

# ECM251 - Linguagens de Programação I

15 - Aplicação de Padrões de Projeto

Prof. Murilo Zanini de Carvalho

Prof. Tiago Sanches da Silva

# Mapa da Aula

1. Finalizar aplicações de CRUD no banco da aula anterior.
2. Entender o conceito de padrões de projeto.
3. Refatorar o projeto de CRUD no banco.
4. Fazer os exercícios.

Retirado de  
(<https://i.pinimg.com/originals/f9/8b/3f/f98b3f924902009a4f6686102c122af4.jpg>), em 08/09/2019



# Relembrar é viver: CRUD



CREATE

C



READ

R



UPDATE

U

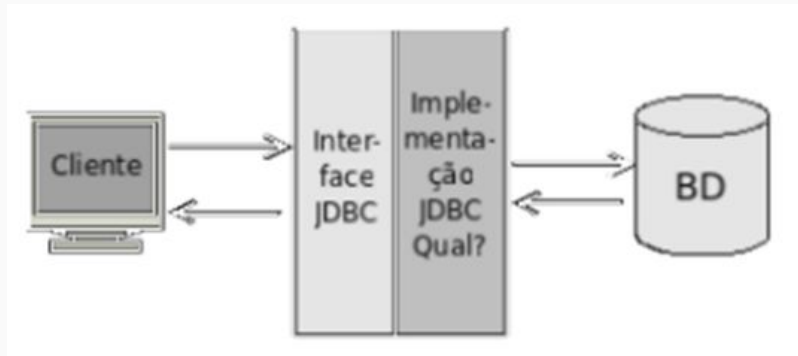


DELETE

D

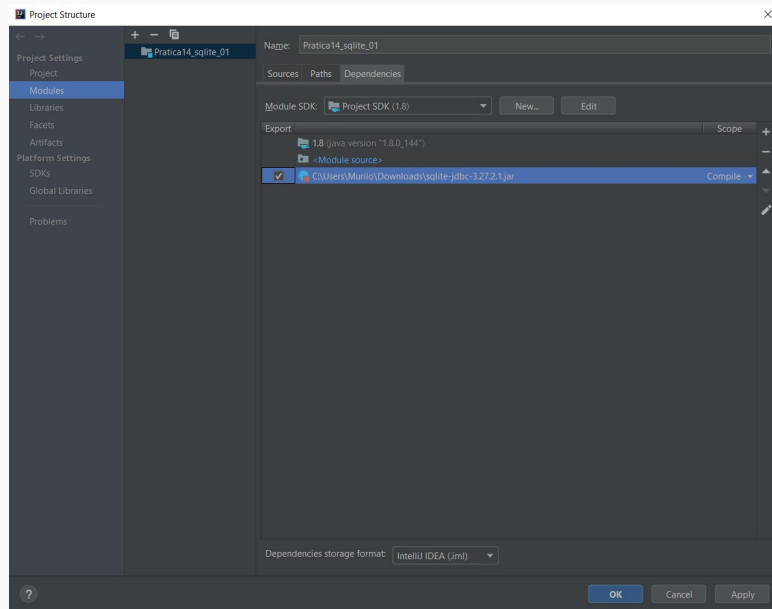
# Relembrar é viver: Driver JDBC

Esse conjunto de classes concretas é quem fará a ponte entre o código cliente que usa a API JDBC e o banco de dados. São essas classes que sabem se comunicar através do protocolo proprietário do banco de dados. Esse conjunto de classes recebe o nome de driver.



# Relembrar é viver: Como adicionar um JAR no projeto

Em “Project Structure”, dentro de “Modules”, é possível adicionar os arquivos JAR do JDBC.



# Retomando projeto anterior

Operações realizadas até o momento:

- Inserir usuários.
- Pegar todos os usuários do banco.
- Interface mínima para usar o sistema.

# Operação de Update

Vamos fazer o update pela chave primária do usuário, contudo, na interface, vamos procurar pelo nome do usuário, para atualizar o nome e o e-mail.

# Operação de Deletar

Igual a operação de atualizar, mudando a QUERY SQL que será realizada para retirar o usuário do banco apenas.



# Design Patterns

Design Patterns são uma forma de organizar os códigos produzidos que não só facilita sua implementação como permite utilizar um modelo “padronizado” de solução para um dado problema.

Foram propostos originalmente para resolver problemas de arquitetura e ficaram conhecidos no desenvolvimento de software quando o livro “Padrões de Projeto – Soluções Reutilizáveis de Software Orientado a Objetos” foi lançado. Seus autores são conhecidos como GoF - *Gang of Four*.

# Design Patterns

<u><i>Criação</i></u>	<u><i>Estrutural</i></u>	<u><i>Comportamental</i></u>	
Abstract Factory	Adapter	Chain of Responsibility	Observer
Builder	Facade	Command	State
Factory Method	Bridge	Interpreter	Strategy
Prototype	Decorator	Iterator	Template Method
Singleton	Flyweight	Mediator	Visitor
	Composite	Memento	
	Proxy		

Retirado de  
([https://www.devmedia.com.br/imagens/articles/169202/estrutura\\_de\\_sign\\_pattern.png](https://www.devmedia.com.br/imagens/articles/169202/estrutura_de_sign_pattern.png))  
, em 08/09/2019

# Design Patterns

Para saber mais:

- <https://www.opus-software.com.br/design-patterns/>
- <https://www.devmedia.com.br/introducao-design-pattern/18838>
- <https://refactoring.guru/design-patterns>
- <https://refactoring.guru/design-patterns/java>

# Design Patterns

Pattern que já estamos utilizando:

- DAO - *Data Access Object*  
(<https://www.profissionaisti.com.br/2016/12/design-pattern-criando-uma-classe-dao-generica/>)
- MVC - Model View Controller (quando trabalhamos com interfaces gráficas).

# Design Patterns

Pattern que VAMOS utilizar:

- Singleton (<https://refactoring.guru/design-patterns/singleton> ).
  - Possui vantagens e desvantagens, discutir primeiro.
  - Onde o padrão pode ser aplicado?

# Refatorar!

Ajustar o projeto desenvolvido até aqui para que ele implemente o pattern que discutimos.

Aproveite e verifique se existe mais alguma oportunidade de refatorar o código.

Retirado de  
(<https://res.infoq.com/presentations/refatorar-codigo/pt/slides/sl1.jpg>), em 08/09/2019

## Refatorar!

Porque ninguém gosta de código que cheira mal

# Exercício - 1

Seu trabalho será construir um banco de dados que permita que o pesquisador armazene o nome, onde ele encontrou (lat e lon), a altura e o peso (esse pesquisador é famoso por definir isso apenas com um olhar) e o tipo do animal avistado.

Como, por questões de Marketing, o tipo do pok... Animal muda a todo momento, seu programa deve permitir que ele atualize os dados inseridos.

## Exercício - 2

Supondo que você chegou até o exercício 2, logo sua implementação do exercício 1 foi um sucesso! Agora é necessário limitar o acesso a esses dados todos. Para isso, crie uma nova tabela no seu banco de dados com os usuários cadastrados.

O seu programa agora deve pedir para que os usuários se identifiquem primeiro antes de entrar no sistema.