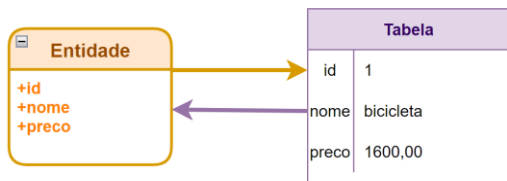




Para que o **JPA entenda como trabalhar com o banco de dados** usando as **classes de JAVA** precisamos fazer o **mapeamento** das **mesmas**. A ideia do **JPA** sempre foi resolver esse problema de **banco de dados relacional** e orientação a objetos:



Esse mapeamento é feito através de **anotações** nas **classes Java** e nos atributos dela, as anotações fazem qualquer tipo de mapeamento, **seja relacional, do ID, constraints de colunas, nomes e etc...**

```
@Entity
@Table(name = "produtos")
public class Produto {

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

- **@Entity**: diz para o JPA que essa **classe representa uma tabela no banco de dados**.
- **@Table**: permite **especificar o nome da tabela no banco de dados**, útil para tabela com nome diferente da classe.
- **@Id**: diz qual **atributo da classe é o ID** na tabela do banco de dados
- **@GeneratedValue**: diz para o JPA que **a geração do ID é responsabilidade do banco de dados** além de permitir **especificar a estratégia de geração do ID**, isso depende do banco, pode ser *SEQUENCE*, *IDENTITY* e etc

****Pelo JPA é necessário adicionar no xml de persistence a tag class com o nome completo da classe para que ele possa mapear, o Hibernate faz isso sozinho, mas talvez outras implementações não façam****