

Relacionamentos do tipo EAGER

Transcrição

Para finalizar, na classe `Livro`, tivemos um problema no primeiro módulo do treinamento JSF, e para resolver esse problema fizemos um mapeamento que não é o ideal:

```
@Entity
public class Livro implements Serializable {

    @ManyToMany(fetch = FetchType.EAGER)
    private List<Autor> autores = new ArrayList<Autor>();

    // restante do código
}
```

Muitos livros podem ter muitos autores, por isso temos o relacionamento *many-to-many* e o tipo *EAGER* significa que quando um livro é carregado, automaticamente são carregados os autores.

Normalmente esse tipo de relacionamento é uma má prática, pois nem sempre quando carregamos um livro, queremos carregar os autores, e isso pode gerar um custo de memória muito grande, em um projeto maior, por exemplo.

Vamos voltar o relacionamento para *LAZY*, que já é o padrão do mesmo:

```
@Entity
public class Livro implements Serializable {

    @ManyToMany
    private List<Autor> autores = new ArrayList<Autor>();

    // restante do código
}
```

Isso significa que ao carregar o livro, não serão carregados automaticamente os autores. O problema que isso gerará para nós é que, ao tentar carregar um livro para alterá-lo, seus autores não estarão carregados, ou seja, não conseguiremos exibi-los, por isso receberemos um `LazyInitializationException`.

Pare resolver isso, temos que alterar o método `carregar` da classe `LivroBean`, pois é esse método que é executado quando carregamos um livro. Atualmente ele está assim:

```
public void carregar(Livro livro) {
    System.out.println("Carregando livro");
    this.livro = livro;
}
```

Ou seja, colocamos o livro da tabela no nosso formulário, e é esse livro da tabela não terá os autores carregados. Por isso vamos carregar explicitamente o livro da tabela, através de uma nova conexão, utilizando o `LivroDao`. Nessa nova

conexão, carregando o livro novamente, o JPA conseguirá carregar os autores:

```
public void carregar(Livro livro) {  
    System.out.println("Carregando livro");  
    this.livro = this.livroDao.buscaPorId(livro.getId());  
}
```

Podemos testar agora, e tentar alterar um livro. Repare agora que os seus autores são carregados, e nenhuma exceção é lançada no console! Problema resolvido.