



Roger Siqueira
Anderson Fuhr



Agenda



Flyway:

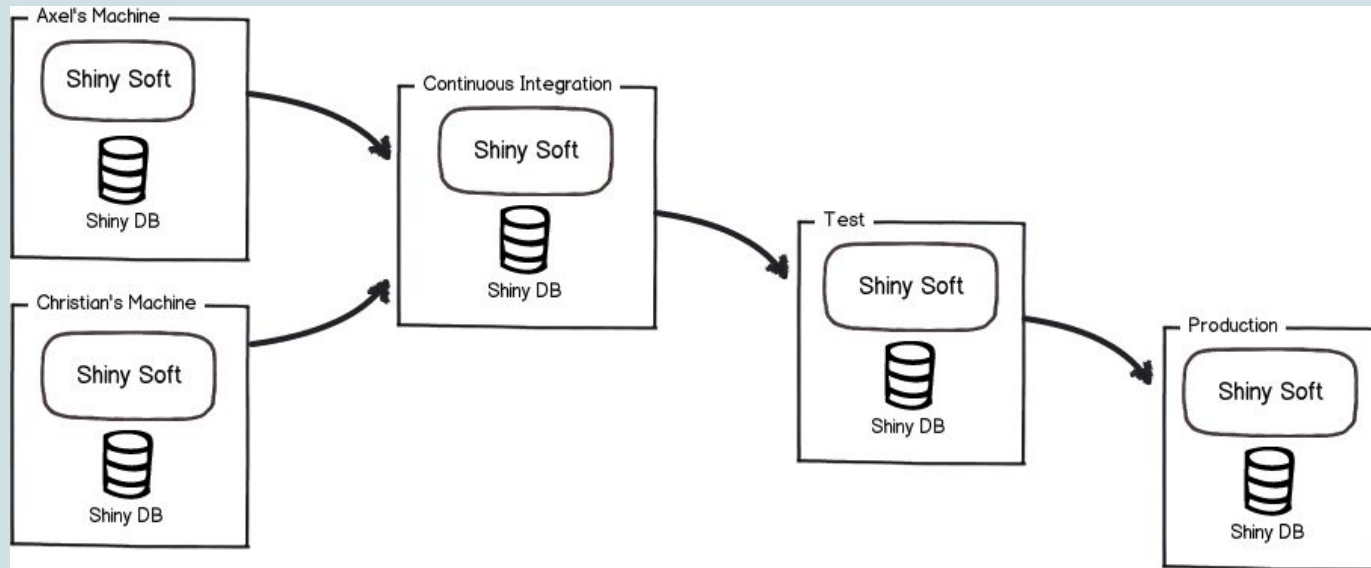
- Definição e como funciona;
- Instalação;
- Callbacks;
- Testes;
- Implementação e casos de uso;
- Debate / Dúvidas;

Flyway - Definição



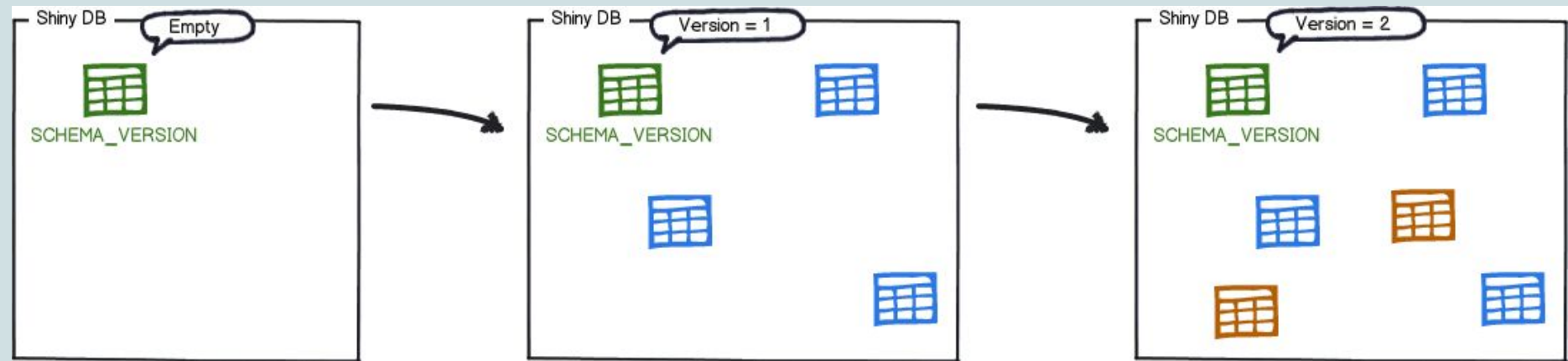
- Ferramenta de versionamento de banco de dados (BD);
- Gerencia e automatiza alterações no BD;
- Código aberto e licenciado pela Apache License 2.0.

Flyway - Por que versionar o banco de dados?



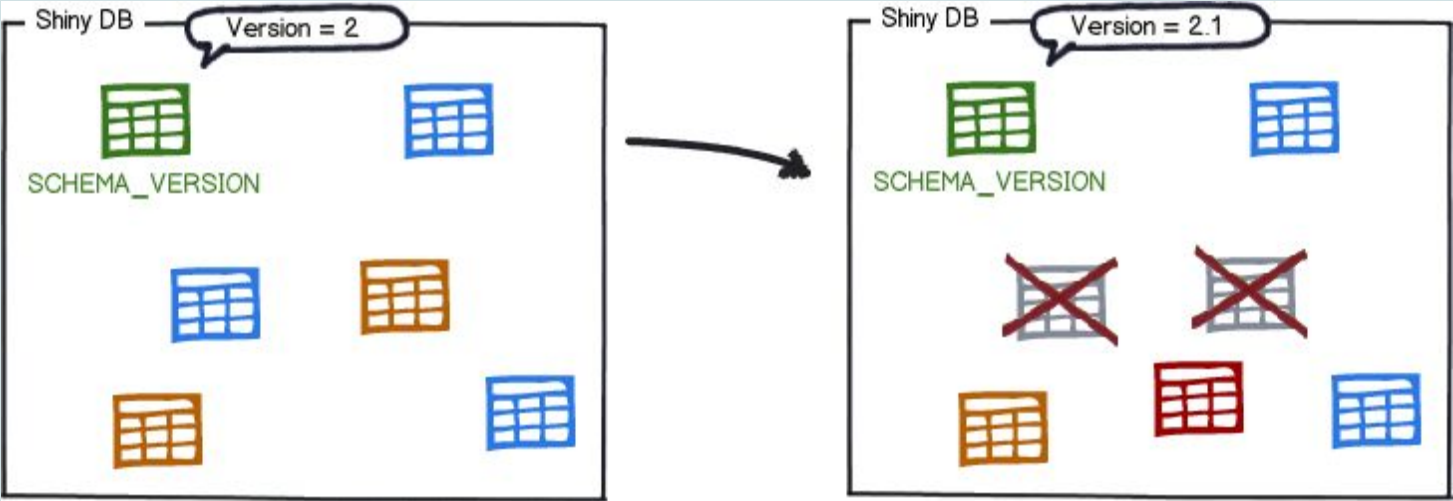
- Replicar e gerenciar um BD;
- Deixar claro o estado atual;

Flyway - Como funciona?



version	description	type	script	checksum	installed_by	installed_on	execution_time	success
1	Initial Setup	SQL	V1__Initial_Setup.sql	1996767037	axel	2016-02-04 22:23:00.0	546	true
2	First Changes	SQL	V2__First_Changes.sql	1279644856	axel	2016-02-06 09:18:00.0	127	true

Flyway - Como funciona?



vers...	description	type	script	checksum	installed_by	installed_on	execution_time	suc...
1	Initial Setup	SQL	V1__Initial_Setup.sql	1996767037	axel	...22:23:00.0	546	true
2	First Changes	SQL	V2__First_Chang.sql	1279644856	axel	... 09:18:00.0	127	true
2.1	Refactoring	JDBC	V2_1__Refactoring	1549856922	axel	... 17:45:05.4	251	true

Flyway - Como funciona?





- **Validate** - verifica se os scripts aplicados no BD correspondem aos disponíveis na pasta migrate (checksum)
- **Baseline** - Define o ponto de partida para o migrate
- **Migrate** - Migra o Schema do BD para a última versão
- **Check** - Análise pré ou pós migrate
- **Clean** - Dropa os dados configurados
- **Undo** - Desfaz a aplicação do script mais recente



Flyway - Instalação

- Pré requisitos: BD configurado.
- Adicionar dependência ao gerenciador;
- Instanciar o Flyway no método main do projeto:

```
Flyway flyway = Flyway.configure()  
    .dataSource("jdbc:h2:file:./target/foobar", "sa", null).load();
```

- Executar os comandos no método main:

```
flyway.migrate();
```



Flyway - Instalação

Projeto Spring Boot:

- Pré requisitos: BD configurado.
- Adicionar dependência ao gerenciador;
- Configuração do Flyway no application.properties;
- Execução automática dos comandos pelo Spring.



Flyway - Instalação

- application.properties

```
spring.flyway.url=${database_url}  
spring.flyway.user=${usuario_do_banco_de_dados}  
spring.flyway.password=${senha_do_banco_de_dados}  
flyway.baseline-on-migrate=true  
spring.flyway.baseline-version=1  
flyway.locations=classpath:/db/migration
```

- configurar jpa para não gerenciar o BD

```
spring.jpa.hibernate.ddl-auto=none  
spring.jpa.generate-ddl=false
```



Flyway - CallBacks

- SQL Callbacks
- Java CallBacks
- Script Callbacks



- Arquivos SQL que são executados conforme a convenção de nomenclatura: nome do evento seguido pelo sufixo de migração SQL.
- Exemplos:
beforeMigrate.sql, beforeEachMigrate.sql, afterMigrate....



- Interface Callback: se utiliza de três métodos, que são:
 - supports: eventos que serão manipulados
 - canHandleInTransaction: decide se a manipulação será tratada de forma transacional
 - handler: lógica que será executada em cada tipo de evento

Flyway - Callbacks Java



```
public static class MeuFlywayCallback implements Callback {  
    @Override  
    public boolean supports(Event event, Context context) {  
        // Retorne true para os eventos de callback que você deseja manipular  
        return event == Event.BEFORE_CLEAN  
            || event == Event.AFTER_CLEAN  
            || event == Event.BEFORE_MIGRATE  
            || event == Event.AFTER_MIGRATE;  
    }  
  
    @Override  
    public boolean canHandleInTransaction(Event event, Context context) {  
        /* Retorne true se você deseja manipular o evento dentro de uma transação,  
        caso contrário, retorne false */  
        return false;  
    }  
  
    @Override  
    public void handle(Event event, Context context) {  
        // Lógica a ser executada em cada evento de callback  
        switch (event) {  
            case BEFORE_CLEAN:  
                // Lógica a ser executada antes da limpeza do banco de dados  
                System.out.println("Before clean");  
                break;  
            case AFTER_CLEAN:  
                // Lógica a ser executada após a limpeza do banco de dados  
                System.out.println("After clean");  
                break;  
            // Adicione mais casos conforme necessário para outros eventos  
        }  
    }  
}
```



Flyway - Callbacks Scripts

- Oferece suporte às extensões de arquivos:
 - .ps1 -> PowerShell Windows
 - .bat -> Windows
 - .cdm -> Windows
 - .sh -> Unix/Linux
 - .bash -> Unix/Linux
 - .py -> Python

- Para não utilizar o contexto das migrações nos testes, desabilitar o Flyway em `src/test/resources/application.properties`

```
spring.flyway.enabled=false  
spring.jpa.hibernate.ddl-auto=create
```

- Para utilizar o Flyway em testes:

```
flyway.url=jdbc:h2:mem:testedb  
flyway.user=sa  
flyway.password=  
flyway.locations=classpath:db/migration
```

Flyway - Testes



```
public class MyTest {  
  
    private Flyway flyway;  
  
    @Before  
    public void setUp() {  
        flyway = Flyway.configure()  
            .dataSource("jdbc:h2:mem:test", "sa", null)  
            .locations("db/migration")  
            .load();  
        flyway.migrate();  
    }  
  
    @After  
    public void tearDown() {  
        flyway.clean();  
    }  
  
    @Test  
    public void myTest() {  
        // teste aqui  
    }  
}
```



Flyway - Implementação e casos de uso

- Casos de uso no Sicredi:
 - Resolução de incidentes
 - Versionamento de BD
 - ...

- Uso do Flyway na própria API ou API separada?
- Sempre devemos versionar?
- Sua equipe mantém uma documentação sobre o versionamento do banco de dados?
- Quais outras ferramentas para versionamento além do Flyway?



Roger Siqueira - roger.siqueira@db.tec.br
Anderson Fuhr - anderson.souza@db.tec.br

