

Persistindo dados com **Java** + **PostgreSQL**





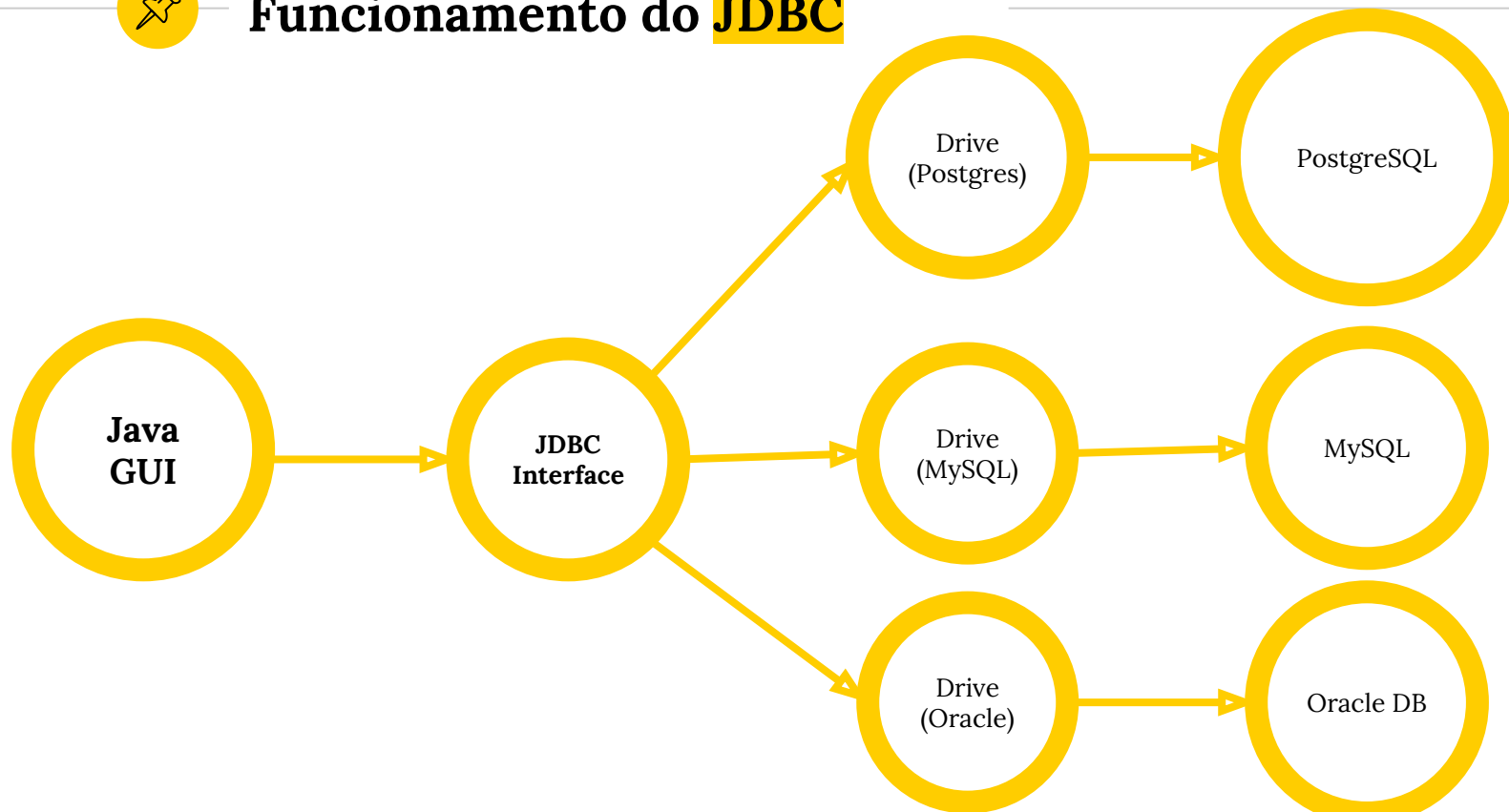
Conceitos

O processo de armazenamento de dados é também chamado de **persistência**.

A biblioteca de persistência em banco de dados relacionais do Java é chamada **JDBC** (Java Database Connectivity).

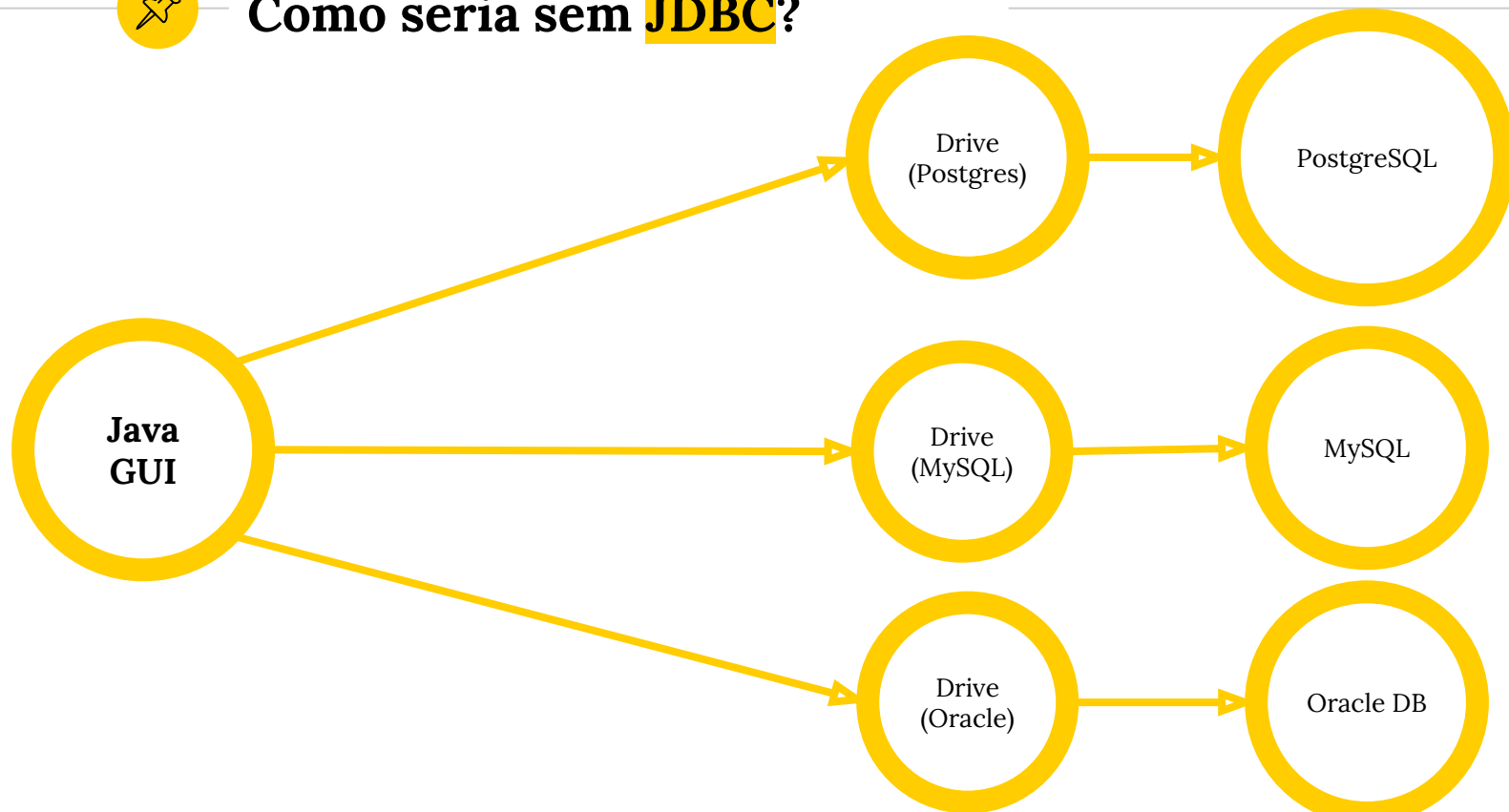


Funcionamento do JDBC



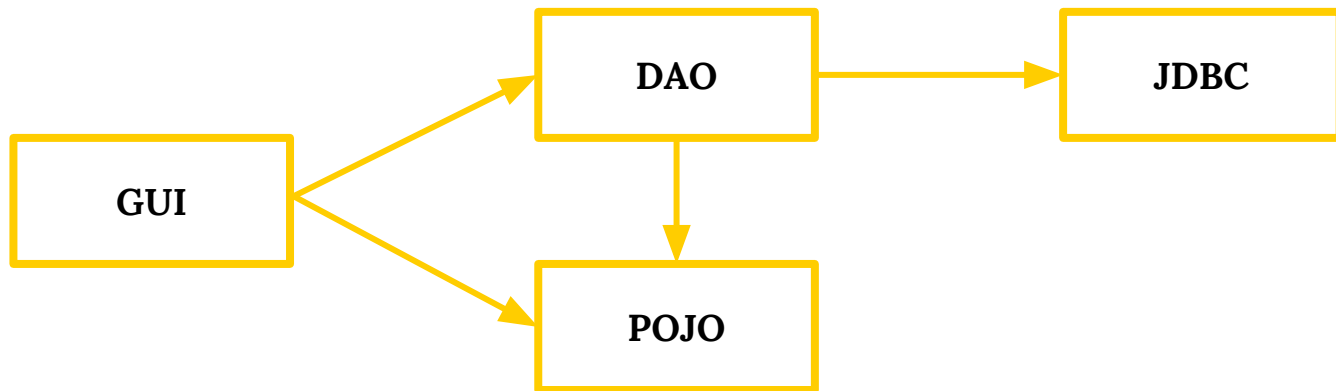


Como seria sem JDBC?

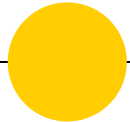




Estrutura do projeto



DAO (*Data Access Object*)

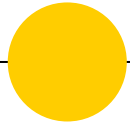




Por que usar **DAOs**?

A ideia é **remover o código de acesso ao banco de dados de suas classes da lógica** e colocá-lo em uma classe responsável pelo acesso aos dados. Assim o código de acesso ao banco de dados fica em um lugar só, tornando **mais fácil a manutenção**.

POJO (*Plain Old Java Objects*)







O que é **POJO**?

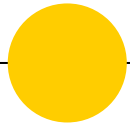
É simplesmente uma denominação que se dá para um **objeto "normal", sem nada de especial**. Ele tem seus campos, seus métodos, seus construtores, etc, mas não segue [necessariamente] nenhum padrão pré-estabelecido.



Exemplos de **POJO**

- Usuário 
- Livro 
- Usuario-has-livro

Prática





O que é necessário ter instalado?

- JDK
- Eclipse IDE
- PostgreSQL



Criando o projeto

- Eclipse > New > Java Project
- Preencha o nome do projeto
- Finish



Baixando o drive PostgreSQL JDBC

<https://jdbc.postgresql.org/download.html>



Adicionando o drive ao projeto

- Projeto criado > Botão direito > Properties > Java Build Path > Add External JARs > Seleciona o driver > OK

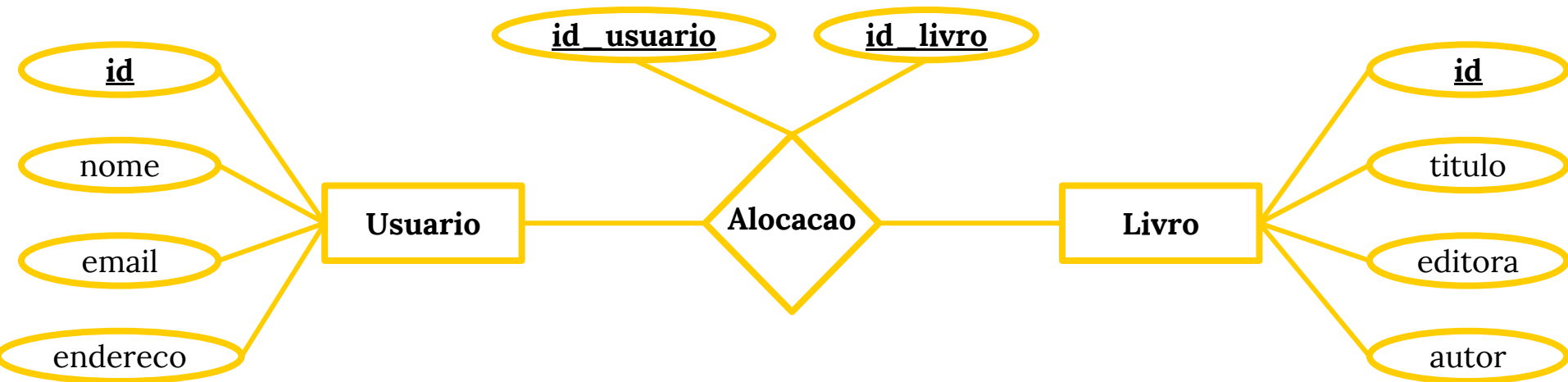


Projeto simples de uma biblioteca

Uma bibliotecária precisa de um sistema simples de locação de livros, onde é preciso apenas cadastrar um **usuário** (com **nome**, **email** e **endereço**), um **livro** (com **título**, **editora** e **autor**) e a relação de **alocação**.



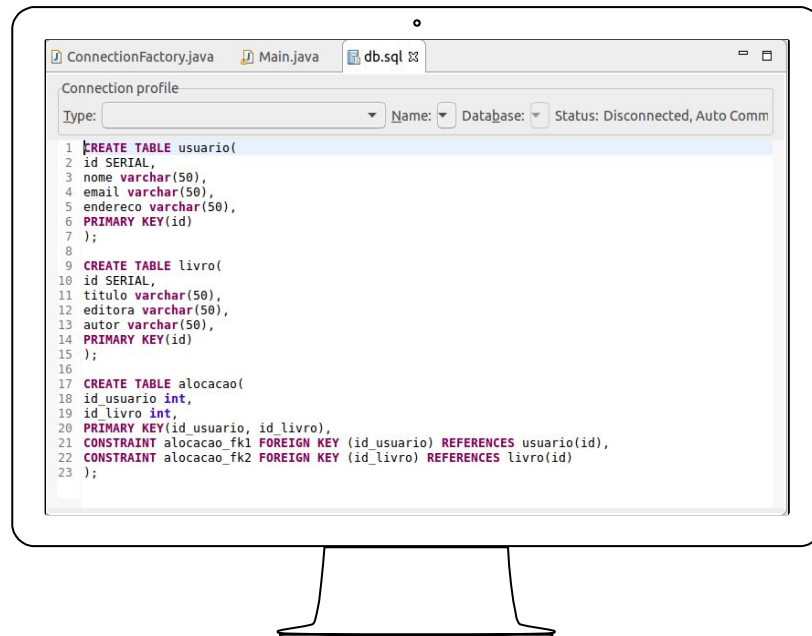
Modelo Entidade Relacionamento





Banco de dados

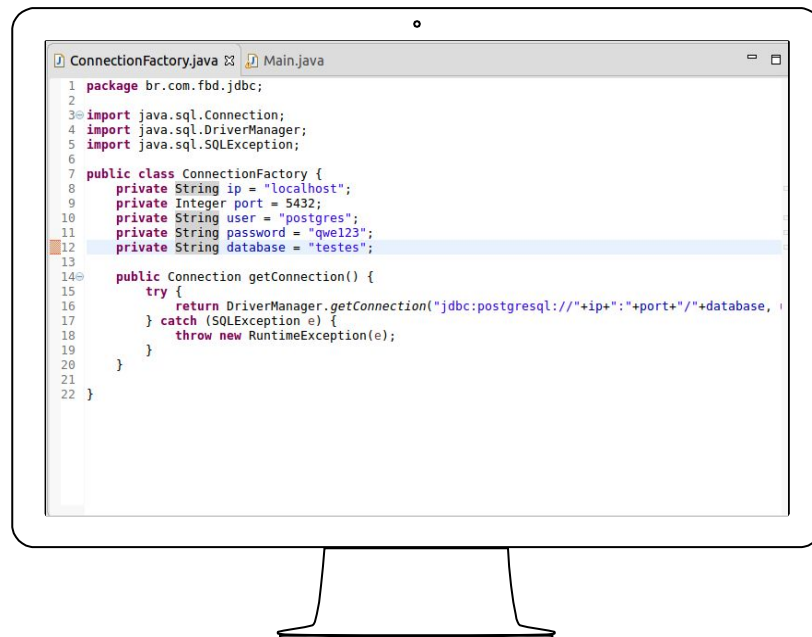
- Duas tabelas que guardam **POJOs**
- Uma tabela com as relações **usuário** – **livro**





Classe **ConnectionFactory**

- Fábrica de conexões
- Responsável por abrir a conexão com o banco de dados





Classes

- Pacote **POJO**

- Usuario
- Livro

- Pacote **DAO**

- UsuarioDAO
- LivroDAO
- AlocacaoDAO

- Pacote **JDBC**

- ConnecionFactory



Dúvidas?



Referências

<https://www.caelum.com.br/apostila-java-web/bancos-de-dados-e-jdbc/#2-3-a-conexao-em-java>

<http://www.devmedia.com.br/manipulando-dados-com-jdbc-em-java/27287>