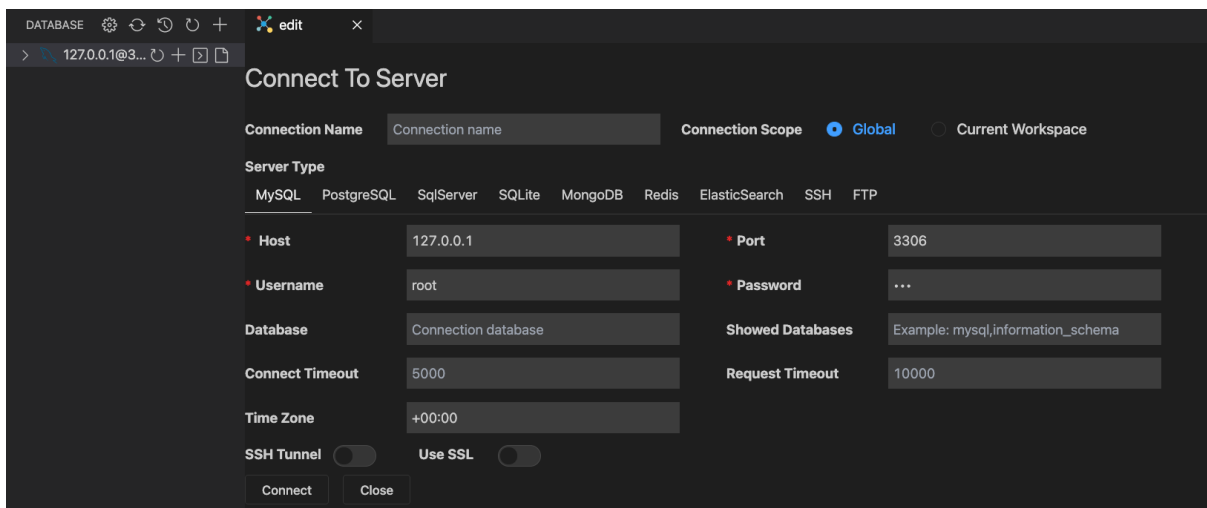


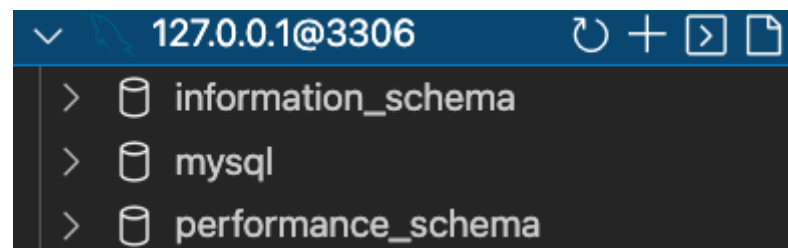


Esse tutorial tem como objetivo, revisar os conceitos vistos no curso de Java Avançado na Apex. Siga atentamente os passos abaixo, tendo dúvida é só enviar e-mail para: ralf.lima@gmail.com, bons estudos.

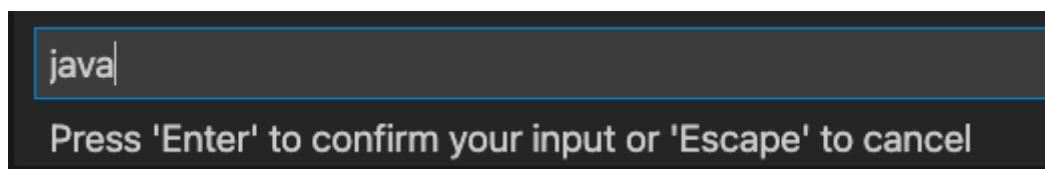
1. Abra o Visual Studio Code, certifique-se que tenha instalado o banco de dados e a extensão para Visual Studio Code chamada **MySQL**, que deverá estar na lateral esquerda do editor.
2. Faça a conexão, fornecendo o usuário e a senha. Por padrão o usuário é **root**.



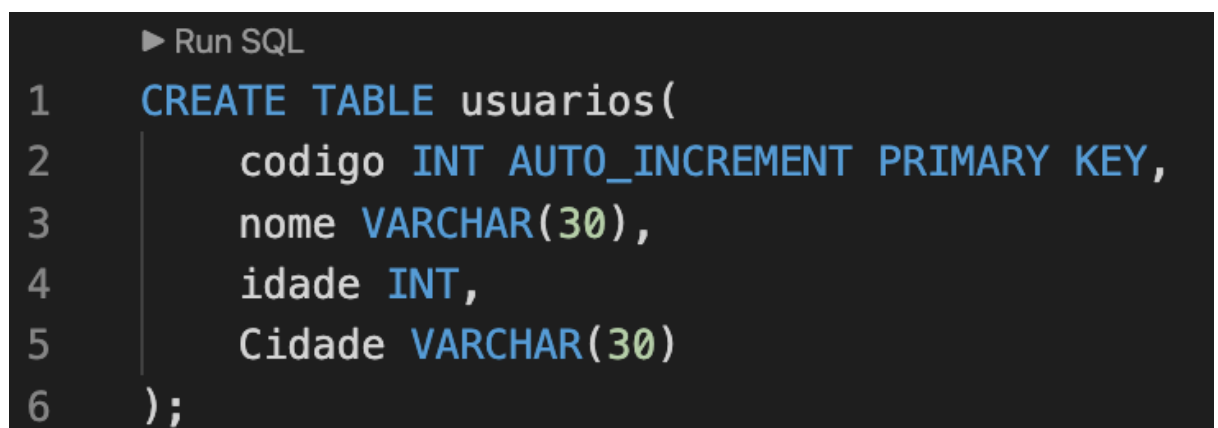
3. Crie uma base de dados, ela é representada pelo sinal de soma, perto do host (127.0.0.1):



4. Abrirá uma caixa para poder digitar o nome da base, neste exemplo colocaremos simplesmente Java:

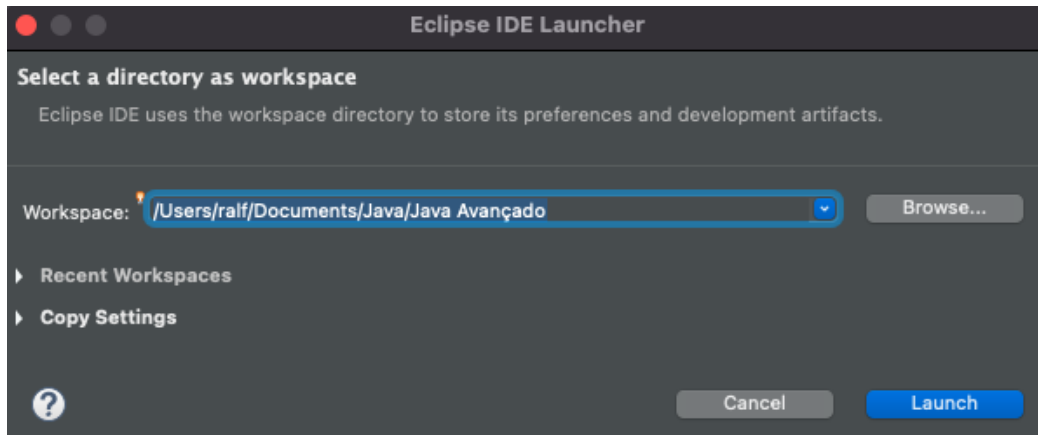


5. Agora podemos criar a tabela, para isso, na lateral esquerda onde há sua base de dados Java, terá o ícone de uma folha, clique nela e digite a seguinte estrutura:

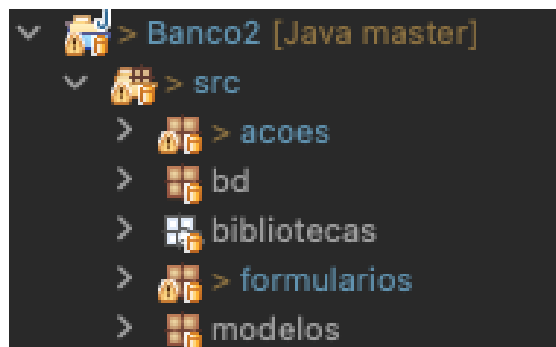


6. Clique no link Run SQL na parte superior do seu código, a tabela estará criada.

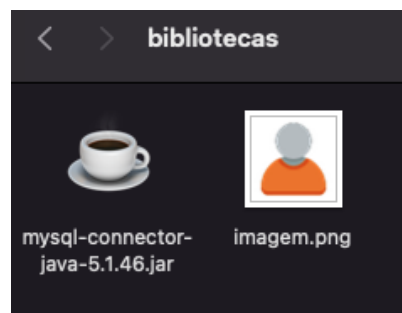
7. Agora vamos para o Eclipse, para isso, crie uma pasta e em seguida abra o Eclipse e selecione essa pasta.



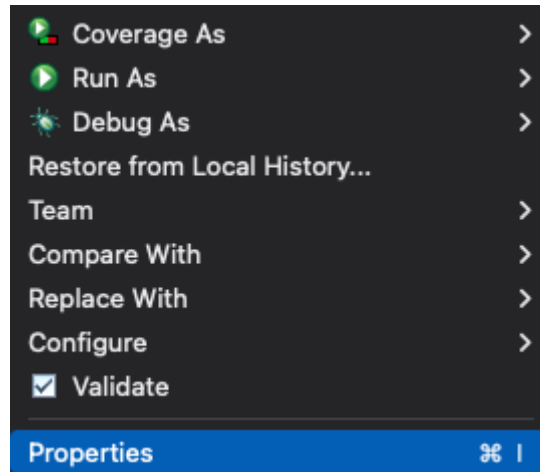
8. Crie um novo projeto e atribua os seguintes pacotes:



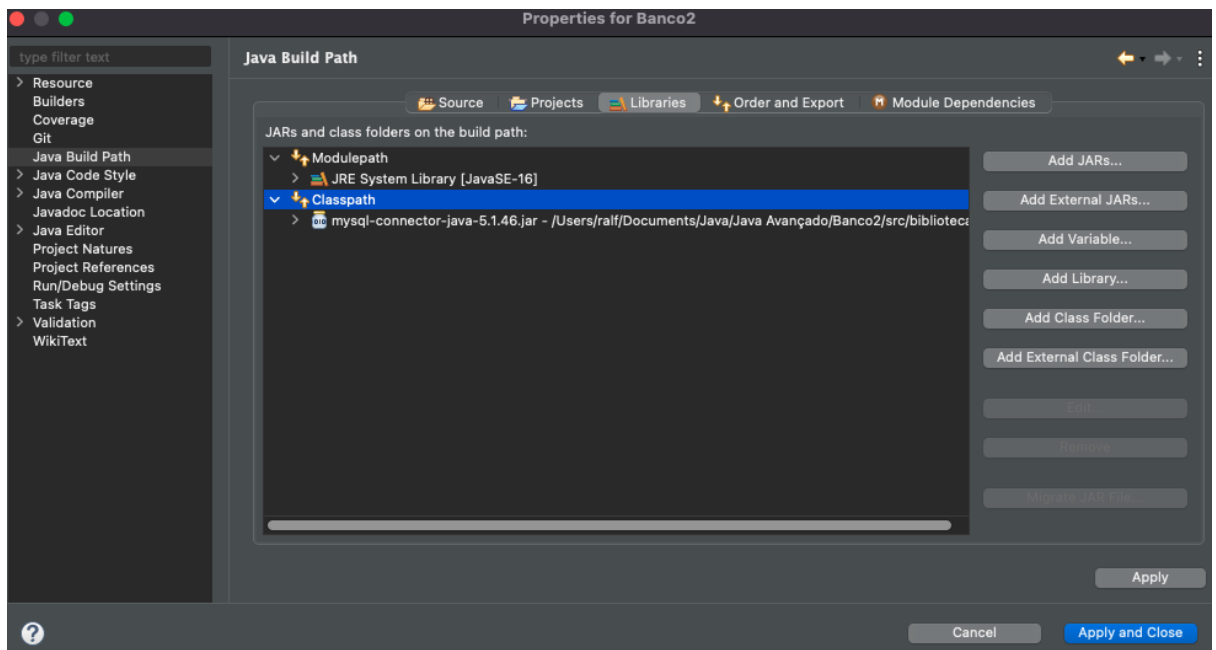
9. Abra o explorador de arquivos e procure pela pasta bibliotecas, adicione o driver de conexão e qualquer outro elemento complementar que queira utilizar, neste tutorial será adicionado um ícone:



10. Vamos importar o driver de conexão para o banco MySQL, para isso, clique com o botão direito sobre o projeto no Eclipse e selecione a opção propriedades:



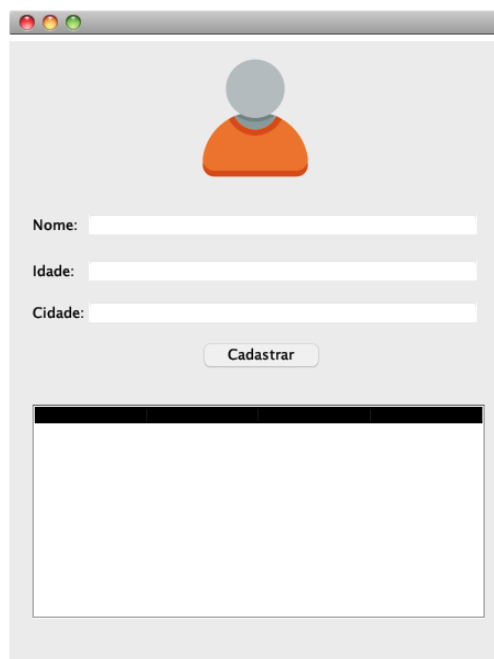
11. Na lateral esquerda selecione a opção **Java Build Path**, na nova aba procure por **Libraries**, clique em **Classpath**, e em seguida na lateral direita selecione **Add External JARs**, procure pelo driver de conexão na pasta bibliotecas e em seguida clique em **Apply and Close**.



12. No pacote **bd**, crie uma classe chamada **Conexao**, com a seguinte estrutura:

```
1 package bd;
2
3 import java.sql.Connection;
4
5
6 public class Conexao {
7
8     // Método para conectar
9     public static Connection conectar() {
10
11         // Atributo
12         Connection con = null;
13
14         // Tentativa
15         try {
16             con = DriverManager.getConnection("jdbc:mysql://localhost/java", "root", "password");
17         } catch (Exception falha) {
18             System.out.println("Falha ao conectar: "+falha.getMessage());
19         }
20
21         // Retornar
22         return con;
23     }
24 }
25
26 }
```

13. No pacote **formularios**, crie um elemento JFrame, para isso clique no pacote com o botão direito -> new -> Other. Procure pela opção WindowBuilder -> Swing Designer -> JFrame. Coloque o nome que quiser neste arquivo e crie a seguinte estrutura:



The image shows a Java Swing window titled "Cadastro" (Registration). It features a light gray background and a standard macOS-style title bar with red, yellow, and green window control buttons. At the top center is a placeholder for a user profile picture, represented by a gray circle above an orange rounded rectangle. Below this are three text input fields labeled "Nome:", "Idade:", and "Cidade:". Underneath the fields is a button labeled "Cadastrar" (Register). At the bottom of the window is a large, empty rectangular area, likely intended for a list of registered users or additional information.

14. No pacote **modelo**, crie uma classe chamada Cliente com a seguinte estrutura:

```
1 package modelo;
2
3 public class Cliente {
4
5     // Atributos
6     private int codigo, idade;
7     private String nome, cidade;
8
9     // Construtores
10    public Cliente() {}
11
12    public Cliente(int codigo, String nome, int idade, String cidade) {
13        this.codigo = codigo;
14        this.nome = nome;
15        this.idade = idade;
16        this.cidade = cidade;
17    }
18
19 }
```

Importante! Não se esqueça de implementar os Getters e Setters. Na parte superior do Eclipse procure por Source -> Generate Getters and Setters.

15. No pacote de **ações**, crie uma classe para poder implementar as funcionalidades de banco de dados. Iniciamos com o cadastro:

```
15 // Cadastrar
16 public static void cadastrar(Cliente c) {
17
18     try {
19
20         Connection conexao = Conexao.conectar();
21
22         String sql = "INSERT INTO clientes (nome, idade, cidade) VALUES (?, ?, ?)";
23
24         PreparedStatement pstmt = conexao.prepareStatement(sql);
25         pstmt.setString(1, c.getNome());
26         pstmt.setInt(2, c.getIdade());
27         pstmt.setString(3, c.getCidade());
28
29         pstmt.execute();
30
31     } catch (Exception erro) {
32         System.out.println("Falha ao cadastrar " + erro.getMessage());
33     }
34
35 }
```

16. Agora com a seleção:

```
37 // Listar
38 public static DefaultTableModel listar() {
39
40     DefaultTableModel dtm = new DefaultTableModel();
41     dtm.addColumn("Código");
42     dtm.addColumn("Nome");
43     dtm.addColumn("Idade");
44     dtm.addColumn("Cidade");
45
46
47     try {
48
49         Connection conexao = Conexao.conectar();
50
51         String sql = "SELECT * FROM clientes";
52
53         Statement stmt = conexao.createStatement();
54
55         ResultSet rs = stmt.executeQuery(sql);
56
57         while(rs.next()) {
58             dtm.addRow(new Object[] {
59                 rs.getInt(1),
60                 rs.getString(2),
61                 rs.getInt(3),
62                 rs.getString(4)
63             });
64         }
65
66     } catch (Exception erro) {
67         System.out.println("Falha ao listar");
68     }
69
70     return dtm;
71
72 }
```

17. Em seguida com a exclusão:

```
74 // Remover cliente
75 public static void remover(int codigo) {
76
77     // Tentativa
78     try {
79
80         Connection conexao = Conexao.conectar();
81
82         String sql = "DELETE FROM clientes WHERE codigo = ?";
83
84         PreparedStatement pstmt = conexao.prepareStatement(sql);
85         pstmt.setInt(1, codigo);
86         pstmt.execute();
87
88     } catch (Exception erro) {
89         System.out.println("Falha ao excluir");
90     }
91 }
92
93 }
```

18. Nessa etapa vamos focar no funcionamento do formulário, para isso no botão cadastrar, iremos adicionar o seguinte código:

```
92 JButton btnCadastrar = new JButton("Cadastrar");
93 btnCadastrar.addActionListener(new ActionListener() {
94     public void actionPerformed(ActionEvent e) {
95
96         try {
97             String nome = txtNome.getText();
98             int idade = Integer.parseInt(txtIdade.getText());
99             String cidade = txtCidade.getText();
100
101             Cliente c = new Cliente(0, nome, idade, cidade);
102
103             AcoesCliente.cadastrar(c);
104
105             table.setModel(AcoesCliente.listar());
106
107             txtNome.setText("");
108             txtIdade.setText("");
109             txtCidade.setText("");
110
111             JOptionPane.showMessageDialog(null, "Cadastro realizado com sucesso!");
112         } catch (Exception erro) {
113             JOptionPane.showMessageDialog(null, "Falha ao cadastrar");
114         }
115     }
116 }
117 });
```


19. Nessa etapa, precisamos chamar o método responsável por listar o dados em nossa tabela, procure o seu componente de tabela e chame o método **listar()**, criado no passo 16:

```
127
128         table = new JTable(AcoesCliente.listar());
129
```

20. Por fim, iremos implementar a ação de exclusão. Iremos adicionar um evento de clique em nossa tabela, para isso teremos a seguinte estrutura:

```
131•         table.addMouseListener(new MouseAdapter() {
132•             @Override
133             public void mouseClicked(MouseEvent e) {
134
135                 // Obter a linha selecionada
136                 int linha = table.getSelectedRow();
137
138                 // Obter o código do cliente
139                 int codigo = Integer.parseInt(table.getValueAt(linha,0).toString());
140
141                 // Efetuar a exclusão
142                 AcoesCliente.remover(codigo);
143
144                 // Atualizar tabela
145                 table.setModel(AcoesCliente.listar());
146             }
147         });
```

Esse exemplo se encontra no GitHub: <https://github.com/ralfslima/java>