





# Welcome to MySQL Workbench

MySQL Workbench is the official graphical user interface (GUI) tool for MySQL. It allows you to design, create and browse your database schemas, work with database objects and insert data as well as design and run SQL queries to work with stored data. You can also migrate schemas and data from other database vendors to your MySQL database.

[Browse Documentation >](#)[Read the Blog >](#)[Discuss on the Forums >](#)

## MySQL Connections

Local instance MySQL80

 root

 localhost:3306





# Welcome to MySQL Workbench

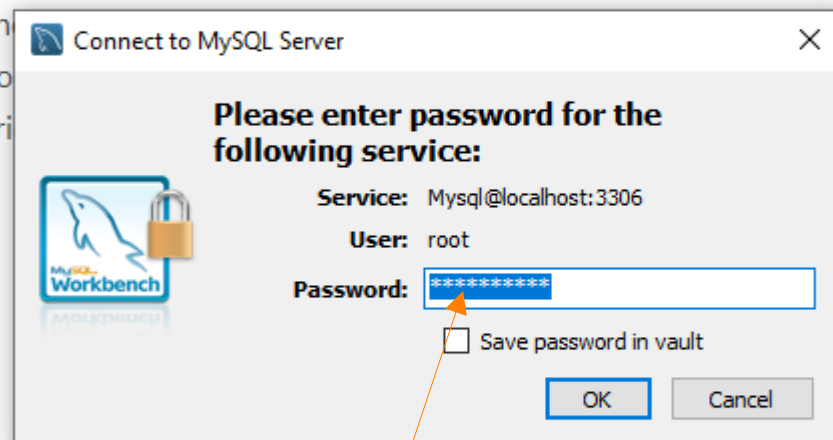
MySQL Workbench is the  
create and browse your  
design and run SQL queries

[Browse Documentation >](#)

## MySQL Connections

Local instance MySQL80

 root  
 localhost:3306



Akillies101

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: agenda x

**SCHEMAS**

Filter objects

- dbagenda
  - Tables
    - contatos
  - Views
  - Stored Procedures
  - Functions
- sakila
- sys
- world

Administration Schemas

Information

No object selected

Limit to 1000 rows

```
1 • CREATE DATABASE dbagenda;
2 • show databases;
3 • use dbagenda;
4 • create table contatos(
5     idcon int primary key auto_increment,
6     nome varchar(50) not null,
7     fone varchar(15) not null,
8     email varchar(50)
9 );
10 • show tables;
11 • describe contatos;
```

Output

Action Output

#	Time	Action	Message
1	15:45:43	use dbagenda	0 row(s) af

Ctrl + enter



Navigator

## SCHEMAS

Filter objects

- dbagenda
  - Tables
    - contatos
  - Views
  - Stored Procedures
  - Functions
- sakila
- sys
- world

Administration Schemas

Information

agenda\* x

Limit to 1000 rows

```
8      email varchar(50)
9    );
10 • show tables;
11 • describe contatos;
12 /* CRUD CREATE */
13
14 /* CRUD READ */
15 • select * from contatos;
```

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:

	idcon	nome	fone	email
*	NULL	NULL	NULL	NULL

contatos 1 x

Apply

Revert

Output

Action Output

#	Time	Action	Message
---	------	--------	---------

Result Grid

Form Editor

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator:

SCHEMAS

Filter objects

dbagenda

- Tables
  - contatos
- Views
- Stored Procedures
- Functions

sakila

sys

world

Administration Schemas

Information

No object selected

agenda\* x

Limit to 1000 rows

```
6 nome varchar(50) not null,  
7 fone varchar(15) not null,  
8 email varchar(50)  
9 );  
10 • show tables;  
11 • describe contatos;  
12 /* CRUD CREATE */  
13 • insert into contatos (nome,fone,email) values ('Bill Gates', '99999-1111', 'Gates@outlook.com');  
14 /* CRUD READ */  
15 • select * from contatos;
```

Result Grid

	idcon	nome	fone	email
*	NULL	NULL	NULL	NULL

contatos 1 x

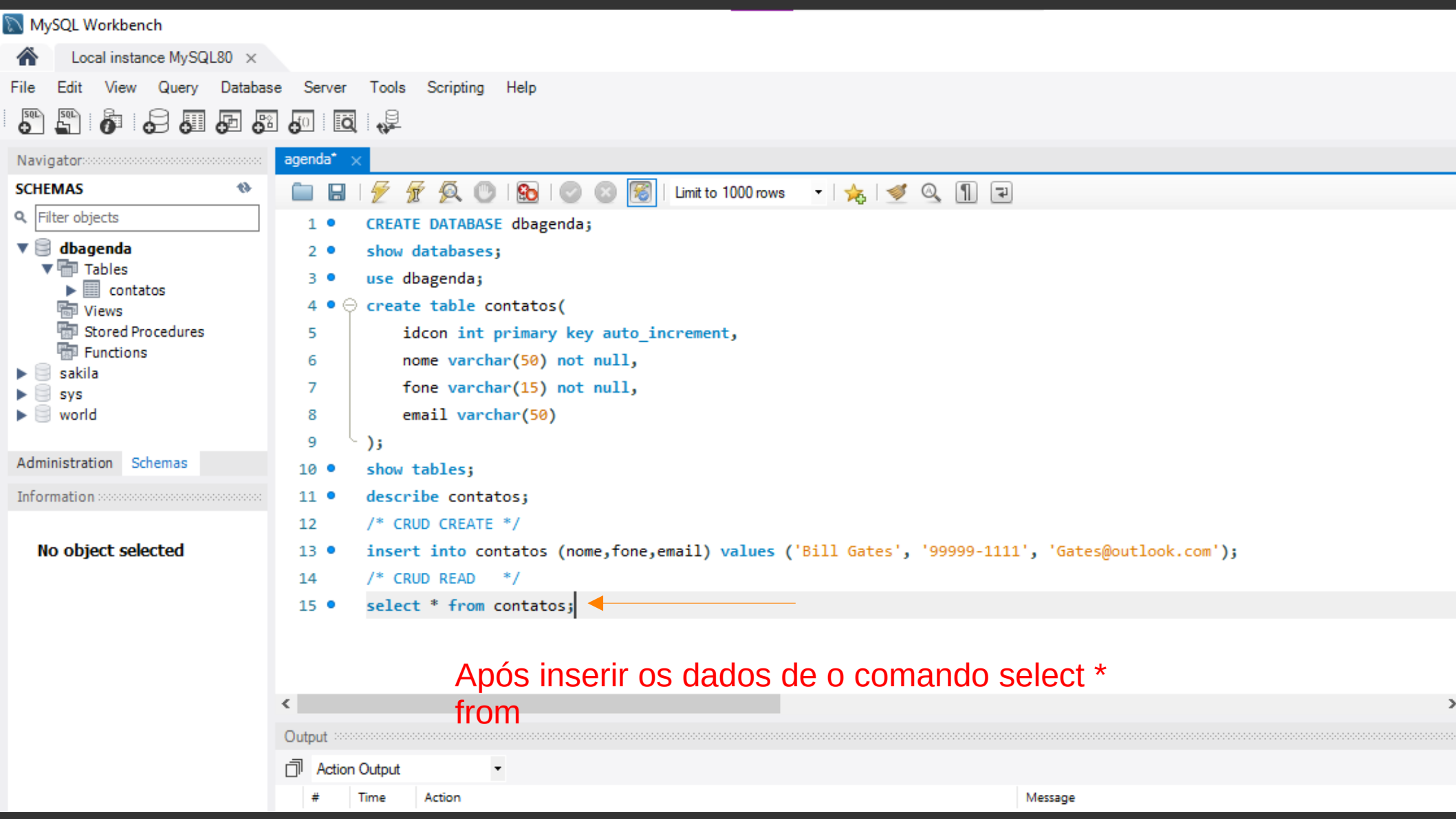
Apply Revert

Output

Action Output

#	Time	Action	Message
✓ 1	15:45:43	use dbagenda	0 row(s) affected
✓ 2	15:51:08	select * from contatos LIMIT 0, 1000	0 row(s) returned

Inserindo os dados manualmente no banco de dados



agenda\*

```
1 • CREATE DATABASE dbagenda;
2 • show databases;
3 • use dbagenda;
4 • create table contatos(
5     idcon int primary key auto_increment,
6     nome varchar(50) not null,
7     fone varchar(15) not null,
8     email varchar(50)
9 );
10 • show tables;
11 • describe contatos;
12 /* CRUD CREATE */
13 • insert into contatos (nome,fone,email) values ('Bill Gates', '99999-1111', 'Gates@outlook.com');
14 /* CRUD READ */
15 • select * from contatos;
```

Após inserir os dados de o comando select \*  
from



## SCHEMAS

Filter objects

- dbagenda
  - Tables
    - contatos
  - Views
  - Stored Procedures
  - Functions
- sakila
- sys
- world

Administration Schemas

Information

agenda x

```
6      nome varchar(50) not null,  
7      fone varchar(15) not null,  
8      email varchar(50)  
9  );  
10 • show tables;  
11 • describe contatos;  
12 /* CRUD CREATE */  
13 • insert into contatos (nome,fone,email) values ('Bill Gates', '99999-1111', 'Gates@outlook.com');  
14 /* CRUD READ */  
15 • select * from contatos;
```

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:

	idcon	nome	fone	email
▶	1	Bill Gates	99999-1111	Gates@outlook.com
*	NULL	NULL	NULL	NULL

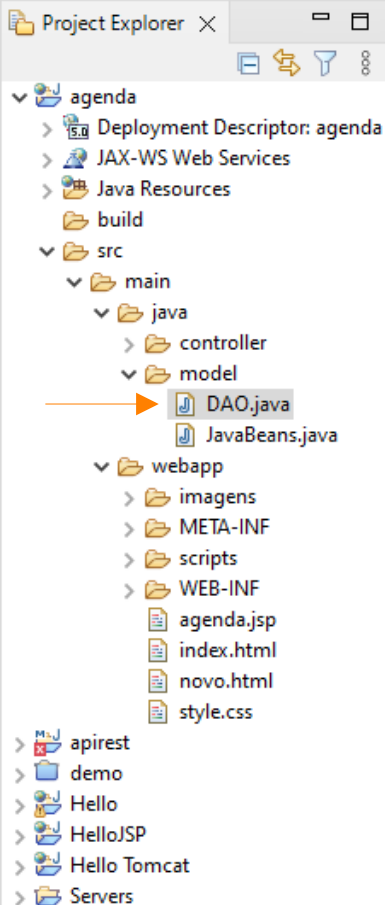
Dados inseridos

contatos 2 x

Apply

Output





```
1 package model;
2
3 import java.sql.Connection;
4
5
6 public class DAO {
7     // modulo de conexão
8     // parametros de conexão
9     private String driver = "com.mysql.cj.jdbc.Driver";
10    private String url      = "jdbc:mysql://127.0.0.1:3306/dbagenda?useTimezone
11    private String user     = "root";
12    private String password = "Akillles101";
13    // metodo de conexão
14    private Connection conectar() {
15        Connection con = null;
16        try {
17            Class.forName(driver);
18            con = DriverManager.getConnection(url, user, password);
```



Project Explorer

- agenda
  - Deployment Descriptor: agenda
  - JAX-WS Web Services
  - Java Resources
  - build
  - src
    - main
      - java
        - controller
        - model
          - DAO.java
          - JavaBeans.java
    - webapp
      - imagens
      - META-INF
      - scripts
      - WEB-INF
        - agenda.jsp
        - index.html
        - novo.html
        - style.css
- apirest
- demo
- Hello
- HelloJSP
- Hello Tomcat
- Servers

novo.html

validador.js

Controller.java

agenda.jsp

index.html

JavaBeans.java

\*DAO.java X

```
27 public void testeConexao() {
28     try {
29         Connection con = conectar();
30         System.out.println(con);
31         con.close();
32     } catch (Exception e) {
33         System.out.println(e);
34         // TODO: handle exception
35     }
36 }
37
38 /* CRUD CREATE */
39 public void inserirContato(JavaBeans contato) {
40
41 }
42 }
43
```

Markers Properties Servers Data Source Explorer Snippets Console X

No consoles to display at this time.



Project Explorer × novo.html validator.js \*Controller.java × agenda.jsp index.html JavaBeans.java DAO.java

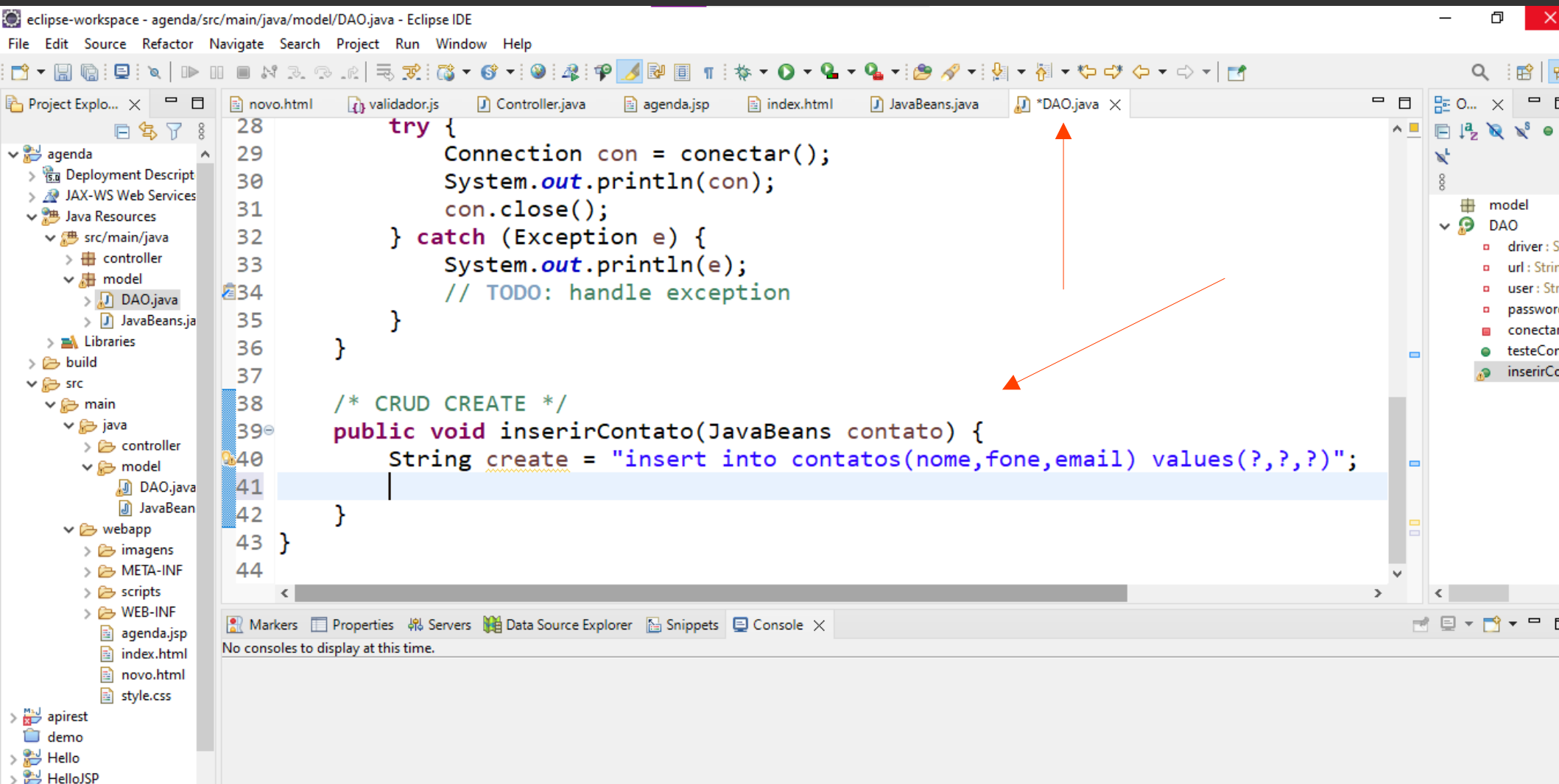
Project Explorer

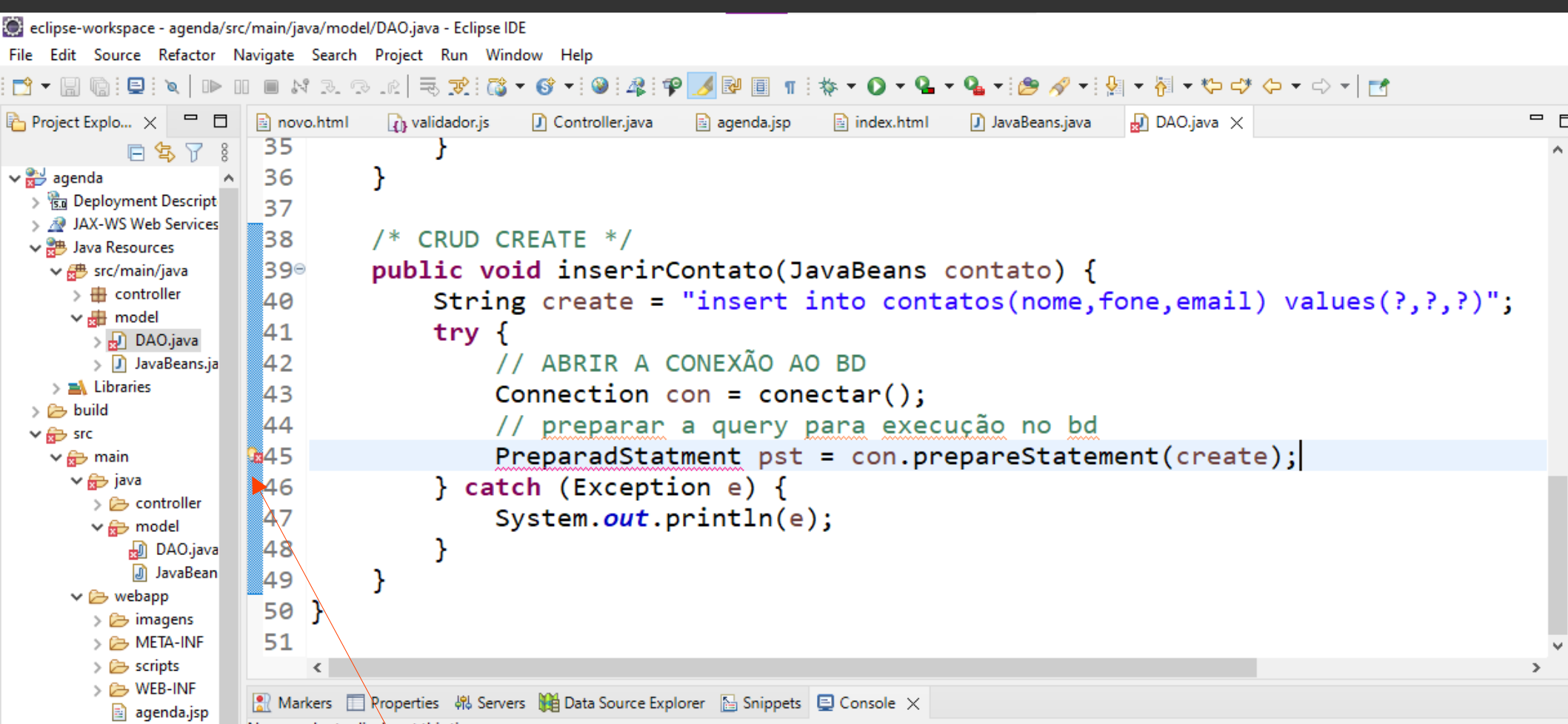
- agenda
  - Deployment Descriptor: ager
  - JAX-WS Web Services
  - Java Resources
    - src/main/java
      - controller
      - model
        - DAO.java
        - JavaBeans.java
    - Libraries
    - build
    - src
      - main
        - java
          - controller
          - model
            - DAO.java
            - JavaBeans.java
        - webapp
          - imagens
          - META-INF
          - scripts
          - WEB-INF
            - agenda.jsp
            - index.html

```
38
39 // Novo contato
40 protected void novoContato(HttpServletRequest request, HttpServletResponse
41 //teste de recebimento dos dados do formulario
42 //System.out.println(request.getParameter("nome"));
43 //System.out.println(request.getParameter("fone"));
44 //System.out.println(request.getParameter("email"));
45 // setar as variaveis JavaBeans
46 contato.setNome(request.getParameter("nome"));
47 contato.setFone(request.getParameter("fone"));
48 contato.setEmail(request.getParameter("email"));
49 // inserir o metodo inserirContato passando o objeto contato
50 dao.inserirContato(contato);
51 }
52
53 }
54
```

Markers Properties Servers Data Source Explorer Snippets Console ×

No consoles to display at this time.





Esse recurso precisa ser importado no eclipse ctrl + shift + o  
faz o import

```
43 Connection con = conectar();
44 // preparar a query para execução no bd
45 Statement pst = con.prepareStatement(create);
46
47
48 }
49
50 }
```

Multiple markers at this line

- PreparedStatement cannot be resolved to a type
- PreparadStatment cannot be resolved to a type

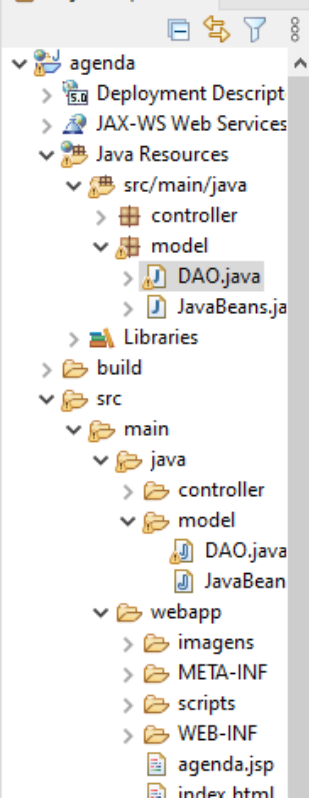
Import 'PreparedStatement' (java.sql)

- Create class 'PreparedStatement'
- Create interface 'PreparedStatement'
- Change to 'ClientPreparedStatement' (com.mysql.cj.jdbc)
- Change to 'DelegatingPreparedStatement' (org.apache.tomcat.jdbc)
- Change to 'EmptyStatement' (org.eclipse.jdt.internal.compiler.ast)
- Change to 'JdbcPreparedStatement' (com.mysql.cj.jdbc)

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
```

Import ctrl + shift + o

```
1 package model;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.PreparedStatement;
6
7 public class DAO {
```

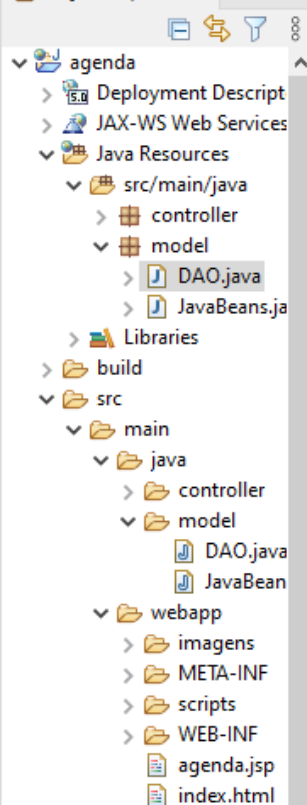


```
40 public void inserirContato(JavaBeans contato) {
41     String create = "insert into contatos(nome,fone,email) values(?,?,?)";
42     try {
43         // ABRIR A CONEXÃO AO BD
44         Connection con = conectar();
45         // preparar a query para execução no bd
46         PreparedStatement pst = con.prepareStatement(create);
47         // substituir os parametros ? pelo conteudo das variaveis javabeans
48         pst.setString(1, contato.getNome());
49         pst.setString(1, contato.getFone());
50         pst.setString(1, contato.getEmail());
51     } catch (Exception e) {
52         System.out.println(e);
53     }
54 }
55 }
56
```





Project Explorer



```
43 // ABRIR A CONEXÃO AO BD
44 Connection con = conectar();
45 // preparar a query para execução no bd
46 PreparedStatement pst = con.prepareStatement(create);
47 // substituir os parametros ? pelo conteudo das variaveis javabeans
48 pst.setString(1, contato.getNome());
49 pst.setString(1, contato.getFone());
50 pst.setString(1, contato.getEmail());
51 // EXECUTAR A QUERY
52 pst.executeUpdate();
53 // ENCERRAR A CONEXÃO AO BANCO DE DADOS
54 con.close();
55 } catch (Exception e) {
56     System.out.println(e);
57 }
58 }
59 }
```

Markers Properties Servers Data Source Explorer Snippets Console

No consoles to display at this time.



```

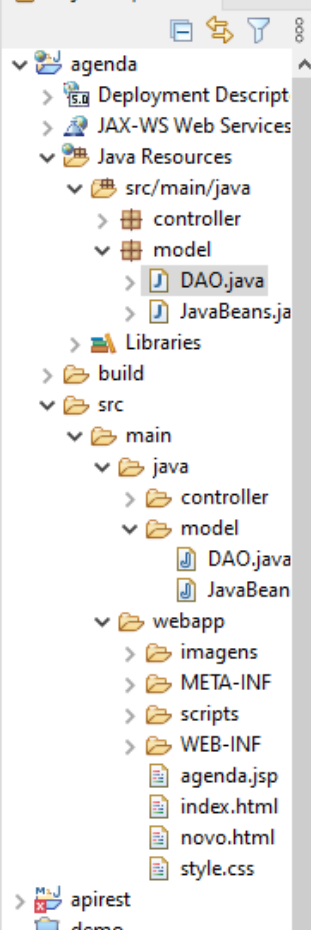
package model;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;

public class DAO {
    // modulo de conexão
    // parametros de conexão
    private String driver = "com.mysql.cj.jdbc.Driver";
    private String url = "jdbc:mysql://127.0.0.1:3306/dbagenda?useTimezone=true&serverTimezone=UTC";
    private String user = "root";
    private String password = "Akillies101";
    // metodo de conexão
    private Connection conectar() {
        Connection con = null;
        try {
            Class.forName(driver);
            con = DriverManager.getConnection(url, user, password);
            return con;
        } catch (Exception e) {
            System.out.println(e);
            return null;
            // TODO: handle exception
        }
    }
    // teste de conexão
    public void testeConexao() {
        try {
            Connection con = conectar();
            System.out.println(con);
            con.close();
        } catch (Exception e) {
            System.out.println(e);
            // TODO: handle exception
        }
    }

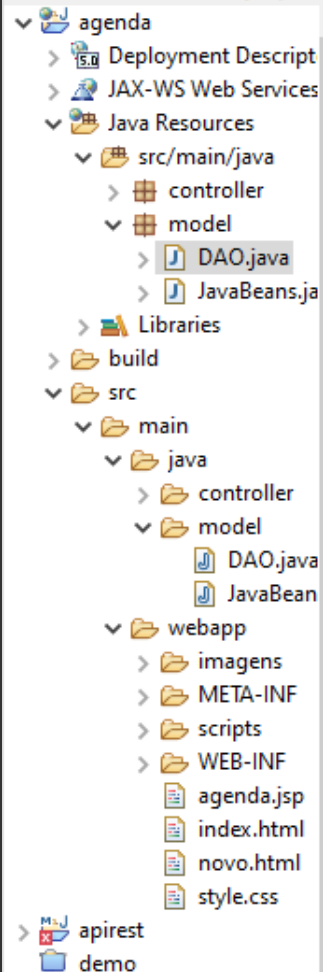
    /* CRUD CREATE */
    public void inserirContato(JavaBeans contato) {
        String create = "insert into contatos(nome,fone,email) values(?,?,?)";
        try {
            // ABRIR A CONEXÃO AO BD
            Connection con = conectar();
            // preparar a query para execução no bd
            PreparedStatement pst = con.prepareStatement(create);
            // substituir os parametros ? pelo conteudo das variaveis javaBeans
            pst.setString(1, contato.getNome());
            pst.setString(2, contato.getFone());
            pst.setString(3, contato.getEmail());
            // EXECUTAR A QUERY
            pst.executeUpdate();
            // ENCERRAR A CONEXÃO AO BANCO DE DADOS
            con.close();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}

```



```
38
39 // Novo contato
40 protected void novoContato(HttpServletRequest request, HttpServletResponse response) {
41     //teste de recebimento dos dados do formulario
42     //System.out.println(request.getParameter("nome"));
43     //System.out.println(request.getParameter("fone"));
44     //System.out.println(request.getParameter("email"));
45     // setar as variaveis JavaBeans
46     contato.setName(request.getParameter("nome"));
47     contato.setFone(request.getParameter("fone"));
48     contato.setEmail(request.getParameter("email"));
49     // inserir o metodo inserirContato passando o objeto contato
50     dao.inserirContato(contato);
51     // redirecionar para o documento agenda.jsp
52     response.sendRedirect("main");
53 }
54
```

No consoles to display at this time.



```
38
39 // Novo contato
40 protected void novoContato(HttpServletRequest request, HttpServletResponse response) {
41     //teste de recebimento dos dados do formulario
42     //System.out.println(request.getParameter("nome"));
43     //System.out.println(request.getParameter("fone"));
44     //System.out.println(request.getParameter("email"));
45     // setar as variaveis JavaBeans
46     contato.setNome(request.getParameter("nome"));
47     contato.setFone(request.getParameter("fone"));
48     contato.setEmail(request.getParameter("email"));
49     // inserir o metodo inserirContato passando o objeto contato
50     dao.inserirContato(contato);
51     // redirecionar para o documento agenda.jsp
52     response.sendRedirect("main");
53 }
54
```

Salve tudo

The screenshot shows the Eclipse IDE interface. The Project Explorer on the left displays the project structure: agenda > Deployment Descriptor: agenda > JAX-WS Web Services > Java Resources > src/main/java > controller > model > DAO.java > JavaBeans.java > Libraries > build > src > main > java > controller > model > DAO.java > JavaBeans.java > webapp > imagens > META-INF > scripts > WEB-INF > agenda.jsp > index.html > novo.html > style.css. The index.html file is selected. The main editor shows the following HTML code:

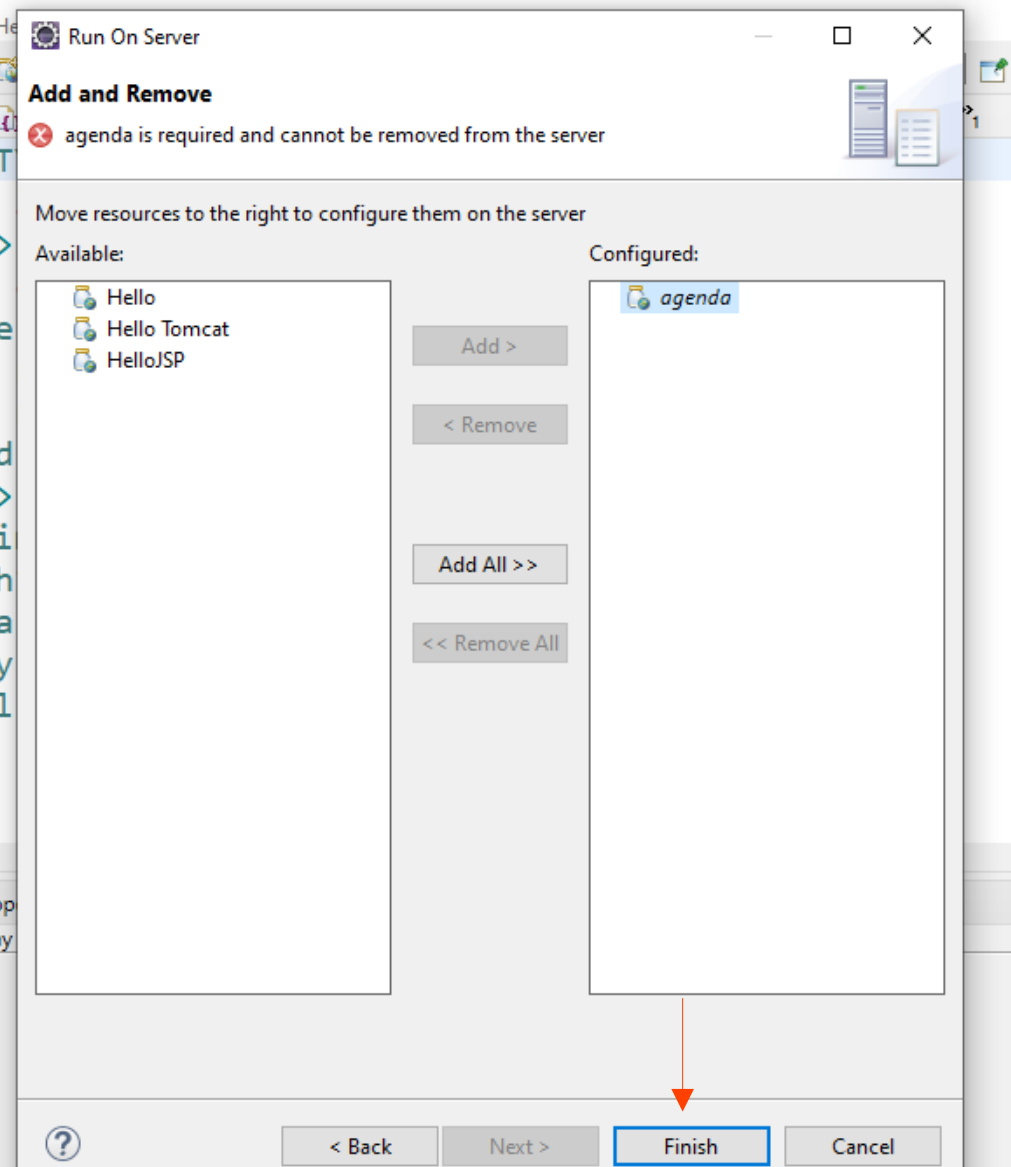
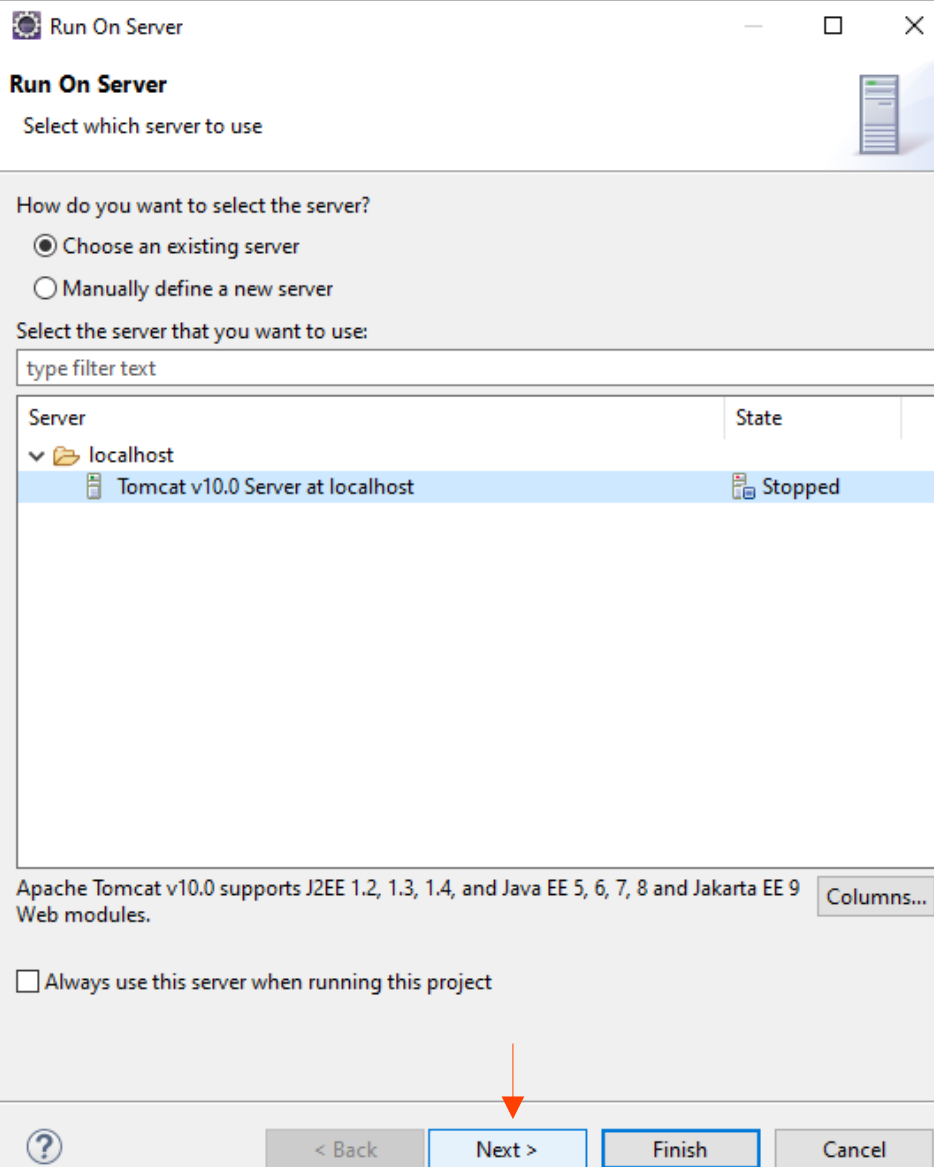
```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4 <meta charset="utf-8">
5 <title>Agenda de contatos</title>
6 <link rel="icon" href="imagens/favicon.png">
7 <link rel="stylesheet" href="style.css">
8 </head>
9 <body>
10 
11 <h1>Agenda de contatos</h1>
12 <a href="main" class="Botao1">Acessar</a>
13 </body>
14 </html>
```

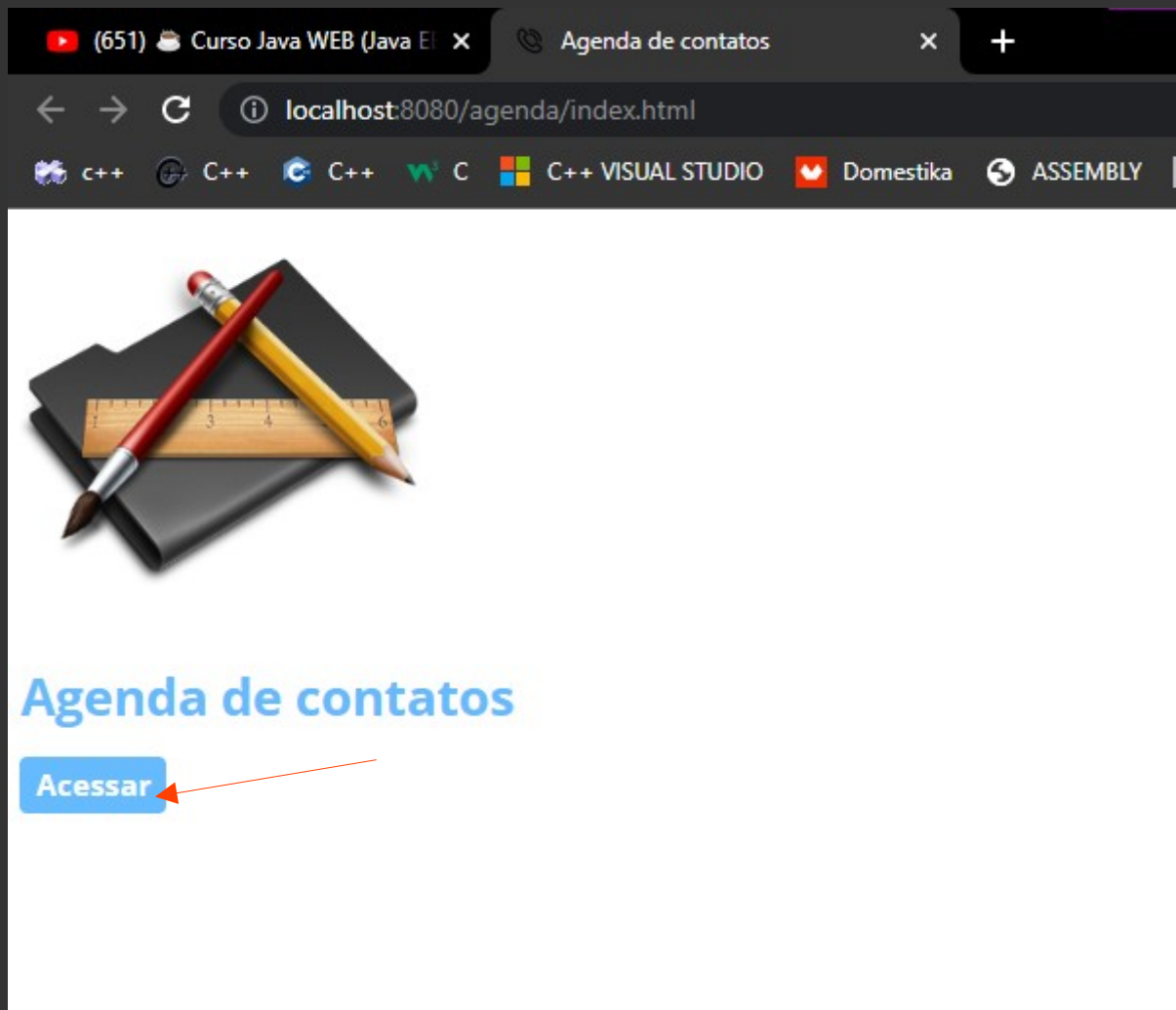
Red arrows point to the 'Run' button in the toolbar and the 'index.html' file in the Project Explorer. The text 'Salve tudo escolha o index.html e execute' is overlaid on the bottom right of the editor area.

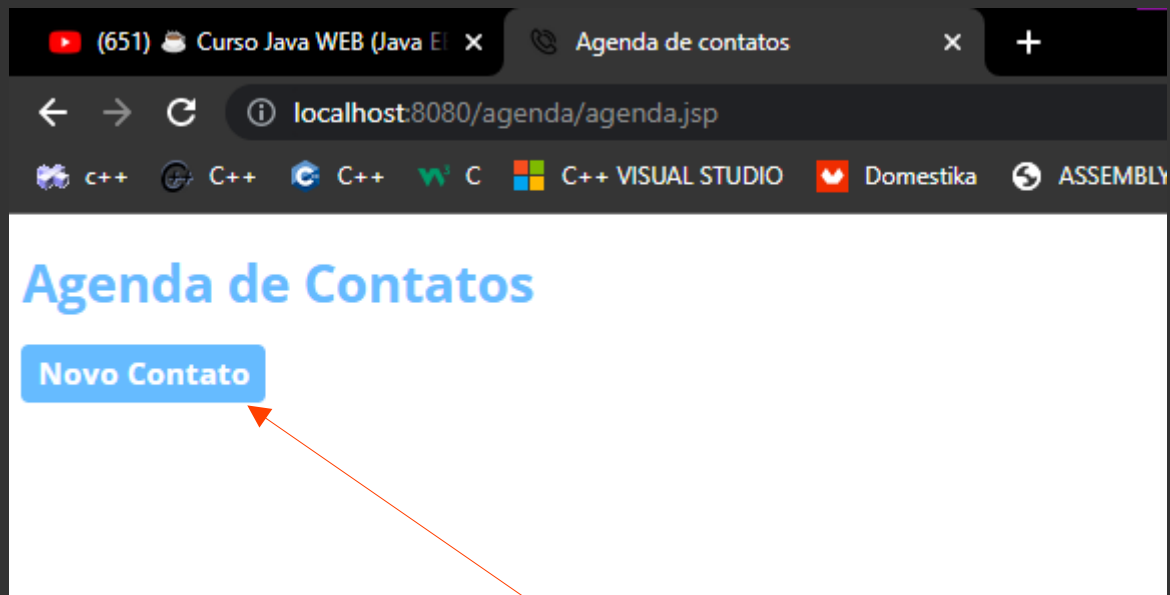
Salve tudo escolha o index.html e execute


Markers Properties Servers Data Source Explorer Snippets Console


No consoles to display at this time.















(651)  Curso Java WEB (Java E x

 Agenda de contatos x +

← → ↺

localhost:8080/agenda/novo.html

 c++  C++  C++  C  C++ VISUAL STUDIO  Domestika  ASSEMBLY 

## Criar novo contato

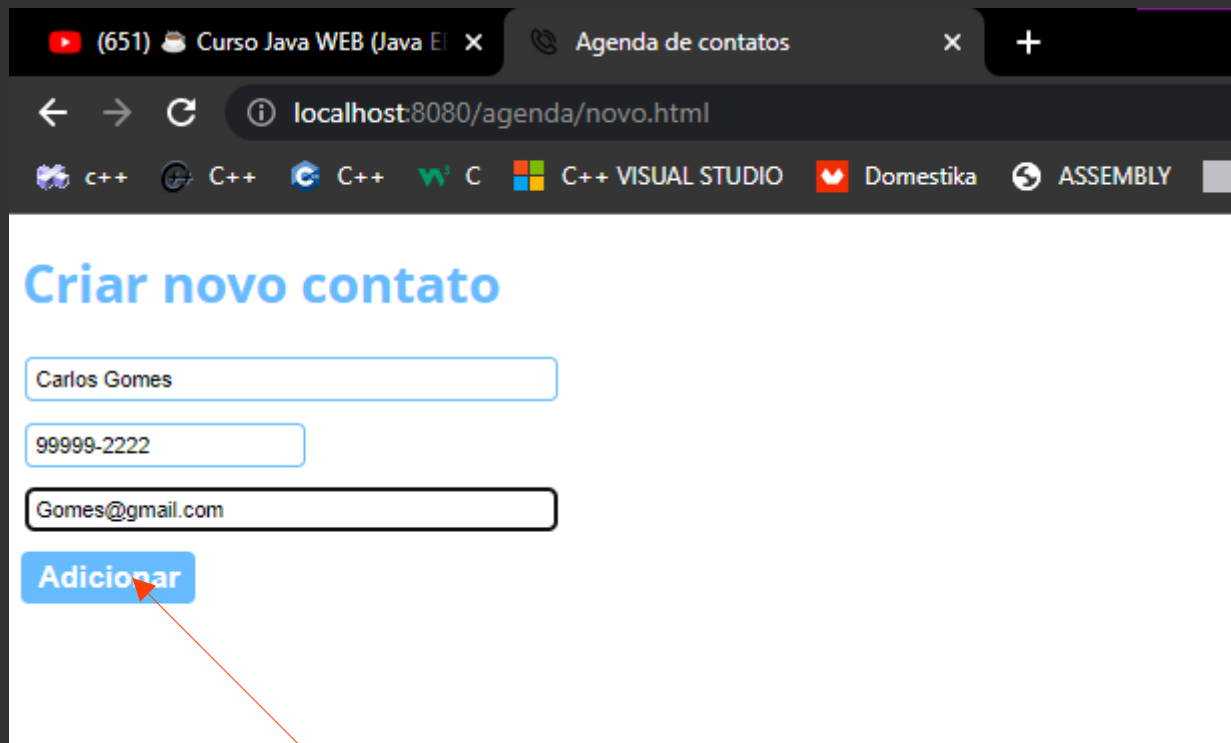
Nome

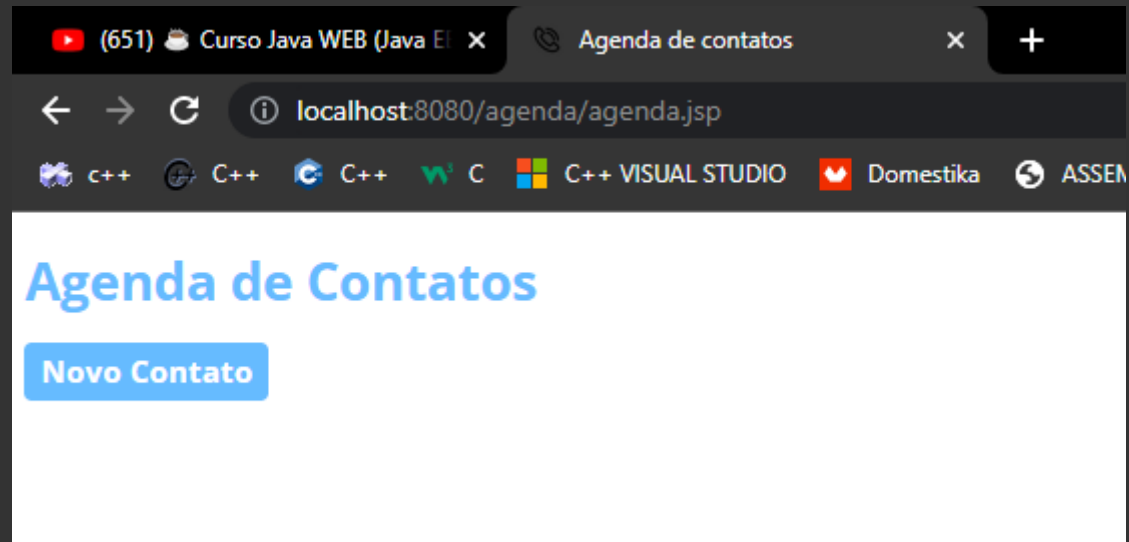
Ende

Email

Adicionar







Agora vamos abrir o mysql



Navigator

## SCHEMAS

Filter objects

- dbagenda
  - Tables
    - contatos
  - Views
  - Stored Procedures
  - Functions
- sakila
- sys
- world

Administration Schemas

Information

agenda x

Limit to 1000 rows

```
6      nome varchar(50) not null,  
7      fone varchar(15) not null,  
8      email varchar(50)  
9  );  
10 • show tables;  
11 • describe contatos;  
12 /* CRUD CREATE */  
13 • insert into contatos (nome,fone,email) values ('Bill Gates', '99999-1111', 'Gates@outlook.com');  
14 /* CRUD READ */  
15 • select * from contatos;
```

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:

	idcon	nome	fone	email
▶	1	Bill Gates	99999-1111	Gates@outlook.com
*	NULL	NULL	NULL	NULL

contatos 2 x

Output

Action Output

Ctrl + enter  
Para ver o contato que adicionamos

Apply

Revert



## SCHEMAS

Filter objects

- dbagenda
  - Tables
    - contatos
  - Views
  - Stored Procedures
  - Functions
- sakila
- sys
- world

Administration Schemas

No object selected

```
6      nome varchar(50) not null,  
7      fone varchar(15) not null,  
8      email varchar(50)  
9  );  
10 • show tables;  
11 • describe contatos;  
12 /* CRUD CREATE */  
13 • insert into contatos (nome,fone,email) values ('Bill Gates', '99999-1111', 'Gates@outlook.com');  
14 /* CRUD READ */  
15 • select * from contatos;
```

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:

idcon	nome	fone	email
1	Bill Gates	99999-1111	Gates@outlook.com
NULL	NULL	NULL	NULL

Como sempre deu erro null

contatos 3 x

Apply

Revert

Output

Action Output

O erro foi meu mesmo tinha que mudar isso 1 2 e 3  
Mudei e tentei denovo

```
// substituir os parametros ? pelo conteudo  
pst.setString(1, contato.getNome());  
pst.setString(2, contato.getFone());  
pst.setString(3, contato.getEmail());  
// EXECUTAR A QUERY  
pst.executeUpdate();  
// FECHAR A CONEXÃO AO BANCO DE DADOS
```

(651)

Curso Java WEB (Java E

Agenda de contatos

localhost:8080/agenda/novo.html

c++

C++

C++

C

C++ VISUAL STUDIO

Domestika

AS

## Criar novo contato

Cris Oliveira

99999-2222

cris@gmail.com

Adicionar



## SCHEMAS

Filter objects

- dbagenda
  - Tables
    - contatos
  - Views
  - Stored Procedures
  - Functions
- sakila
- sys
- world

Administration Schemas



```
6      nome varchar(50) not null,  
7      fone varchar(15) not null,  
8      email varchar(50)  
9  );  
10 • show tables;  
11 • describe contatos;  
12 /* CRUD CREATE */  
13 • insert into contatos (nome,fone,email) values ('Bill Gates', '99999-1111', 'Gates@outlook.com');  
14 /* CRUD READ */  
15 • select * from contatos;
```

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:

	idcon	nome	fone	email
▶	1	Bill Gates	99999-1111	Gates@outlook.com
	2	Cris Oliveira	99999-2222	cris@gmail.com
*	NULL	NULL	NULL	NULL

Resolvido eram os numeros mesmo

