

ATIVIDADE INDIVIDUAL

Cap 4 - Traduzindo objeto para o modelo relacional - Mãos a obra

 DE 06/02/2023 A 27/02/2023

PRAZO ENCERRADO



ENTREGAR ATIVIDADE

Atividades entregues **até 3 dias após o prazo** receberão até **70% da nota**.
O cálculo é feito automaticamente pelo sistema, o professor não tem controle sobre o percentual da nota atribuída.

INTRODUÇÃO



Atividade – Mãos à obra!

Faça o upload de sua atividade na plataforma FIAP ON, na seção de atividades e aguarde a nota e o feedback do professor.

Atenção: Verifique se o arquivo do upload está correto, não é possível enviar um outro arquivo após fechamento da entrega na plataforma ou correção do professor.

Até agora vimos o conceito de ORM e alguns detalhes sobre sua realização. Ficamos a par também das informações iniciais sobre o JPA e Hibernate, que são as ferramentas mais usadas em Java para o Mapeamento Objeto-Relacional. Agora chegou a hora de praticar!

Indicamos a criação de um novo projeto com JPA/Hibernate utilizando o Maven. Conforme visto neste capítulo, configure o persistence.xml para utilizar um banco de dados e crie a classe de ORM Estabelecimento. Sinta-se livre para usar o projeto-base disponibilizado no capítulo.

O código-fonte abaixo apresenta um exemplo de classe para o banco de dados Oracle. Caso utilize outro banco, não esqueça de ajustar a criação automática de índices (sequence), alguns bancos de dados relacionais não utilizam esse conceito.

```
package br.com.fiap.smartcities.domain;
```

```
import java.util.Calendar;
```

```
import javax.persistence.Column;
```

```
import javax.persistence.Entity;
```

```
import javax.persistence.GeneratedValue;
```


[← VOLTAR À LISTA](#)

George F. R. Santos – Classes do teste

Fonte: Elaborado pelo autor (2021)

Certifique-se de que o parâmetro do método `createEntityManagerFactory` (“smartcities-orm”) é o mesmo que está configurado no arquivo `persistence.xml`, o nome da `persistence unit`. Depois, é só executar! Como o arquivo `persistence.xml` está configurado para criar as tabelas e exibir os SQL enviados para o banco, o resultado esperado é a criação da tabela, que pode ser verificada no console do eclipse.

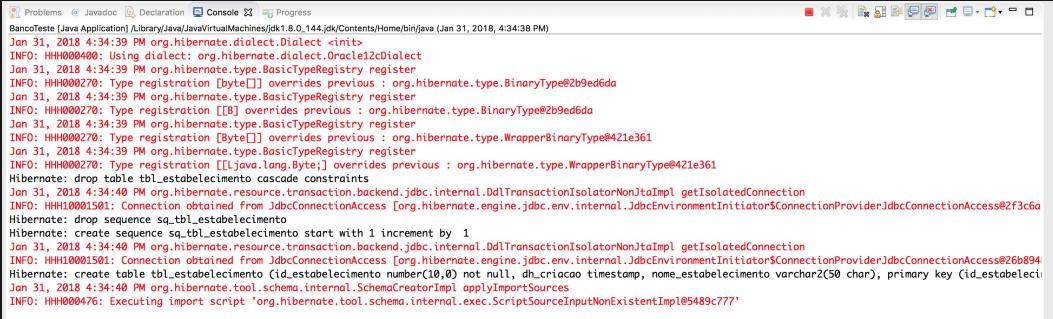


Figura 1 – Resultado da execução do programa

Fonte: Elaborado pelo autor (2021)

Observe que, primeiramente, o programa envia os SQL para apagar a tabela e a sequência, comportamento-padrão para a configuração “create”. Não se preocupe se ocorrer um erro neste ponto, porque possivelmente você não tem a tabela e a sequência no banco para serem apagadas. Depois, consegue-se ver as queries que criam a tabela e a sequência.

Fique à vontade para entrar no banco de dados verificar a tabela criada. Tente também modificar a classe de ORM, adicionando novos atributos. Crie uma classe e execute o programa. Veja que o banco de dados será modificado!

Sucesso! Já temos a entidade de estabelecimento em nosso sistema e também em nosso banco de dados. Agora, para completar a entrega, crie mais **duas entidades** (classe domain do Java) de um sistema imaginário (Ex.: para um sistema de controle de alunos, poderíamos ter as entidades de aluno, disciplinas, professores, tutores e outros). Execute a classe **main** e acompanhe a criação das tabelas no banco de dados.

O próximo passo é realizar as operações básicas do CRUD, que será tema do próximo capítulo!



Envio

ENTREGA PENDENTE



Avaliação

AVALIAÇÃO PENDENTE

PRAZO DE ENTREGA

Segunda-feira, 27 de Fevereiro de 2023, às 23h59

TEMPO RESTANTE

⚠ A tarefa está atrasada 259 dias e 23 horas

ÚLTIMA MODIFICAÇÃO

-

