

## Atividade em Grupo 24-05-2025

Criar um novo projeto maven usando o Spring com as seguintes dependências:

- Spring Web
- Spring Data JPA
- Postgres
- H2
- Dev Tools
- 

### **application.properties**

Configurar o arquivo de propriedades **application.properties**, adicionar dois perfis um para uso do banco de dados **postgres (production)** e outro para o banco de dados **h2 (developer)** salvando o banco em um arquivo.

### **Classes**

Criar as classes(entidades) abaixo com os seguintes atributos e fazer os mapeamentos e relacionamentos entre classes.

#### **Paciente**

id – Long  
nome – String  
consultas – List<Consulta>

#### **Medico**

id – Long  
nome – String  
consultas – List<Consulta>

#### **Consulta**

id – Long  
dataConsulta – LocalDate  
paciente – Paciente  
medico – Medico  
procedimentos – List<Procedimentos>

#### **Procedimentos**

id – Long  
quantidade – Integer  
valorProcedimento – Double  
consulta -Consulta  
exame – Exame

#### **Exame**

id – Long  
tipo – String  
valor - Double

## POPULAR TABELAS: usar o script abaixo para inserir em um banco no Dbeaver ou PgAdmin

```
create table paciente(id_paciente serial primary key, nome varchar(60));
insert into paciente (nome) values('João');
insert into paciente (nome) values('José');
insert into paciente (nome) values('Maria');
```

```
create table medico(id_medico serial primary key, nome varchar(60));
insert into medico (nome) values('Maria de Fátima');
insert into medico (nome) values('Joana');
```

```
create table exame(id_exame serial primary key , tipo varchar(50), valor float);
insert into exame (tipo, valor) values('Sangue', 50);
insert into exame (tipo, valor) values('Raio X', 100);
insert into exame (tipo, valor) values('Urina',25);
insert into exame (tipo, valor) values('Ultrassom',300);
```

```
create table consulta(id_consulta serial primary key, data_consulta date, id_paciente integer,id_medico integer,
foreign key (id_paciente) REFERENCES paciente (id_paciente),
foreign key (id_medico) REFERENCES medico (id_medico));
```

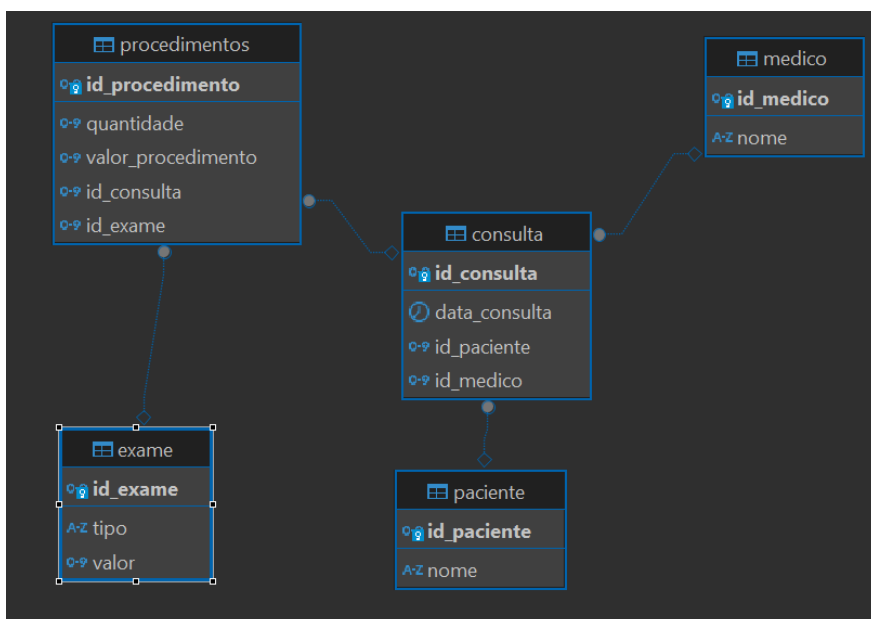
```
insert into consulta (data_consulta, id_paciente, id_medico) values('2021-10-10',1,1);
insert into consulta (data_consulta, id_paciente, id_medico) values('2021-10-11',2,1);
```

```
create table procedimentos(id_procedimento serial primary key, quantidade integer,
valor_procedimento float,
id_consulta integer, foreign key (id_consulta) references consulta(id_consulta),
id_exame integer, foreign key (id_exame) references exame(id_exame));
```

```
insert into procedimentos (quantidade,valor_procedimento,id_consulta,id_exame) values(1,50,1,1);
insert into procedimentos (quantidade,valor_procedimento,id_consulta,id_exame) values(1,100,1,2);
insert into procedimentos (quantidade,valor_procedimento,id_consulta,id_exame) values(2,250,1,4);
```

```
insert into procedimentos (quantidade,valor_procedimento,id_consulta,id_exame) values(2,50,2,1);
insert into procedimentos (quantidade,valor_procedimento,id_consulta,id_exame) values(1,100,2,2);
insert into procedimentos (quantidade,valor_procedimento,id_consulta,id_exame) values(1,200,2,3);
```

## Diagrama MER



## Repository

Somente para classe de Consulta

## Service

Somente para classe de Consulta

Criar os métodos para inserir, buscar e listar as consultas

**Obs: Faça o uso da programação funcional onde for possível**

## Controller

Somente para classe de Consulta

## DTO

**ConsultaDTO** – uma consulta possui uma lista de procedimentos

dataConsulta

nomePaciente

procedimentosDTO

totalGeral

## ProcedimentoDTO

tipoExame

quantidade

valorProcedimento

subtotal

## Postman

Fazer um get no endpoint para buscar uma determinada consulta e retorna seus procedimentos.

