Aplicação Java Spring Boot para CRUD em PostgreSQL

Configurações do projeto:

1. Dependências Maven (pom.xml):

- Spring Web
- Spring Data JPA
- PostgreSQL Driver
- Lombok (opcional para simplificar o código)

```xml
<dependencies>
    <!-- Spring Web -->
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <!-- Spring Data JPA -->
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>

    <!-- PostgreSQL Driver -->
    <dependency>
        <groupId>org.postgresql</groupId>
```

```
        <artifactId>postgresql</artifactId>
        <scope>runtime</scope>
    </dependency>

    <!-- Lombok (opcional) -->
    <dependency>
        <groupId>org.projectlombok</groupId>
        <artifactId>lombok</artifactId>
        <scope>provided</scope>
    </dependency>
</dependencies>
```

2. Configuração de conexão com PostgreSQL (application.properties):

```properties
spring.datasource.url=jdbc:postgresql://localhost:5432/seu_banco_de_dados
spring.datasource.username=seu_usuario
spring.datasource.password=sua_senha
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
```

1. Camada Model (Produto.java):

```java
package com.example.demo.model;

import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import lombok.Data;

@Data
@Entity
public class Produto {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String nome;
    private String descricao;
    private float preco;
    private String url;
}
```

2. Camada Repository (ProdutoRepository.java):
```java
package com.example.demo.repository;
```

```
import org.springframework.data.jpa.repository.JpaRepository;
import com.example.demo.model.Produto;

public interface ProdutoRepository extends JpaRepository<Produto, Long> {
}
```

3. Camada DTO (ProdutoDTO.java):
```java
package com.example.demo.dto;

import lombok.Data;

@Data
public class ProdutoDTO {
    private Long id;
    private String nome;
    private String descricao;
    private float preco;
    private String url;
}
```

4. Camada Service (ProdutoService.java):
```java
```

```
package com.example.demo.service;

import com.example.demo.dto.ProdutoDTO;
import com.example.demo.model.Produto;
import com.example.demo.repository.ProdutoRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;
import java.util.Optional;

@Service
public class ProdutoService {

    @Autowired
    private ProdutoRepository produtoRepository;

    public List<Produto> listarTodos() {
        return produtoRepository.findAll();
    }

    public Optional<Produto> buscarPorId(Long id) {
        return produtoRepository.findById(id);
    }

    public Produto salvar(ProdutoDTO produtoDTO) {
        Produto produto = new Produto();
```

```
        produto.setNome(produtoDTO.getNome());
        produto.setDescricao(produtoDTO.getDescricao());
        produto.setPreco(produtoDTO.getPreco());
        produto.setUrl(produtoDTO.getUrl());
        return produtoRepository.save(produto);
    }

        public Optional<Produto> atualizar(Long id, ProdutoDTO produtoDTO) {
        return produtoRepository.findById(id).map(produto -> {
            produto.setNome(produtoDTO.getNome());
            produto.setDescricao(produtoDTO.getDescricao());
            produto.setPreco(produtoDTO.getPreco());
            produto.setUrl(produtoDTO.getUrl());
            return produtoRepository.save(produto);
        });
    }

    public void deletar(Long id) {
        produtoRepository.deleteById(id);
    }
}
```

5. Camada Controller (ProdutoController.java):

```java
package com.example.demo.controller;
```

```
import com.example.demo.dto.ProdutoDTO;
import com.example.demo.model.Produto;
import com.example.demo.service.ProdutoService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import java.util.List;
import java.util.Optional;

@RestController
@RequestMapping("/api/produtos")
public class ProdutoController {

    @Autowired
    private ProdutoService produtoService;

    @GetMapping
    public List<Produto> listarTodos() {
        return produtoService.listarTodos();
    }

    @GetMapping("/{id}")
    public ResponseEntity<Produto> buscarPorId(@PathVariable Long
id) {
        Optional<Produto> produto = produtoService.buscarPorId(id);
```

```
        return produto.map(ResponseEntity::ok)
                .orElse(ResponseEntity.notFound().build());
    }

    @PostMapping
            public ResponseEntity<Produto> salvar(@RequestBody
ProdutoDTO produtoDTO) {
        Produto produto = produtoService.salvar(produtoDTO);
        return ResponseEntity.ok(produto);
    }

    @PutMapping("/{id}")
     public ResponseEntity<Produto> atualizar(@PathVariable Long id,
@RequestBody ProdutoDTO produtoDTO) {
                        Optional<Produto> produtoAtualizado =
produtoService.atualizar(id, produtoDTO);
        return produtoAtualizado.map(ResponseEntity::ok)
                .orElse(ResponseEntity.notFound().build());
    }

    @DeleteMapping("/{id}")
    public ResponseEntity<Void> deletar(@PathVariable Long id) {
        produtoService.deletar(id);
        return ResponseEntity.noContent().build();
    }
}
```