

Descargar JAVA SDK 11

<https://www.oracle.com/java/technologies/javase/jdk11-archive-downloads.html>

ORACLE		Products	Industries	Resources	Customers	Partners	Developers	Events	Company	View Accounts	Contact Sales
Linux x64 Debian Package		138.40 MB								 jdk-11.0.16.1_linux-x64_bin.deb	
Linux x64 RPM Package		144.59 MB								 jdk-11.0.16.1_linux-x64_bin.rpm	
Linux x64 Compressed Archive		161.08 MB								 jdk-11.0.16.1_linux-x64_bin.tar.gz	
macOS Arm 64 Compressed Archive		153.36 MB								 jdk-11.0.16.1_macos-aarch64_bin.tar.gz	
macOS x64 DMG Installer		152.83 MB								 jdk-11.0.16.1_macos-aarch64_bin.dmg	
macOS x64 Compressed Archive		155.48 MB								 jdk-11.0.16.1_macos-x64_bin.tar.gz	
x64 DMG Installer		154.96 MB								 jdk-11.0.16.1_macos-x64_bin.dmg	
Solaris SPARC Compressed Archive		184.66 MB								 jdk-11.0.16.1_solaris-sparcv9_bin.tar.gz	
 Windows x64 Installer		140.55 MB								 jdk-11.0.16.1_windows-x64_bin.exe	
Windows x64 Compressed Archive		158.30 MB								 jdk-11.0.16.1_windows-x64_bin.zip	

Descargar JAVA SDK 11

<https://www.oracle.com/java/technologies/javase/jdk11-archive-downloads.html>

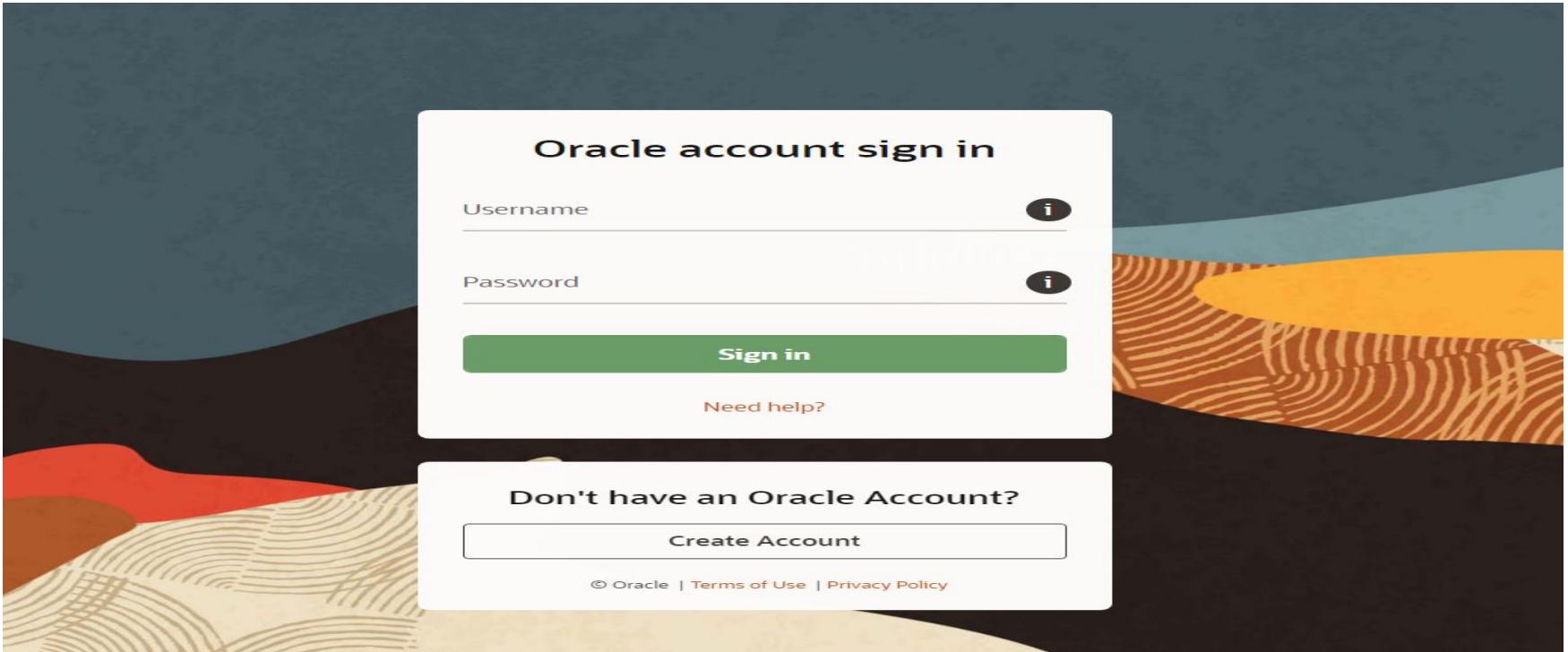
The screenshot shows the Oracle Java SE Downloads page. The main content area displays a table with two rows of download options:

Format	File Name	Size	Action
Compressed Archive	jdk-11.0.16.1_linux-x64_bin.tar.gz	161.08 MB	Download

Below the table, there is a note: "You must accept the Oracle Technology Network License Agreement for Oracle Java SE to download this software." A checkbox labeled "I reviewed and accept the Oracle Technology Network License Agreement for Oracle Java SE" is checked, with the note "Required" underneath it. A green button at the bottom right says "Download jdk-11.0.16.1_windows-x64_bin.exe".

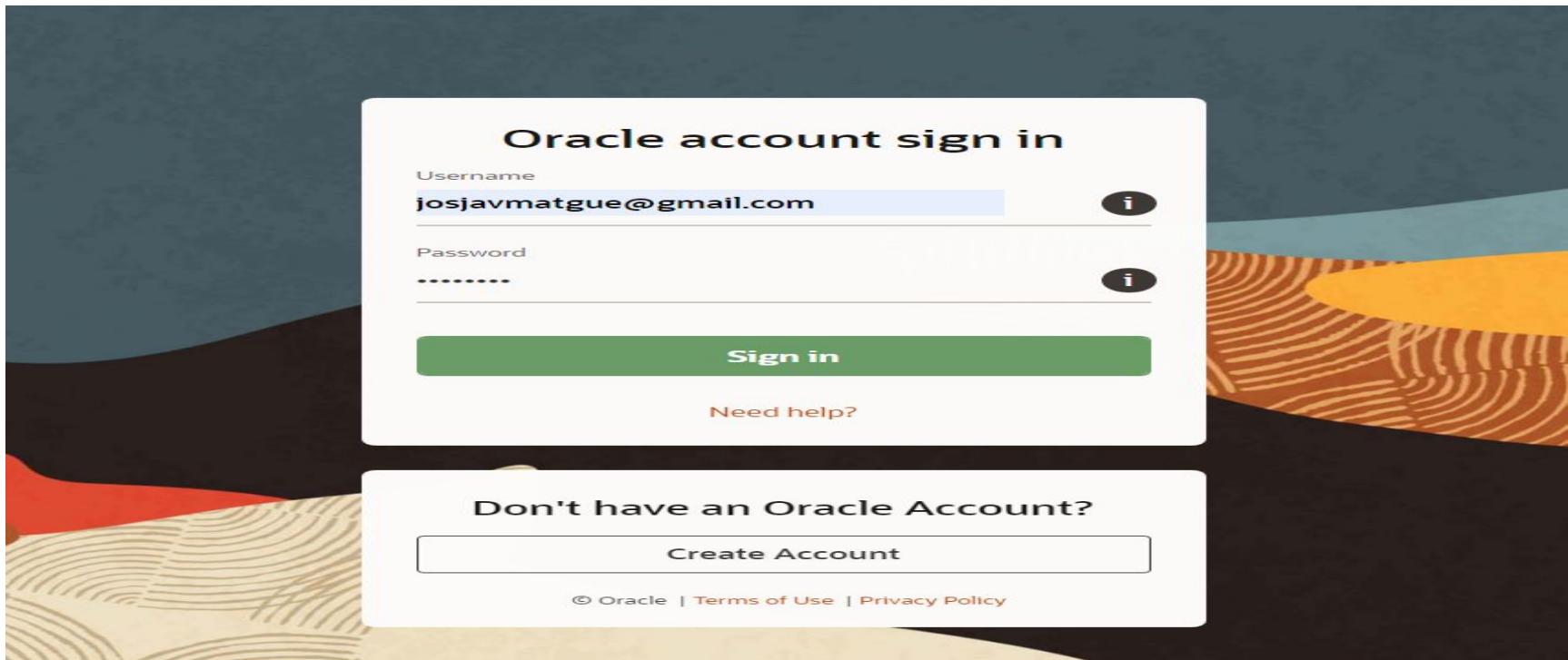
Descargar JAVA SDK 11

<https://www.oracle.com/java/technologies/javase/jdk11-archive-downloads.html>



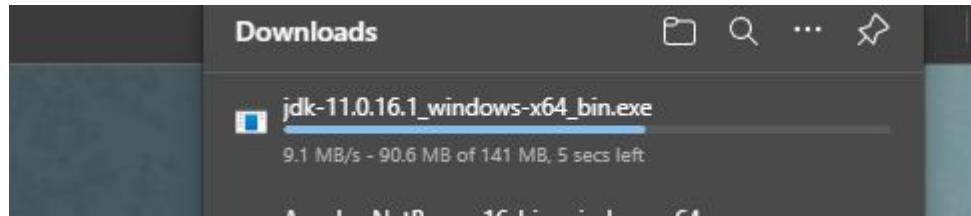
Descargar JAVA SDK 11

<https://www.oracle.com/java/technologies/javase/jdk11-archive-downloads.html>



Descargar JAVA SDK 11

<https://www.oracle.com/java/technologies/javase/jdk11-archive-downloads.html>



Descargar JAVA SDK 11

<https://www.oracle.com/java/technologies/javase/jdk11-archive-downloads.html>



Descargar JAVA SDK 11

<https://www.oracle.com/java/technologies/javase/jdk11-archive-downloads.html>



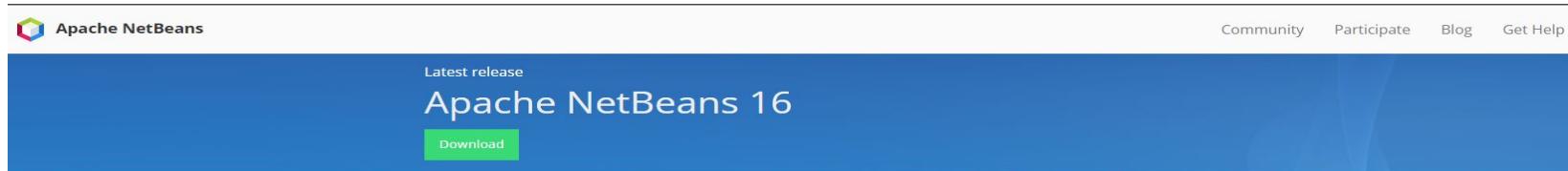
Descargar JAVA SDK 11

<https://www.oracle.com/java/technologies/javase/jdk11-archive-downloads.html>



Descargar NetBeans

<https://netbeans.apache.org/download/index.html>



The screenshot shows the Apache NetBeans download page. At the top, there's a navigation bar with the Apache NetBeans logo, 'Community', 'Participate', 'Blog', and 'Get Help'. Below the navigation, a large blue banner features the text 'Latest release' and 'Apache NetBeans 16' in white. A green 'Download' button is positioned below the main title. The background of the page has a subtle blue gradient.

Apache NetBeans Releases

Apache NetBeans is released four times a year. For details, see [full release schedule](#).

Apache NetBeans 16

Latest version of the IDE, released on November 30, 2022.

[Download](#)

Older releases

Older Apache NetBeans releases can still be downloaded, but are no longer supported.

[Find out more](#)

Daily builds and building from source

All Apache NetBeans source code is freely available to build yourself, or you can download (unsupported) daily development builds.

[Find out more](#)

Descargar NetBeans

<https://netbeans.apache.org/download/index.html>

NetBeans

Community Participate

Downloading Apache NetBeans 16

Apache NetBeans 16 was released on November 30, 2022.

Apache NetBeans 16 is available for download from your closest Apache mirror.

Binaries (Platform Independent):

- [netbeans-16-bin.zip \(SHA-512, PGP ASC\)](#)

Installers and Packages:

- [Apache-NetBeans-16-bin-windows-x64.exe \(SHA-512, PGP ASC\)](#)
- [Apache-NetBeans-16-bin-macosx.dmg \(SHA-512, PGP ASC\)](#)
- [apache-netbeans_16-1_all.deb \(SHA-512, PGP ASC\)](#)
- [apache-netbeans-16-0.noarch.rpm \(SHA-512, PGP ASC\)](#)
- [Linux snap package](#)

Source:

- [netbeans-16-source.zip \(SHA-512, PGP ASC\)](#)

Officially, it is important that you [verify the integrity](#) of the downloaded files using the PGP signatures (.asc file) or a hash (.sha512

[Community Installers](#)

[Deployment Platforms](#)

[Known Issues](#)

[Building from Source](#)

[Community Approval](#)

[Earlier Releases](#)

Descargar NetBeans

<https://netbeans.apache.org/download/index.html>



The Runtime JDK NetBeans uses does not influence the JDK range projects can use.

Known Issues

- Gradle projects in Apache NetBeans 16 are currently not supported when running the IDE on JDK 19.
- [All Issues on GitHub](#)
- [How to Report an Issue](#)

Building from Source

To build Apache NetBeans 16 from source you need:

1. [A distribution of OpenJDK 11.](#)
2. [Apache Ant 1.10 or greater \(<https://ant.apache.org>\).](#)

Once you have everything installed then:



1. [Unzip netbeans-16-source.zip in a directory of your liking.](#)
2. `cd` to that directory, and then run `ant` to build the Apache NetBeans IDE. Once built you can run the IDE by typing
`./nbbuild/netbeans/bin/netbeans`

For more details refer to the README.

Descargar NetBeans

<https://netbeans.apache.org/download/index.html>



A dark blue horizontal navigation bar with white text and icons. It includes links for News, About, Make a Donation, The Apache Way, Join Us, Downloads, and a search icon.



COMMUNITY-LED DEVELOPMENT "THE APACHE WAY"

Projects ▾ People ▾ Community ▾ License ▾ Sponsors ▾



We suggest the following site for your download:

<https://dlcdn.apache.org/netbeans/netbeans-installers/16/Apache-NetBeans-16-bin-windows-x64.exe>

Alternate download locations are suggested below.

It is essential that you verify the integrity of the downloaded file using the PGP signature ([.asc](#) file) or a hash ([.md5](#) or [.sha*](#) file).



<https://dlcdn.apache.org/netbeans/netbeans-installers/16/Apache-NetBeans-16-bin-windows-x64.exe>

BACKUP SITE

<https://dlcdn.apache.org/netbeans/netbeans-installers/16/Apache-NetBeans-16-bin-windows-x64.exe>

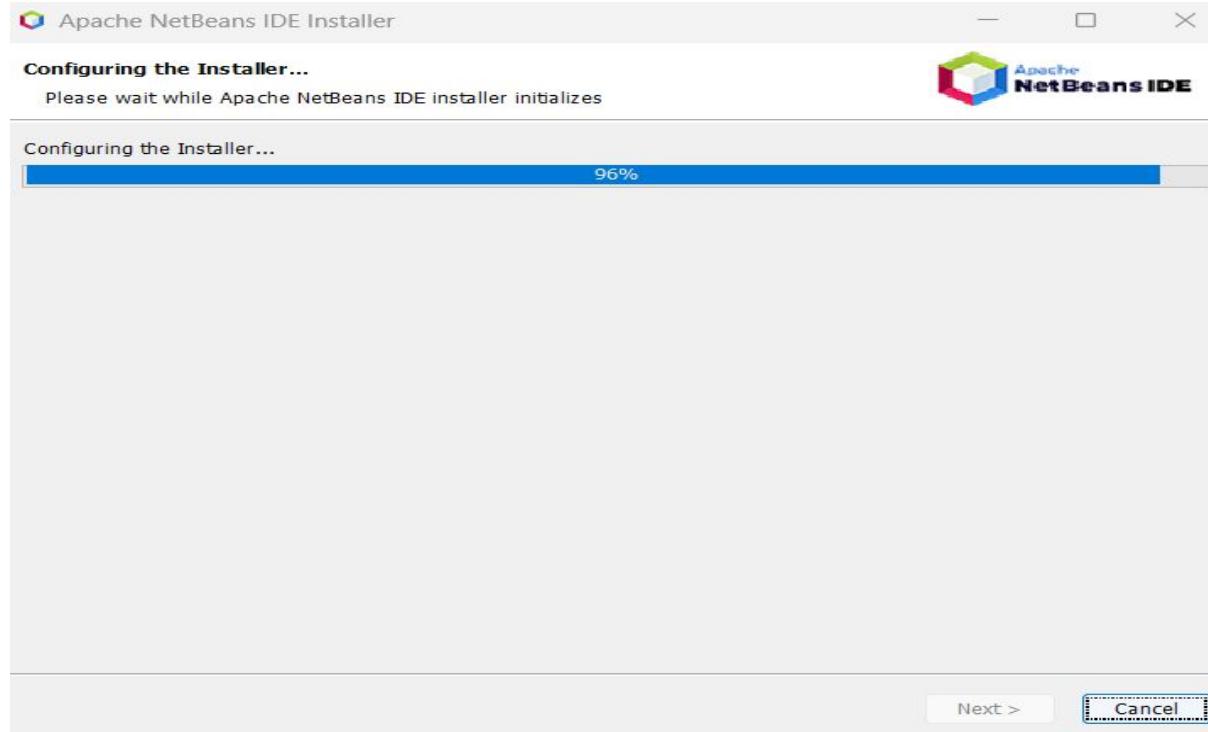
Descargar NetBeans

<https://netbeans.apache.org/download/index.html>



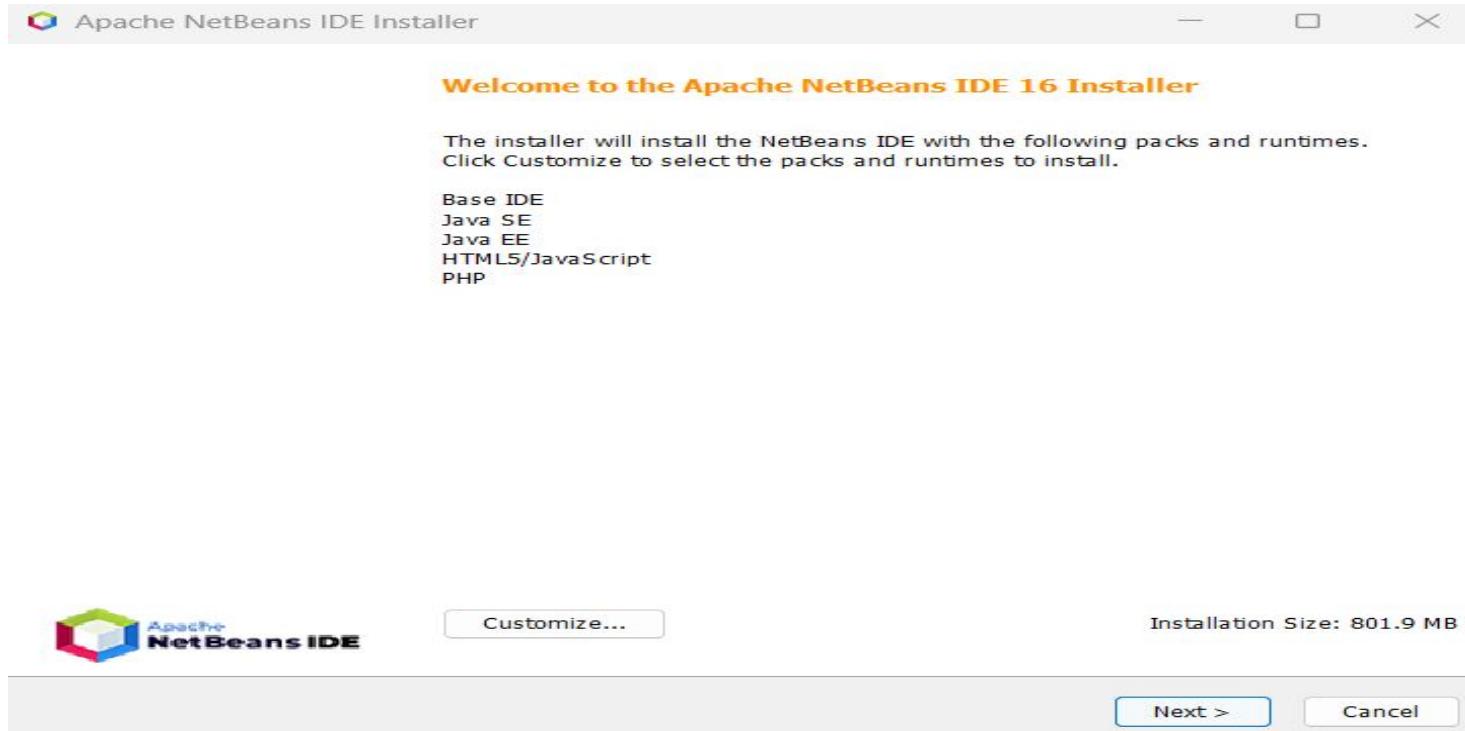
Descargar NetBeans

<https://netbeans.apache.org/download/index.html>



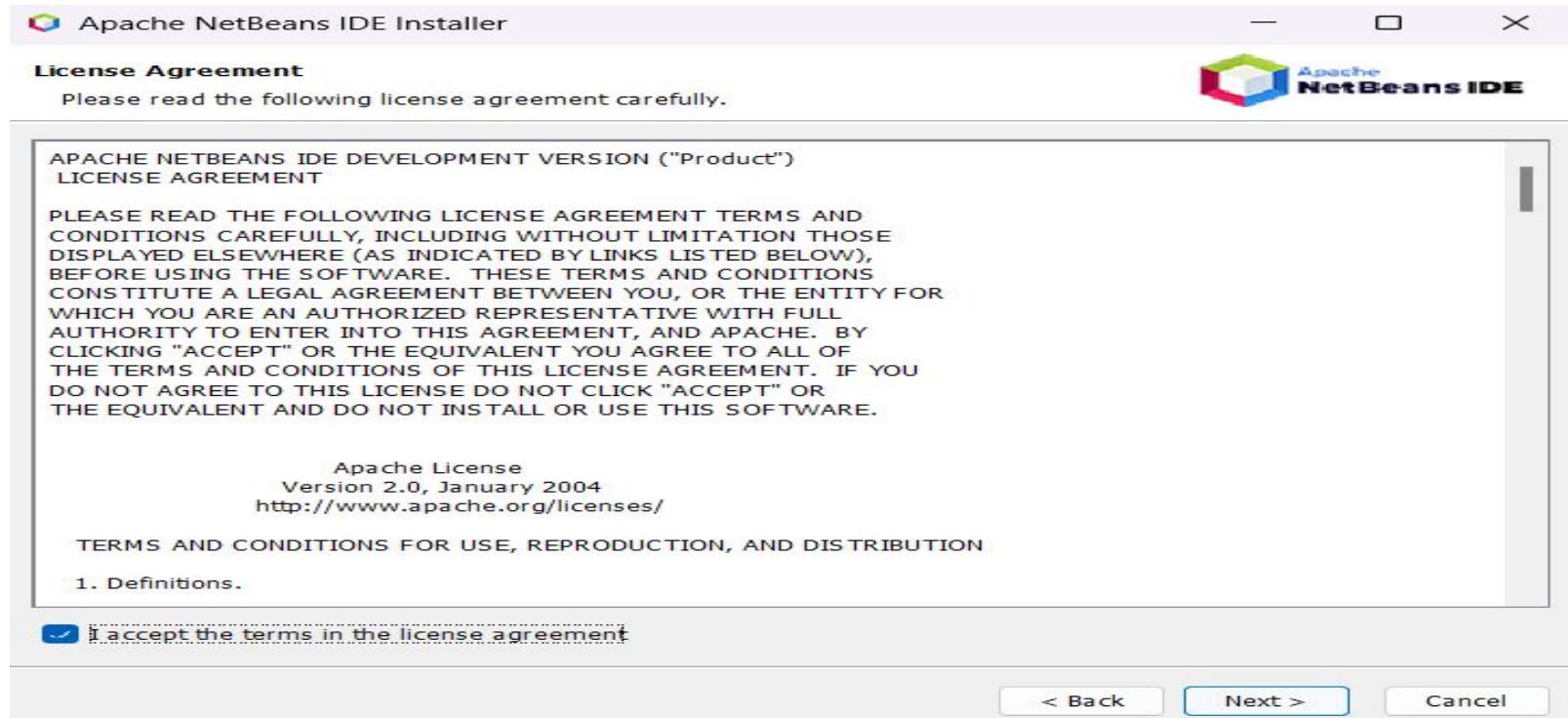
Descargar NetBeans

<https://netbeans.apache.org/download/index.html>



Descargar NetBeans

<https://netbeans.apache.org/download/index.html>

A screenshot of the Apache NetBeans IDE Installer window. The title bar says "Apache NetBeans IDE Installer". The main content area is titled "License Agreement" and contains the following text:
**APACHE NETBEANS IDE DEVELOPMENT VERSION ("Product")
LICENSE AGREEMENT**
PLEASE READ THE FOLLOWING LICENSE AGREEMENT TERMS AND CONDITIONS CAREFULLY, INCLUDING WITHOUT LIMITATION THOSE DISPLAYED ELSEWHERE (AS INDICATED BY LINKS LISTED BELOW), BEFORE USING THE SOFTWARE. THESE TERMS AND CONDITIONS CONSTITUTE A LEGAL AGREEMENT BETWEEN YOU, OR THE ENTITY FOR WHICH YOU ARE AN AUTHORIZED REPRESENTATIVE WITH FULL AUTHORITY TO ENTER INTO THIS AGREEMENT, AND APACHE. BY CLICKING "ACCEPT" OR THE EQUIVALENT YOU AGREE TO ALL OF THE TERMS AND CONDITIONS OF THIS LICENSE AGREEMENT. IF YOU DO NOT AGREE TO THIS LICENSE DO NOT CLICK "ACCEPT" OR THE EQUIVALENT AND DO NOT INSTALL OR USE THIS SOFTWARE.

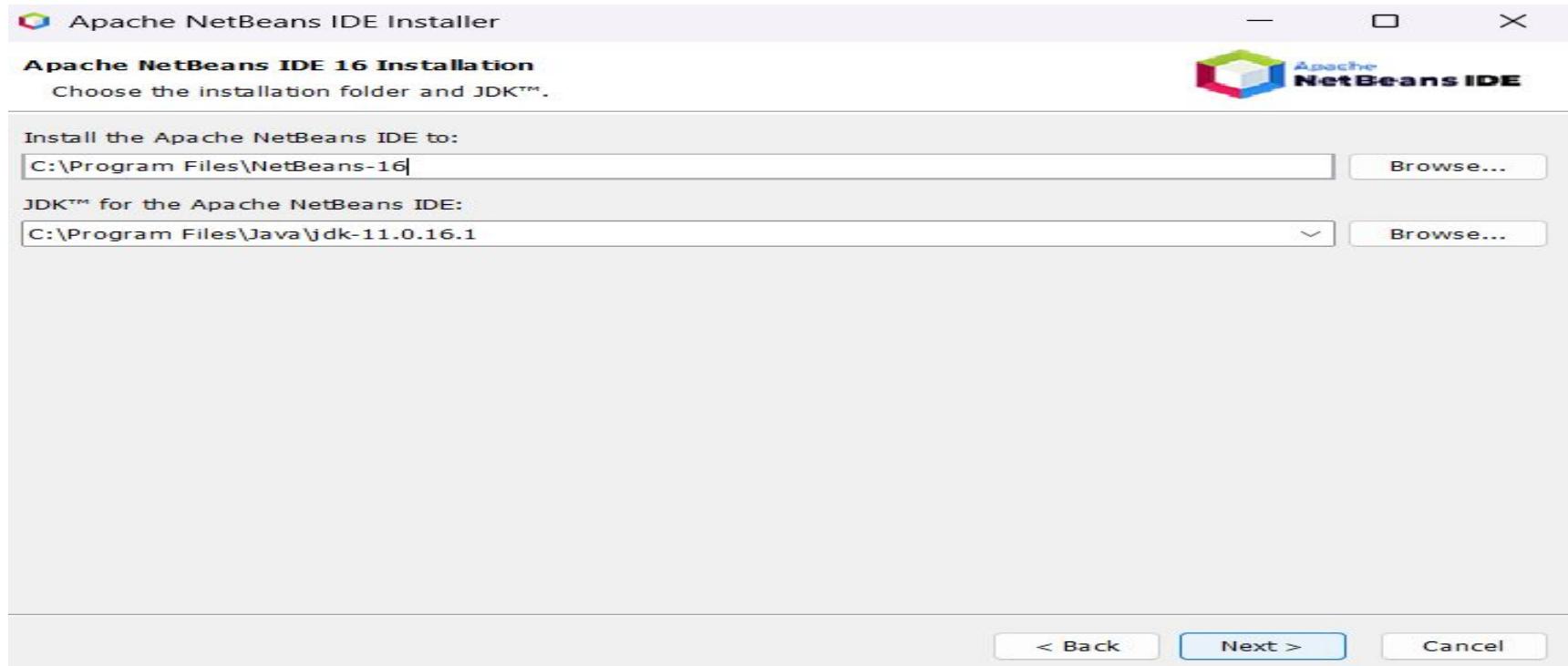
Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.

 I accept the terms in the license agreement

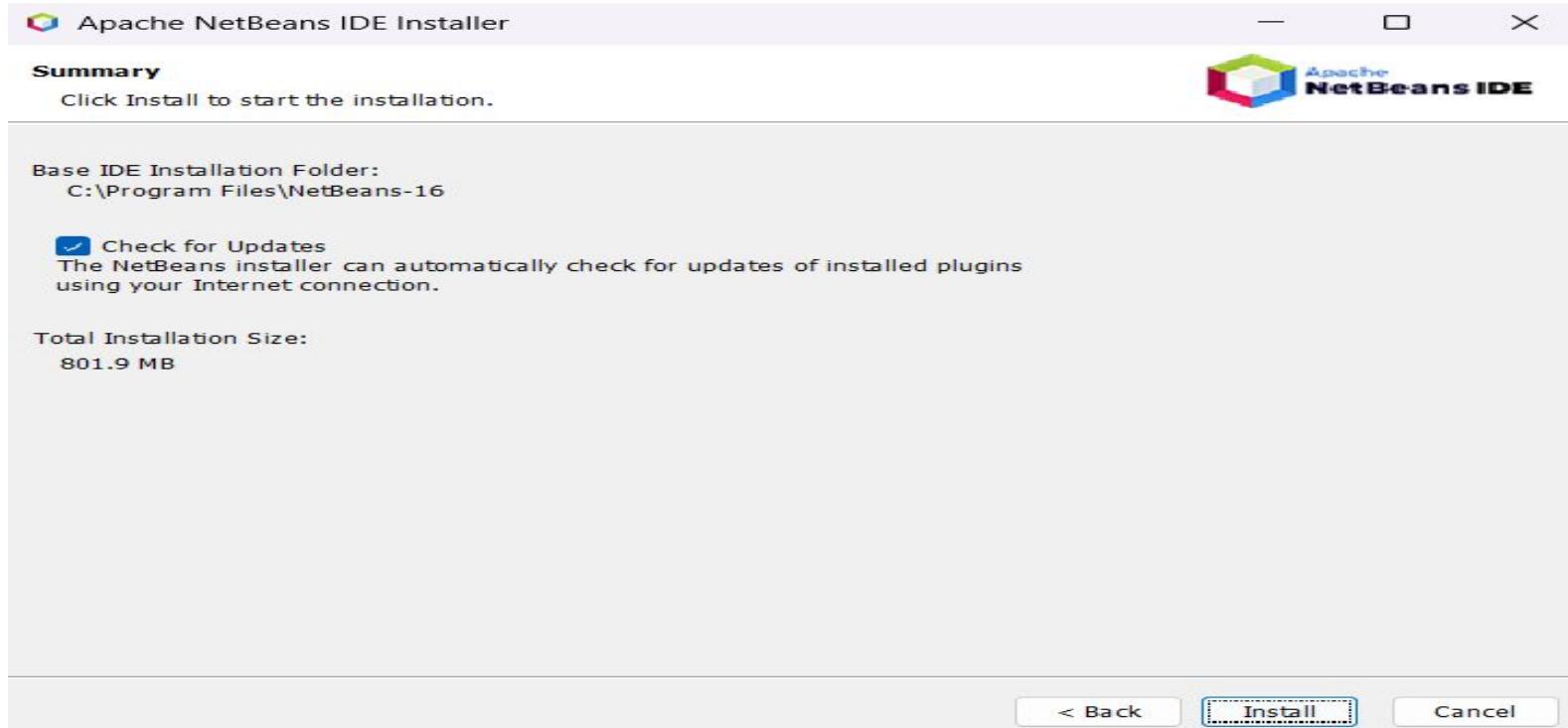
Descargar NetBeans

<https://netbeans.apache.org/download/index.html>



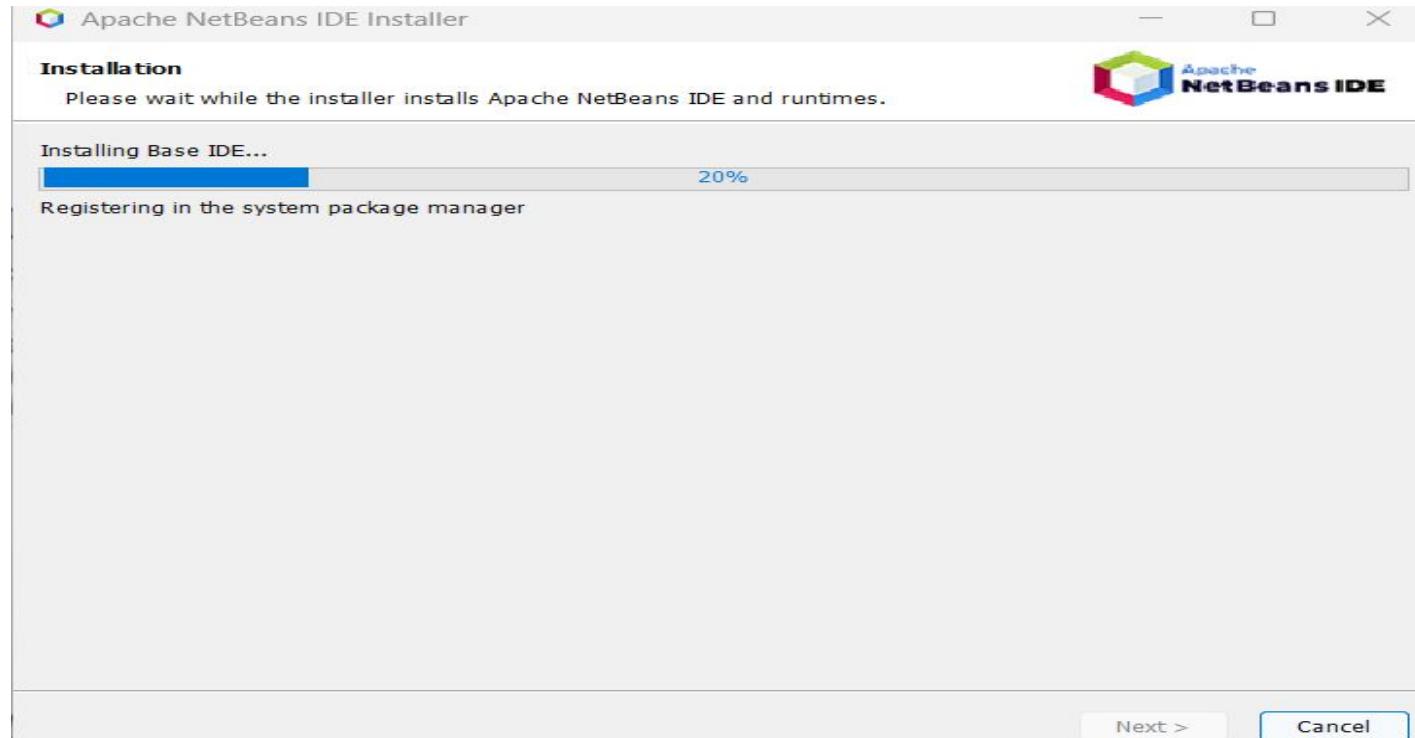
Descargar NetBeans

<https://netbeans.apache.org/download/index.html>



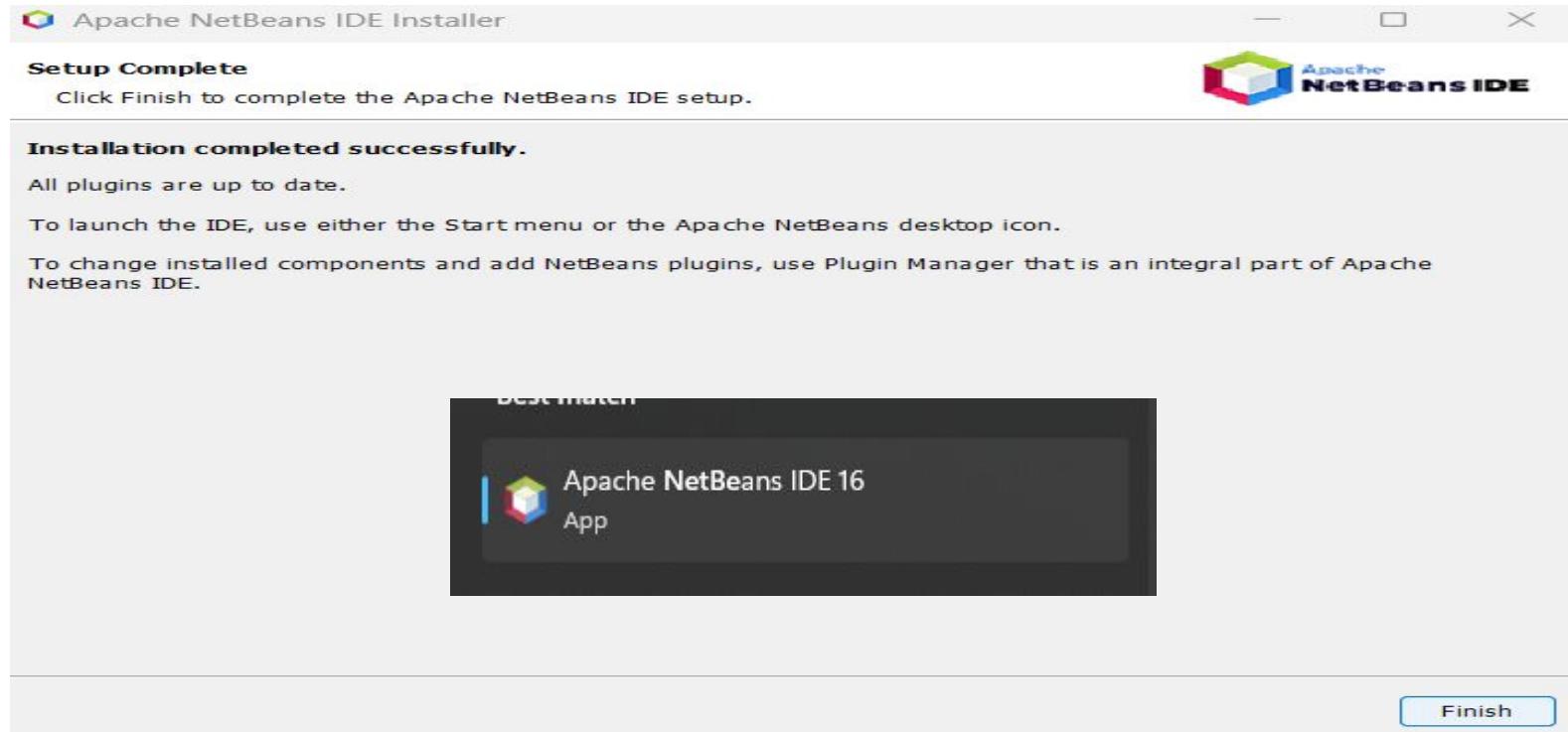
Descargar NetBeans

<https://netbeans.apache.org/download/index.html>



Descargar NetBeans

<https://netbeans.apache.org/download/index.html>





File View Debug Profile Team Tools Window Help

Apache NetBeans IDE 16



▼ Search (Ctrl+I)

— □ ×

Start Page

◀ ▶ ⌂ □



Learn & Discover

My NetBeans

What's New

Show On Startup

Learn & Discover

[Take a Tour](#)

Try a Sample Project

What's New

Community Corner

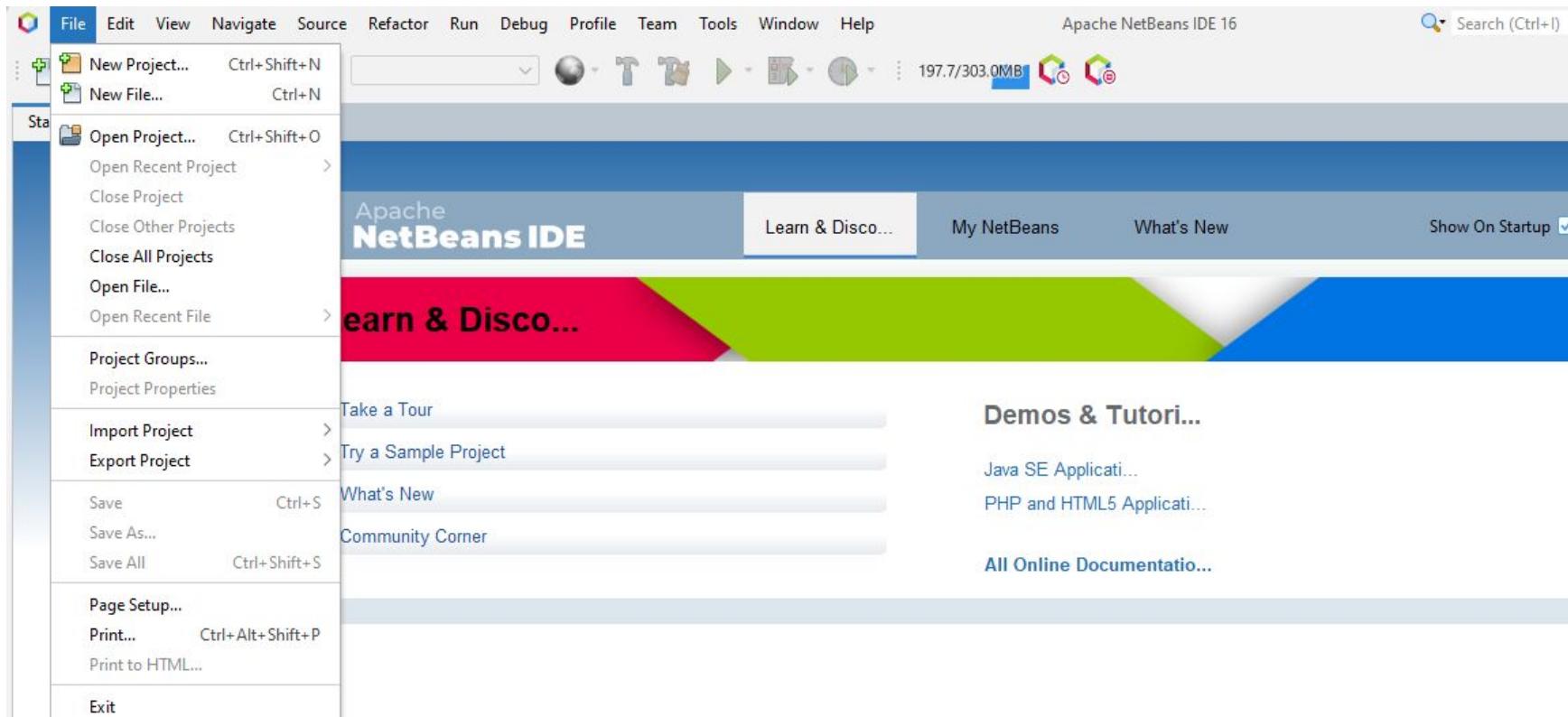
Demos & Tutorials

Java SE Applications

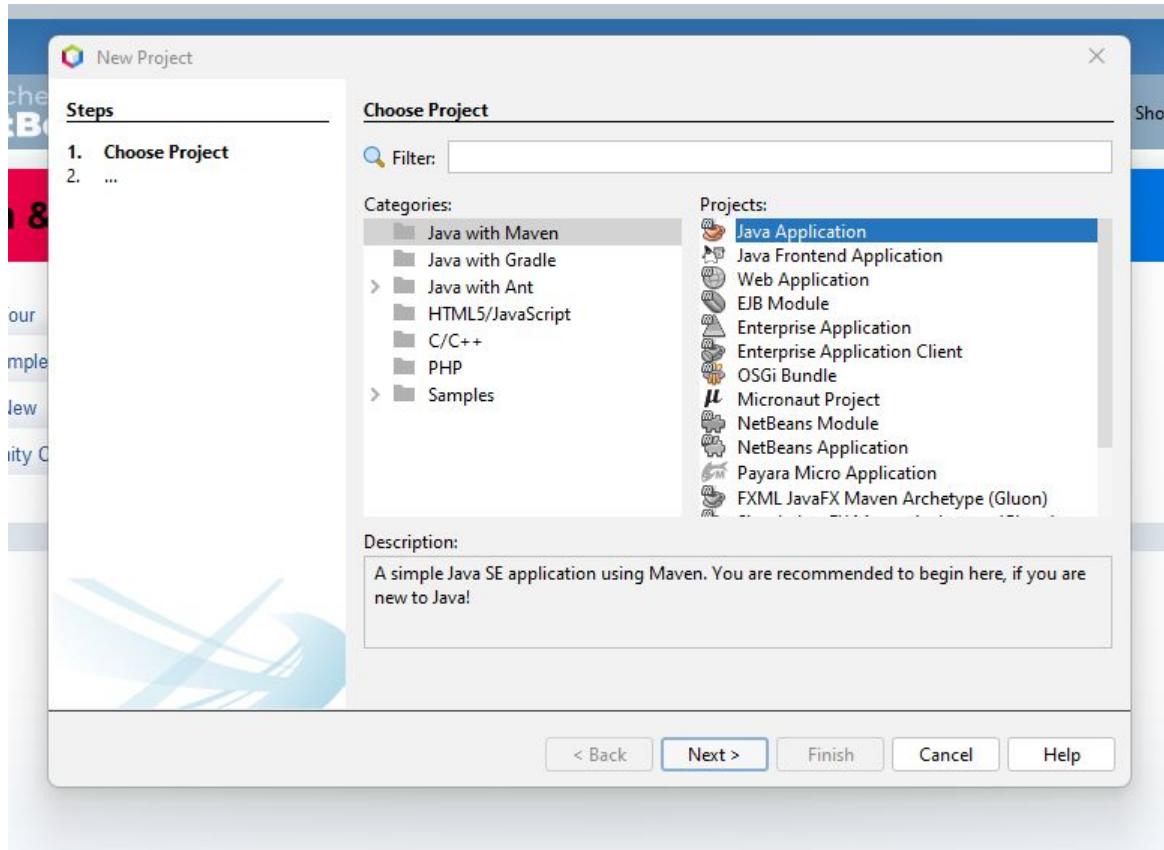
PHP and HTML5 Applications

All Online Documentation >>

Primer Programa Hola Mundo



Primer Programa Hola Mundo



Primer Programa Hola Mundo

New Java Application

Steps

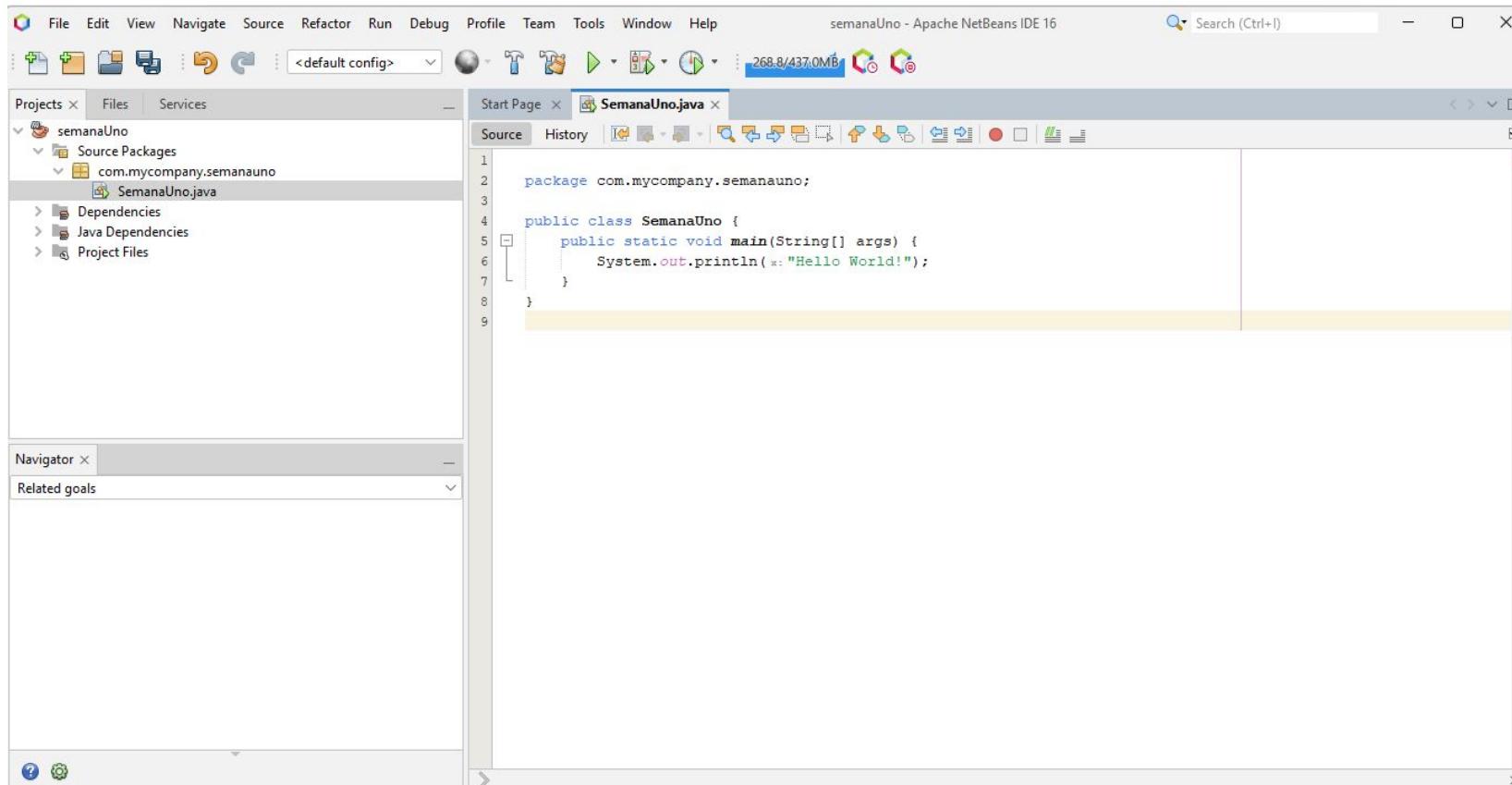
1. Choose Project
2. **Name and Location**

Name and Location

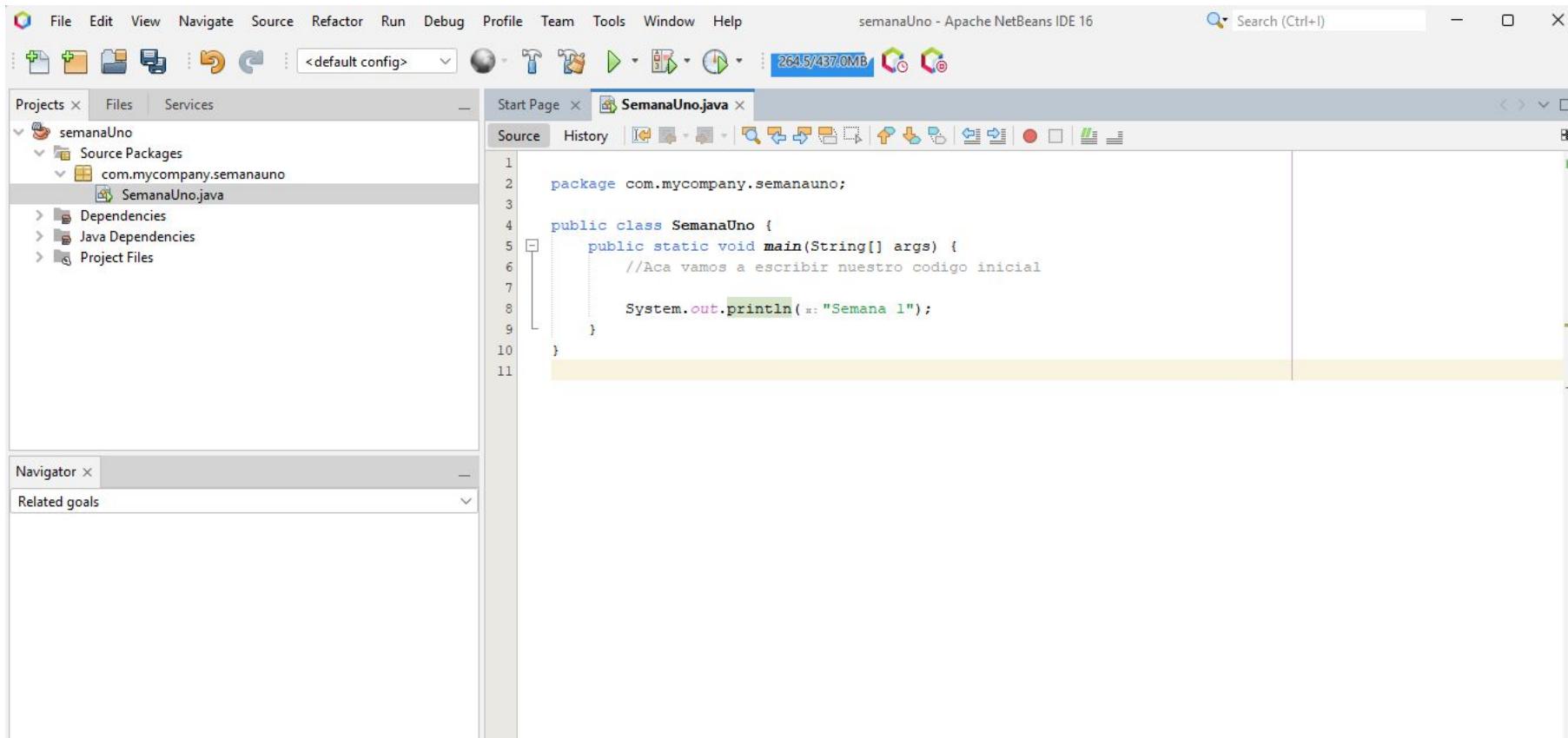
Project Name:	semanaUno	
Project Location:	C:\1\Java	Browse...
Project Folder:	C:\1\Java\semanaUno	
Artifact Id:	semanaUno	
Group Id:	com.mycompany	
Version:	1.0-SNAPSHOT	
Package:	com.mycompany.semanauno	(Optional)

< Back Next > [Finish](#) Cancel Help

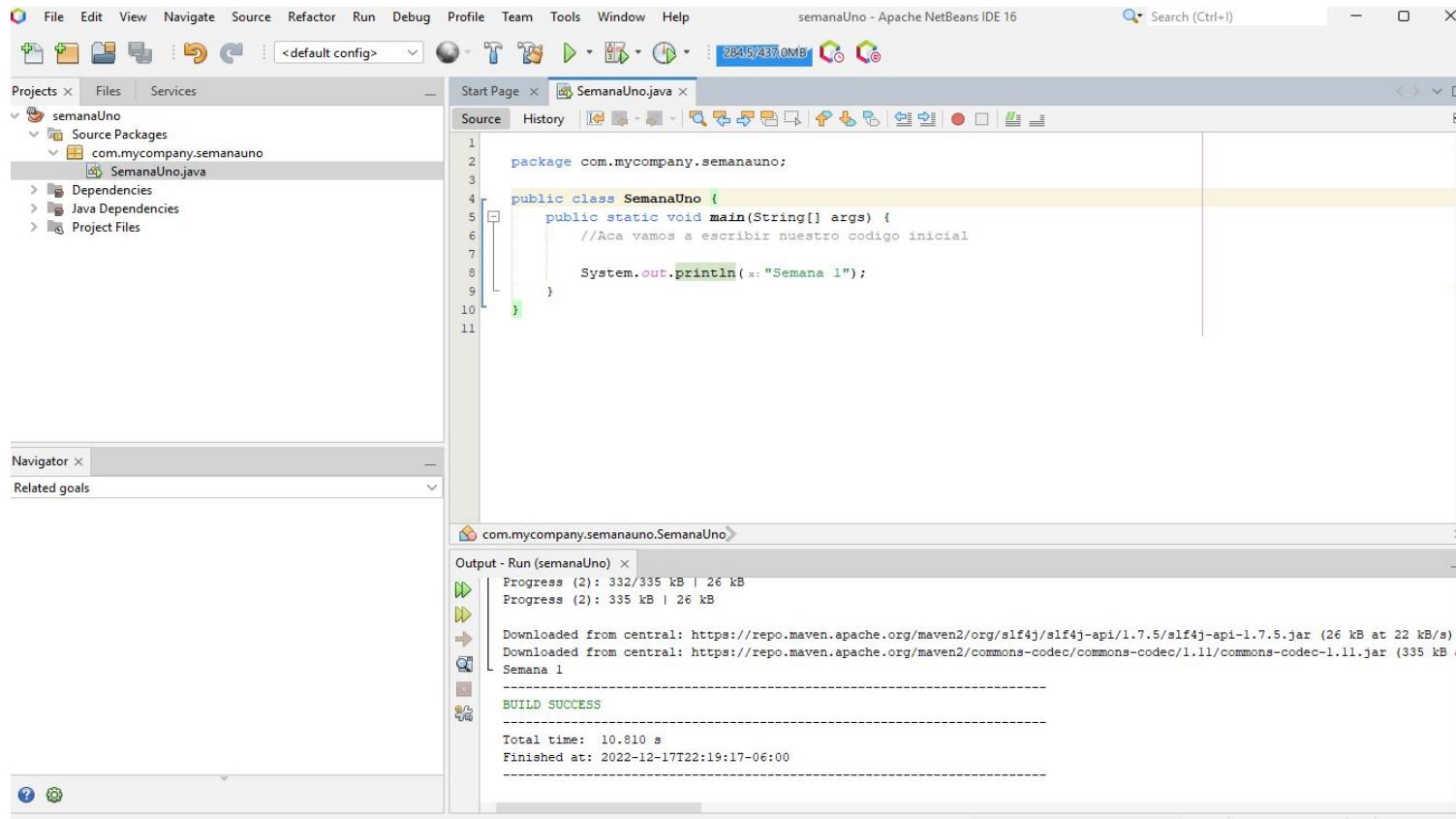
Primer Programa Hola Mundo



Primer Programa Hola Mundo



Primer Programa Hola Mundo



ALMACENAMIENTO DE DATOS

Start Page

SemanaUno.java

Source

History



```
1
2     package com.mycompany.semanauno;
3
4     public class SemanaUno {
5         public static void main(String[] args) {
6             //Aca vamos a escribir nuestro codigo inicial
7
8
9             //PRIMERO DE DEFINE EL TIPO DE VARIABLE, SEGUIDO DEL NOMBRE DE VARIABLE Y POSTERIOR EL VALOR DE ESTA
10            int var_entero = 1;
11            double var_double = 5.4;
12            char var_char = 'a';
13            String var_string = "Frase";
14
15
16            System.out.println("Semana 1");
17        }
18    }
```

CONDICIONALES

Start Page

SemanaUno.java

Source History



```
1
2     package com.mycompany.semanauno;
3
4     public class SemanaUno {
5         public static void main(String[] args) {
6             //Aca vamos a escribir nuestro codigo inicial
7
8
9             //LOS CONDICIONALES SE USAN PARA EVALUAR VARIAS OPCIONES
10            int edad = 18;
11
12            if(edad>=18){
13                System.out.println("Eres mayor de edad");
14            }else{
15                System.out.println("Eres menor de edad");
16            }
17        }
18    }
19 }
```

Source History



```
1
2 package com.mycompany.semanauno;
3
4 public class SemanaUno {
5     public static void main(String[] args) {
6         //Aca vamos a escribir nuestro codigo inicial
7
8
9         //LOS CONDICIONALES SE USAN PARA EVALUAR VARIAS OPCIONES
10        int var_posiciones = 0;
11        int var_max_campos = 20;
12        int var_pos = 0;
13
14        while(var_posiciones <= var_max_campos){
15            var_posiciones = var_posiciones+1;
16            System.out.println("Posición"+var_pos);
17            var_pos = var_pos +1;
18        }
19    }
20}
```

Start Page × SemanaUno.java ×

Source History |

```
4 public class SemanaUno {  
5     public static void main(String[] args) {  
6         //Aca vamos a escribir nuestro codigo inicial  
7  
8  
9         //LOS CONDICIONALES SE USAN PARA EVALUAR VARIAS OPCIONES  
10        int var_posiciones = 0;  
11        int var_max_campos = 20;  
12        int var_pos = 0;  
13  
14        while(var_posiciones <= var_max_campos){  
15            var_posiciones = var_posiciones+1;  
16            //var_posiciones++;  
17            System.out.println("Posición"+var_pos);  
18            var_pos = var_pos +1;  
19        }  
}
```

com.mycompany.semanauno.SemanaUno > main >

Start Page

SemanaUno.java

Source

History



```
1
2     package com.mycompany.semanauno;
3
4     public class SemanaUno {
5         public static void main(String[] args) {
6             //Aca vamos a escribir nuestro codigo inicial
7
8
9             //LOS CONDICIONALES SE USAN PARA EVALUAR VARIAS OPCIONES
10            int var_posiciones = 0;
11            int var_max_campos = 20;
12
13            for (int i = var_posiciones; i < var_max_campos; i++) {
14
15                System.err.println("Número "+i);
16            }
17
18        }
19    }
```

Funciones = Métodos

(La idea es no repetir código y más bien reciclar)

Start Page

SemanaUno.java

Source History

```
1
2 package com.mycompany.semanauno;
3
4 public class SemanaUno {
5     public static void main(String[] args) {
6         //Aca vamos a escribir nuestro codigo inicial
7
8         met_mostrar(); //aca añadimos nuestros métodos
9     }
10    //public static, en resumen es para ver quien puede accesar, o que visibilidad puede tener
11    //void es para indicar que no devuelve nada
12    public static void met_mostrar(){
13        System.out.println("Hola estoy desde el metodo met_mostrar");
14    }
15 }
16
```

Ejemplo método tipo entero sin parámetro

Start Page

SemanaUno.java

Source

History



```
1
2     package com.mycompany.semanauno;
3
4     public class SemanaUno {
5         public static void main(String[] args) {
6             //Aca vamos a escribir nuestro codigo inicial
7
8             //metodoEntero();
9
10            System.out.println( metodoEntero());
11
12
13
14        }
15
16        //public static, en resumen es para ver quien puede accesar, o que visibilidad puede tener
17        //void es para indicar que no devuelve nada
18        public static int metodoEntero(){
19            int variable = 0;
20            return variable;
21
22        }
23    }
24 }
```

Ejemplo método tipo entero con parámetro

Start Page

SemanaUno.java

Source

History



```
1
2 package com.mycompany.semanauno;
3
4 public class SemanaUno {
5     public static void main(String[] args) {
6         //Aca vamos a escribir nuestro codigo inicial
7
8         //metodoEntero();
9
10        System.out.println( ::metodoEntero( numero: 3 ) );
11
12
13    }
14
15
16    //public static, en resumen es para ver quien puede accesar, o que visibilidad puede tener
17    //void es para indicar que no devuelve nada
18    public static int metodoEntero(int numero){
19        int variable = numero;
20        return variable;
21
22    }
23
24 }
```

The screenshot shows a Java code editor with the following code:

```
1 package com.mycompany.semanauno;
2
3 public class SemanaUno {
4     public static void main(String[] args) {
5         //Aca vamos a escribir nuestro codigo inicial
6
7         //metodoEntero();
8
9         System.out.println(x.metodoSuma( numero1: 3, numero2: 5));
10
11
12
13     }
14
15
16     //public static, en resumen es para ver quien puede accesar, o que visibilidad puede tener
17
18     public static int metodoSuma(int numero1, int numero2){
19         int total = numero1 + numero2;
20         return total;
21
22     }
23
24 }
```

The code editor has tabs for "Source" and "History". The toolbar includes icons for copy, paste, find, search, and other common operations. The code is syntax-highlighted, with keywords in blue and comments in gray. A yellow bar at the bottom indicates the current line of code.

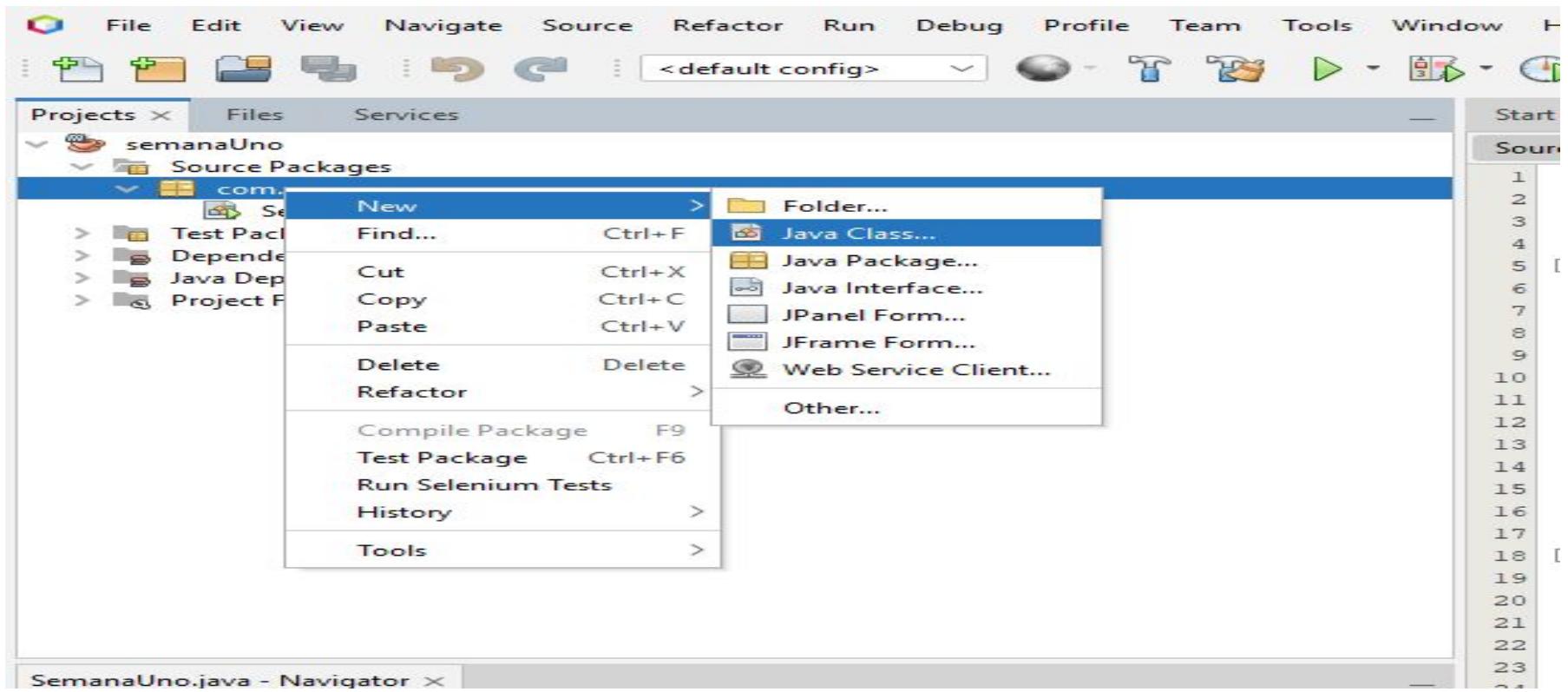
```
Output - Run (semanaUno) ×

cd C:\1\Java\semanaUno; "JAVA_HOME=C:\\Program Files\\Java\\jdk-11.0.16.1" cmd /c "\"C:\\Program Files\\NetBeans-16\\netbeans\\java\\maven\\bin\\mvn.cmd\" -Dexec.vmw
Running NetBeans Compile On Save execution. Phase execution is skipped and output directories of dependency projects (with Compile on Save turned on) will be used instead.
Scanning for projects...

-----< com.mycompany:semanaUno >-----
[INFO] Building semanaUno 1.0-SNAPSHOT
[INFO]   [jar]
[INFO] --- exec-maven-plugin:3.0.0:exec (default-cli) @ semanaUno ---
[INFO] 8
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time:  0.817 s
[INFO] Finished at: 2023-05-10T10:45:28-03:00
[INFO]  .
```

Programación orientada a Objetos

Digamos que queremos aislar una parte del código y que se vea en otro archivo



Esto nos permite crear una nueva clase

New Java Class



Steps

1. Choose File Type
2. Name and Location

Name and Location

Class Name:

Project:

Location: ▾

Package: ▾

Created File:

Superclass:

Interfaces:

< Back

Next >

Finish

Cancel

Help

Start Page × SemanaUno.java × met_Pers

Source History

```
1 package com.mycompany.semanauno;  
2  
3  
4 public class met_Persona {  
5  
6 }  
7  
8 //esta es una clase que despues podemos referenciar en SemanaUno.java
```

Start Page × SemanaUno.java ×

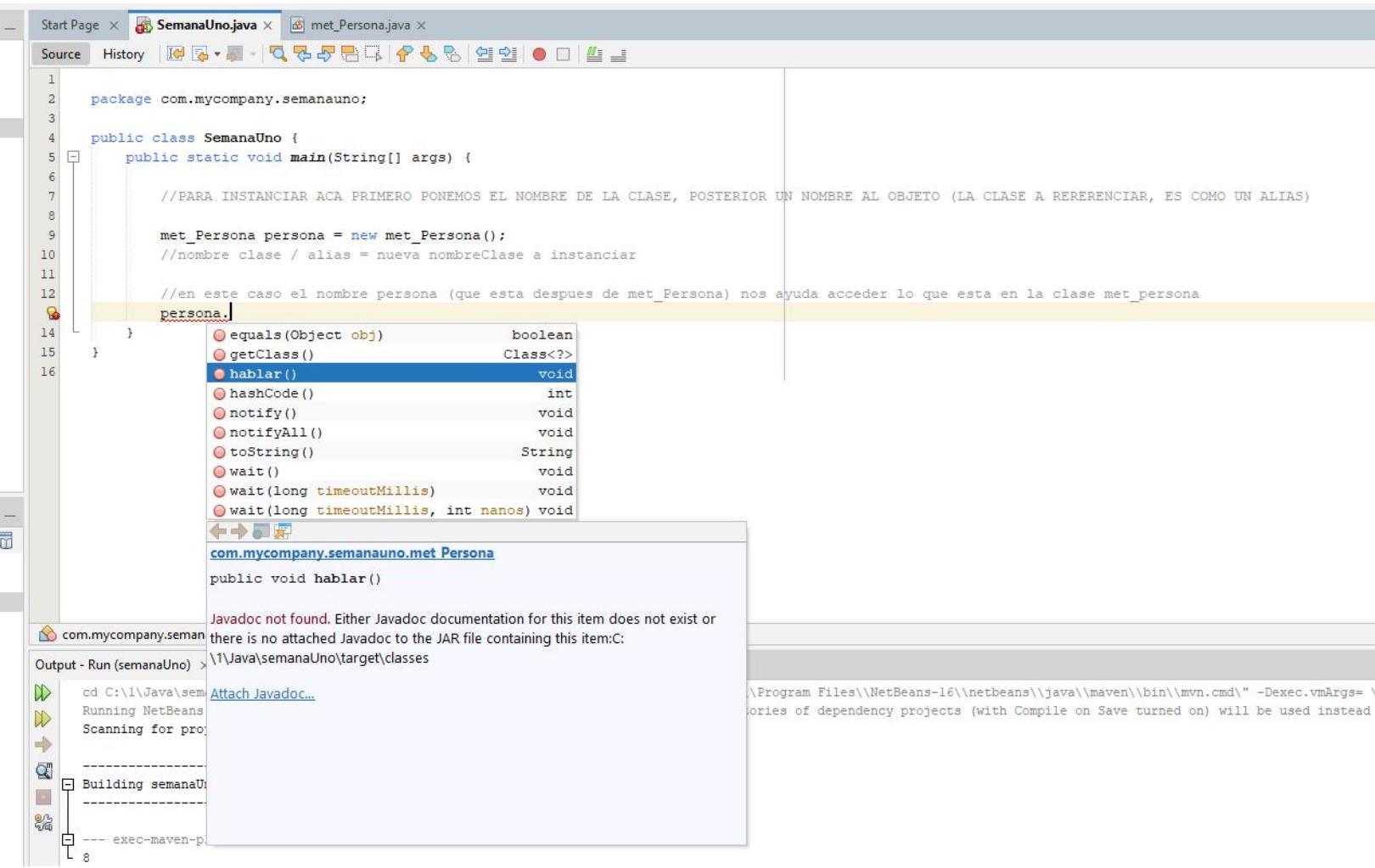
met_Persona.java ×

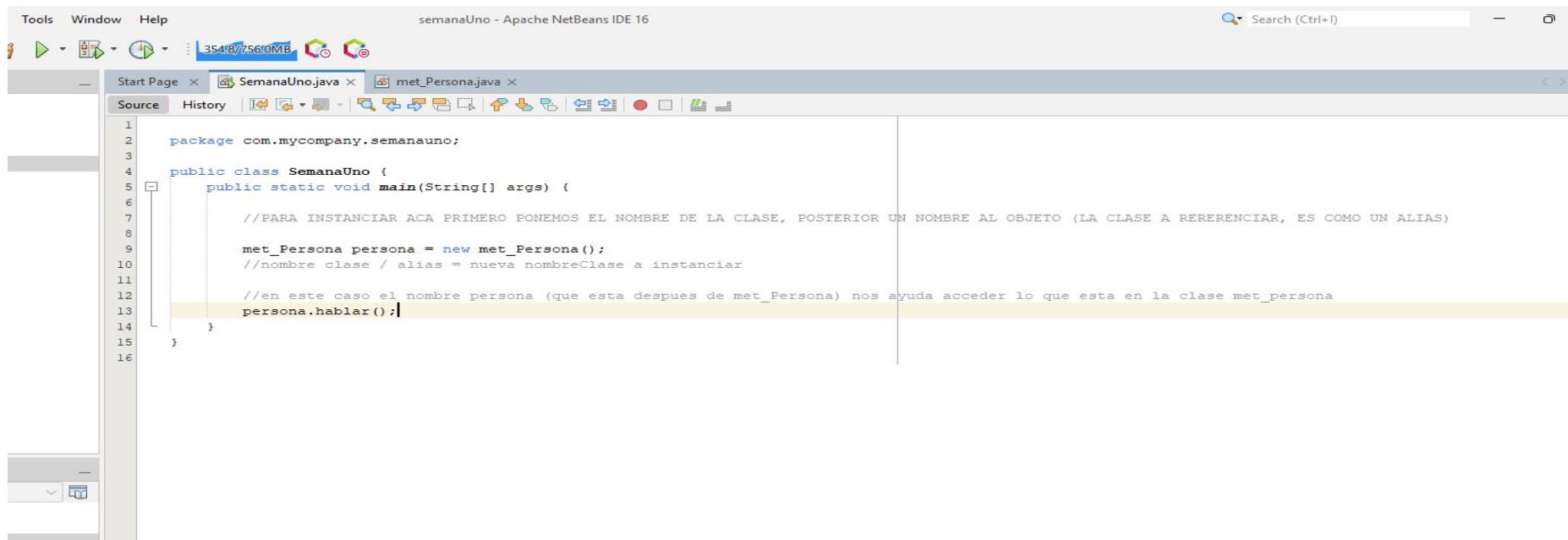
Source History



```
1
2 package com.mycompany.semanauno;
3
4 public class met_Persona {
5
6     public void hablar(){
7         System.out.println("Hola");
8     }
9
10 }
11
12 //esta es una clase que despues podemos referenciar en SemanaUno.java
13
```

Para agregar a la clase SemanaUno.java hay que primero hacer referencia a la clase y despues referencia al método, eso se llama instancia un objeto





Hagamos otro método más en la met_Persona para probar

ols Window Help

semanaUno - Apache NetBeans IDE 16



Start Page x SemanaUno.java x

met_Persona.java x

Source History

```
1
2 package com.mycompany.semanauno;
3
4 public class met_Persona {
5
6     public void hablar(){
7         System.out.println("Hola");
8     }
9
10    public int edad(int edad){
11        return edad;
12    }
13
14 }
15
16 //esta es una clase que despues podemos referenciar en SemanaUno.java
17
```



Start Page × SemanaUno.java × met_Persona.java ×

Source History

```
1 package com.mycompany.semanauno;
2
3 public class SemanaUno {
4     public static void main(String[] args) {
5
6         //PARA INSTANCIAR ACA PRIMERO PONEMOS EL NOMBRE DE LA CLASE, POSTERIOR UN NOMBRE AL OBJETO (LA CLASE A REFERENCIAR, ES COMO UN ALIAS)
7
8         met_Persona persona = new met_Persona();
9         //nombre clase / alias = nueva nombreClase a instanciar
10
11         //en este caso el nombre persona (que esta despues de met_Persona) nos ayuda acceder lo que esta en la clase met_persona
12         persona.hablar();
13         persona.edad(edad:22);
14         System.out.println("::persona.edad(edad:22)");
15     }
16 }
17
18 }
```

Constructores



<default config>



475.6 / 756.0MB



Projects X Files Services

- semanaUno
- Source Packages
 - com.mycompany.semanauno
 - SemanaUno.java
 - met_Persona.java
- Test Packages
- Dependencies
- Java Dependencies
- Project Files

Start Page X

SemanaUno.java X

met_Persona.java X

Source History



1

New Java Class

Steps

- Choose File Type
- Name and Location

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

Superclass:

Interfaces:

Navigator X

Members

- met_Persona
 - met_Persona()
 - edad(int edad) : int
 - hablar()

BUILD SUCCESS



Start Page ×

SemanaUno.java ×

met_Persona.java ×

clas_Perro.java ×

Source

History

```
1 package com.mycompany.semanauno;  
2  
3 public class clas_Perro {  
4  
5 }  
6  
7
```

Output - Run (semanaUno) ×

```
cd C:\1\Java\semanaUno; "JAVA_HOME=C:\\\\Program Files\\\\Java\\\\jdk-11.0.16.1" cmd /c "\"C:\\\\Program Files\\\\NetBeans-16\\\\netbear  
Running NetBeans Compile On Save execution. Phase execution is skipped and output directories of dependency projects (with C  
Scanning for projects...
```

```
-----< com.mycompany:semanaUno >-----
```

```
Building semanaUno 1.0-SNAPSHOT
```

```
-----[ jar ]-----
```

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ semanaUno ---
```

```
Hola
```

```
22
```

The screenshot shows the Apache NetBeans IDE interface with the following details:

- Title Bar:** Shows "Apache NetBeans 10.0" and a search bar.
- Toolbar:** Includes icons for file operations (New, Open, Save, Print), project management (Project Navigator, File Navigator, Task List, Properties, Tools, Help), and code navigation (Search, Find, Replace, Go To, Go To Type, Go To Definition).
- Tab Bar:** Displays four tabs: "Start Page", "SemanaUno.java", "met_Persona.java", and "clas_Perro.java".
- Source Tab:** Active tab showing the Java code for the "clas_Perro" class.
- Code Area:** Contains the following Java code:

```
1 package com.mycompany.semanauno;
2
3 public class clas_Perro { //se indica para inicializar variables, nada más
4
5     //atributos
6
7     String nombre;
8     int edad;
9
10
11    //metodos o funciones
12
13    public clas_Perro (String nombre, int edad){
14
15        this.nombre = nombre;
16        this.edad = edad;
17    }
18
19
20    public void mostrarDatos(){
21        System.out.println("nombre: "+this.nombre);
22        System.out.println("Edad: "+this.edad);
23    }
24
25}
26
```

The code defines a class named "clas_Perro" with two attributes: "nombre" (String) and "edad" (int). It includes a constructor that takes "nombre" and "edad" as parameters and assigns them to the respective attributes. It also includes a method named "mostrarDatos" that prints the value of "nombre" and "edad" to the console.

Start Page

SemanaUno.java

met_Persona.java

clas_Perro.java

Source History | | | |

```
1
2     package com.mycompany.semanauno;
3
4     public class SemanaUno {
5         public static void main(String[] args) {
6
7             //PARA INSTANCIAR ACA PRIMERO PONEMOS EL NOMBRE DE LA CLASE, POSTERIOR UN NOMBRE AL OBJETO (LA CLA
8
9             //Iniciamos o instanciamos aca la clase PERRO
10
11            clas_Perro Perro = new clas_Perro(nombre:"Docker", edad:8);
12
13            Perro.mostrarDatos();
14
15        }
16    }
```

GETTERS y SETTERS

Modificar clases, siempre van antes de los métodos que tenemos armados, o en su defecto puede ir después, siempre y cuando estén después de los constructores

Vamos a la clase en cuestión que queremos modificar



Start Page × SemanaUno.java × met_Persona.java × clas_Perro.java ×

Source History

```
1 package com.mycompany.semanauno;
2
3 public class clas_Perro { //se indica para inicializar variables, nada más
4
5     //atributos
6
7     String nombre;
8     int edad;
9
10
11
12     //metodos o funciones
13
14     public clas_Perro (String nombre, int edad){
15
16         this.nombre = nombre;
17         this.edad = edad;
18     }
19
20
21     ge
22     @getEdad() - generate int
23     @getNombre() - generate String
24     Imported Items: Press 'Ctrl+SPACE' Again for All Items
25         System.err.println("nombre: "+this.nombre);
26         System.err.println("Edad: "+this.edad);
27     }
28
29 }
```

ge
@getEdad() - generate int
@getNombre() - generate String

Imported Items: Press 'Ctrl+SPACE' Again for All Items



Start Page × SemanaUno.java × met_Persona.java × clas_Perro.java ×

Source History

```
1 package com.mycompany.semanauno;
2
3 public class clas_Perro { //se indica para inicializar variables, nada más
4
5     //atributos
6
7     String nombre;
8     int edad;
9
10
11
12     //metodos o funciones
13
14     public clas_Perro (String nombre, int edad){
15
16         this.nombre = nombre;
17         this.edad = edad;
18     }
19
20
21     ge
22     @getEdad() - generate int
23     @getNombre() - generate String
24     Imported Items: Press 'Ctrl+SPACE' Again for All Items
25         System.err.println("nombre: "+this.nombre);
26         System.err.println("Edad: "+this.edad);
27     }
28
29 }
```

ge
@getEdad() - generate int
@getNombre() - generate String

Imported Items: Press 'Ctrl+SPACE' Again for All Items

indow Help semanaUno - Apache NetBeans IDE 16 Search (Ctrl+I) 413.2 / 756.0MB

Start Page × SemanaUno.java × met_Persona.java × clas_Perro.java ×

Source History |

```
7
8     String nombre;
9     int edad;
10
11     //metodos o funciones|
12
13     public clas_Perro (String nombre, int edad) {
14
15         this.nombre = nombre;
16         this.edad = edad;
17     }
18
19     public int getEdad() { //traer
20         return edad;
21     }
22
23     public void setEdad(int edad) { //colocar, setear
24         this.edad = edad;
25     }
26
27
28
29
30
31     public void mostrarDatos(){
32         System.out.println("nombre: "+this.nombre);
33         System.out.println("Edad: "+this.edad);
34     }
35
36
37 }
38
```

get = voy a traer los datos y hacer operaciones con ellos.

set = yo puedo colocar o establecer valores a cada uno de estos atributos

Set= no retorna ningún valor, solo la recibi y lo asigna

Ejemplo con get en la clase perro

File Help

semanaUno - Apache NetBeans IDE 16

Search (Ctrl+I)

- X



Start Page SemanaUno.java clas_Perro.java

Source History



```
1 package com.mycompany.semanauno;
2
3 public class SemanaUno {
4     public static void main(String[] args) {
5
6         //PARA INSTANCIAR ACA PRIMERO PONEMOS EL NOMBRE DE LA CLASE, POSTERIOR UN NOMBRE AL OBJETO (LA CLASE A REFERENCIAR, ES COMO UN ALIAS)
7
8         //Iniciamos o instanciamos aca la clase PERRO
9
10        clas_Perro Perrol = new clas_Perro(nombre: "Docker", edad: 8);
11        clas_Perro Perro2 = new clas_Perro(nombre: "Maylo", edad: 3);
12
13        int var_edadPerro2 = Perro2.getEdad();
14        System.out.println("La edad de Maylo es: "+var_edadPerro2); //y aca la estamos mostrando
15
16        //Perro2.getEdad(); //Aca estamos obteniendo la edad de Maylo
17        //System.out.println("La edad de Maylo es: "+Perro2.getEdad()); //y aca la estamos mostrando
18
19    }
20
21 }
```

com.mycompany.semanauno.SemanaUno > main > Perro2 >

Output - Run (semanaUno) >

```
cd C:\1\Java\semanaUno; "JAVA_HOME=C:\\Program Files\\Java\\jdk-11.0.16.1" cmd /c "\"C:\\Program Files\\NetBeans-16\\netbeans\\java\\maven\\bin\\mvn.cmd\" -Dexec.vmArgs= \"-Dexec.args= "
Running NetBeans Compile On Save execution. Phase execution is skipped and output directories of dependency projects (with Compile on Save turned on) will be used instead of their jars
Scanning for projects...
```

```
-----< com.mycompany:semanaUno >-----
Building semanaUno 1.0-SNAPSHOT
-----[ jar ]-----
--- exec-maven-plugin:3.0.0:exec (default-cli) @ semanaUno ---
La edad de Maylo es: 3
```

Ejemplo con set en la clase perro

Source History

```
7
8     String nombre;
9     int edad;
10
11
12     //metodos o funciones
13
14     public clas_Perro (String nombre, int edad){//inicializa las variables
15
16         this.nombre = nombre;
17         this.edad = edad;
18     }
19
20
21     public void setEdad(int edad) {//colocar, setear el valor con la variable inicializada
22         this.edad = edad;
23     }
24
25     public int getEdad() { //muestra la edad
26         return edad;
27     }
28
29
30     public void mostrarDatos(){
31         System.err.println("nombre: "+this.nombre);
32         System.err.println("Edad: "+this.edad);
33     }
34
```

The screenshot shows a Java IDE interface with the following details:

- Project Bar:** Shows "Start Page" as the current project.
- File List:** Lists "SemanaUno.java" and "clas_Perro.java".
- Toolbar:** Includes standard icons for file operations like Open, Save, Print, and a search icon.
- Source View:** Displays the code for the `SemanaUno` class. The code creates a new instance of the `clas_Perro` class named `perrol`, sets its age to 10, and prints its name and age.

```
1 package com.mycompany.semanauno;
2
3 public class SemanaUno {
4     public static void main(String[] args) {
5         //ejemplo SET
6
7         //instanciamos la clase PERRO
8         clas_Perro perrol = new clas_Perro(nombre: "Maylo", edad: 9);
9         perrol.setEdad(edad: 10);
10        System.err.println("Edad de Maylo es: "+perrol.getEdad());
11
12    }
13
14
15 }
```

```
Output - Run (semanaUno) ×

cd C:\1\Java\semanaUno; "JAVA_HOME=C:\\\\Program Files\\\\Java\\\\jdk-11.0.16.1" cmd /c "\"C:\\\\Program Files\\\\NetBeans-16\\\\netbeans\\\\java\\\\maven\\\\Running NetBeans Compile On Save execution. Phase execution is skipped and output directories of dependency projects (with Compile on Save Scanning for projects...

-----< com.mycompany:semanaUno >-----
Building semanaUno 1.0-SNAPSHOT
[ jar ]
----- exec-maven-plugin:3.0.0:exec (default-cli) @ semanaUno ---
Edad de Maylo es: 10
-----
BUILD SUCCESS
```

Hagamos lo mismo con la variable nombre



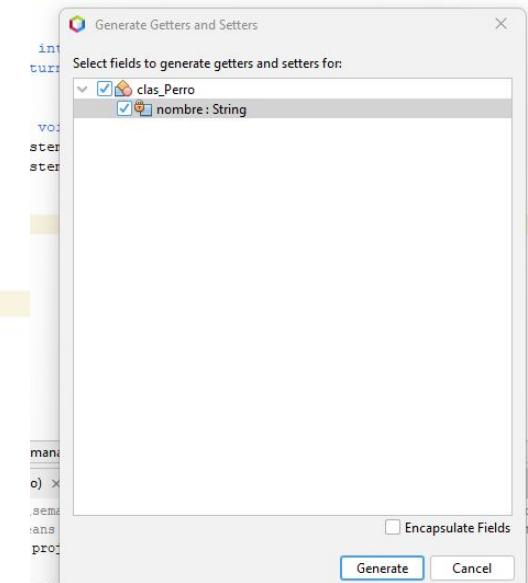
Source

History

File Edit View Tools Window Help

```
16         this.nombre = nombre;
17         this.edad = edad;
18     }
19
20     public void setEdad(int edad) { //colocar, setear el valor con la variable inicializada
21         this.edad = edad;
22     }
23
24     public int getEdad() { //muestra la edad
25         return edad;
26     }
27
28     public void mostrarDatos(){
29         System.out.println("nombre: "+this.nombre);
30         System.out.println("Edad: "+this.edad);
31     }
32
33 }
```

- Generate
- Constructor...
- Logger...
- Getter...
- Setter...
- Getter and Setter...**
- equals() and hashCode()...
- toString()...
- Delegate Method...
- Override Method...
- Add Property...



semanaUno 1.0-SNAPSHOT

Output - Run (semanaUno)



Start Page ×

SemanaUno.java ×

clas_Perro.java ×

Source

History

```
16         this.nombre = nombre;
17         this.edad = edad;
18     }
19
20     public void setEdad(int edad) //colocar, setear el valor con la variable inicial
21         this.edad = edad;
22     }
23
24     public int getEdad() { //muestra la edad
25         return edad;
26     }
27
28     public void mostrarDatos(){
29         System.out.println("nombre: "+this.nombre);
30         System.out.println("Edad: "+this.edad);
31     }
32
33     public String getNombre() {
34         return nombre;
35     }
36
37     public void setNombre(String nombre) {
38         this.nombre = nombre;
39     }
40
41
42 
```

Start Page

SemanaUno.java

clas_Perro.java

Source History



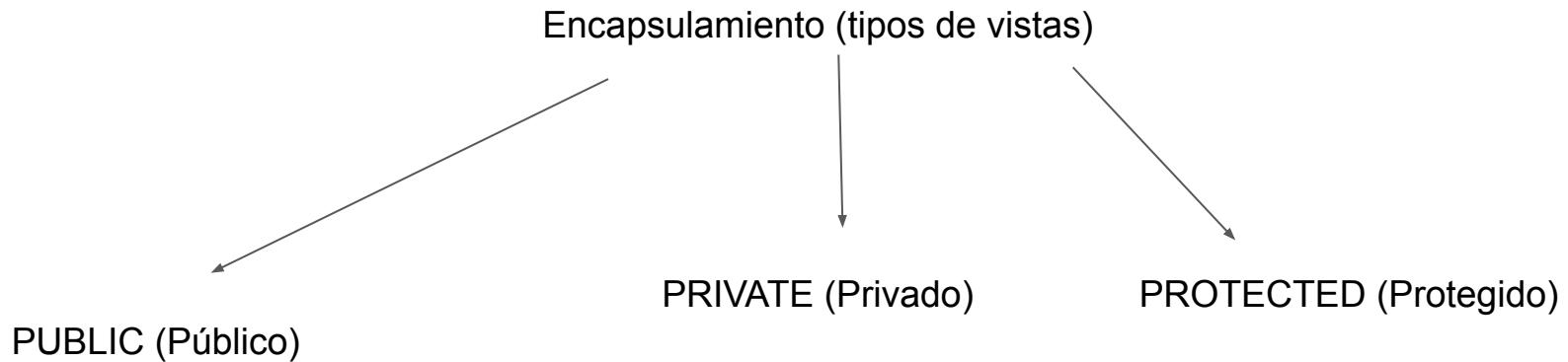
```
1
2 package com.mycompany.semanauno;
3
4 public class SemanaUno {
5     public static void main(String[] args) {
6         //ejemplo SET
7
8         //instanciamos la clase PERRO
9         clas_Perro perrol = new clas_Perro(nombre: "Maylo", edad: 9);
10        perrol.setEdad(edad: 10);
11        perrol.setNombre(nombre: "Maylo");
12        perrol.getNombre();
13        System.err.println("Edad de "+ perrol.getNombre()+" es: "+perrol.getEdad());
14    }
15 }
16
17
```

com.mycompany.semanauno.SemanaUno > main >

Output - Run (semanaUno)

```
cd C:\1\Java\semanaUno; "JAVA_HOME=C:\\Program Files\\Java\\jdk-11.0.16.1" cmd /c "\"C:\\Program Files\\NetBeans-16.0\\java\\bin\\java.exe\" -jar C:\\Users\\User\\AppData\\Local\\Temp\\NetBeans\\semanaUno\\target\\semanaUno-1.0-SNAPSHOT.jar"
Running NetBeans Compile On Save execution. Phase execution is skipped and output directories of dependency project
Scanning for projects...
-----< com.mycompany:semanaUno >-----
Building semanaUno 1.0-SNAPSHOT
-----[ jar ]-----
--- exec-maven-plugin:3.0.0:exec (default-cli) @ semanaUno ---
Edad de Maylo es: 10
-----
```

Encapsulamiento (tipos de vistas)



EJEMPLO PUBLIC y PRIVATE

The screenshot shows a Java code editor with the file Alumno.java open. The code defines a class with private attributes and public methods for interacting with those attributes.

```
private String apellido_uno;
private String apellido_dos;

//Creo un metodo sin variables, que sera accesido desde otra clase.
public Alumno() {
}

//creo el metodo publico para que pueda ser accedido desde otra clase.
public Alumno(int edad, String nombre, String apellido_uno, String apellido_dos) {
    this.edad = edad;
    this.nombre = nombre;
    this.apellido_uno = apellido_uno;
    this.apellido_dos = apellido_dos;
}

//voy a mostrar las variables
public int getEdad() {
    return edad;
}
public String getNombre() {
    return nombre;
}
public String getApellido_uno() {
    return apellido_ultimo;
}
public String getApellido_dos() {
    return apellido_dos;
}

//voy a editar o setear las variables
public void setEdad(int edad) {
    this.edad = edad +1;
}
public void setNombre(String nombre) {
    this.nombre = nombre;
}
public void setApellido_uno(String apellido_ultimo) {
    this.apellido_ultimo = apellido_ultimo;
}
public void setApellido_dos(String apellido_dos) {
    this.apellido_dos = apellido_dos;
}
```

Los atributos se ponen privados para proteger que o no pueda acceder a ellas.

Los atributos se ponen privados para proteger que otra clase no pueda acceder a ellas.

Start Page

SemanaUno.java

Alumno.java

clas_Perro.java

Source History



```
1
2 package com.mycompany.semanauno;
3
4 public class SemanaUno {
5
6
7     public static void main(String[] args) {
8         Alumno Alumno_uno = new Alumno();
9         Alumno Alumno_dos = new Alumno(edad: 33, nombre: "Javier", apellido_uno: "Mata", apellido_dos: "Guerrero");
10        // Alumno(int edad, String nombre, String apellido_uno, String apellido_dos)
11    }
12
13 }
```

Probemos cambiando los constructores publicos por privados a ver que pasa

Source History

```
4  /*
5   package com.mycompany.semanauno;
6
7   public class Alumno{ //creo que constructor que tendra las variables iniciales
8
9     private int edad;
10    private String nombre;
11    private String apellido_uno;
12    private String apellido_dos;
13
14    //Creo un metodo sin variables, que sera accesido desde otra clase.
15    public Alumno(){
16
17    }
18
19    //creo el metodo publico para que pueda ser accedido desde otra clase.
20    private Alumno(int edad, String nombre, String apellido_uno, String apellido_dos){
21      this.edad = edad;
22      this.nombre = nombre;
23      this.apellido_uno = apellido_uno;
24      this.apellido_dos = apellido_dos;
25    }
26
27    //voy a mostrar las variables
28    public int getEdad() {
29      return edad;
30    }
31    public String getNombre() {
32      return nombre;
```

Start Page ×

SemanaUno.java ×

Alumno.java ×

clas_Perro.java ×

Source History



```
1
2     package com.mycompany.semanauno;
3
4     public class SemanaUno {
5
6
7         public static void main(String[] args) {
8             Alumno Alumno_uno = new Alumno();
9             Alumno Alumno_dos = new Alumno(33, "Javier", "Mata", "Guerrero");
10            // Alumno(int edad, String nombre, String apellido_uno, String apellido_dos)
11        }
12    }
```

Volvamos a ponerlos públicos para que no de el error.

Start Page × SemanaUno.java × Alumno.java × clas_Perro.java ×



Source History |

```
1 package com.mycompany.semanauno;
2
3 public class SemanaUno {
4
5
6     public static void main(String[] args) {
7         Alumno Alumno_uno = new Alumno();
8         Alumno Alumno_dos = new Alumno( edad:33, nombre: "Javier", apellido_uno: "Mata", apellido_dos: "Guerrero");
9         // Alumno(int edad, String nombre, String apellido_uno, String apellido_dos)
10
11         System.out.println("EL nombre del estudiante es:"+Alumno_dos.getNombre()+" "+Alumno_dos.getApellido_uno()+" "+Alumno_dos.getApellido_dos());
12     }
13 }
14
15 }
```

com.mycompany.semanauno.SemanaUno > main >

Output - Run (semanaUno) ×

cd C:\1\Java\semanaUno; "JAVA_HOME=C:\\Program Files\\Java\\jdk-11.0.16.1" cmd /c "\"C:\\Program Files\\NetBeans-16\\netbeans\\java\\maven\\bin\\mvn.cmd\" -Dexec.vmArgs= \"-Dexec.args= Running NetBeans Compile On Save execution. Phase execution is skipped and output directories of dependency projects (with Compile on Save turned on) will be used instead of their jars Scanning for projects...

-----< com.mycompany:semanaUno >-----

----- Building semanaUno 1.0-SNAPSHOT -----

-----[jar]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ semanaUno ---

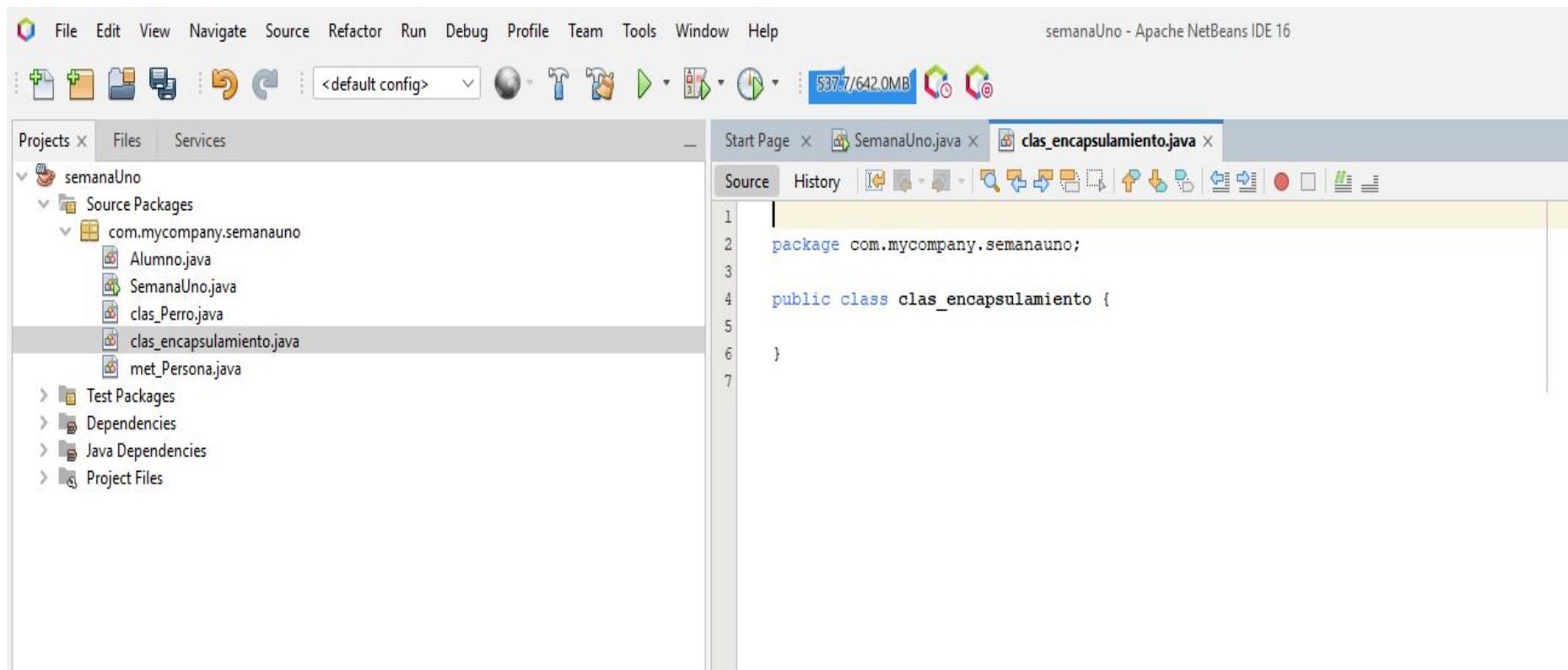
EL nombre del estudiante es:Javier Mata Guerrero

BUILD SUCCESS

Total time: 0.675 s

Finished at: 2023-01-10T20:42:48-06:00

EJEMPLO NUEVO PUBLIC, PRIVADO Y PROTECTED



EJEMPLO NUEVO PUBLIC, PRIVADO Y PROTECTED

The screenshot shows the Apache NetBeans IDE 16 interface. The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The title bar indicates the project is "semanaUno - Apache NetBeans IDE 16". The toolbar contains various icons for file operations like New, Open, Save, and Build. The left sidebar shows the project structure under "Projects": "semanaUno" (selected), "Source Packages" (expanded) containing "com.mycompany.semanauno" with files "Alumno.java", "SemanaUno.java", "clas_Perro.java", "clas_encapsulamiento.java" (selected), and "met_Persona.java"; "Test Packages", "Dependencies", "Java Dependencies", and "Project Files". The main workspace displays the Java code for "clas_encapsulamiento.java". The code defines a class with four types of variables:

```
1 package com.mycompany.semanauno;
2
3 public class clas_encapsulamiento {
4     //Ingreso variables de ejemplo en cada tipo de encapsulamiento
5     public int var_publica;
6     private int var_privada;
7     protected int var_protected;
8     static int var_static;
9 }
10
11
```

The code editor has tabs for "Source" and "History", and a toolbar with various icons for navigation and selection.

EJEMPLO NUEVO PUBLIC, PRIVADO Y PROTECTED

The screenshot shows the Apache NetBeans IDE 16 interface. The title bar reads "semanaUno - Apache NetBeans IDE 16". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The toolbar has various icons for file operations like New, Open, Save, and Build. The Projects tab shows a project named "semanaUno" with a package "com.mycompany.semanauno" containing files Alumno.java, SemanaUno.java, clas_Perro.java, clas_encapsulamiento.java, and met_Persona.java. The current file is "SemanaUno.java". The code editor displays:

```
1 package com.mycompany.semanauno;
2
3 public class SemanaUno {
4
5     public static void main(String[] args) {
6         //Instanciamos o llamamos la clase
7         clas_encapsulamiento obj_clas_encapsulamiento = new clas_encapsulamiento();
8     }
9 }
10
11
12
13
14 }
```

The code editor has tabs for Source and History, and a toolbar with various icons for navigation and search. The status bar at the bottom shows "SemanaUno.java - Navigator" and "Members <empty>".

EJEMPLO NUEVO PUBLIC, PRIVADO Y PROTECTED

The screenshot shows an IDE interface with the following details:

- Projects:** A tree view showing a project named "semanaUno" with a "Source Packages" folder containing "com.mycompany.semanauno" which includes files like "Alumno.java", "SemanaUno.java", "clas_Perro.java", "clas_encapsulamiento.java", and "met_Persona.java".
- Files:** The file "SemanaUno.java" is open in the editor.
- Code:** The code defines a public class "SemanaUno" with a main method. Inside the main method, an object of type "clas_encapsulamiento" is created and assigned to a variable "obj_clas_encapsulamiento".
- Completion:** A code completion tooltip is displayed at the cursor position, listing the following members:
 - ? var_protected int
 - var_publica int
 - equals (Object obj) boolean
 - getClass () Class<??>

Notar que solo se pueden ver las variables de protected y la pública (estos se pueden ver desde el objeto)

EJEMPLO NUEVO PUBLIC, PRIVATE Y PROTECTED

The screenshot shows a Java development environment with the following details:

- Project Structure:** The project is named "semanaUno". It contains a "Source Packages" folder with a package named "com.mycompany.semanauno". Inside this package are five files: "Alumno.java", "SemanaUno.java", "clas_Perro.java", "clas_encapsulamiento.java", and "met_Persona.java".
- Code Editor:** The file "SemanaUno.java" is open. The code defines a public class "SemanaUno" with a main method. In the main method, an object of type "clas_encapsulamiento" is created and assigned to the variable "obj_clas_encapsulamiento".
- IntelliJ IDEA's Completion Feature:** A tooltip is displayed over the variable "obj_clas_encapsulamiento", showing completion suggestions. The suggestions include "var static int", "class", and "com.mycompany.semanauno.clas_encapsulamiento". Below these suggestions, a note states: "Javadoc not found. Either Javadoc documentation for this item does not exist or there is no attached Javadoc to the JAR file containing this item:C:\1\Java\semanaUno\target\classes".

Notar que la variable estática solo se puede ver desde la clase en sí, Es decir static sólo es visible desde la clase misma.

EJEMPLO NUEVO PUBLIC, PRIVATE Y PROTECTED

The screenshot shows a Java development environment with the following details:

- Projects View:** A tree view showing a project named "semanaUno". Under "Source Packages", there is a package "com.mycompany.semanauno" containing files: Alumno.java, SemanaUno.java, clas_Perro.java, clas_encapsulamiento.java, and met_Persona.java.
- Code Editor:** The file "SemanaUno.java" is open. The code is as follows:

```
1 package com.mycompany.semanauno;
2
3 public class SemanaUno {
4
5     public static void main(String[] args) {
6
7         //Instanciamos o llamamos la clase
8         clas_encapsulamiento obj_clas_encapsulamiento = new clas_encapsulamiento();
9
10        clas_encapsulamiento.var_static = 3;
11
12    }
13
14 }
15
16
17 }
```

- Toolbars:** Standard Java editor toolbars for navigation, search, and code manipulation.
- Status Bar:** Shows "SemanaUno.java - Navigator" at the bottom left.

Notar que la variable estática solo se puede ver desde la clase en sí, Es decir static sólo es visible desde la clase misma.

EJEMPLO NUEVO PUBLIC, PRIVATE Y PROTECTED

The screenshot shows a Java development environment with the following details:

- Project Explorer (Left):** A tree view of the project structure under "semanaUno".
 - Source Packages:** com.mycompany.semanauno
 - Alumno.java
 - SemanaUno.java (selected)
 - clas_Perro.java
 - clas_encapsulamiento.java
 - met_Persona.java
 - Test Packages
 - Dependencies
 - Java Dependencies
 - Project Files
- Editor Area (Center):** The code editor displays SemanaUno.java.

```
1 package com.mycompany.semanauno;
2
3 public class SemanaUno {
4
5     public static void main(String[] args) {
6
7         //Instanciamos o llamamos la clase
8         clas_encapsulamiento obj_clas_encapsulamiento = new clas_encapsulamiento();
9         clas_encapsulamiento.var_static = 3;
10
11         obj_clas_encapsulamiento.var_protected = 3;
12         obj_clas_encapsulamiento.var_publica = 3;
13
14     }
15
16 }
17
18 }
```
- Bottom Navigation Bar:** Shows "SemanaUno.java - Navigator" and other tabs.

Notar que la variable estática solo se puede ver desde la clase en sí, Es decir static sólo es visible desde la clase misma.

EJEMPLO NUEVO PUBLIC, PRIVATE Y PROTECTED

The screenshot shows a Java development environment with the following interface elements:

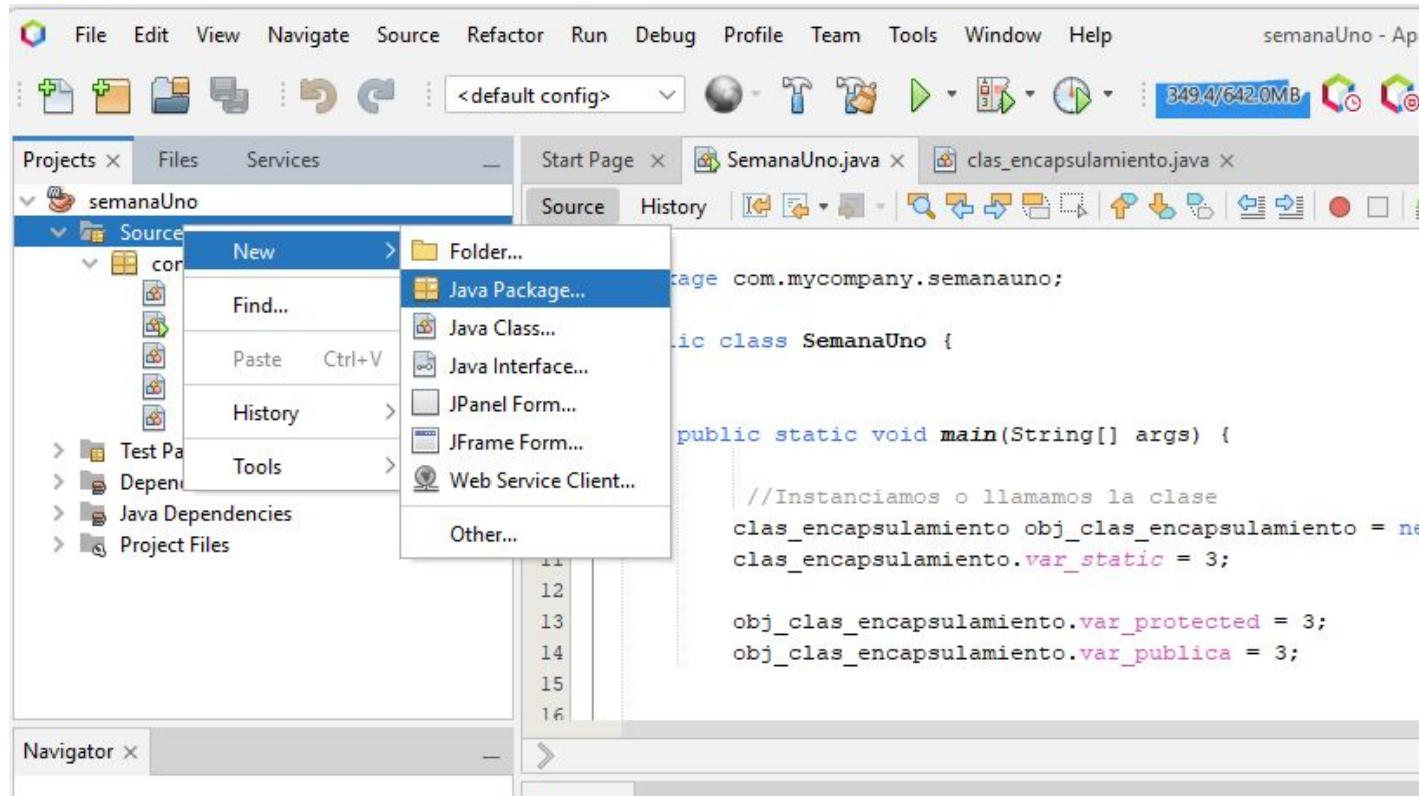
- Projects** tab: Shows a project named "semanaUno" containing "Source Packages" like "com.mycompany.semanauno" which includes files "Alumno.java", "SemanaUno.java", "clas_Perro.java", "clas_encapsulamiento.java", and "met_Persona.java".
- Start Page** tab: Active tab, showing the file "SemanaUno.java".
- Source** tab: Selected tab, showing the code for "SemanaUno.java".
- History** tab: Unselected tab.
- Toolbar**: Standard Java development toolbar with icons for file operations, search, and navigation.
- Code Editor**: Displays the following Java code:

```
1 package com.mycompany.semanauno;
2
3 public class SemanaUno {
4
5     public static void main(String[] args) {
6
7         //Instanciamos o llamamos la clase
8         clas_encapsulamiento obj_clas_encapsulamiento = new clas_encapsulamiento();
9         clas_encapsulamiento.var_static = 3;
10
11         obj_clas_encapsulamiento.var_protected = 3;
12         obj_clas_encapsulamiento.var_publica = 3;
13
14     }
15
16 }
17
18 }
19
```

The code demonstrates the use of `public`, `private`, and `protected` access modifiers. The variable `var_static` is `private` to the class and can only be accessed within its own scope. The variables `var_protected` and `var_publica` are `protected` and can be accessed from within the same package or from subclasses in other packages.

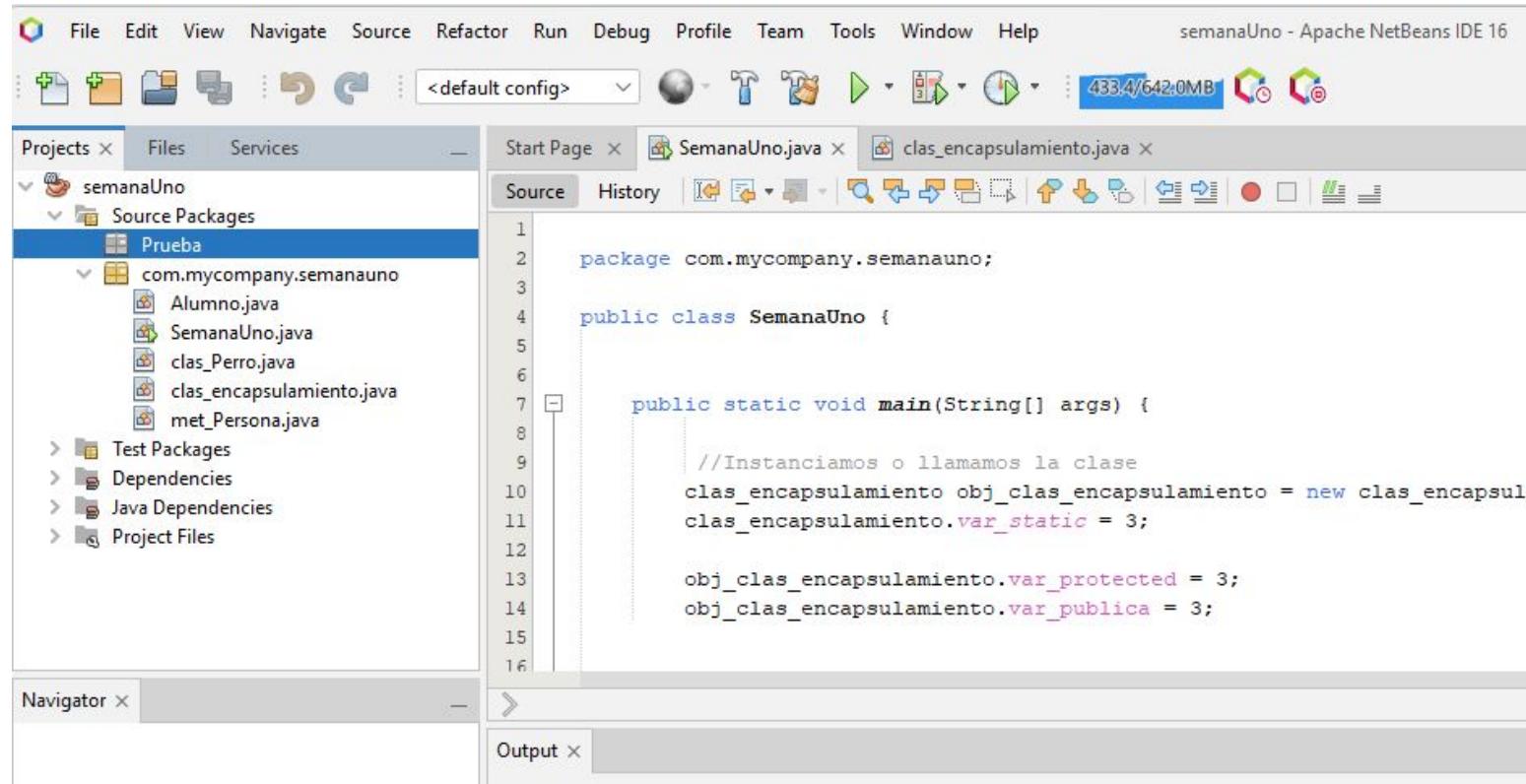
La variable PROTECTED solo puede ser vista dentro del mismo paquete en donde se encuentra la clase, verifiquemos en la siguiente filmina

EJEMPLO NUEVO PUBLIC, PRIVATE Y PROTECTED



Creamos un nuevo paquete, para crearle una clase y así probar el concepto de Protected

EJEMPLO NUEVO PUBLIC, PRIVATE Y PROTECTED

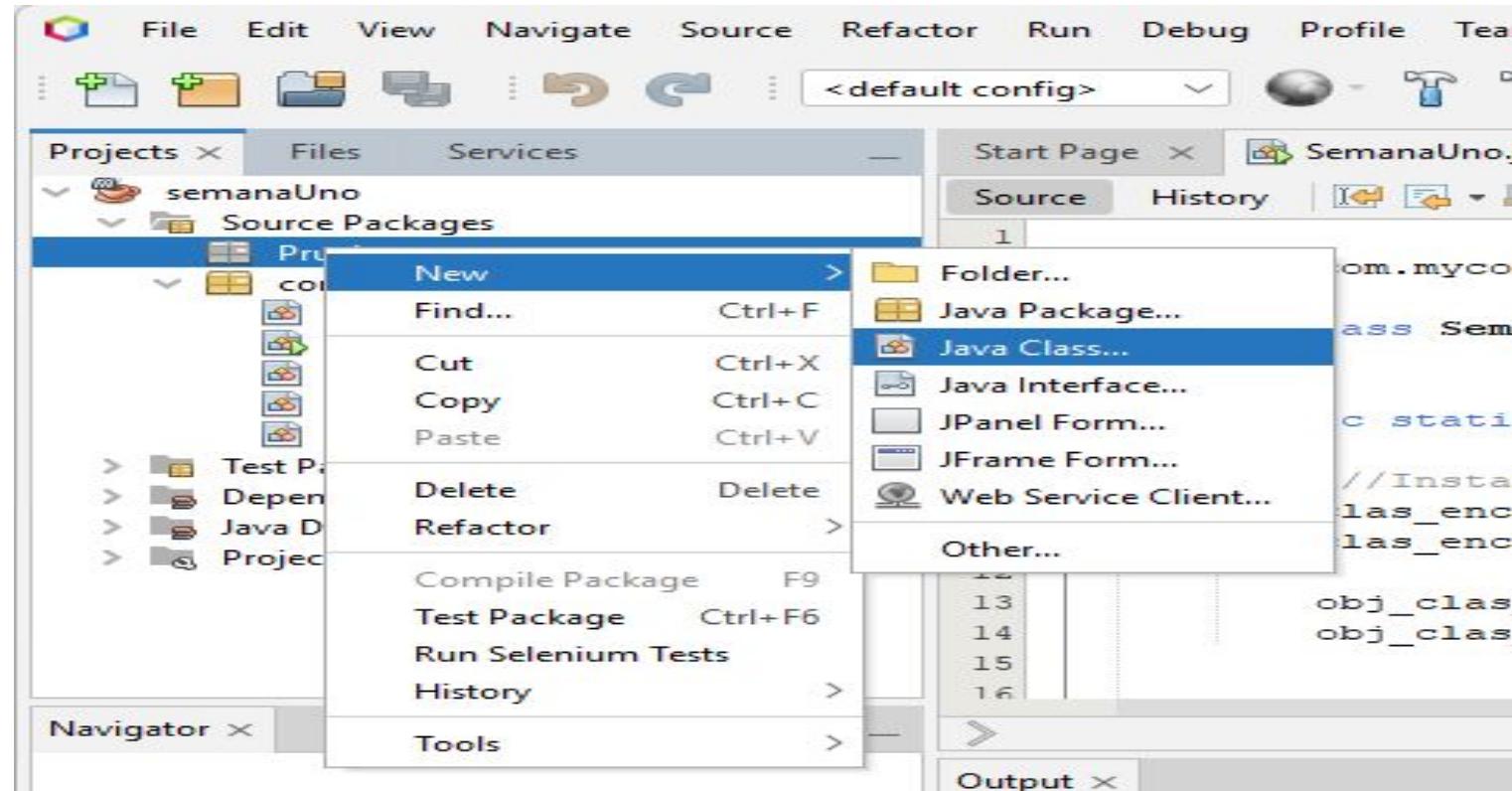


The screenshot shows the Apache NetBeans IDE 16 interface. The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The title bar indicates "semanaUno - Apache NetBeans IDE 16". The toolbar contains various icons for file operations like New, Open, Save, and Build. The Projects tab shows a project named "semanaUno" with a "Source Packages" node expanded, containing a package named "Prueba" which contains several Java files: Alumno.java, SemanaUno.java, clas_Perro.java, clas_encapsulamiento.java, and met_Persona.java. The Files tab is also visible. The central workspace displays the Java code for "SemanaUno.java". The code defines a package named "com.mycompany.semanauno" and a public class "SemanaUno". It includes a main method that creates an object of "clas_encapsulamiento" and sets its static variable "var_static" to 3. It also sets two protected variables ("var_protected" and "var_publica") to 3. The code editor has line numbers from 1 to 16. Below the code editor are tabs for Navigator and Output.

```
1 package com.mycompany.semanauno;
2
3 public class SemanaUno {
4
5
6     public static void main(String[] args) {
7
8         //Instanciamos o llamamos la clase
9         clas_encapsulamiento obj_clas_encapsulamiento = new clas_encapsula
10        clas_encapsulamiento.var_static = 3;
11
12        obj_clas_encapsulamiento.var_protected = 3;
13        obj_clas_encapsulamiento.var_publica = 3;
14
15
16 }
```

Allí está el paquete listo, ahora le fabricamos una clase a ese paquete

EJEMPLO NUEVO PUBLIC, PRIVATE Y PROTECTED



Allí está el paquete listo, ahora le fabricamos una clase a ese paquete