Abaixo, teremos as versões finais de nossas entidades, já com seus relacionamentos devidamente mapeados:

Fornecedor.java

```
package entidades;

import java.util.Set;
import javax.persistence.CascadeType;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.Id;
import javax.persistence.ManyToMany;
import javax.persistence.Table;

@Entity
@Table(name = "Fornecedor")
public class Fornecedor {
	@Id
	private String cnpj;
	private String nomeFactasia;
	private String endereco;
	private String telefone;

	// Uma fornecedor fornece muitos produtos
	// A melhor forma de representar é com um implementação da interface Set
	// CascadeType.ALL indica que qualquer operação sobre um fornecedor de
	// propaga para os objetos referenciados,
	// por exemplo, se um fornecedor for excluído, todos os produtos desse
	// fornecedor também serão
	// FetchType.LAZY indica que o carregamento dos produtos é "atrasada",
	// quando um fornecedor for carregado
	// seus produtos somente serão carregados casso seja necessário, isto é,
	// sob demanda
	@ManyToMany(cascade = CascadeType.ALL, fetch = FetchType.LAZY)
	private Set<Produto> produtos;

	public String getNomeFactasia() {
		return nomeFactasia;
	}

	public void setNomeFactasia(String nomeFactasia) {
		this.nomeFactasia = nomeFactasia;
	}

	public String getEndereco() {
		return endereco;
	}

	public void setEndereco(String endereco) {
		this.endereco = endereco;
	}

	public String getTelefone() {
		return telefone;
	}
```

```
	public void setTelefone(String telefone) {
		this.telefone = telefone;
	}

	public String getCnpj() {
		return cnpj;
	}

	public void setCnpj(String cnpj) {
		this.cnpj = cnpj;
	}

	public Set<Produto> getProdutos() {
		return produtos;
	}

	public void setProdutos(Set<Produto> produtos) {
		this.produtos = produtos;
	}


}
```

Fabricante.java

```
package entidades;

import java.util.Set;
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.Id;
import javax.persistence.OneToMany;
import javax.persistence.Table;

@Entity
@Table(name = "Fabricante")
public class Fabricante {

	@Id
	private String cnpj;

	@Column(length = 30, name = "nome", nullable = false)
	private String nomeFactasia;

	@Column(length = 40, name = "endereco", nullable = false)
	private String endereco;

	@Column(length = 10, name = "descricao_produto", nullable = false)
	private String telefone;

	// Um fabricante fabrica muitos produtos
	// A melhor forma de representar é com um implementação da interface Set
	// CascadeType.PERSIST inca que a propagação ocorrerá somente na
```

```
	// persistência, com isso se existir um produto não persistido associado ao
	// fabricante que está sendo persistido ele também será persistido
	// FetchType.LAZY indica que o carregamento dos produtos é "atrazada",
	// quando um fabricante for carregado
	// seus produtos somente serão carregados casso seja preciso, isto é,
	// sob demanda
	// mappedBy="fabricante" indica que na Entidade Produto, existe um
	// atributo chamado fabricante que faz
	// referência a seu fabricante
	@OneToMany(cascade = CascadeType.PERSIST, fetch = FetchType.LAZY,
mappedBy = "fabricante")
	private Set<Produto> produtos;

	public String getCnpj() {
		return cnpj;
	}

	public void setCnpj(String cnpj) {
		this.cnpj = cnpj;
	}

	public String getNomeFactasia() {
		return nomeFactasia;
	}

	public void setNomeFactasia(String nomeFactasia) {
		this.nomeFactasia = nomeFactasia;
	}

	public String getEndereco() {
		return endereco;
	}

	public void setEndereco(String endereco) {
		this.endereco = endereco;
	}

	public String getTelefone() {
		return telefone;
	}

	public void setTelefone(String telefone) {
		this.telefone = telefone;
	}

	public Set<Produto> getProdutos() {
		return produtos;
	}

	public void setProdutos(Set<Produto> produtos) {
		this.produtos = produtos;
	}


}
```

Produto.Java

```java
package entidades;

import java.util.Date;
import java.util.Set;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.ManyToMany;
import javax.persistence.ManyToOne;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;

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

	@Id
	@GeneratedValue(strategy = GenerationType.AUTO)
	private Integer codigo;

	@Column(length = 20, name = "descricao_produto", nullable = false)
	private String descricao;

	@Column(scale = 8, precision = 2)
	private double valor;

	@Temporal(TemporalType.DATE)
	private Date dataCadastro;

	// Um produto possui muitos fornecedores
	// A melhor forma de representar é com um implementação da interface Set
	// FetchType.EAGER indica que o carregamento dos fornecedores é ocorrerá
	// juntamente com o carregamento do produto
	// mappedBy="produtos" indica que na Entidade Fornecedor, existe um
	// atributo chamado produtos que faz referência a seus produtos
	@ManyToMany(fetch = FetchType.EAGER, mappedBy = "produtos")
	private Set<Fornecedor> fornecedores;

	// Um produto possui um fabricante
	// A melhor forma de representar é com um implementação da interface Set
	// FetchType.EAGER indica que o carregamento dos fornecedores é ocorrerá
	// juntamente com o carregamento do produto
	// Para o relacionamento @ManyToOne não existe o elemento mappedBy
	@ManyToOne(fetch = FetchType.EAGER)
	private Fabricante fabricante;

	public Integer getCodigo() {
		return codigo;
	}
```

```
	public void setCodigo(Integer codigo) {
		this.codigo = codigo;
	}

	public String getDescricao() {
		return descricao;
	}

	public void setDescricao(String descricao) {
		this.descricao = descricao;
	}

	public double getValor() {
		return valor;
	}

	public void setValor(double valor) {
		this.valor = valor;
	}

	public Date getDataCadastro() {
		return dataCadastro;
	}

	public void setDataCadastro(Date dataCadastro) {
		this.dataCadastro = dataCadastro;
	}

	public Set<Fornecedor> getFornecedores() {
		return fornecedores;
	}

	public void setFornecedores(Set<Fornecedor> fornecedores) {
		this.fornecedores = fornecedores;
	}

	public Fabricante getFabricante() {
		return fabricante;
	}

	public void setFabricante(Fabricante fabricante) {
		this.fabricante = fabricante;
	}



}
```