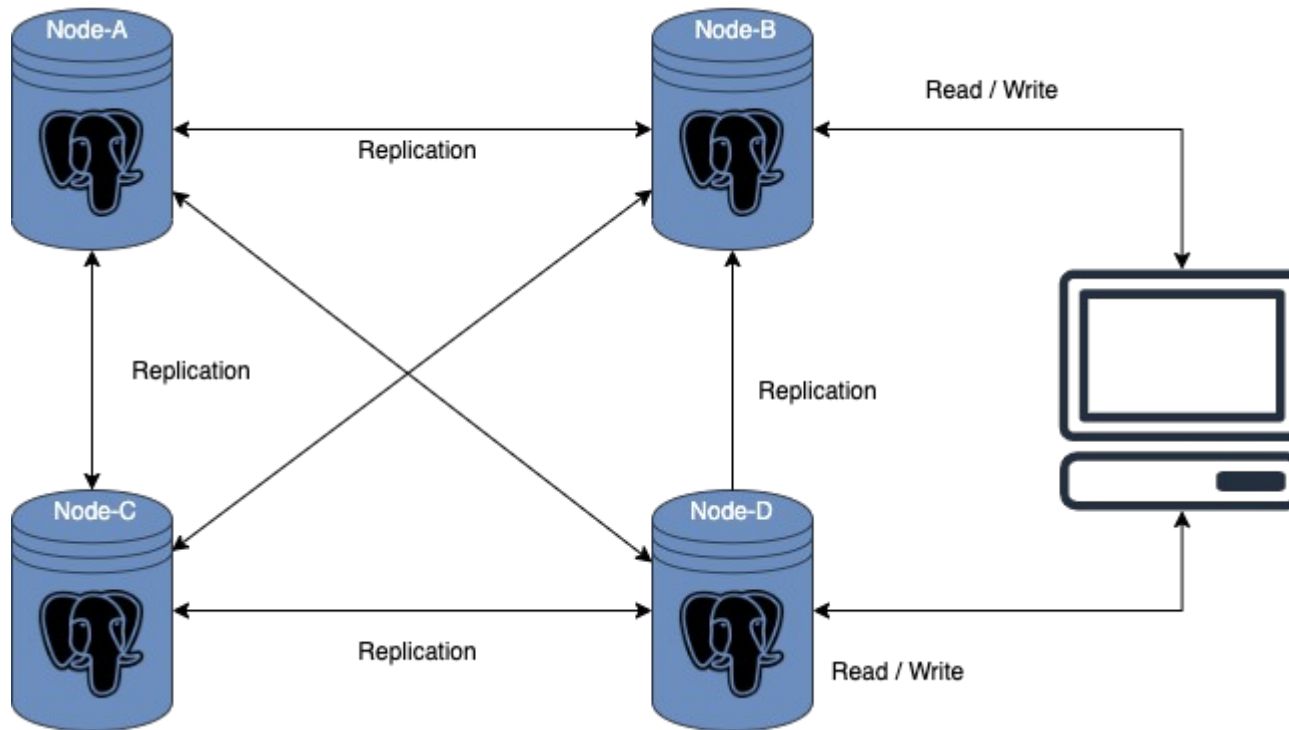


# Nosso Ecossistema de Serviços

Big Data e Data Science	CRM e CMS	ECM e BPM	Business Intelligence
Análise de Dados da IoT Análise Preditiva Processamento Distribuído Banco de Dados Colunares	Marketing e Vendas Fidelização SAC e Pós-vendas Portais de Conteúdo	Gestão de Documentos Gerenciamento de Mídias Processo de Negócio BPMN e BPMS	Painéis de Indicadores Cubos de Análise Relatórios Gerenciais Tomada de Decisão
Big Data & Data Lake Big Data Analytics Machine Learning	Customer Relationship Management Content Management System Pesquisa de Mercado & SLA	Enterprise Content Management Records Management Business Process Management	Business Intelligence & Analytics Dashboards e OLAP Data Integration & Data Mining
Consultoria   Treinamento   Projeto	Consultoria   Treinamento   Projeto	Consultoria   Treinamento   Projeto	Consultoria   Treinamento   Projetos



# Replicação Multi-Master



## Prós

- No caso de um mestre falhar, o outro mestre está lá para servir a atualização e a inserção.
- Os nós mestres estão localizados em vários locais diferentes, portanto, as chances de falha de todos os mestres são mínimas.
- As atualizações de dados são possíveis em vários servidores.
- Não requer rotear o tráfego para apenas um único mestre.

## Contra

- A principal desvantagem da replicação multi-master é sua complexidade.
- A resolução de conflitos é muito difícil porque gravações simultâneas em vários nós são possíveis.
- Às vezes, a intervenção manual é necessária em caso de conflito.
- Possibilidade de inconsistência de dados.



## Contra

- O PostgreSQL nativo não implementa Master-Master
- Tabela de Opções:

Serial Number	Replication	Company	Open-Source	Language	Type	Failover	Synchronous	Version	Active
1	BDR3	2ndQuadrant	N	C	Extension	Y	ASync	PostgreSQL 12	Y
2	BDR2	2ndQuadrant	Y	C	Extension	Y	ASync	PostgreSQL 9.6	Y
3	xDB	EnterpriseDB	N	Java	Application	Y		PostgreSQL 12	Y
4	PostgreSQL-XL	2ndQuadrant*	Y	C	Fork	Y		PostgreSQL 10	Y
5	PostgreSQL-XC/XC2	EnterpriseDB and NTT	Y	C	Fork	Y			Y
6	Rubyrep	Arndt Lehmann	Y		Application	Y			N
7	Bucardo	End Point Corporation	Y	PERL	Application	Y		Postgres 12	Y

- Fonte: <https://www.percona.com/blog/multi-master-replication-solutions-for-postgresql/>





## Bucardo

Multi-master Database  
Replication



## check\_postgres

Nagios/Icinga-friendly  
PostgreSQL monitoring



## DBIx::Safe

Restricted Database Handles  
in Perl



## tail\_n\_mail

Live logfile monitoring



## Boxinfo

Gather host info easily



## pgsi

Wiki-ready query  
performance reports

## Conceito

- Sistema de replicação para o PostgreSQL assíncrono, permitindo operações multi-master e multi-slave.
- Desenvolvido no Backcountry.com por Jon Jensen e Greg Sabino Mullane da End Point Corporation.
- Software Livre de código aberto lançado sob a licença BSD \*.
  - \* 2 clausulas (citar autor e isenção de responsabilidade)
- Desenvolvido em Perl.
- Versão mais atual 5.6.0.



## Origem

- Referência ao Bucardo, uma subespécie de cabra selvagem que vivia nas montanhas dos Pireneus.
- Foi declarado extinto em 2000, mas, em 2003, cientistas tentaram reviver a espécie com técnicas de clonagem.
- A experiência teve um sucesso temporário, criando o primeiro animal “desextinto”, mas o clone viveu apenas alguns minutos.



## Conceito

- Multi-master suporta PostgreSQL.
- Para replicação a origem deve ser PostgreSQL, mas os destinos podem ser PostgreSQL, MySQL, Redis, Oracle, MariaDB, SQLite ou MongoDB.



PostgreSQL



ORACLE



redis



## Requisitos

- Perl e libs Perl e postgresql-perl.
- Requer um banco de dados dedicado e é executado como um daemon Perl que se comunica com esse banco de dados e todos os outros bancos de dados envolvidos na replicação.
- Todas as informações específicas que o daemon são armazenadas no banco de dados bucardo principal, incluindo uma lista de todos os bancos de dados envolvidos na replicação e como alcançá-los, todas as tabelas que devem ser replicadas e como cada uma deve ser replicada.

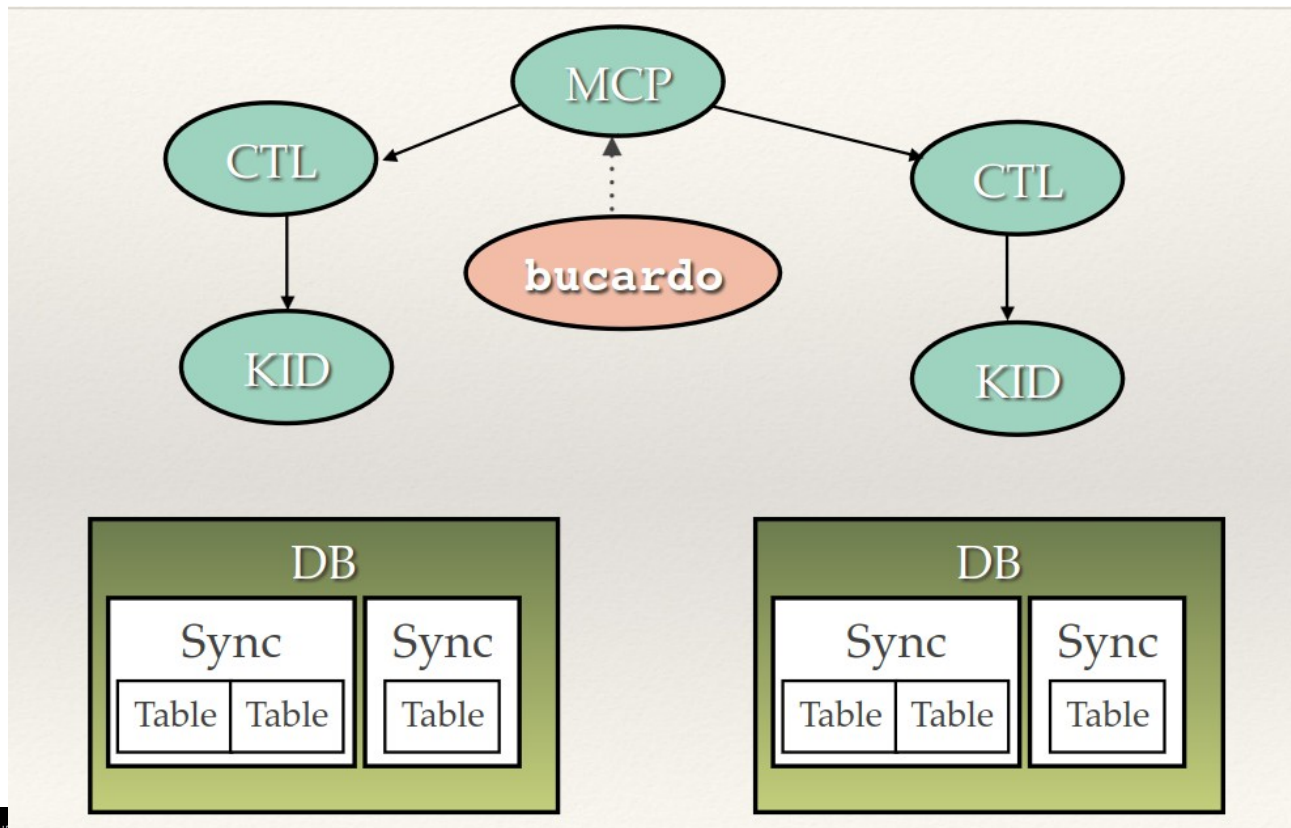


## Multi-master

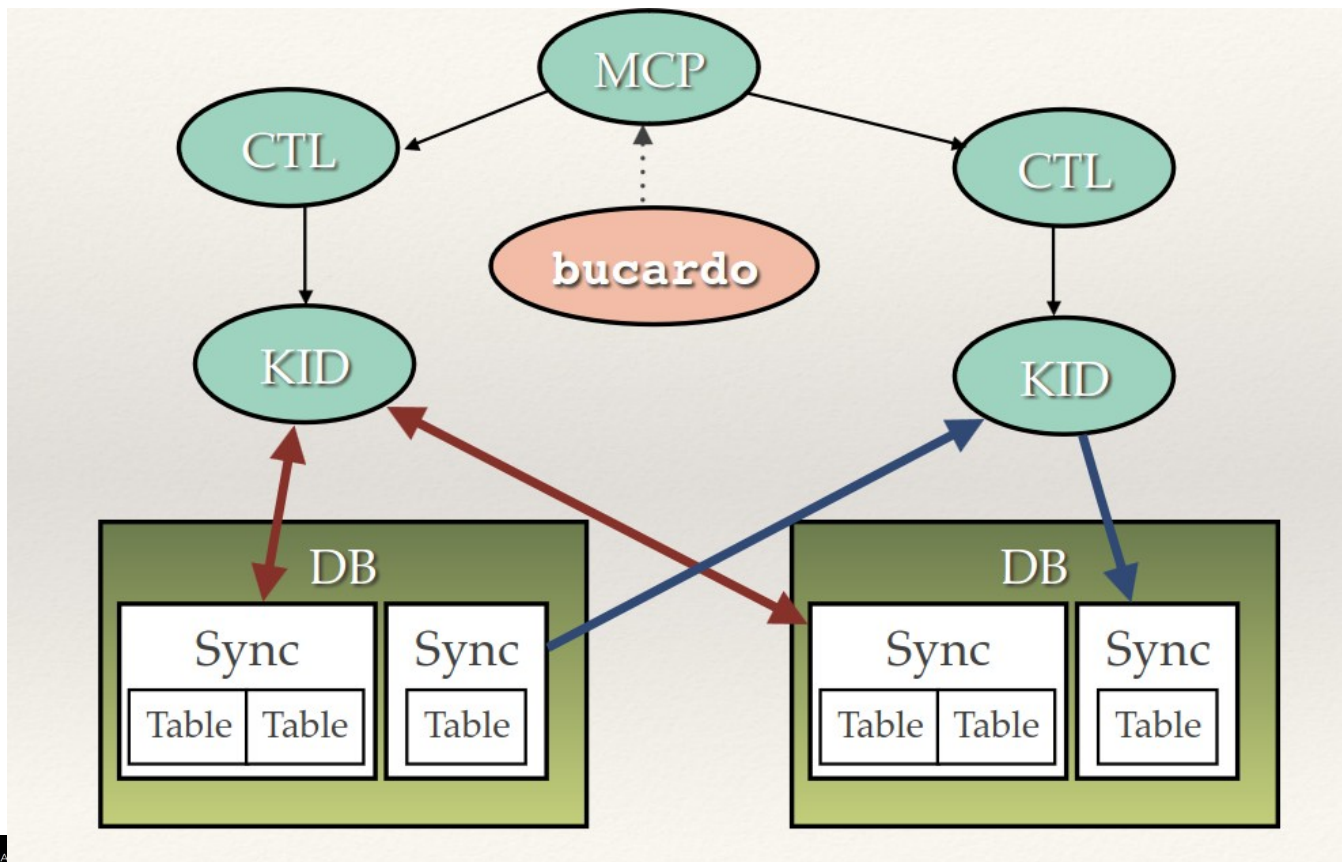
- A replicação multi-master usa dois ou mais bancos de dados, com resolução de conflitos (opções padrão ou sub-rotinas personalizadas) para lidar com a mesma atualização em ambos os lados.
- A replicação mestre-slave envolve uma ou mais origens indo para um ou mais destinos.



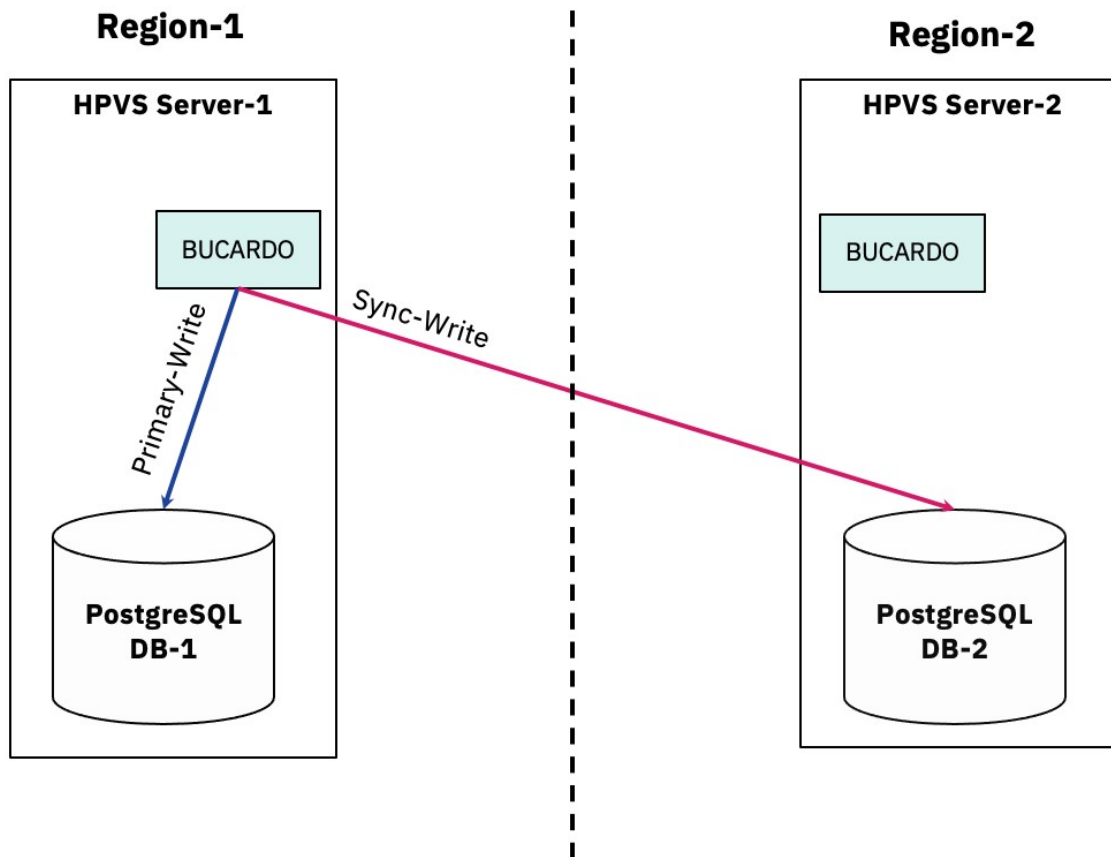
# Bucardo - Master-Master



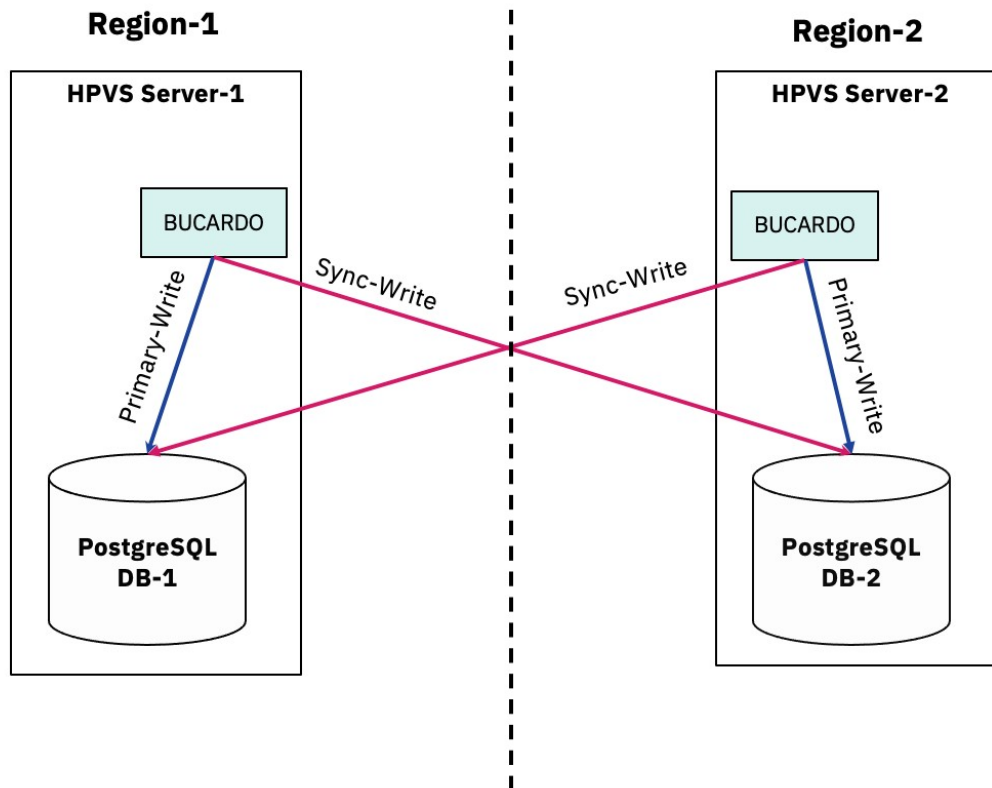
# Bucardo - Master-Master



# Bucardo - Master-Master



# Bucardo - Master-Master



## Workflow

- A primeira etapa na execução é adicionar dois ou mais bancos de dados ao banco de dados principal do Bucardo.
- Feito isso, são adicionadas informações sobre quais tabelas devem ser replicadas, bem como quaisquer agrupamentos de tabelas.
- Em seguida, as sincronizações são adicionadas. As sincronizações são ações de replicação nomeadas, copiando um conjunto específico de tabelas de um servidor para outro servidor ou grupo de servidores.
- Uma vez configurado o Bucardo, as triggers começam a armazenar informações sobre quais linhas foram alteradas em todas as tabelas de interesse.

## Sincronização de troca (multi-master)

- Uma alteração é feita na tabela e registrada na tabela bucardo\_delta.
- Um aviso é enviado ao daemon principal, informando alteração.
- O daemon notifica o controlador para essa sincronização e volta a escutar.
- O controlador cria um “kid” para lidar com a replicação.
- O kid inicia uma nova transação e desabilita triggers e regras nas tabelas.
- Em seguida, reúne uma lista de quais linhas foram alteradas desde a última replicação e compara as duas para descobrir o que deve ser feito.
- Se houver um conflito, o manipulador de conflitos padrão ou personalizado, definido por tabela, será executado para resolver as coisas.
- Triggers e regras são reativados e a transação é confirmada.
- Se a transação falhar, todos os manipuladores de exceção personalizados serão executados.
- O kid sinaliza para o controlador que terminou.



## Cli commands

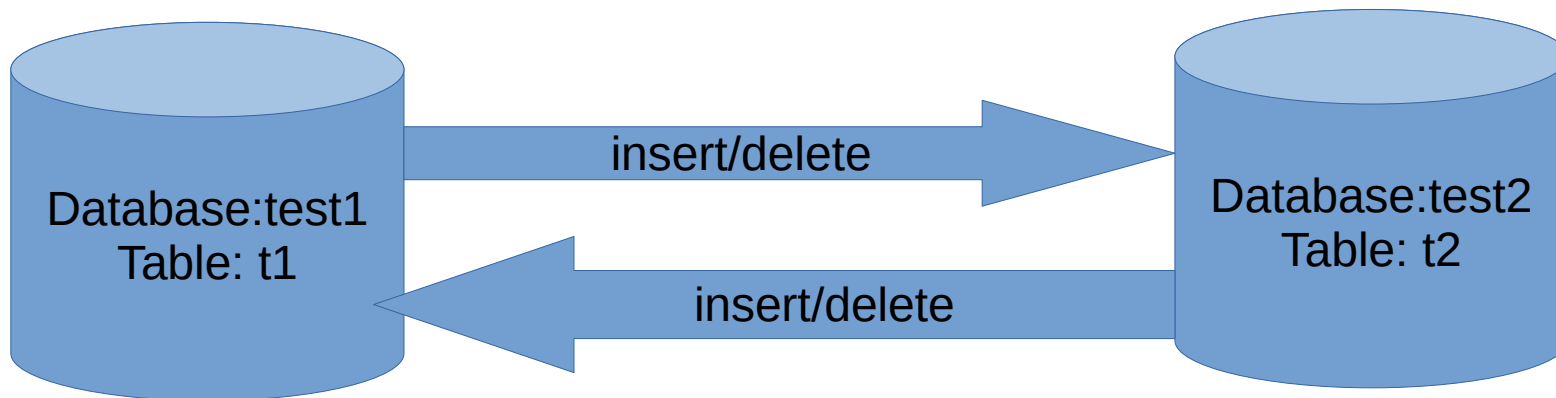
- bucardo install
- bucardo add
- bucardo status
- bucardo kick
- bucardo help

## Limitações

- Sem manipulação automática de DDL
- Requer PKs

## O que faremos

- Instalar o Bucardo.
- Criar dois databases (no mesmo server para teste)
- Apresentar Replicações.



# Obrigado

Marcio Junior Vieira

[marcio@ambientelivre.com.br](mailto:marcio@ambientelivre.com.br)

@marviojvieira @ambientelivre

<https://www.linkedin.com/in/mvieira1/>