

**FIAP GRADUAÇÃO**

# DIGITAL BUSINESS ENABLEMENT

Prof. Me. Thiago T. I. Yamamoto

#06 – RELACIONAMENTOS E ENUM



thiagoyama



thiagoyama@gmail.com

## #06 – RELACIONAMENTOS E ENUM

---

- Relacionamento
- Tags para select
- Enum

# RELACIONAMENTO MANY-TO-ONE COM SELECT

Vamos trabalhar com duas entidades com o relacionamento um para muitos, o nosso exemplo serão as classes **Produto** e **Categoria**.

```
@Entity
@SequenceGenerator(name="seqProduto", sequenceName="SEQ_PRODUTO", allocationSize=1)
public class Produto {

    @Id
    @GeneratedValue(strategy=GenerationType.SEQUENCE,generator="seqProduto")
    private int codigo;

    @ManyToOne
    private Categoria categoria;

    //...

}
```

```
@Entity
@SequenceGenerator(name="seqCategoria", sequenceName="SEQ_PRODUTO", allocationSize=1)
public class Categoria {

    @Id
    @GeneratedValue(strategy=GenerationType.SEQUENCE,generator="seqCategoria")
    private int codigo;

    private String descricao;

    //..

}
```

- Vamos utilizar as tags `<form:select>` e `<form:options>` para listar as categorias cadastradas no formulário de cadastro de um produto;

```
<div class="form-group">
  <form:label path="categoria">Categoria</form:label>
  <form:select path="categoria.codigo" cssClass="form-control">
    <form:options items="${categorias}" itemLabel="descricao" itemValue="codigo"/>
  </form:select>
  <form:errors path="categoria"/>
</div>
```

- **items** – lista de categorias;
- **itemLabel** – atributo da categoria que será exibida para o usuário;
- **itemValue** – atributo da categoria que será enviado para o servidor;

- Agora o controller precisa enviar a lista de categorias para preencher as opções do select:

## Controller

```
@GetMapping("cadastrar")  
public ModelAndView abrirForm(Produto produto){  
    ModelAndView retorno = new ModelAndView("produto/cadastro");  
    retorno.addObject("categorias", categoriaDao.list());  
    return retorno;  
}
```

## JSP



```
<form:options items="${categorias}" itemLabel="descricao" itemValue="codigo"/>
```

# SPRING MVC ENUM



- Vamos adicionar um enum na classe Produto para determinar o tipo de produto;
- Note que o enum possui um atributo e um construtor que recebe um valor para o *label*. Esse atributo será utilizado para ser exibido para o usuário.

```
public enum TipoProduto {  
    NOVO("Novo"), USADO("Usado");  
  
    private final String label;  
  
    private TipoProduto(String label) {  
        this.label = label;  
    }  
  
    public String getLabel() {  
        return label;  
    }  
  
}
```

- Vamos adicionar um atributo do tipo do enum na classe produto:

```
@Entity
@SequenceGenerator(name="seqProduto", sequenceName="SEQ_PRODUTO", allocationSize=1)
public class Produto {

    //...

    private TipoProduto tipo;

    //...

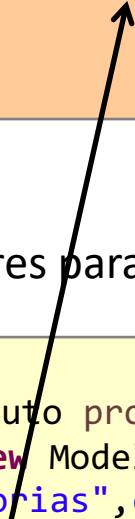
}
```

- Vamos utilizar as mesmas tags para exibir as opções do enum na tela, note que não é necessário definir o itemValue:

```
<div class="form-group">
    <form:label path="tipo">Tipo Produto</form:label>
    <form:select path="tipo" cssClass="form-control">
        <form:options items="${tipoProdutos}" itemLabel="label"/>
    </form:select>
</div>
```

- No controller é preciso enviar os valores para o select:

```
@GetMapping("cadastrar")
public ModelAndView abrirForm(Produto produto){
    ModelAndView retorno = new ModelAndView("produto/cadastro");
    retorno.addObject("categorias", categoriaDao.list());
    retorno.addObject("tipoProdutos", TipoProduto.values());
    return retorno;
}
```



**Copyright © 2017 - 2018 - Prof. Me. Thiago T. I. Yamamoto**

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).

*“Há cinco degraus para se alcançar a sabedoria: Calar, ouvir, lembrar, sair, estudar.”*