Linguagem

De

Programação

**Aluno :** Alex Fonseca Ramos

**Curso :** Técnico em Informática

**Turno :** Manhã

**Profº :** Sérgio Cruz

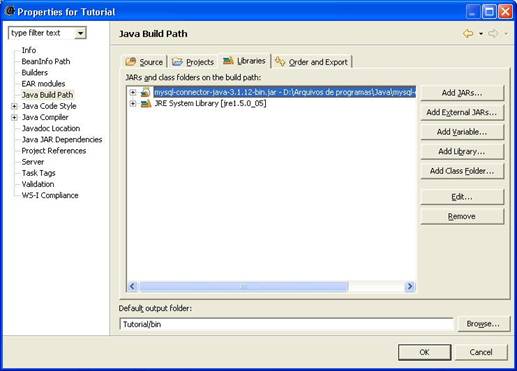
**Relatório sobre configuração da conexão MySQL com Java.**

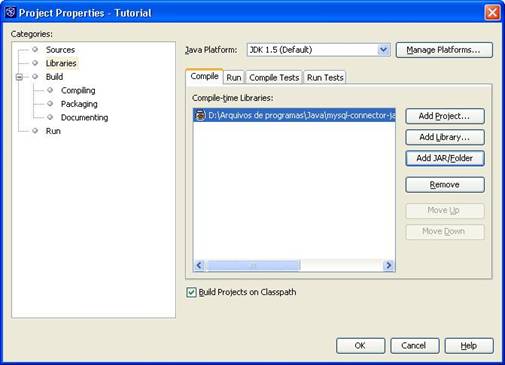
**JDBC tipo 1**: foi o primeiro tipo a ser criado, não faz uma conexão real com o banco de dados, mas sim uma conexão com ODBC. Não é muito utilizado hoje em dia por ser escrito em linguagem nativa, o que sacrifica a portabilidade e exige configuração extra no cliente.

**JDBC tipo 2**: esse tipo de driver eliminou a dependência de ODBC, mas ainda é escrito em linguagem nativa, esse código nativo permite fazer chamadas a uma API cliente do SGBD.

**JDBC tipo 3**: totalmente escrito em [**Java**](https://www.devmedia.com.br/guia/linguagem-java/38169), eliminou a necessidade de bibliotecas de código nativo favorecendo a portabilidade.

**JDBC tipo 4**: também totalmente escrito em Java, utiliza o protocolo de rede proprietário do SGBD, convertendo as chamadas JDBC para chamadas diretas ao SGBD dispensando uma API cliente intermediaria.





**Compreender e integrar a conexão das linguagens JAVA e SQL. Criar conexão ao SGBD.**

**A conexão em Java**

A conexão a um banco de dados é feita de maneira elegante com Java. Para evitar que cada banco tenha a sua própria API e um conjunto de classes e métodos, temos um único conjunto de interfaces muito bem definidas que devem ser implementadas. Esse conjunto de interfaces fica dentro do pacote java.sql e nos referiremos a ele como **JDBC**.

## Fábrica de Conexões

Em determinado momento de nossa aplicação, gostaríamos de ter o controle sobre a construção dos objetos da nossa classe. Muito pode ser feito através do construtor, como saber quantos objetos foram instanciados ou fazer o log sobre essas instanciações.

Às vezes, também queremos controlar um processo muito repetitivo e trabalhoso, como abrir uma conexão com o banco de dados. Tomemos como exemplo a classe a seguir, que seria responsável por abrir uma conexão com o banco:

**public** **class** **ConnectionFactory** {

**public** Connection **getConnection**() {

**try** {

**return** DriverManager.getConnection(

"jdbc:mysql://localhost/fj21", "root", "<SENHA DO BANCO AQUI>");

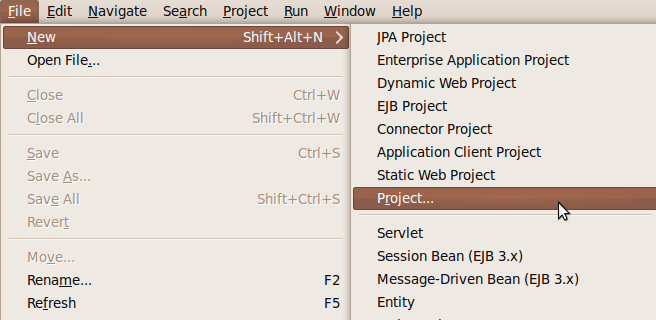
} **catch** (SQLException e) {

**throw** **new** RuntimeException(e);

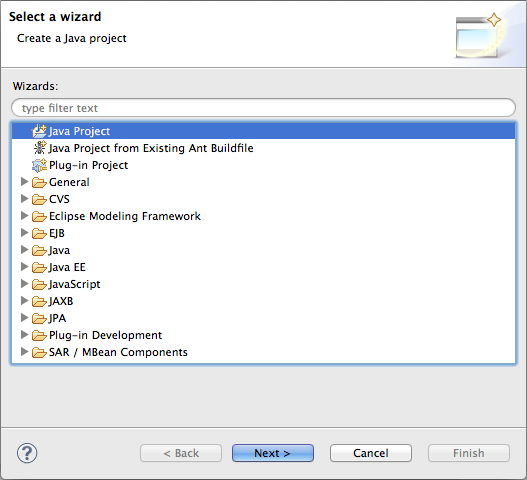
}

**Exercícios: ConnectionFactory**

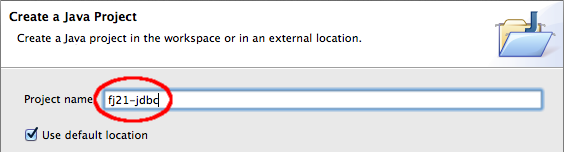
Vá em **File -> New -> Project**:



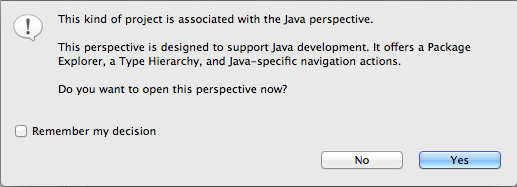
Selecione **Java Project** e clique em **Next**:



Coloque o nome do projeto como **fj21-jdbc** e clique em **Finish**:



Uma nova janela surgirá perguntando se você quer criar um novo **module-info.java**, clique em **Don't Create** aceite a mudança de perspectiva:



Pode-se dizer que é uma API que reúne conjuntos de classes e interfaces escritas na linguagem Java na qual possibilita se conectar através de um driver específico do banco de dados desejado.