

Camada MODEL

eclipse-workspace - agenda/src/main/java/model/JavaBeans.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help



Project Explorer

- agenda
 - Deployment Descriptor: ager
 - JAX-WS Web Services
 - Java Resources
 - src/main/java
 - controller
 - model
 - DAO.java
 - JavaBeans.java
 - Libraries
 - build
 - src
 - apiREST
 - demo
 - Hello
 - HelloJSP
 - Hello Tomcat
 - Servers

Controller.java

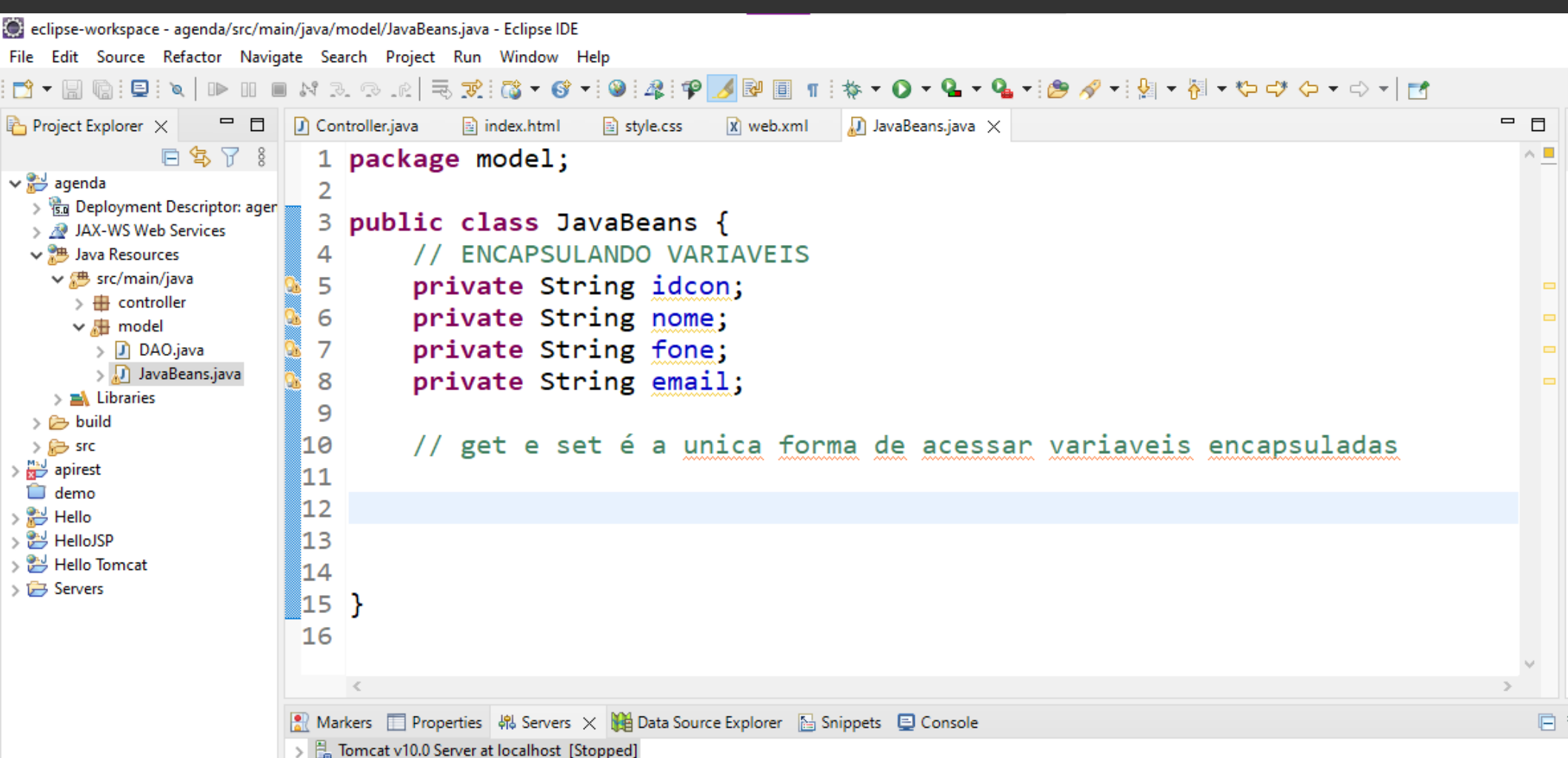
index.html

style.css

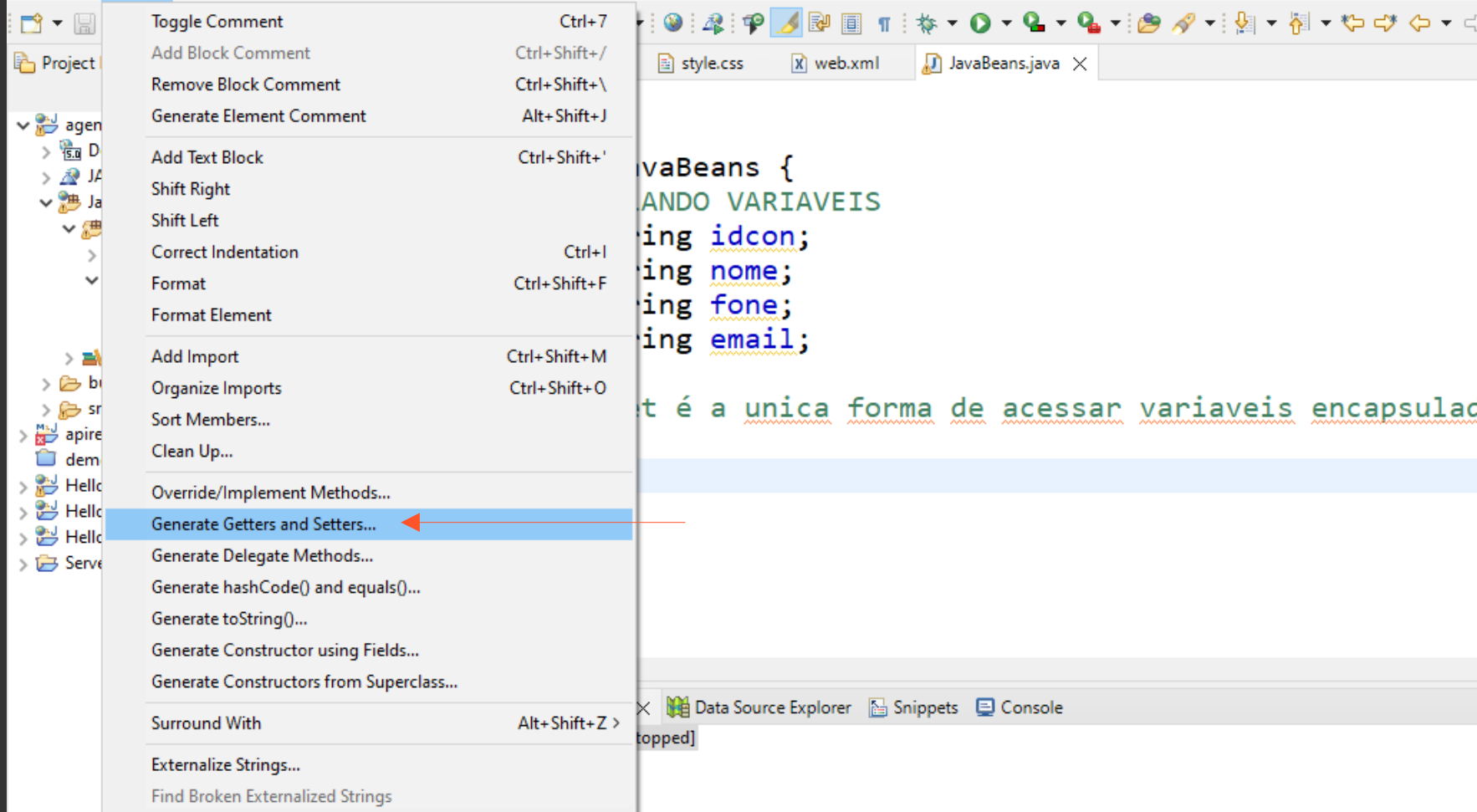
web.xml

JavaBeans.java

```
1 package model;  
2  
3 public class JavaBeans {  
4  
5 }  
6
```



Essas variáveis são tiradas dos campos do banco de dados



The screenshot shows the Eclipse IDE interface. The 'Source' menu is open, displaying various code editing and refactoring options. The 'Generate Getters and Setters...' option is highlighted with a blue background and a red arrow pointing to it. The menu items are organized into sections, with some having keyboard shortcuts listed to the right. The background shows a project explorer on the left and a code editor on the right with a JavaBeans.java file open.

- Toggle Comment Ctrl+7
- Add Block Comment Ctrl+Shift+ /
- Remove Block Comment Ctrl+Shift+ \
- Generate Element Comment Alt+Shift+J
- Add Text Block Ctrl+Shift+ '
- Shift Right
- Shift Left
- Correct Indentation Ctrl+I
- Format Ctrl+Shift+F
- Format Element
- Add Import Ctrl+Shift+M
- Organize Imports Ctrl+Shift+O
- Sort Members...
- Clean Up...
- Override/Implement Methods...
- Generate Getters and Setters...**
- Generate Delegate Methods...
- Generate hashCode() and equals()...
- Generate toString()...
- Generate Constructor using Fields...
- Generate Constructors from Superclass...
- Surround With Alt+Shift+Z >
- Externalize Strings...
- Find Broken Externalized Strings

```
vaBeans {  
    ANDO VARIAVEIS  
    ing idcon;  
    ing nome;  
    ing fone;  
    ing email;
```

```
t é a unica forma de acessar variaveis encapsulad
```

Data Source Explorer Snippets Console

topped]



Generate Getters and Setters



Select getters and setters to create:

- > ☒ ☐ email
- > ☒ ☐ fone
- > ☒ ☐ idcon
- > ☒ ☐ nome

Select All

Deselect All

Select Getters

Select Setters

☐ Allow setters for final fields (remove 'final' modifier from fields if necessary)

Insertion point:

After 'email' ▼

Sort by:

Fields in getter/setter pairs ▼

Access modifier

☒ public

☐ protected

☐ package

☐ private

☐ final

☐ synchronized

☐ Generate method comments

The format of the getters/setters may be configured on the [Code Templates](#) preference page.

8 of 8 selected.

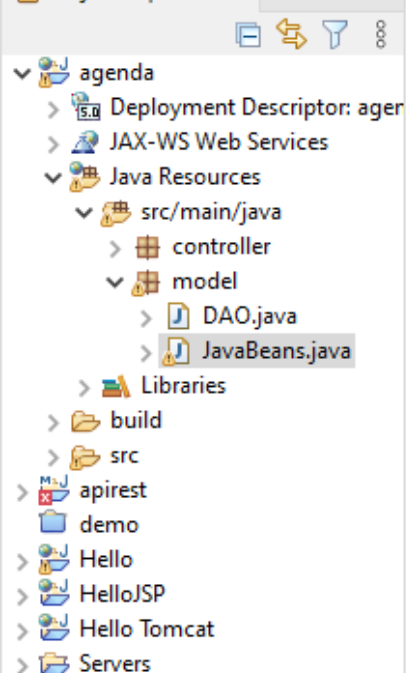


Generate

Cancel



Project Explorer



Controller.java index.html style.css web.xml *JavaBeans.java

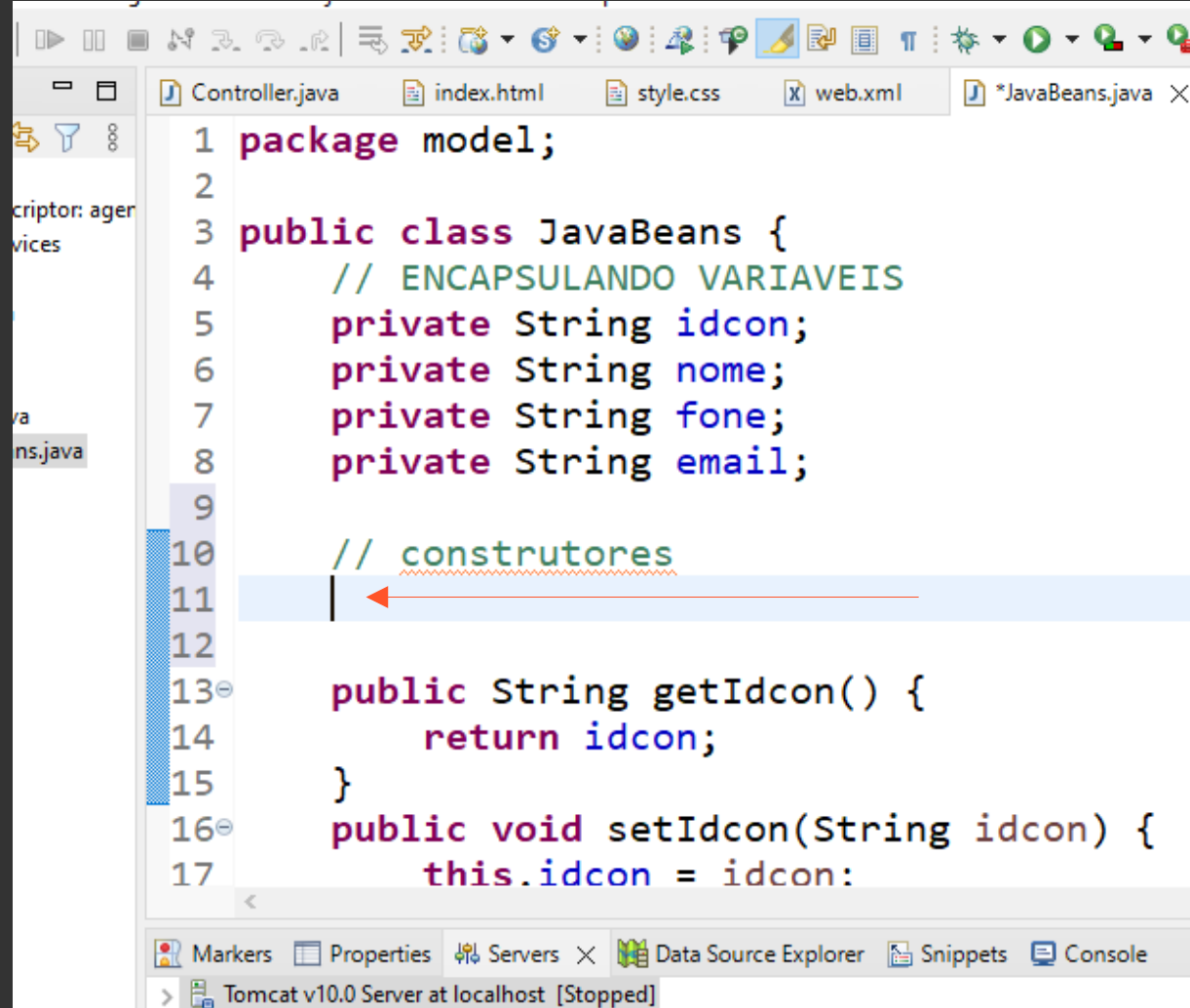
```
10         return idcon;
11     }
12     public void setIdcon(String idcon) {
13         this.idcon = idcon;
14     }
15     public String getNome() {
16         return nome;
17     }
18     public void setNome(String nome) {
19         this.nome = nome;
20     }
21     public String getFone() {
22         return fone;
23     }
24     public void setFone(String fone) {
25         this.fone = fone;
26     }
```

Os metodos
getters e setters
são gerados

Markers Properties Servers Data Source Explorer Snippets Console

> Tomcat v10.0 Server at localhost [Stopped]

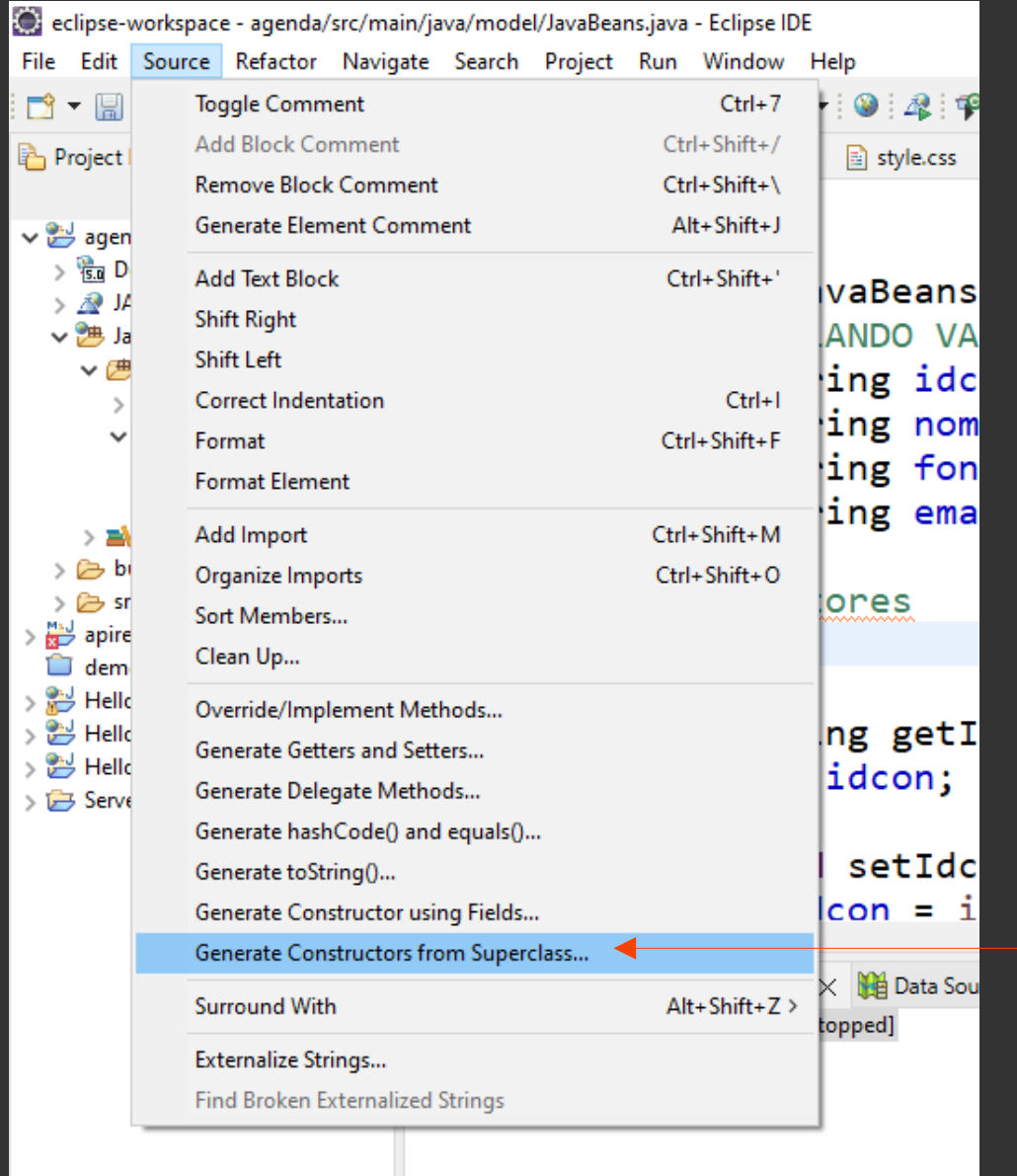
Abra um
espaço para
os
construtores

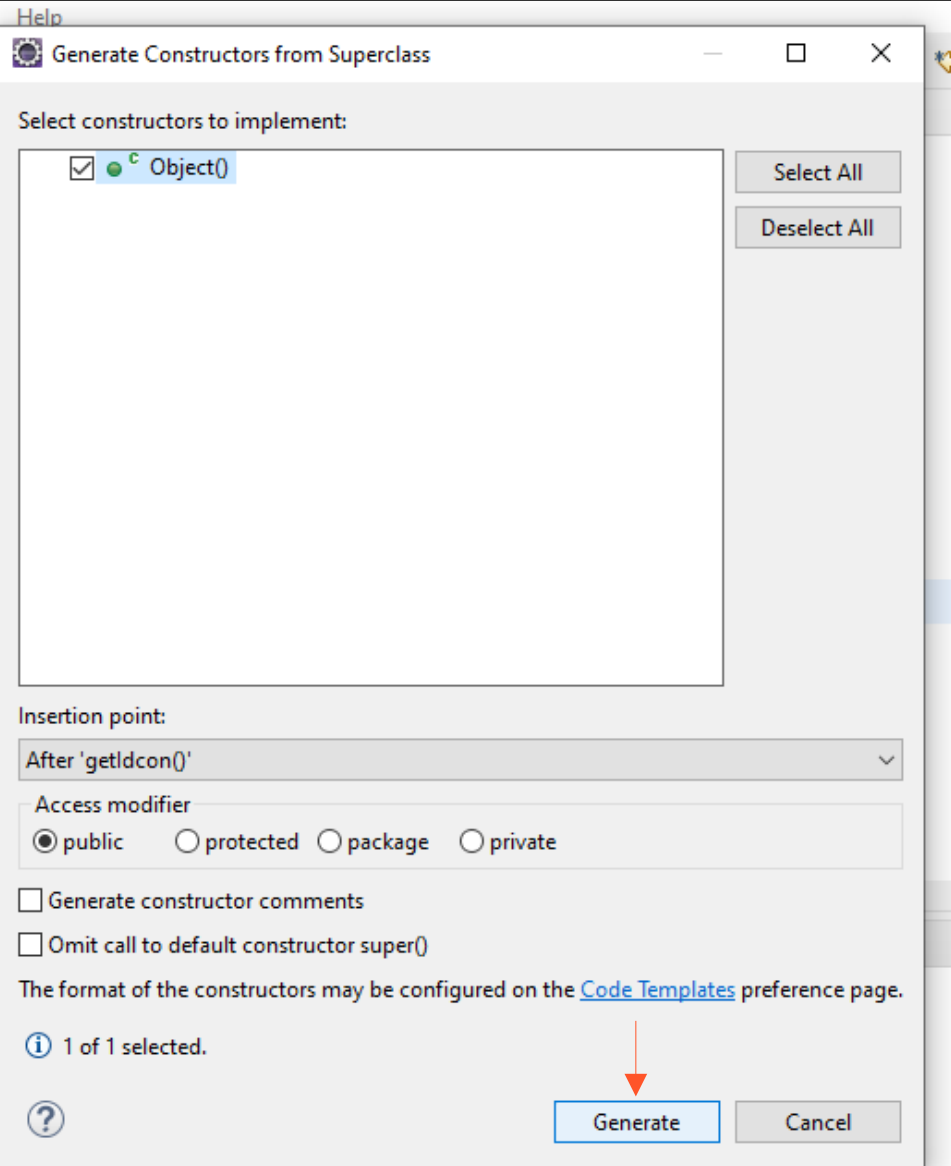


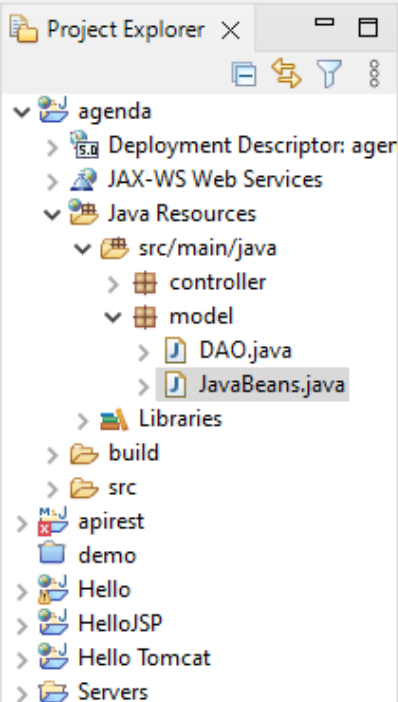
```
1 package model;
2
3 public class JavaBeans {
4     // ENCAPSULANDO VARIAVEIS
5     private String idcon;
6     private String nome;
7     private String fone;
8     private String email;
9
10    // construtores
11
12
13    public String getIdcon() {
14        return idcon;
15    }
16    public void setIdcon(String idcon) {
17        this.idcon = idcon;
```

Markers Properties Servers Data Source Explorer Snippets Console

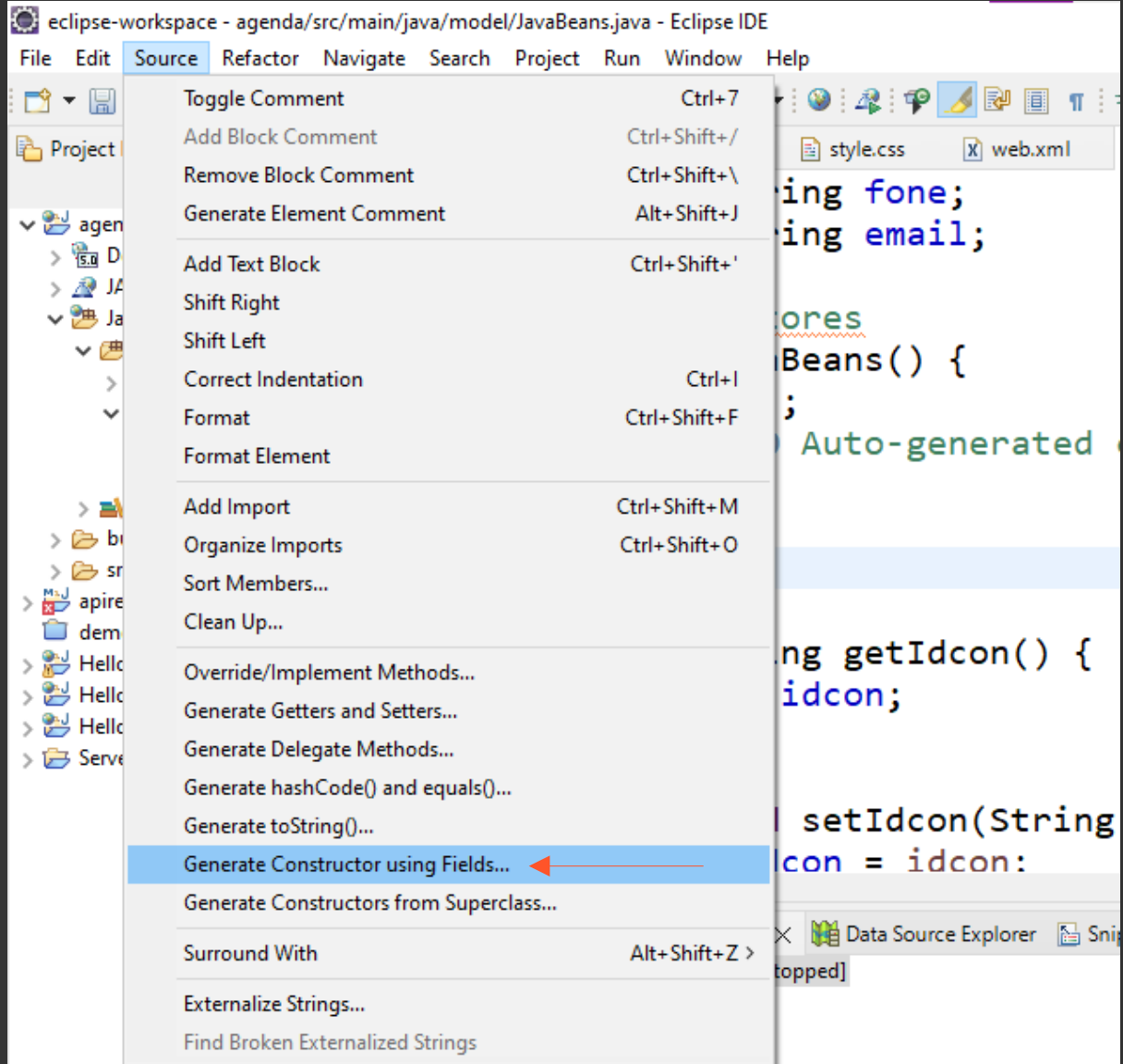
> Tomcat v10.0 Server at localhost [Stopped]







```
Controller.java  index.html  style.css  web.xml  JavaBeans.java X
7      private String fone;
8      private String email;
9
10     // construtores
11     public JavaBeans() {
12         super();
13         // TODO Auto-generated constructor stub
14     }
15
16     public String getIdcon() {
17         return idcon;
18     }
19
20     public void setIdcon(String idcon) {
21         this.idcon = idcon;
22     }
23     public String getNome() {
```



Generate Constructor using Fields

Select super constructor to invoke:

Object()

Select fields to initialize:

<input checked="" type="checkbox"/>	<input type="checkbox"/>	idcon
<input checked="" type="checkbox"/>	<input type="checkbox"/>	nome
<input checked="" type="checkbox"/>	<input type="checkbox"/>	fone
<input checked="" type="checkbox"/>	<input type="checkbox"/>	email

Select All

Deselect All

Up

Down

Insertion point:

After 'JavaBeans()'


Access modifier


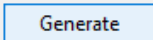
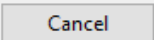
☒ public ☐ protected ☐ package ☐ private

☐ Generate constructor comments

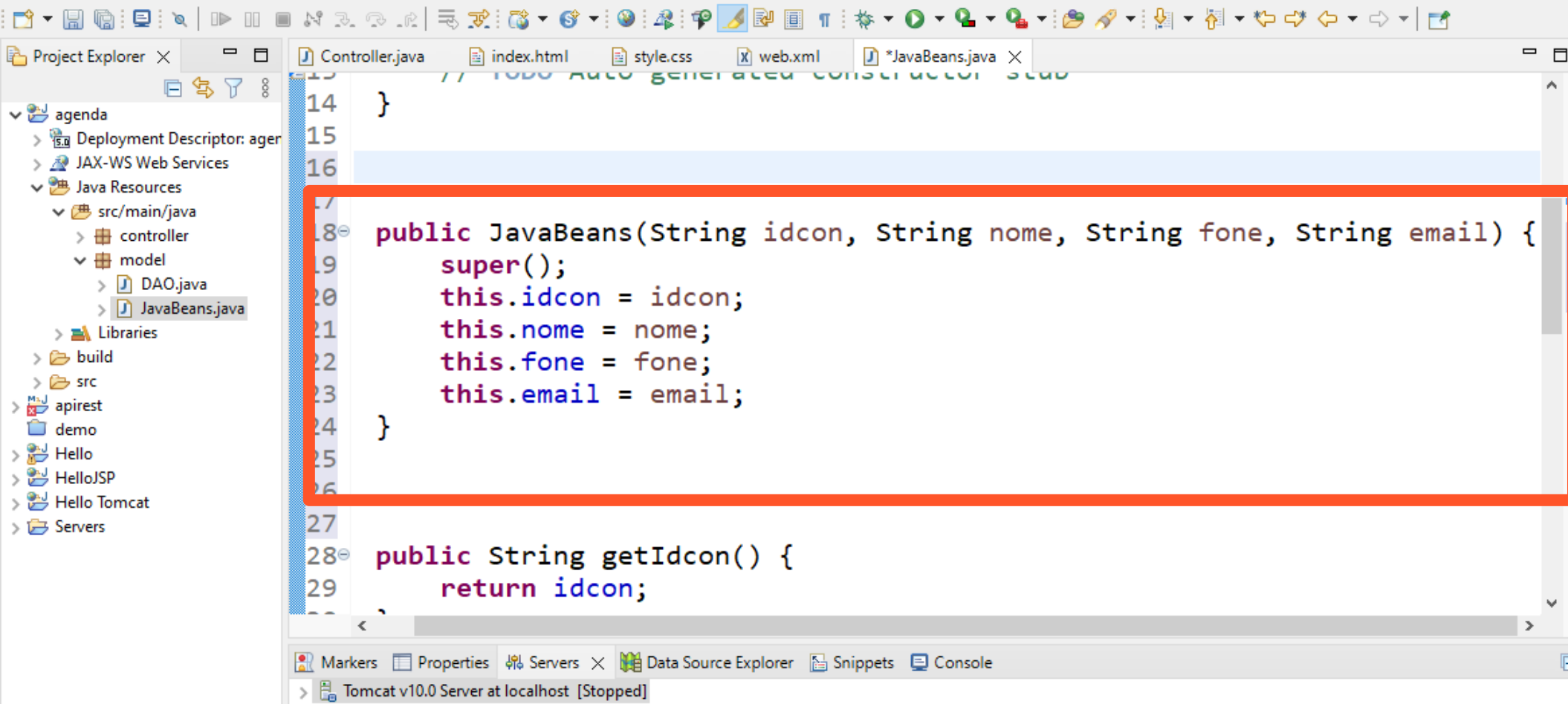
☐ Omit call to default constructor super()

The format of the constructors may be configured on the [Code Templates](#) preference page.

 4 of 4 selected.

A red arrow points from the text "The format of the constructors may be configured on the Code Templates preference page." to the "Generate" button.



The screenshot displays the Eclipse IDE interface. On the left, the Project Explorer shows the project structure: **agenda** > **Deployment Descriptor: ager** > **JAX-WS Web Services** > **Java Resources** > **src/main/java** > **controller** > **model** > **DAO.java** > **JavaBeans.java**. The main editor window shows the code for **JavaBeans.java**. The code is as follows:

```
// TODO Auto-generated constructor stub
14 }
15
16
17
18 public JavaBeans(String idcon, String nome, String fone, String email) {
19     super();
20     this.idcon = idcon;
21     this.nome = nome;
22     this.fone = fone;
23     this.email = email;
24 }
25
26
27
28 public String getIdcon() {
29     return idcon;
30 }
```

The status bar at the bottom indicates the server status: **Tomcat v10.0 Server at localhost [Stopped]**.

