



CAPACITAÇÕES
**JAVA PARA
A PDPJ-Br**

JAVA AVANÇADO

CADERNO DE ATIVIDADES - TRILHA 3

Ronaldo Pinheiro Gonçalves Junior



ATIVIDADES DO CADERNO

Atividade 1: Instalação do SGBD PostgreSQL

Objetivo:

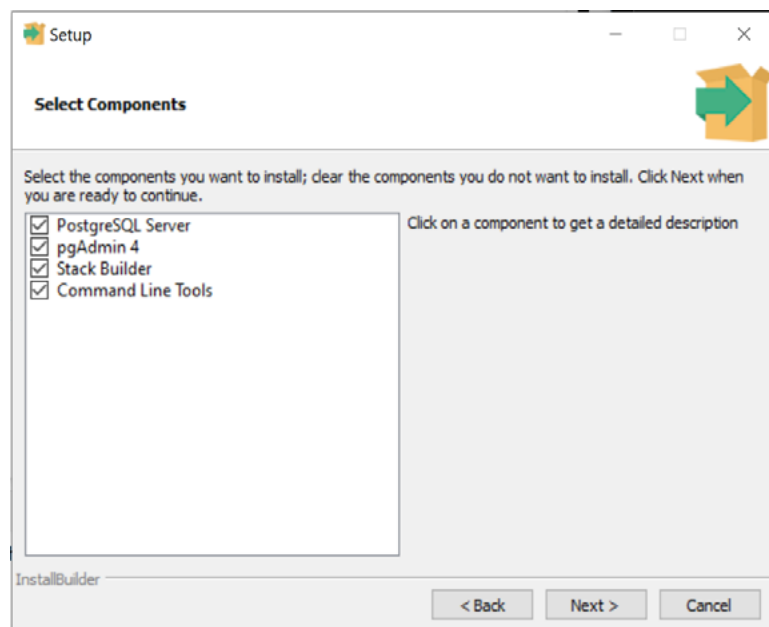
Instalar o Sistema de Gerência de Banco de Dados (SGBD) PostgreSQL.

Requisitos:

Não tem.

Descrição da atividade 1:

1. Acesse o *site* oficial: <https://www.postgresql.org/download/>
2. Selecione a opção de instalação correspondente ao seu sistema operacional:
 - Nesta instalação, utilizaremos a versão “16.2”;
3. Inicie o processo de instalação:
 - Caso apareça, selecione o servidor PostgreSQL e o pgAdmin. Eles serão utilizados para gerenciamento via interface gráfica. Você pode selecionar também o Stack Builder e o Command Line Tools, caso queira ter acesso a ferramentas adicionais de gerenciamento, integração, migração e assim por diante;



Fonte: Elaborado pelo autor, 2024.

4. Durante o processo de instalação, você precisará definir uma senha para o usuário administrador “postgres”;

5. Escolha a porta em que o PostgreSQL irá executar:
 - Nesta instalação, utilizaremos a porta 5432;
6. Aguarde e finalize o processo de instalação.

Atividade 2: Configuração da ferramenta pgAdmin do SGBD PostgreSQL

Objetivo:

Configurar a ferramenta pgAdmin para a execução de comandos SQL do PostgreSQL.

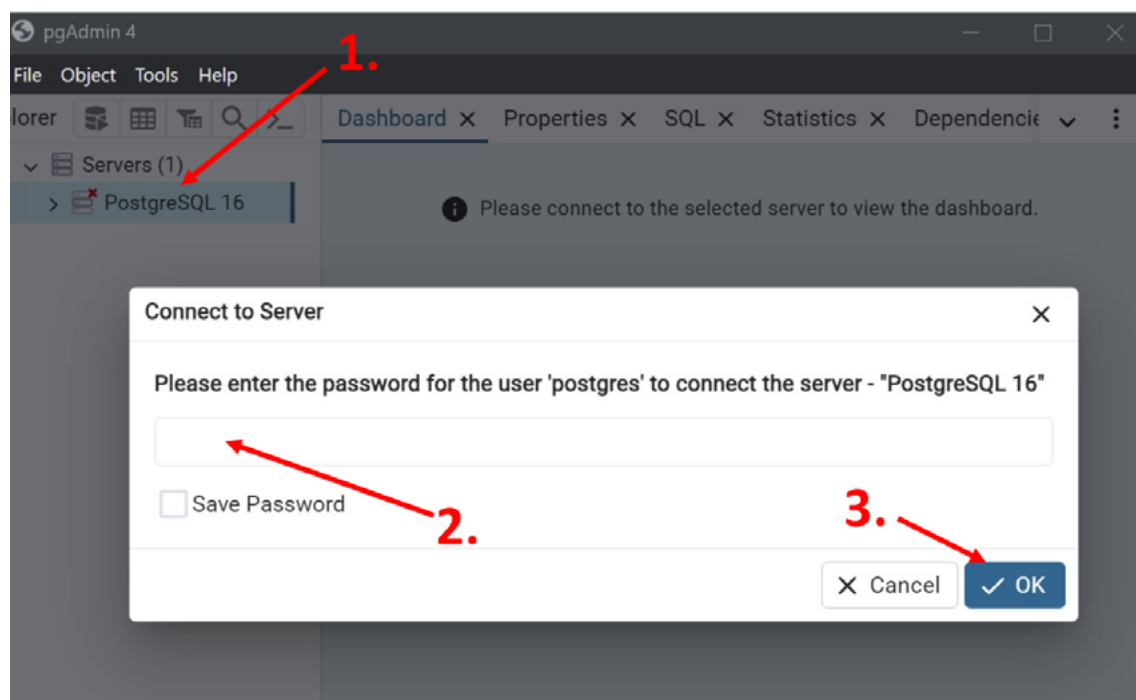
Requisito:

- Realização da atividade 1.

Descrição da atividade 2:

Após a instalação do PostgreSQL, vamos configurar a ferramenta pgAdmin e testar o acesso ao servidor PostgreSQL, para executar as próximas atividades. Abra a ferramenta pgAdmin que acabamos de instalar e siga os passos abaixo:

1. Selecione o servidor PostgreSQL, na lista de servidores;
2. Informe a senha definida no passo 4 da instalação;
3. Confirme o preenchimento para acessar o servidor.

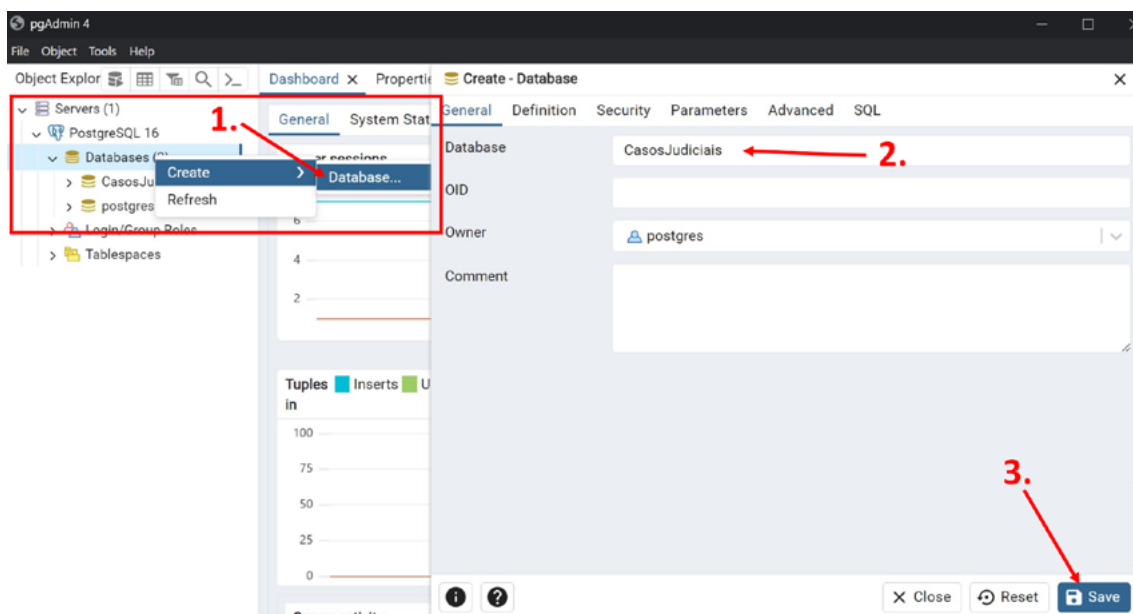


Fonte: Elaborada pelo autor, 2024.



Agora, com um acesso ao servidor, vamos criar um novo banco de dados e abrir a janela de execução de comandos SQL:

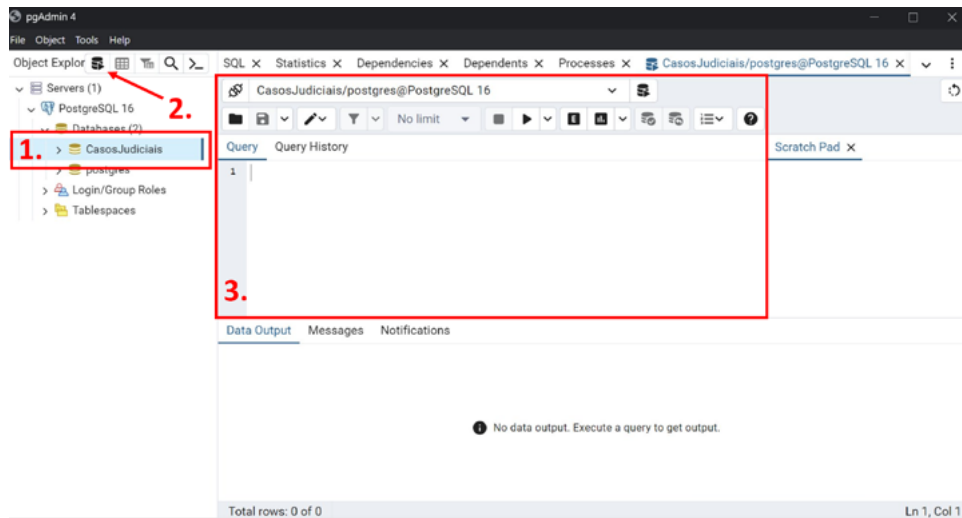
1. Com o servidor PostgreSQL expandido, clique com o botão direito em Databases e acesse a opção de criação de um novo banco de dados;
2. Preencha o campo nome do banco de dados:
 - Na imagem exemplo, utilizamos o nome “CasosJudiciais”;
3. Selecione a opção para salvar o banco de dados.



Fonte: Elaborada pelo autor, 2024.

Com o banco de dados criado, precisamos abrir a ferramenta de busca, para a execução de comandos SQL, nas próximas atividades:

1. Selecione o banco de dados criado nos passos anteriores;
2. Escolha a opção de abrir a ferramenta de busca SQL;
3. Confirme que a área de consulta “Query” agora está visível.



Fonte: Elaborada pelo autor, 2024.

Atividade 3: Utilização da linguagem DDL no SGBD PostgreSQL

Objetivo:

Realizar operações de definição de dados, utilizando o executor de comandos SQL do PostgreSQL.

Requisito:

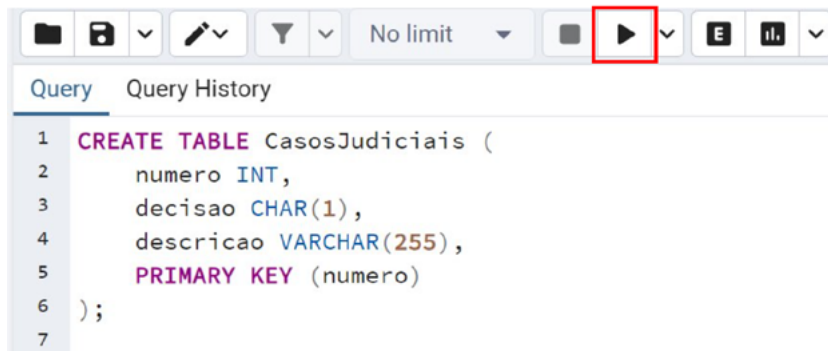
- Realização da atividade 2.

Descrição da atividade 3:

Escreva um comando na linguagem DDL, para a criação de uma nova tabela. Você pode utilizar o código abaixo:

```
CREATE TABLE CasosJudiciais (  
    numero INT,  
    decisao CHAR(1),  
    descricao VARCHAR(255),  
    PRIMARY KEY (numero)  
);
```

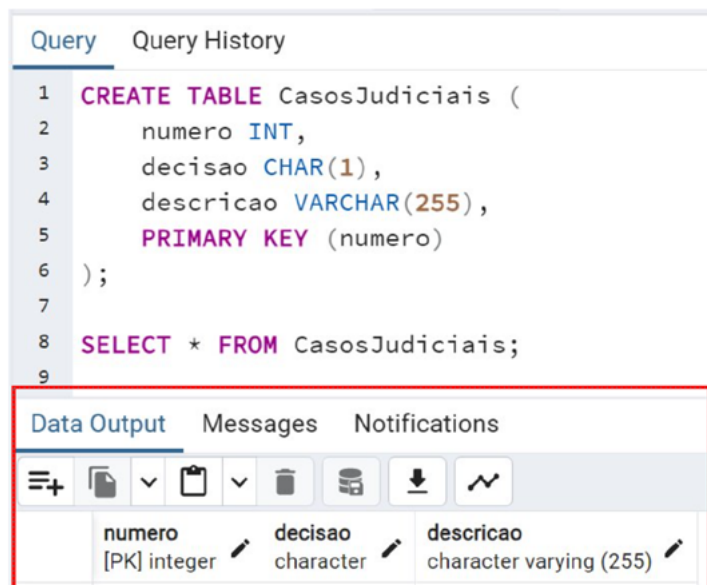
Para executar o comando, aperte a tecla F5 ou selecione a opção executar, conforme mostra a seguinte imagem.



Fonte: Elaborada pelo autor, 2024.

Você pode verificar a criação da tabela, executando o simples comando DQL: *SELECT * FROM CasosJudiciais;*

O resultado pode ser visto na aba de saída de dados "Data Output".



Fonte: Elaborada pelo autor, 2024.

Após a criação da tabela, execute os comandos a seguir e verifique a aba de saída de dados, ao executar cada um deles:

```
ALTER TABLE CasosJudiciais ADD dataAbertura DATE;
```

```
SELECT * FROM CasosJudiciais;
```

```
ALTER TABLE CasosJudiciais DROP dataAbertura;
```

```
SELECT * FROM CasosJudiciais;
```

```
ALTER TABLE CasosJudiciais RENAME TO CasoJudicial;
```

```
SELECT * FROM CasosJudiciais;
```

```
ALTER TABLE CasoJudicial RENAME TO CasosJudiciais;
```

```
SELECT * FROM CasosJudiciais;
```

```
DROP TABLE CasosJudiciais;
```

```
SELECT * FROM CasosJudiciais;
```

Execute novamente o comando de criação da tabela, antes de finalizar essa atividade, para que a tabela possa ser utilizada nas próximas atividades.

Atividade 4: Utilização das linguagens DQL e DDL no SGBD PostgreSQL

Objetivo:

Realizar operações de consulta e manipulação de dados, utilizando o executor de comandos SQL do PostgreSQL.

Requisito:

- Realização da atividade 3.

Descrição da atividade 4:

Execute os comandos a seguir e verifique a saída de dados, após cada um deles:

```
INSERT INTO CasosJudiciais (numero, decisao, descricao)
```

```
VALUES (1, 'A', 'Caso 1');
```

```
SELECT * FROM CasosJudiciais;
```

```
DELETE FROM CasosJudiciais WHERE decisao = 'A';
```

```
SELECT * FROM CasosJudiciais;
```

```
INSERT INTO CasosJudiciais (numero, decisao, descricao)
```

VALUES

(2, 'I', 'Descrição do caso 2'),

(3, 'C', 'Descrição do caso 3'),

(4, 'A', 'Descrição do caso 4');

SELECT * FROM CasosJudiciais;

UPDATE CasosJudiciais SET decisao = 'C' WHERE decisao = 'A';

SELECT * FROM CasosJudiciais

WHERE decisao = 'C'

ORDER BY numero DESC;

Atividade 5: Instalação da ferramenta FlyWay

Objetivo:

Efetuar a instalação da ferramenta FlyWay, com as opções de interface gráfica Desktop e linha de comando.

Requisito:

- Realização da atividade 4.

Descrição da atividade 5:

1. Acesse o *site* oficial:
 - <https://flywaydb.org/>
2. Acesse a opção de *download* do FlyWay:
 - Utilizaremos a versão gratuita “FlyWay Community”;
3. Selecione a opção de instalação correspondente ao seu sistema operacional;
4. Preencha os campos obrigatórios de endereço de email, nome e sobrenome:
 - Selecione a opção de download do FlyWay Community;
5. Execute a instalação do FlyWay:
 - O diretório de instalação escolhido neste exemplo foi: “C:\Program Files\Red Gate\Flyway Desktop”.

A versão mais atual do FlyWay conta com uma ferramenta gráfica que nos auxiliará na tarefa de migração e controle de versão, na camada de banco de dados. As próximas atividades do caderno usam essa instalação, mas, alternativamente, você pode optar por acessar o diretório de instalação diretamente, para configurar o FlyWay de forma manual e executar o FlyWay pela linha de comando, que também acabamos de instalar.

Atividade 6: Criação de um projeto FlyWay para o PostgreSQL

Objetivos:

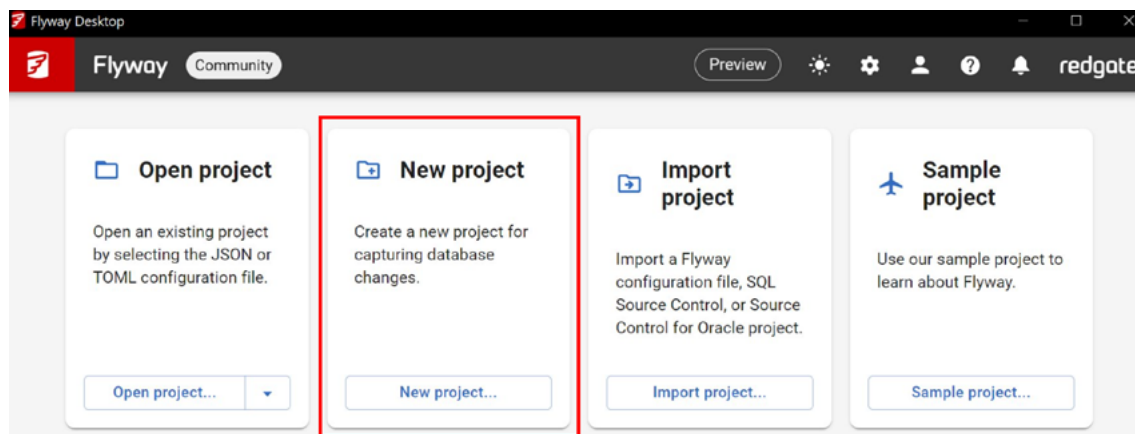
Criar um projeto para migração e controle de versão de banco de dados e conectar ao PostgreSQL.

Requisito:

- Realização da atividade 5.

Descrição da atividade 6:

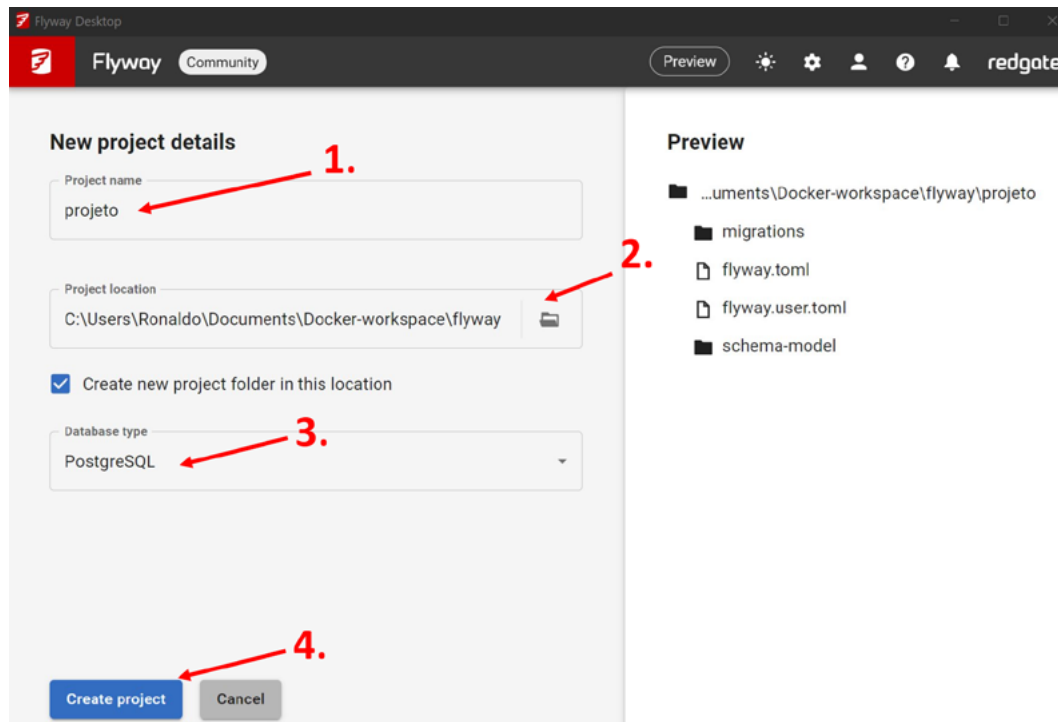
Execute a aplicação FlyWay Desktop, que instalamos anteriormente, e escolha a opção de criação de um novo projeto, conforme mostra a imagem a seguir:



Fonte: Elaborada pelo autor, 2024.

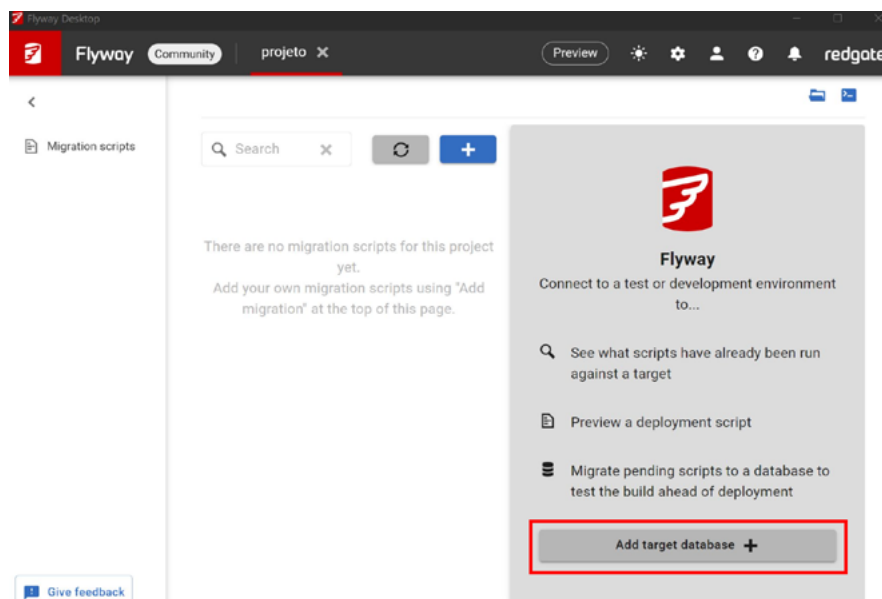
Em seguida, preencha as informações do projeto, levando em consideração a instalação do PostgreSQL feita nas atividades anteriores. Para isso, siga os passos abaixo:

1. Escolha um nome para o projeto;
2. Escolha um diretório para o projeto;
3. Selecione "PostgreSQL", para a opção tipo de banco de dados;
4. Confirme a criação do projeto.



Fonte: Elaborada pelo autor, 2024.

O próximo passo é realizar a conexão do projeto FlyWay com nossa instalação do servidor PostgreSQL. Para isso, selecione a opção de adicionar um novo banco de dados, conforme mostra a imagem a seguir:



Fonte: Elaborada pelo autor, 2024.

A última etapa para a configuração da nossa instalação é fornecer os dados de conexão ao banco de dados que criamos anteriormente. Siga o passo a passo, para concluir a conexão:

1. Preencha um nome, para identificação da conexão;
2. Escreva "localhost" no campo "Host";
3. Escreva "public" no campo "schemas":
 - 3.1. Neste caderno de atividades, não criamos um schema por meio do comando "CREATE SCHEMA". Logo, por padrão, o schema utilizado pelo PostgreSQL é o "public". Se você criou algum schema anteriormente no banco de dados existente, pode optar por usá-lo nesse momento.
4. Coloque "postgres" no campo "Username";
5. Preencha o campo de senha, com a senha definida na instalação do PostgreSQL;
6. Efetue o teste de conexão:
 - 6.1. Se tudo estiver preenchido corretamente, o FlyWay Desktop conseguirá testar a conexão com o banco de dados Postgre e mostrar uma mensagem de sucesso.
7. Selecione a opção de testar e salvar a conexão.

Atividade 7: Configuração da primeira versão e migração do banco de dados FlyWay para o PostgreSQL

Objetivos:

Estabelecer uma primeira versão do banco de dados, gerar e executar uma nova versão de migração, utilizando FlyWay conectado a um banco de dados PostgreSQL.

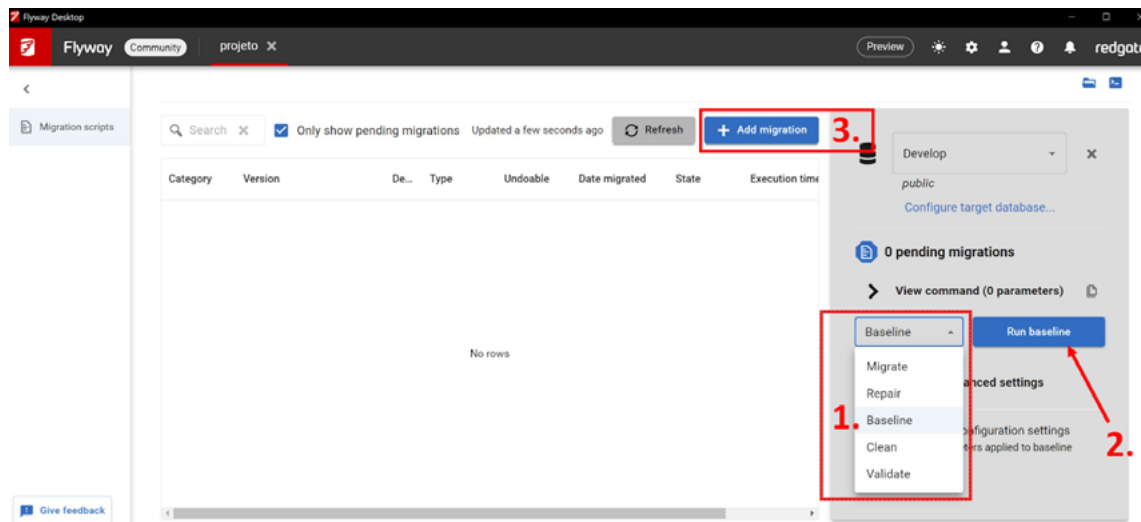
Requisitos:

- Realização da atividade 6.

Descrição da atividade 7:

O primeiro passo para iniciar as migrações e o controle de versões de banco de dados é o estabelecimento de uma primeira versão da estrutura. Siga os passos abaixo, para atingir esse objetivo:

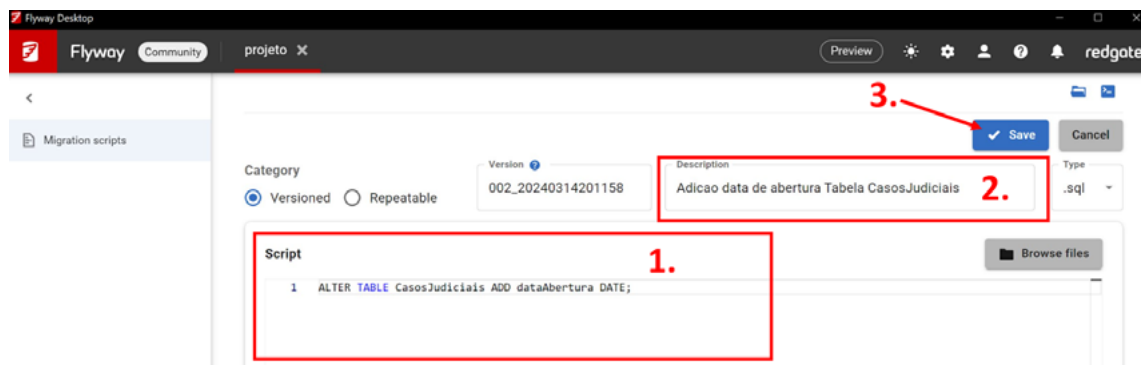
1. Selecione a ação de "Baseline";
2. Execute a ação de "Baseline";
3. Uma vez que o processo de criação da primeira versão da estrutura seja concluído com sucesso, selecione a opção de adição de migração.



Fonte: Elaborada pelo autor, 2024.

Agora que temos nossa conexão com o PostgreSQL e estabelecemos a atual estrutura do banco de dados como uma primeira versão, podemos criar novas versões ou aplicar modificações à estrutura do banco de dados, através da adição de migração. Vamos executar a adição de uma migração, conforme o seguinte exemplo:

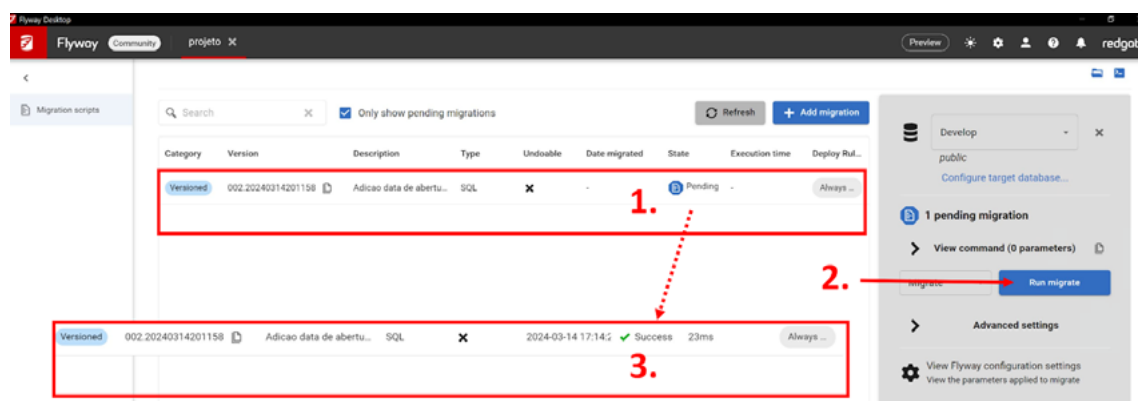
1. Escreva o comando SQL, para atualização do banco de dados:
 - Neste caso, estamos mexendo na estrutura da tabela CasosJudiciais, para adicionar uma nova coluna.
2. Descreva a mudança a ser aplicada:
 - Por convenção, devemos utilizar uma mensagem descritiva, para identificar as mudanças sendo aplicadas.
3. Selecione a opção de salvar a migração.



Fonte: Elaborada pelo autor, 2024.

Para concluirmos o processo, precisamos agora aplicar a ação de migração em todas as modificações com uma versão adicionada. Neste exemplo, temos apenas uma migração, mas o processo pode ser replicado por vários membros da equipe de desenvolvimento, para várias novas versões do banco de dados. Siga os passos abaixo, para aplicar as mudanças no banco de dados PostgreSQL:

1. Verifique que a migração possui uma versão, uma descrição e um estado “Pending”;
2. Execute a ação de migração;
3. Após a conclusão, verifique que a migração agora tem um estado “Success”.



Fonte: Elaborada pelo autor, 2024.

P.S.: Caso deseje, abra a ferramenta pgAdmin, para verificar se as mudanças realmente foram aplicadas.

Atividade 8: Configuração do projeto para Spring Data JPA e PostgreSQL de dados FlyWay para o PostgreSQL

Objetivo:

Configurar o projeto para utilizar Spring Data JPA e PostgreSQL.

Requisito:

- Realização da atividade 1.

Descrição da atividade 8:

O primeiro passo de configuração do projeto Spring Boot será a adição das dependências do Spring Data JPA e PostgreSQL:

```
<dependency>
```

```
<groupId>org.springframework.boot</groupId>
```

```
<artifactId>spring-boot-starter-data-jpa</artifactId>
```

```
</dependency>  
<dependency>  
    <groupId>org.postgresql</groupId>  
    <artifactId>postgresql</artifactId>  
    <scope>runtime</scope>  
</dependency>
```

Após adicionar as dependências, devemos configurar no arquivo `application.properties` as informações necessárias para conexão com o SGBD PostgreSQL. Primeiramente, devemos adicionar as linhas com o endereço do servidor do banco de dados e as credenciais:

```
spring.datasource.url=jdbc:postgresql://localhost:5432/CasosJudiciais  
spring.datasource.username= postgres  
spring.datasource.password= Seguro10!
```

Em seguida, como o Spring Boot utiliza o Hibernate como implementação da JPA, vamos adicionar o dialeto do PostgreSQL para o Hibernate:

```
spring.jpa.properties.hibernate.dialect=  
org.hibernate.dialect.PostgreSQLDialect
```

Finalmente, vamos configurar a propriedade `ddl-auto` para “update”, pois queremos que entidades sejam criadas e atualizadas automaticamente:

```
spring.jpa.hibernate.ddl-auto= update
```

P.S.: Não é recomendado que o valor dessa última propriedade permaneça “update”, para ambientes de produção. Em vez disso, recomenda-se o valor “validate”, que evita alterações no banco de dados.

Atividade 9: Instalação do servidor de mensagens RabbitMQ

Objetivo:

Instalar o servidor de mensagens RabbitMQ e suas dependências.

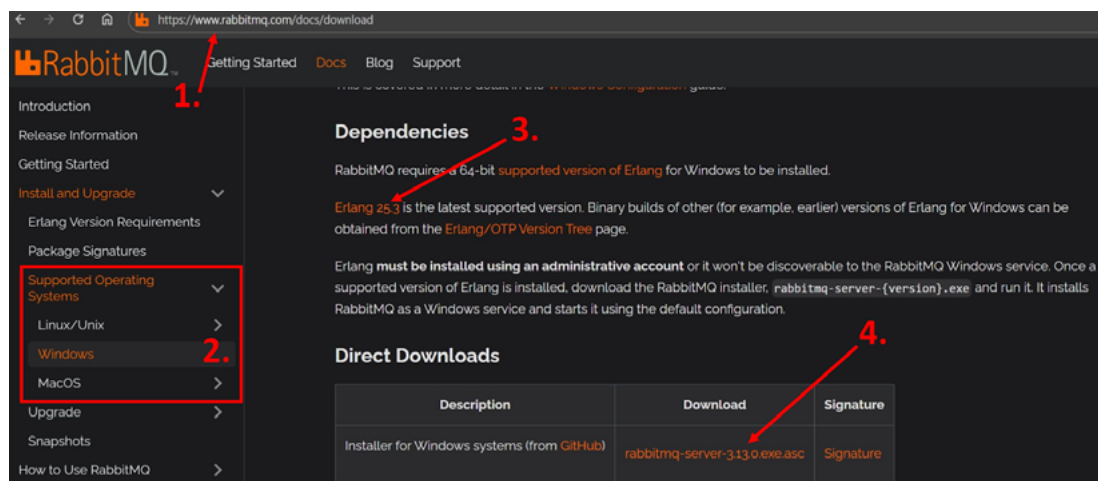
Requisitos:

- Erlang deve ser instalado, por um usuário administrador do sistema operacional;
- Possuir apenas uma versão do Erlang instalado.

Descrição da atividade 9:

1. Acesse o site oficial:
 - a. <https://www.rabbitmq.com/docs/download>

2. Selecione a opção de instalação correspondente ao seu sistema operacional;
3. Identifique a seção de dependências e faça a instalação do Erlang:
 - a. Realize esse passo apenas se o Erlang não estiver instalado;
 - b. Nesta atividade, foi utilizada a versão “26.2.3”;
 - c. Utilize um usuário administrador do seu sistema operacional, para efetuar a instalação.
4. Selecione a opção de instalação do RabbitMQ:
 - a. Nesta atividade, foi utilizada a versão “3.13.0”;
 - b. Utilize um usuário administrador do seu sistema operacional, para efetuar a instalação.



Fonte: Elaborada pelo autor, 2024.

Atividade 10: Configuração do projeto Spring Boot para o RabbitMQ

Objetivo:

Configurar o projeto Spring Boot, para integrar o servidor de mensagens RabbitMQ.

Requisito:

- Realização da atividade 9.

Descrição da atividade 10:

1. Adicione a dependência ao arquivo “pom.xml”:


```
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-amqp</artifactId>
</dependency>
```

2. No arquivo `application.properties`, adicione as linhas de configuração, ao final do arquivo:
- ```
spring.rabbitmq.host=localhost
spring.rabbitmq.port=5672
spring.rabbitmq.username=guest
spring.rabbitmq.password=guest
```

3. Altere a classe principal de execução do projeto, adicionando a anotação do RabbitMQ, conforme código exemplo abaixo:

```
@EnableRabbit
@SpringBootApplication
public class ProjetoApplication {
 public static void main(String[] args) {
 SpringApplication.run(ProjetoApplication.class, args);
 }
}
```

4. Crie um arquivo de configuração para a fila:

```
@Configuration
public class QueueConfig {
 public static final String NOME_FILA = "Casos";
 public static final String NOME_EXCHANGE = "ExchangeCasos";
 public static final String ROUTING_KEY = "CriarCaso";
 @Bean
 DirectExchange usuarioExchange() {
 return new DirectExchange(NOME_EXCHANGE);
 }

 @Bean
 Queue queue() {
 return QueueBuilder.durable(NOME_FILA).build();
 }
 @Bean
 Binding binding() {
 return BindingBuilder.bind(queue()).to(usuarioExchange())
 .with(ROUTING_KEY);
 }
}
```

5. Crie um arquivo de configuração para o Producer:

```
@Configuration
public class ProducerConfig {
 @Value(QueueConfig.NOME_FILA)
 private String message;
```





```
@Bean
public Queue producerQueue() {
 return new Queue(message, true);
}
```

6. Finalmente, crie um controlador novo simples, para fins de testes:

```
@RestController
@RequestMapping("/fila")
public class FilaController {
 @Autowired
 private Producer producer;
 @GetMapping
 public String send(){
 producer.send("ENVIANDO ISSO!");
 return "Enviado.";
 }
}
```

Opcional – Habilitar a interface web de gerenciamento de filas:

1. Abra seu terminal de comandos, como administrador;
2. Navegue até a past "sbin" do diretório de instalação;
  - Neste tutorial, o caminho completo para uma instalação Windows: C:\Program Files\RabbitMQ Server\rabbitmq\_server-3.13.0\sbin
3. Execute o comando, para habilitar a interface web de gerenciamento: "rabbitmq-plugins enable rabbitmq\_management".