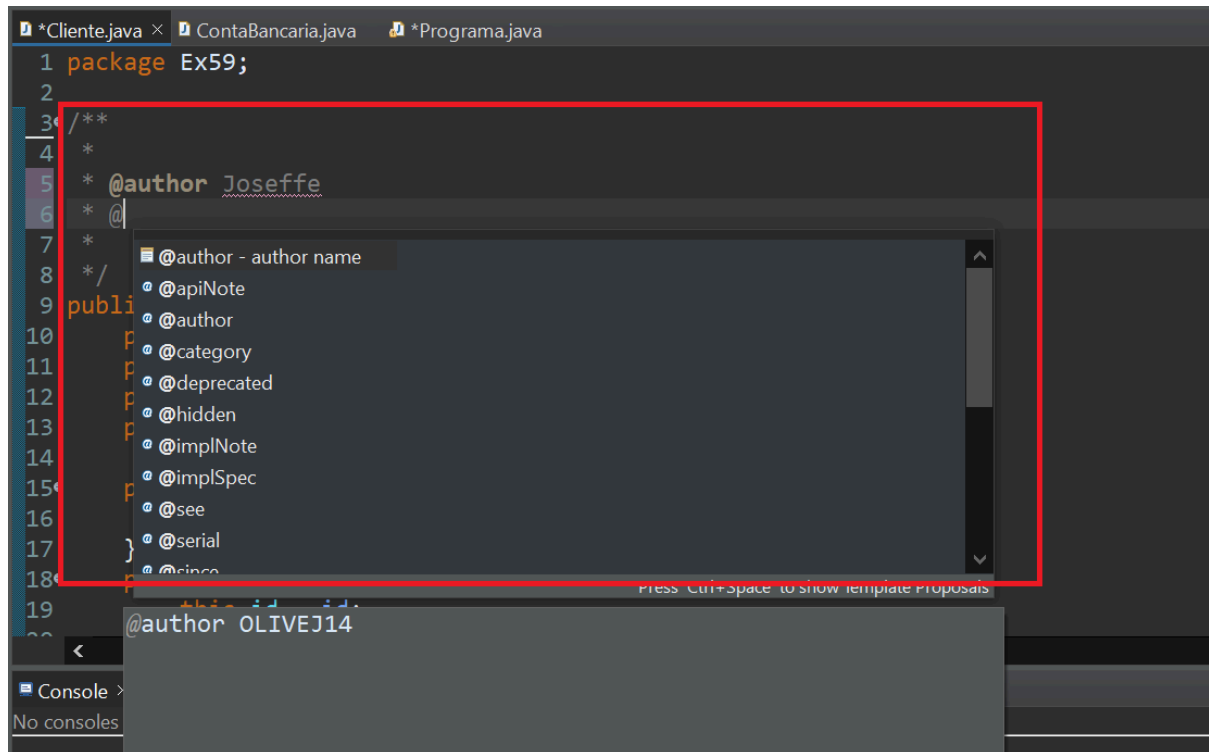


## Aula 19 - JavaDoc

O JavaDoc é um recurso de API do Java responsável para documentar classes e métodos com seus objetivos, autores, versões, etc.

Para utilizar o JavaDoc basta posicionar o curso acima de uma classe ou método e iniciar a documentação com os caracteres `/**`

Com isso, o JavaDoc nos dá várias opções para documentação, como:



### 1. Documentando Classes

Vamos documentar uma classe citando o seu objetivo e seu autor:

```
*Cliente.java  ContaBancaria.java  *Programa.java
1 package Ex59;
2
3 /**
4  * Classe responsavel por representar as Contas Bancarias.
5  *
6  * @author Joseffe Oliveira
7  *
8  */
9 public class ContaBancaria {
10     private String agencia;
11     private String numero;
12     private double saldo;
13
14     public String getAgencia() {
15         return agencia;
16     }
17     public void setAgencia(String agencia) {
18         this.agencia = agencia;
19     }
20 }
```

Com isso, no momento de desenvolvimento, já podemos utilizar o conteúdo documentado apenas passando o mouse sobre o nome da classe, veja:

```
*Cliente.java  ContaBancaria.java  *Programa.java
48     }
49     cliente.setId(id_cliente);
50
51     System.out.print("Digite o seu nome: ");
52     cliente.setNome(ler.next());
53
54     System.out.print("Digite a seu idade: ");
55     cliente.setIdade(ler.nextInt());
56
57     ContaBancaria conta = new ContaBancaria();
58
59     Ex59.ContaBancaria
60     ");
61     Classe responsavel por representar as Contas Bancarias.
62     Author:
63     Joseffe Oliveira
64
65     Press 'F2' for focus
66     ");
```

## 2. Documentando Métodos

Agora vamos documentar um método citando o seu objetivo e seu autor também:

```
*Cliente.java  ContaBancaria.java  *Programa.java
34 public ContaBancaria(String agencia, String numero) {
35     this.agencia = agencia;
36     this.numero = numero;
37     this.saldo = 0;
38 }
39
40 /**
41  * Método responsável por depositar um valor na conta
42  *
43  * @param valor Valor que será depositado
44  */
45 public void Depositar(double valor) {
46     this.saldo = this.saldo + valor;
47     //this.saldo += valor;
48 }
49
50 public void Sacar(double valor) {
51     this.saldo = this.saldo - valor;
52     //this.saldo -= valor;
53 }
```

Com isso, no momento de desenvolvimento, já podemos utilizar o conteúdo documentado apenas passando o mouse sobre o nome da classe, veja:

```
*Cliente.java  ContaBancaria.java  *Programa.java
81 for(Cliete c: listaClientes) {
82     if (c.getId() == id) {
83         indexCliente = listaClientes.indexOf(c);
84         break;
85     }
86 }
87
88 if (indexCliente != -1) {
89     cli = listaClientes.get(indexCliente);
90
91     System.out.print("Digite o valor do depósito: ");
92     valor = ler.nextDouble();
93
94     cli.getConta().Depositar(valor);
95
96     System.out.println("Deposito realizado com sucesso!");
97 }
98 else {
99     System.out.println("Cliente não encontrado!");
100 }
```

void Ex59.ContaBancaria.Depositar(double valor)

Método responsável por depositar um valor na conta

Parameters:

valor Valor que será depositado

Press 'F2' for focus

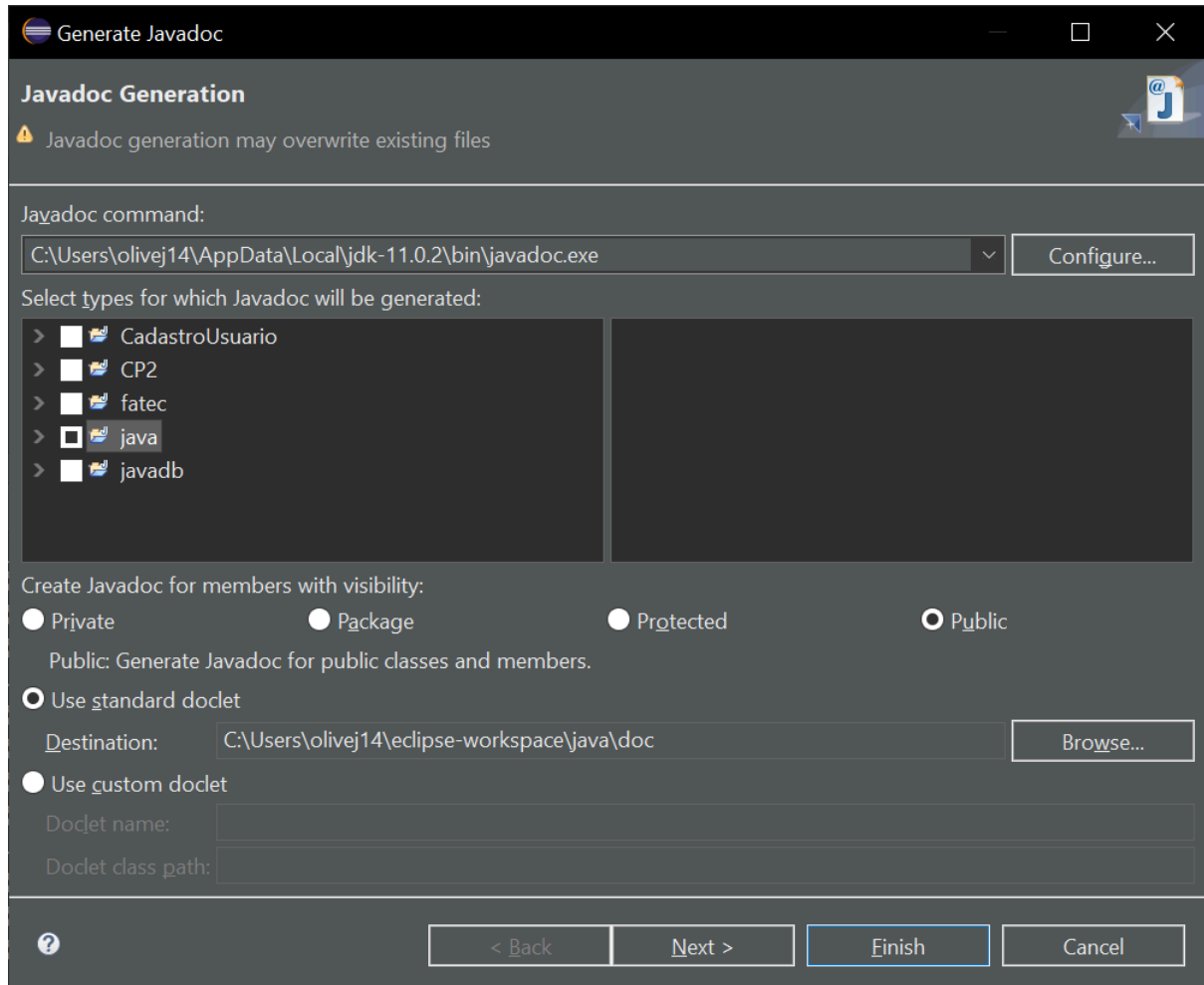
Console ×

No consoles to display at this time.

### 3. Gerando um documento do Javadoc

Ao final é possível gerar um documento contendo todas as documentações de classes e métodos em:

Project > Generate Javadoc > Selecione o seu projeto/package > Next e Finish



Com isso, teremos o seguinte resultado:

```
Cliente.java  ContaBancaria.java  Programa.java x
81         for(Cliente c: listaClientes) {
82             if (c.getId() == id) {
83                 indexCliente = listaClientes.indexOf(c);
84                 break;
85             }
86         }
87
88         if (indexCliente != -1) {
89             cli = listaClientes.get(indexCliente);
90         }
91     }
92 }

Console x
<terminated> Javadoc Generation
Generating C:\Users\olivej14\eclipse-workspace\java\doc\constant-values.html...
Generating C:\Users\olivej14\eclipse-workspace\java\doc\Ex59\class-use\Programa.html...
Generating C:\Users\olivej14\eclipse-workspace\java\doc\Ex59\package-use.html...
Building index for all the packages and classes...
Generating C:\Users\olivej14\eclipse-workspace\java\doc\overview-tree.html...
Generating C:\Users\olivej14\eclipse-workspace\java\doc\index-files\index-1.html...
Generating C:\Users\olivej14\eclipse-workspace\java\doc\index-files\index-2.html...
Generating C:\Users\olivej14\eclipse-workspace\java\doc\index-files\index-3.html...
Building index for all classes...
Generating C:\Users\olivej14\eclipse-workspace\java\doc\allclasses-index.html...
Generating C:\Users\olivej14\eclipse-workspace\java\doc\allpackages-index.html...
Generating C:\Users\olivej14\eclipse-workspace\java\doc\deprecated-list.html...
Building index for all classes...
Generating C:\Users\olivej14\eclipse-workspace\java\doc\allclasses.html...
Generating C:\Users\olivej14\eclipse-workspace\java\doc\allclasses.html...
Generating C:\Users\olivej14\eclipse-workspace\java\doc\index.html...
Generating C:\Users\olivej14\eclipse-workspace\java\doc\help-doc.html...
```

E iremos gerar o seguinte doc no diretório acima:

file:///C:/Users/olivej14/eclipse-workspace/java/doc/Ex59/Programa.html

Getting Started | Whirlpool | Estudos | Integração | Particular | JIRA | Nave | Confluence | DevSecOps Roadmap...

PACKAGE **CLASS** USE TREE DEPRECATED INDEX HELP

ALL CLASSES SEARCH:

SUMMARY: NESTED | FIELD | CONSTR | METHOD    DETAIL: FIELD | CONSTR | METHOD

---

**Package** Ex59

**Class** Programa

java.lang.Object  
Ex59.Programa

---

public class Programa  
extends java.lang.Object

---

**Constructor Summary**

**Constructors**

Constructor	Description
Programa()	

---

**Method Summary**