

# DATABASE MIGRATIONS E SEEDS



# #WHOAMI

## WILLIAM FELIPE WELTER

- Gerente da TI na 4Linux
- Bacharel em Ciência da Computação
- Experiência com desenvolvimento e infraestrutura



**DEVOPS**

# PRINCIPIOS

- Sem intervenções manuais
- Infra as a Code
- Versionamento
- Squads focados no produto
- Metodologias Ageis

# CONTEXTO DEVOPS

- Pipelines
- Continuous Delivery / Continuous Deploy
- Build Automatizado
- Testes Automatizados
- Deploy Automatizado

**MAS, E O BANCO DE DADOS ?**

**O QUE SÃO MIGRATIONS?**

# MIGRATIONS

- Versionamento
- Automatizado
- Alterações de banco dados



# FERRAMENTAS

- Phinx
- FlywayDb
- Dbmate
- Goose
- ...

# PORQUE O PHINX ?

- Open Source
- Suporte a diversos SGBD
- Independente de framework
- Simples

**UTILIZANDO O PHINX**

# INSTALAÇÃO

- Instalar composer
- `php composer.phar require robmorgan/phinx`

# CONFIGURAÇÃO (PHINX.YML)

```
paths:  
  migrations: migrations  
  seeds: seeds  
  
environments:  
  default_migration_table: phinxlog  
  default_database: local  
  production:  
    adapter: pgsql  
    host: PROD_HOSTNAME  
    name: PROD_DBNAME  
    user: PROD_USERNAME  
    pass: 'PROD_PASSWORD'  
    port: PROD_PORT  
    charset: utf8
```

# CONFIGURAÇÃO (PHINX.YML)

- Definir diretórios para armazenar as migration e seeds
- Nome da tabela que ira registrar as migrations aplicadas
- Ambientes

# CRIANDO UMA MIGRATION

```
phinx create CriaTabelaTurmas
Phinx by CakePHP - https://phinx.org. 0.10.8

using config file ./phinx.yml
using config parser yaml
using migration paths
  - /var/www/html/migrations
using seed paths
using migration base class Phinx\Migration\AbstractMigration
using default template
created migrations/20190714220515_cria_tabela_turmas.php
```

# ESCREVENDO UMA MIGRATION

```
use Phinx\Migration\AbstractMigration;

class CriaTabelaTurmas extends AbstractMigration
{
    public function change()
    {
        $class=$this->table("class");
        $class->addColumn('date_begin','date')
            ->addColumn('date_end','date')
            ->addColumn('course_id','integer')
            ->addColumn('deleted','boolean',array(
                'null'=>false,
                'default' => 'false'
            ))
        ->create();
    }
}
```



# METODO CHANGE()

- Reversível - "Automágico"
- Não se deve utilizar SQL puro
- Não se deve inserir dados
- Utilizar `create()` ou `update()`

# METODO UP()

- O que deve ser feito quando migração for aplicada
- Ideal quando quando dados são manipulados (inserção/atualização)
- Permite uso de SQL puro

# UP()

```
use Phinx\Migration\AbstractMigration;

class CriarColunaUpdatedAt extends AbstractMigration
{
    public function up()
    {
        $this->execute("ALTER TABLE class ADD COLUMN updated_at timestamp");
    }
}
```

# METODO DOWN()

- O que deve ser feito quando acontece um rollback de uma migração
- Permite criar rollback não destrutivos
- Não obrigatorio

# DOWN()

```
use Phinx\Migration\AbstractMigration;

class CriarColunaUpdatedAt extends AbstractMigration
{
    public function up()
    {
        $this->execute("ALTER TABLE class ADD COLUMN updated_at timestamp");
    }

    public function down(){
        $this->execute("ALTER TABLE class DROP COLUMN updated_at");
    }
}
```

# UTILIZANDO A API DO PHINX

- + Abstração de SGBD
- + Reversível
- - Flexibilidade
- - Não utiliza recursos específicos do SGBD

# UTILIZANDO A API DO PHINX

- Objeto Table
- execute()
- query()
- fetchRow() / fetchAll()
- insert()

# OBJETO TABLE

- `addColumn()`, `changeColumn()`, `removeColumn()`
- `addIndex()`
- `addForeignKey()`
- `save()` , `create()`, `update()`



**DEMO**

# APLICANDO UMA MIGRATION

- `phinx migrate [-t TARGET] [-e ENVIRONMENT]`

# APLICANDO UMA MIGRATION

```
$ php vendor/bin/phinx migrate
Phinx by CakePHP - https://phinx.org. 0.10.8

using config file ./phinx.yml
using config parser yaml
using migration paths
  - /var/www/html/migrations
using seed paths
warning no environment specified, defaulting to: local
using adapter pgsql
using database postgres

== 20190721015556 CriarColunaUpdatedAt: migrating
== 20190721015556 CriarColunaUpdatedAt: migrated 0.0970s

All Done. Took 0.1143s
```

# ROLLBACK DE MIGRATIONS

- `phinx rollback [-t TARGET] [-e ENVIRONMENT]`

# ROLLBACK DE MIGRATIONS

```
$ php vendor/bin/phinx rollback
Phinx by CakePHP - https://phinx.org. 0.10.8

using config file ./phinx.yml
using config parser yaml
using migration paths
  - /var/www/html/migrations
using seed paths
warning no environment specified, defaulting to: local
using adapter pgsql
using database postgres
ordering by creation time

== 20190721015556 CriarColunaUpdatedAt: reverting
== 20190721015556 CriarColunaUpdatedAt: reverted 0.0377s
```

# CONSULTANDO MIGRATIONS

- `phinx status [-e ENVIRONMENT]`

# CONSULTANDO MIGRATIONS

```
$ php vendor/bin/phinx status
Phinx by CakePHP - https://phinx.org. 0.10.8
```

```
using config file ./phinx.yml
```

```
using config parser yaml
```

```
using migration paths
```

```
- /var/www/html/migrations
```

```
using seed paths
```

```
warning no environment specified, defaulting to: local
```

```
ordering by creation time
```

Status	[Migration ID]	Started	Finished	Mi
up	20190714220515	2019-07-14 23:57:07	2019-07-14 23:57:07	Cr
up	20190721011004	2019-07-21 01:50:50	2019-07-21 01:50:50	Cr
up	20190721015556	2019-07-21 02:07:21	2019-07-21 02:07:21	Cr

**DEMO**



**E A SEGURANÇA ?**

# QUESTÕES DE SEGURANÇA

- GIT garante -> Rastreabilidade e auditoria
- Testes garantem -> Que a migração não "quebrou" a aplicação
- Code Review -> Qualidade e acertividade das migrações

# PRECISAMOS ESTAR PREPARADOS PARA O PIOR!

- Replicas Hotstandby
- Backup P.I.T.R
- Replicas Atrasadas

**POSSO APLICAR MIGRAÇÕES SEM  
IMPACTAR A PRODUÇÃO ?**

# LOCKS E CONSISTENCIA

- No PostgreSQL DDLs são transacionais
- Migrations podem ser encapsuladas em transações

# TRUNCATE

- Transaccional
- Não é MVCC safe
- ACCESS EXCLUSIVE LOCK
- DELETE (alternativa)

# ALTER TABLE

- Transacional
- Rewrite / Non Rewrite Operations
- ADD COLUMN - Com valor default (Rewrite  $\leq 10$ ) (Non Rewrite  $\geq 11^*$ )
- TYPE - Rewrite e Reindex
- DROP COLUMN - Non Rewrite
- Rewrite não é MVCC safe
- ACCESS EXCLUSIVE LOCK
- \*Não volátil

# CREATE INDEX / REINDEX

- LOCK para escrita
- CONCURRENTLY sem LOCK



**COMO UTILIZAR NA MINHA  
PIPELINE?**

# PIPELINE

- Incluir na etapa anterior ao deploy
- Encapsular na inicialização da aplicação
- Incluir no processo de deploy

**DEMO**

**O SÃO SEEDS ?**

# SEEDS

- Inicializar uma base com um conjunto de dados
- Util para agilizar testes dependentes de dados no banco
- Não registra o que foi aplicado
- Não segue ordem de execução

# CRIANDO UMA SEED

```
$ phinx seed:create ClassSeeder
Phinx by CakePHP - https://phinx.org. 0.10.8

using config file ./phinx.yml
using config parser yaml
using migration paths
  - /home/william/Documents/pgconf2019/src/migrations
using seed paths
  - /home/william/Documents/pgconf2019/src/seeds
using seed base class Phinx\Seed\AbstractSeed
created ./seeds/ClassSeeder.php
```

# ESCREVENDO UMA SEED

```
use Phinx\Seed\AbstractSeed;

class ClassStatusSeeder extends AbstractSeed
{

    public function run()
    {
        $status=$this->table('status');
        $data=[
            [
                'id' => '101',
                'name' => 'arguando_sala',
                'description' => 'Aguardando liberacao de sala'
            ]
        ];
        $status->insert($data);
    }
}
```

# METODO RUN()

- Não é reversível
- É onde deve ser inseridos os dados



# METODO GETDEPENDENCIES()

- Utilizado para estabelecer dependencias
- Deve retornar array com as classes dependentes

# METODO GETDEPENDENCIES()

```
use Phinx\Seed\AbstractSeed;

class ClassSeeder extends AbstractSeed
{
    public function getDependencies(){
        return [
            'ClassStatusSeeder'
        ];
    }

    public function run()
    {
        $class=$this->table('class');
        $data = [
```

# APLICANDO SEEDS

- `phinx seed:run [-s SEED] [-e ENVIRONMENT]`

# APLICANDO SEEDS

```
phinx seed:run
Phinx by CakePHP - https://phinx.org. 0.10.8

using config file ./phinx.yml
using config parser yaml
using migration paths
  - /var/www/html/migrations
using seed paths
  - /var/www/html/seeds
warning no environment specified, defaulting to: local
using adapter pgsql
using database postgres

== ClassStatusSeeder: seeding
== ClassStatusSeeder: seeded 0.0301s
```

**OBRIGADO**

**WILLIAM FELIPE WELTER**

**WILLIAM.WELTER@4LINUX.COM.BR**

**BR.LINKEDIN.COM/IN/WILLIAMFELIPEWELTE**

**R**

**GITHUB.COM/WFELIPEW**