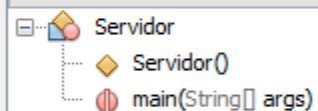
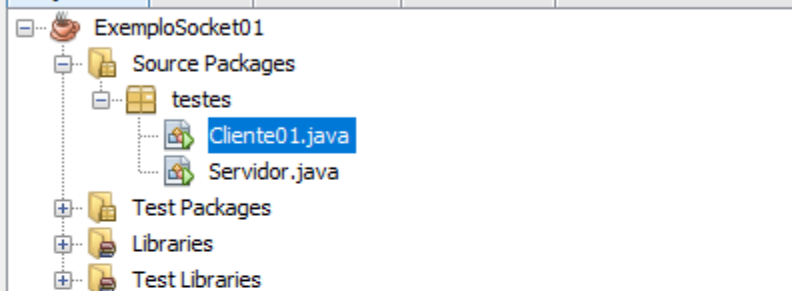


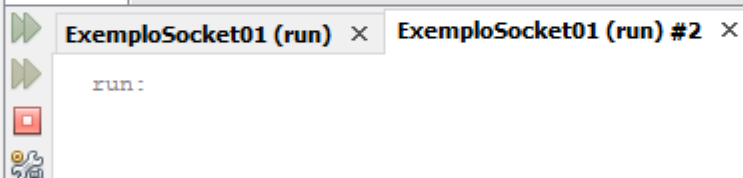
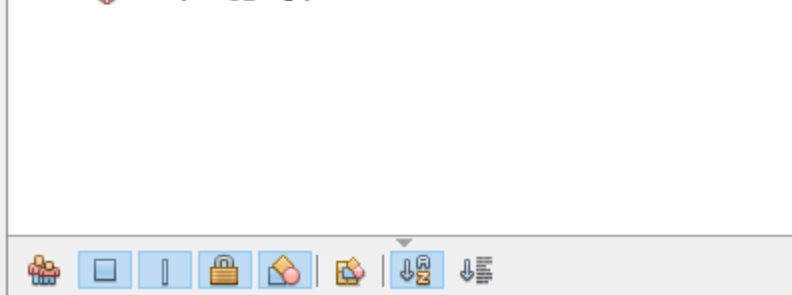
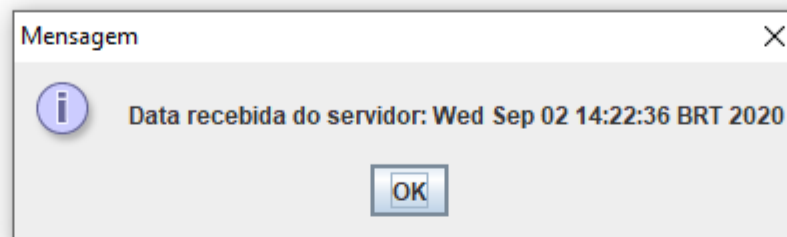
```
1 package testes;
2
3 import java.net.*;
4 import java.io.*;
5 import java.util.Date;
6
7 public class Servidor {
8     public static void main(String[] args) {
9         try {
10             ServerSocket servidor = new ServerSocket(12345);
11             System.out.println("Servidor ativo, aguardando conexões...");
12
13             while(true) {
14                 Socket cliente = servidor.accept();
15                 System.out.println("IP do cliente: " + cliente.getInetAddress().getHostAddress());
16
17                 ObjectOutputStream saida = new ObjectOutputStream(cliente.getOutputStream());
18                 saida.flush();
19                 saida.writeObject(new Date());
20
21                 saida.close();
22                 cliente.close();
23             }
24         } catch (Exception e) {
25             System.out.println("Erro no servidor " + e.getMessage());
26             e.printStackTrace();
27         }
28     }
29 }
30
31
32
33
34
```



```
run:
Servidor ativo, aguardando conexões...
```



```
1 package testes;
2
3 import java.net.*;
4 import java.io.*;
5 import java.util.Date;
6 import javax.swing.JOptionPane;
7
8 public class Cliente01 {
9     public static void main(String[] args) {
10         try {
11             Socket cliente = new Socket("localhost", 12345);
12
13             ObjectInputStream entrada = new ObjectInputStream(cliente.getInputStream());
14             Date dataDoServidor = (Date) entrada.readObject();
15             JOptionPane.showMessageDialog(null, "Data recebida do servidor: " + dataDoServidor.toString());
16
17             entrada.close();
18             System.out.println("Conexão encerrada.");
19
20         } catch (Exception e) {
21             System.out.println("Erro no cliente: " + e.getMessage());
22             e.printStackTrace();
23         }
24     }
25 }
26
```



Projects × **Files** **Services** **Favorites**

- ExemploSocket01
 - Source Packages
 - testes
 - Cliente01.java
 - Servidor.java
 - Test Packages
 - Libraries
 - Test Libraries

Navigador ×

Members <empty>

 - Cliente01
 - Cliente01()
 - main(String[] args)

Source History

```
1 package testes;
2
3 import java.net.*;
4 import java.io.*;
5 import java.util.Date;
6 import javax.swing.JOptionPane;
7
8 public class Cliente01 {
9     public static void main(String[] args) {
10         try {
11             Socket cliente = new Socket("localhost", 12345);
12
13             ObjectInputStream entrada = new ObjectInputStream(cliente.getInputStream());
14             Date dataDoServidor = (Date) entrada.readObject();
15             JOptionPane.showMessageDialog(null, "Data recebida do servidor: " + dataDoServidor.toString());
16
17             entrada.close();
18             System.out.println("Conexão encerrada.");
19
20         } catch (Exception e) {
21             System.out.println("Erro no cliente: " + e.getMessage());
22             e.printStackTrace();
23         }
24     }
25 }
26
```

Output ×

ExemploSocket01 (run) × ExemploSocket01 (run) #2 ×

```
run:
Conexão encerrada.
BUILD SUCCESSFUL (total time: 14 seconds)
```

Projects × **Files** **Services** **Favorites**

- ExemploSocket01
 - Source Packages
 - testes
 - Cliente01.java
 - Servidor.java
 - Test Packages
 - Libraries
 - Test Libraries

Navigador ×

Members <empty>

 - Cliente01
 - Cliente01()
 - main(String[] args)

Icons: Run, Stop, Step Over, Step Into, Step Out, etc.

Source History

```
1 package testes;
2
3 import java.net.*;
4 import java.io.*;
5 import java.util.Date;
6 import javax.swing.JOptionPane;
7
8 public class Cliente01 {
9     public static void main(String[] args) {
10         try {
11             Socket cliente = new Socket("localhost", 12345);
12
13             ObjectInputStream entrada = new ObjectInputStream(cliente.getInputStream());
14             Date dataDoServidor = (Date) entrada.readObject();
15             JOptionPane.showMessageDialog(null, "Data recebida do servidor: " + dataDoServidor.toString());
16
17             entrada.close();
18             System.out.println("Conexão encerrada.");
19
20         } catch (Exception e) {
21             System.out.println("Erro no cliente: " + e.getMessage());
22             e.printStackTrace();
23         }
24     }
25 }
26
```

Output ×

ExemploSocket01 (run) × ExemploSocket01 (run) #2 ×

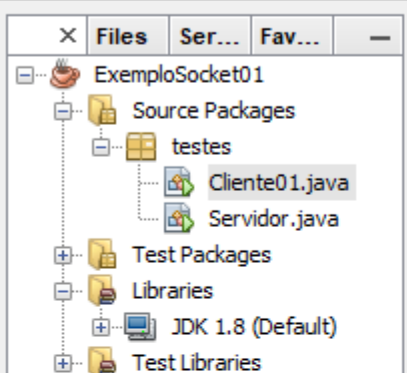
run:
Servidor ativo, aguardando conexões...
IP do cliente: 127.0.0.1



<default config>



510,2/721,0MB



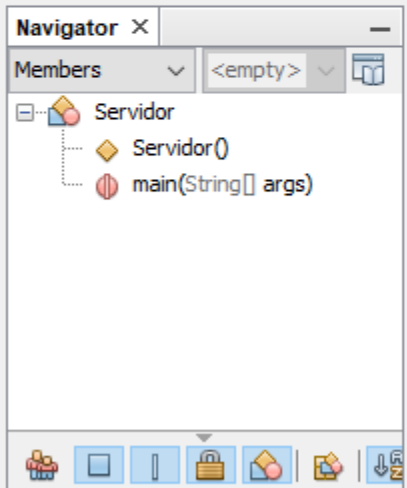
Start Page x Servidor.java x Cliente01.java x

Source

History

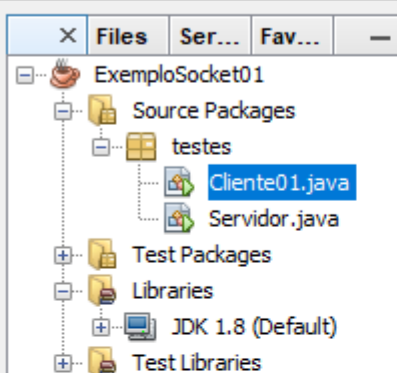


```
1 package testes;
2
3 import java.net.*;
4 import java.io.*;
5 import java.util.Date;
6
7 public class Servidor {
8     public static void main(String[] args) {
9         try {
10             ServerSocket servidor = new ServerSocket(12345);
11             System.out.println("Servidor ativo, aguardando conexões...");
12
13             while(true) {
14                 Socket cliente = servidor.accept();
15                 System.out.println("IP do cliente: " + cliente.getInetAddress().getHostAddress());
16
17                 ObjectInputStream dadosDoCliente = new ObjectInputStream(cliente.getInputStream());
18                 String nome = (String) dadosDoCliente.readObject();
19                 System.out.println("Nome do usuário conectado: " + nome);
20
21                 ObjectOutputStream saida = new ObjectOutputStream(cliente.getOutputStream());
22                 saida.flush();
23                 saida.writeObject(new Date());
24
25                 saida.close();
26                 cliente.close();
27             }
28
29         } catch (Exception e) {
30             System.out.println("Erro no servidor " + e.getMessage());
31             e.printStackTrace();
32         }
33     }
34 }
35
36 }
```



Output - ExemploSocket01 (run) x

```
run:
Servidor ativo, aguardando conexões...
```

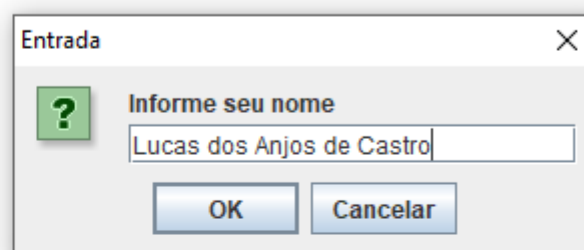



Start Page x Servidor.java x Cliente01.java x

Source

History

```
1 package testes;
2
3 import java.net.*;
4 import java.io.*;
5 import java.util.Date;
6 import javax.swing.JOptionPane;
7
8 public class Cliente01 {
9     public static void main(String[] args) {
10         try {
11             Socket cliente = new Socket("localhost", 12345);
12
13             String nomeCliente = JOptionPane.showInputDialog("Informe seu nome");
14             ObjectOutputStream dadosParaServidor = new ObjectOutputStream(cliente.getOutputStream());
15             dadosParaServidor.flush();
16             dadosParaServidor.writeObject(nomeCliente);
17
18             ObjectInputStream entrada = new ObjectInputStream(cliente.getInputStream());
19             Date dataDoServidor = (Date) entrada.readObject();
20             JOptionPane.showMessageDialog(null, "Data recebida do servidor: " + dataDoServidor.toString());
21
22             entrada.close();
23             System.out.println("Conexão encerrada.");
24
25         } catch (Exception e) {
26             System.out.println("Erro no cliente: " + e.getMessage());
27             e.printStackTrace();
28         }
29     }
30 }
31
```



Navigator x

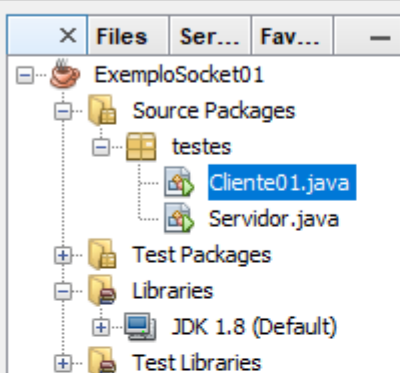
Members <empty>

Cliente01
 Cliente01()
 main(String[] args)

Output x

ExemploSocket01 (run) x ExemploSocket01 (run) #2 x

run:



Start Page X Servidor.java X Cliente01.java X

Source

History

```
1 package testes;
2
3 import java.net.*;
4 import java.io.*;
5 import java.util.Date;
6 import javax.swing.JOptionPane;
7
8 public class Cliente01 {
9     public static void main(String[] args){
10         try {
11             Socket cliente = new Socket("localhost", 12345);
12
13             String nomeCliente = JOptionPane.showInputDialog("Informe seu nome");
14             ObjectOutputStream dadosParaServidor = new ObjectOutputStream(cliente.getOutputStream());
15             dadosParaServidor.flush();
16             dadosParaServidor.writeObject(nomeCliente);
17
18             ObjectInputStream entrada = new ObjectInputStream(cliente.getInputStream());
19             Date dataDoServidor = (Date) entrada.readObject();
20             JOptionPane.showMessageDialog(null, "Data recebida do servidor: " + dataDoServidor.toString());
21
22             entrada.close();
23             System.out.println("Conexão encerrada.");
24
25         } catch (Exception e) {
26             System.out.println("Erro no cliente: " + e.getMessage());
27             e.printStackTrace();
28         }
29     }
30 }
31
```

Mensagem



Data recebida do servidor: Wed Sep 02 14:19:01 BRT 2020

OK

Navigator X

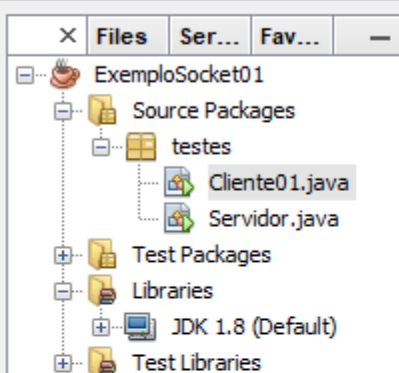
Members <empty>

Cliente01
 Cliente01()
 main(String[] args)

Output X

ExemploSocket01 (run) X ExemploSocket01 (run) #2 X

run:



Start Page x Servidor.java x Cliente01.java x

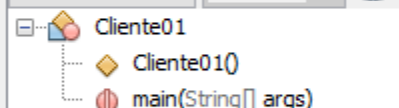
Source

History

```
1 package testes;
2
3 import java.net.*;
4 import java.io.*;
5 import java.util.Date;
6 import javax.swing.JOptionPane;
7
8 public class Cliente01 {
9     public static void main(String[] args) {
10         try {
11             Socket cliente = new Socket("localhost", 12345);
12
13             String nomeCliente = JOptionPane.showInputDialog("Informe seu nome");
14             ObjectOutputStream dadosParaServidor = new ObjectOutputStream(cliente.getOutputStream());
15             dadosParaServidor.flush();
16             dadosParaServidor.writeObject(nomeCliente);
17
18             ObjectInputStream entrada = new ObjectInputStream(cliente.getInputStream());
19             Date dataDoServidor = (Date) entrada.readObject();
20             JOptionPane.showMessageDialog(null, "Data recebida do servidor: " + dataDoServidor.toString());
21
22             entrada.close();
23             System.out.println("Conexão encerrada.");
24
25         } catch (Exception e) {
26             System.out.println("Erro no cliente: " + e.getMessage());
27             e.printStackTrace();
28         }
29     }
30 }
31
```

Navigator x

Members <empty>



Output x

ExemploSocket01 (run) x ExemploSocket01 (run) #2 x

```
run:
Conexão encerrada.
BUILD SUCCESSFUL (total time: 43 seconds)
```


Files Ser... Fav... -

ExemploSocket01

- Source Packages
 - testes
 - Cliente01.java
 - Servidor.java
- Test Packages
- Libraries
 - JDK 1.8 (Default)
- Test Libraries

Navigador

Members <empty>

- Cliente01
 - Cliente01()
 - main(String[] args)

Start Page x Servidor.java x Cliente01.java x

Source History

```
1 package testes;
2
3 import java.net.*;
4 import java.io.*;
5 import java.util.Date;
6 import javax.swing.JOptionPane;
7
8 public class Cliente01 {
9     public static void main(String[] args){
10         try {
11             Socket cliente = new Socket("localhost", 12345);
12
13             String nomeCliente = JOptionPane.showInputDialog("Informe seu nome");
14             ObjectOutputStream dadosParaServidor = new ObjectOutputStream(cliente.getOutputStream());
15             dadosParaServidor.flush();
16             dadosParaServidor.writeObject(nomeCliente);
17
18             ObjectInputStream entrada = new ObjectInputStream(cliente.getInputStream());
19             Date dataDoServidor = (Date) entrada.readObject();
20             JOptionPane.showMessageDialog(null, "Data recebida do servidor: " + dataDoServidor.toString());
21
22             entrada.close();
23             System.out.println("Conexão encerrada.");
24
25         } catch (Exception e) {
26             System.out.println("Erro no cliente: " + e.getMessage());
27             e.printStackTrace();
28         }
29     }
30 }
31
```

Output x

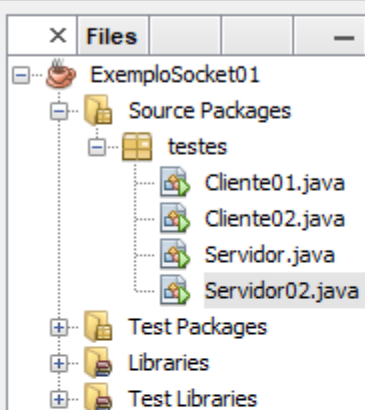
ExemploSocket01 (run) x ExemploSocket01 (run) #2 x

run:

Servidor ativo, aguardando conexões...

IP do cliente: 127.0.0.1

Nome do usuário conectado: Lucas dos Anjos de Castro



Cliente01.java x Servidor.java x Servidor02.java x Cliente02.java x

Source

History




```
1 package testes;
2
3 import java.io.IOException;
4 import java.net.ServerSocket;
5 import java.net.Socket;
6 import java.util.Scanner;
7
8
9
10 public class Servidor02 {
11     public static void main(String[] args) throws IOException {
12         ServerSocket servidor = new ServerSocket(12345);
13         System.out.println("Porta 12345 aberta e aguardando conexões...");
14
15         Socket cliente = servidor.accept();
16         System.out.println("Cliente conectado: " + cliente.getInetAddress().getHostAddress());
17
18         Scanner entrada = new Scanner(cliente.getInputStream());
19         while (entrada.hasNextLine()) {
20             System.out.println(entrada.nextLine());
21         }
22
23         entrada.close();
24         servidor.close();
25     }
26 }
27
```

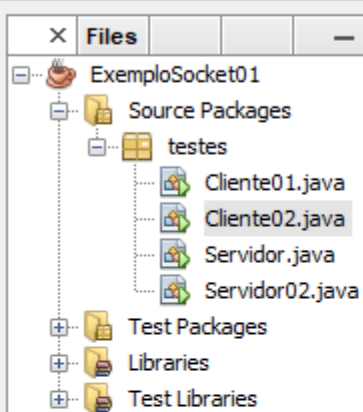
Output - ExemploSocket01 (run) x

```
run:
Porta 12345 aberta e aguardando conexões...
```

Navigator x

Members

 Servidor02
 Servidor02()
 main(String[] args)



Source History

```
1 package testes;
2
3 import java.io.IOException;
4 import java.io.PrintStream;
5 import java.net.Socket;
6 import java.net.UnknownHostException;
7 import java.util.Scanner;
8
9 public class Cliente02 {
10     public static void main(String[] args) throws UnknownHostException, IOException {
11         Socket cliente = new Socket("127.0.0.1", 12345);
12         System.out.println("O cliente se conectou ao servidor!");
13
14         Scanner teclado = new Scanner(System.in);
15         PrintStream saida = new PrintStream(cliente.getOutputStream());
16
17         while (teclado.hasNextLine()) {
18             saida.println(teclado.nextLine());
19         }
20
21         saida.close();
22         teclado.close();
23     }
24 }
25
26
```

Output

ExemploSocket01 (run) ExemploSocket01 (run) #2

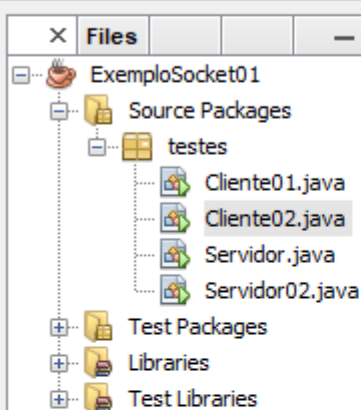
```
run:
O cliente se conectou ao servidor!

ola
teste
testando
1
2
3
```

Navigador

Members

- Cliente02
 - Cliente02()
 - main(String[] args)



```
1 package testes;
2
3 import java.io.IOException;
4 import java.io.PrintStream;
5 import java.net.Socket;
6 import java.net.UnknownHostException;
7 import java.util.Scanner;
8
9 public class Cliente02 {
10     public static void main(String[] args) throws UnknownHostException, IOException {
11         Socket cliente = new Socket("127.0.0.1", 12345);
12         System.out.println("O cliente se conectou ao servidor!");
13
14         Scanner teclado = new Scanner(System.in);
15         PrintStream saida = new PrintStream(cliente.getOutputStream());
16
17         while (teclado.hasNextLine()) {
18             saida.println(teclado.nextLine());
19         }
20
21         saida.close();
22         teclado.close();
23     }
24 }
25
26
```

Output

ExemploSocket01 (run) × ExemploSocket01 (run) #2 ×

```
run:
Porta 12345 aberta e aguardando conexões...
Cliente conectado: 127.0.0.1

ola
teste
testando
1
2
3
BUILD SUCCESSFUL (total time: 1 minute 5 seconds)
```

