



 **VEM SER**
DBC



Spring Data

Aula 4 – Paginação + Postgres

Sumário

- Paginação e Ordenação
- Exercício
- Postgres
- Configuração do banco de dados com Heroku

Paginação

- A paginação costuma ser útil quando temos um grande conjunto de dados e queremos apresentá-lo ao usuário em partes menores.
- Além disso, muitas vezes precisamos classificar esses dados por alguns critérios durante a paginação.

```
public interface ProductRepository extends PagingAndSortingRepository<Product, Integer> {

    List<Product> findAllByPrice(double price, Pageable pageable);

}
```

- Poderíamos ter escolhido estender *JpaRepository*, já que ele estende *PagingAndSortingRepository* também.

Pageable

```
Pageable firstPageWithTwoElements = PageRequest.of(0, 2);

Pageable secondPageWithFiveElements = PageRequest.of(1, 5);
```

```
Pageable sortedByName =
    PageRequest.of(0, 3, Sort.by("name"));

Pageable sortedByPriceDesc =
    PageRequest.of(0, 3, Sort.by("price").descending());

Pageable sortedByPriceDescNameAsc =
    PageRequest.of(0, 5, Sort.by("price").descending().and(Sort.by("name")));
```



show me your code;



Exercício #1

- Criar um controller `PaginacaoController` mapeado para “/paginacao” e com ele faça:
 - uma consulta paginada para contatos ordenado pela descrição
 - uma consulta paginada para endereços ordenado pelo cep
 - uma consulta paginada para endereços filtrada por país

Postgres

- Criar Schema:

```
create schema vem_ser;
```

- Sequence:

```
CREATE SEQUENCE seq_pessoa2  
INCREMENT 1  
START 1;
```

```
INSERT INTO PESSOA  
(ID_PESSOA, NOME, DATA_NASCIMENTO, CPF, EMAIL)  
VALUES(nextval('seq_pessoa2'), 'Maicon Machado Gerardi', TO_DATE('08-09-1991', 'dd-mm-yyyy'),  
'48863250090', 'maicon.gerardi@dbccompany.com.br');
```

Tipo de Dados

- <https://www.postgresql.org/docs/9.5/datatype.html>
- **Texto:** text
- **Números Inteiros:** numeric
- **Decimal:** numeric(x,y) onde x = inteiro, y = casas decimais
- **Datas:** date (somente data) e timestamp (data e hora)



Heroku Com Postgres
show me all;



Dependência

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

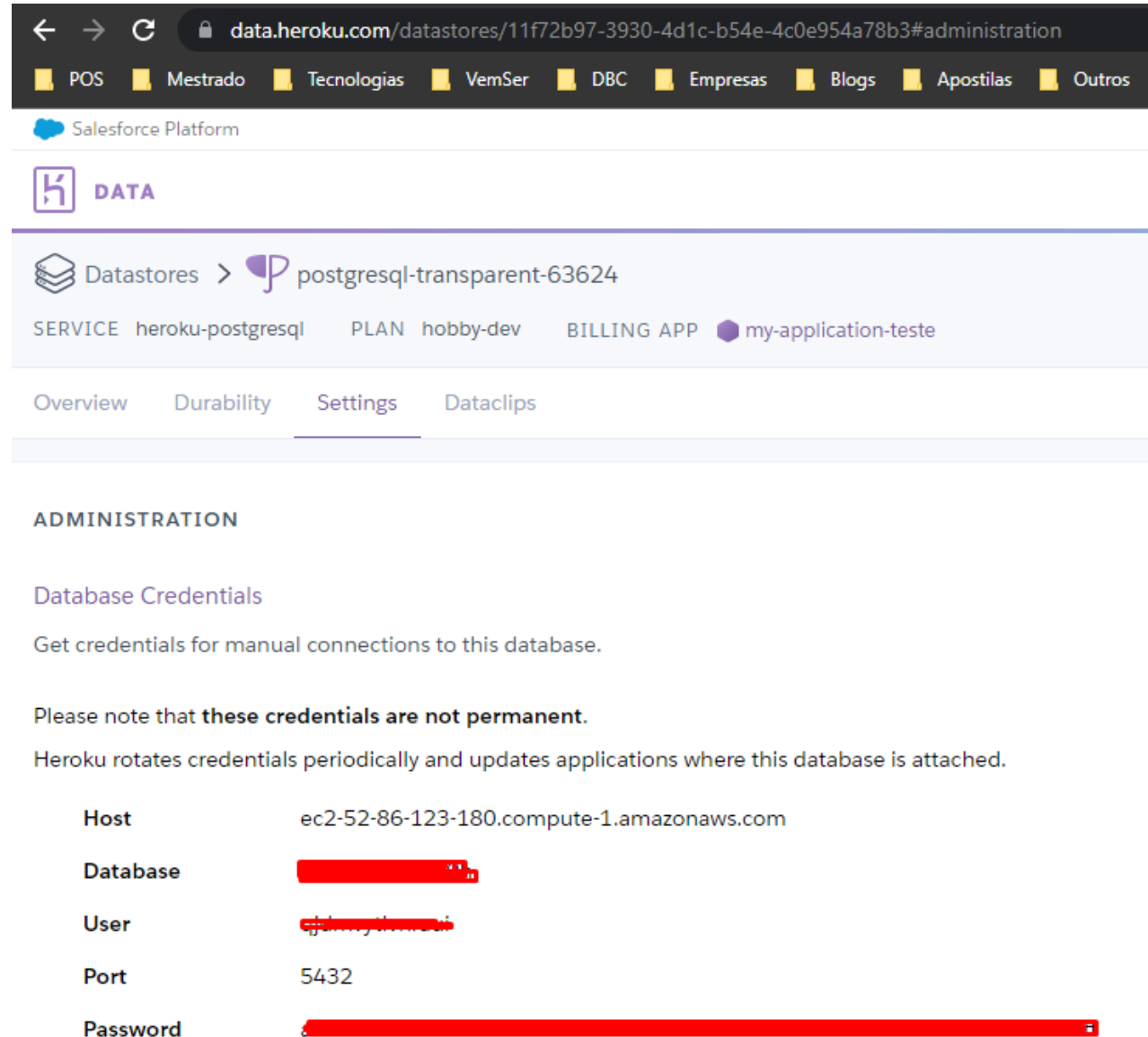
application.properties

```

spring.datasource.url=jdbc:postgresql://HOST:PORTA/DATABASE
spring.datasource.username=user
spring.datasource.password=pass
spring.datasource.hikari.connection-init-sql=set schema 'vem_ser'
spring.datasource.driverClassName=org.postgresql.Driver
spring.jpa.properties.hibernate.default_schema=vem_ser
  
```

JDBC String

- jdbc:postgresql://HOST:PORT/DATABASE



The screenshot shows the Heroku Datastores administration interface. The browser address bar displays the URL: `data.heroku.com/datastores/11f72b97-3930-4d1c-b54e-4c0e954a78b3#administration`. The top navigation bar includes links for POS, Mestrado, Tecnologias, VemSer, DBC, Empresas, Blogs, Apostilas, and Outros. Below this is the Salesforce Platform logo. The main header area shows the 'DATA' icon and the breadcrumb path: Datastores > postgresql-transparent-63624. Below the breadcrumb, there are links for SERVICE (heroku-postgresql), PLAN (hobby-dev), BILLING APP, and my-application-teste. The 'Settings' tab is selected in the sub-navigation bar. The 'ADMINISTRATION' section is active, showing 'Database Credentials'. A note states: 'Please note that **these credentials are not permanent**. Heroku rotates credentials periodically and updates applications where this database is attached.' Below the note is a table of database connection details.

Host	ec2-52-86-123-180.compute-1.amazonaws.com
Database	[REDACTED]
User	[REDACTED]
Port	5432
Password	[REDACTED]

Heroku Command Line

- <https://devcenter.heroku.com/categories/command-line>
- heroku --version
- heroku login
- heroku pg:killall -a <NOME APP>