Colunas do tipo **Date/DateTime/LocalDateTime** já são mapeadas automaticamente pelo **JPA**, então ele já sabe que esse **atributo irá ser convertido em uma coluna desse tipo** no banco de dados:

```
private LocalDate dataCadastro = LocalDate.now();
```

Atributos do tipo **Enum** por **padrão são gravados no banco de dados como do tipo inteiro** onde o valor **é A POSIÇÃO DA CONSTANTE** no **enum**. Visto os **problemas que isso pode causa**r, é interessante especificar esse **Enum:**

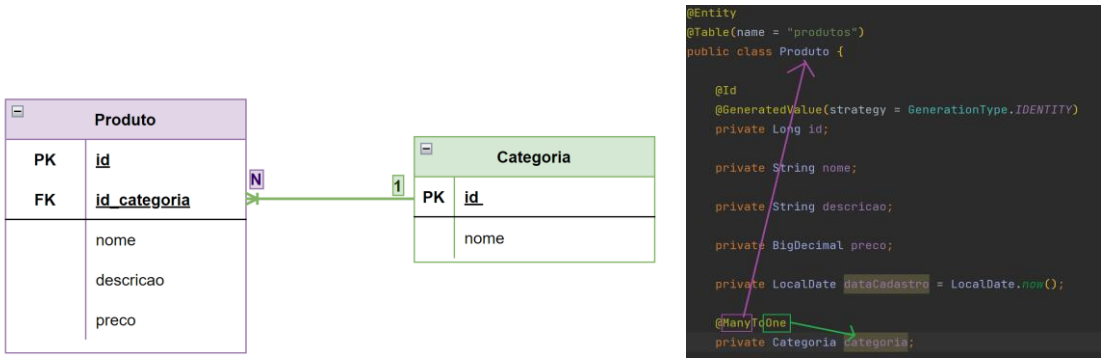
```
@Enumerated(EnumType.STRING)
private Categoria categoria;
```

```
CadastroProduto
fev. 13, 2022 2:05:25 PM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsola
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.inter
Hibernate: create table produtos (id bigint generated by default as identity, categoria varchar(255),
```

- @Enumerated: diz ao **JPA de maneira explicita** que o atributo é um **Enum** podendo ser configurado com o valor que o **enum** deve usar para gravar no banco, ***String ou numérico***.

Enums são pouco flexíveis, então dependendo do caso é melhor usar um Objeto *(isso não exclui o uso do enum em casos que ele é necessário, vale pesar a situação e decidir o que usar).*

Então eu prefiro usar um **relacionamento de entidades nesse caso**, quando usamos uma classe que é uma entidade como atributo de outra entidade, o **JPA** percebe isso e **nos obriga a especificar qual a cardinalidade desse relacionamento,** um Produto tem uma cardinalidade de ***muitos para um*** com **Categoria**, e uma **Categoria** tem a cardinalidade de ***um para muitos*** com Produto:



- @ManyToOne: diz ao JPA que o r**elacionamento entre as entidades são de muitas para uma**, vale ressaltar que apenas a entidade Produto conhece esse relacionamento, portanto, ela é **unidirecional**.

Para salvar **uma entidade** que se relaciona com **outra**, **ambas precisam estar persistidas no banco de dados** antes:

```java
Categoria celulares = new Categoria( nome: "Celulares");
Produto celular = new Produto( nome: "Nokia", descricao: "Velho",new BigDecimal( val: 10.000), celulares);

EntityManager entityManager = JPAUtil.criarEntityManager();
ProdutoDao produtoDao = new ProdutoDao(entityManager);
CategoriaDao categoriaDao = new CategoriaDao(entityManager);

entityManager.getTransaction().begin();

categoriaDao.cadastrar(celulares);
produtoDao.cadastrar(celular);

entityManager.getTransaction().commit();
entityManager.close();
```

```
Hibernate: create table produtos (id bigint generated by default as identity, dataCadastro date, descricao varchar(255), n
Hibernate: alter table produtos add constraint FK8rqw0ljwdaom34jr2t46bjtrn foreign key (categoria_id) references categoria
fev. 13, 2022 2:56:36 PM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
Hibernate: insert into categorias (id, nome) values (null, ?)
Hibernate: insert into produtos (id, categoria_id, dataCadastro, descricao, nome, preco) values (null, ?, ?, ?, ?, ?)
```