

UNIDAD 4

INTRODUCCIÓN A JAVA

GUÍA DE NETBEANS

**Programación
CFGS DAW**

Autores: Carlos Cacho, Raquel Torres, Lionel Tarazon, Fco. Javier Valero

Revisado por:

Paco Aldarias paco.aldarias@ceedcv.es

2020/2021

30/10/20 12:45:42

Licencia



Reconocimiento - NoComercial - CompartirIgual (by-nc-sa):
No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original. NOTA: Esta es una obra derivada de la obra original realizada por Carlos Cacho y Raquel Torres.

Nomenclatura

A lo largo de este tema se utilizarán distintos símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:



Importante



Atención



Interesante

ÍNDICE

1. Introducción.....	4
2. Instalación.....	4
2.1 Descarga de JDK & Netbeans IDE.....	4
2.2 Instalación de JDK & Netbeans IDE.....	5
3. Creación de proyectos.....	8
3.1 Conceptos básicos.....	8
3.2 Creación de un proyecto.....	9
3.3 Creación de ficheros.....	11
4. Compilación y ejecución de aplicaciones.....	13
4.1 Compilación.....	13
4.2 Ejecución.....	15
5. Agradecimientos.....	16

UD4. INTRODUCCIÓN A JAVA

1. INTRODUCCIÓN

NetBeans es un entorno de desarrollo integrado (IDE) libre, desarrollado principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo.

Es importante destacar que es un proyecto de código abierto de gran éxito, con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo.

La plataforma NetBeans, además, permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

NetBeans IDE soporta el desarrollo de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles). Entre sus características se encuentra un sistema de proyectos basado en Ant, control de versiones y refactoring.

2. INSTALACIÓN

Para este módulo es necesario tener instalada la JDK (Java Development Toolkit) y el entorno de desarrollo NetBeans. Es suficiente con tener, en ambos casos, al menos la versión 8. Cualquier otra versión posterior también servirá.

Si ya lo tienes instalado de otros años o por estar cursando el módulo de Entornos de Desarrollo, puedes saltarte este apartado e ir directamente al apartado 3.

2.1 Descarga de JDK & Netbeans IDE

Si no tienes JDK ni NetBeans te aconsejamos que sigas el siguiente enlace de Oracle en el que podrás descargar un instalador que te permite instalar la

versión 8 de ambos fácilmente:

<http://www.oracle.com/technetwork/es/java/javase/downloads/jdk-netbeans-jsp-3413139-esa.html>






Una vez en la página deberemos aceptar la licencia y elegir la descarga del sistema operativo que estemos utilizando.

JDK 8u111 with NetBeans 8.2

This distribution of the JDK includes the Java SE bundle of NetBeans IDE, which is a powerful integrated development environment for developing applications on the Java platform. [Learn more](#)

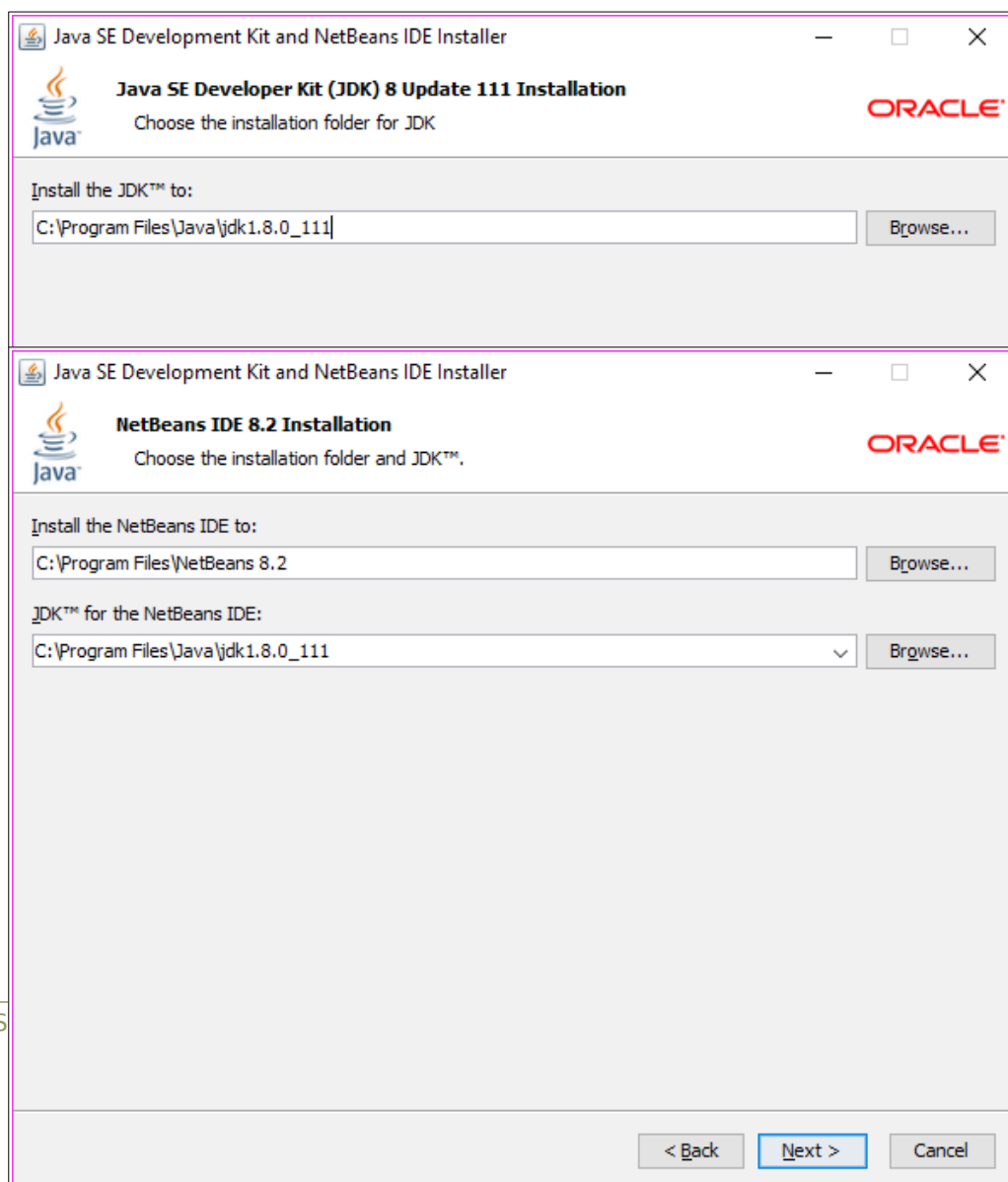
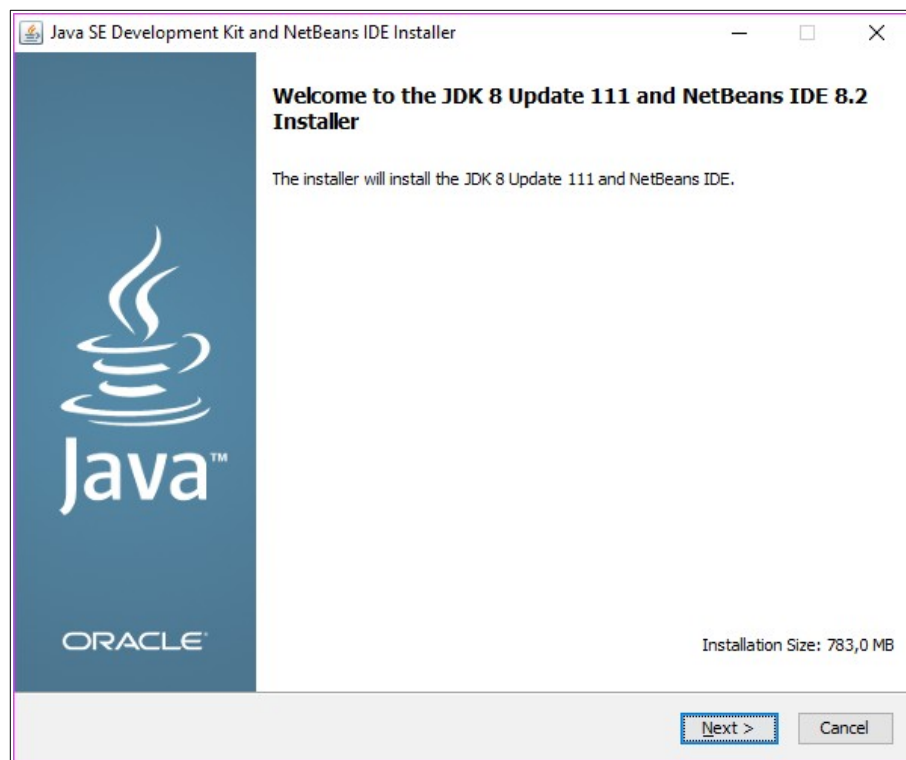
You must accept the JDK 8u111 and NetBeans 8.2 Cobundle License Agreement to download this software.

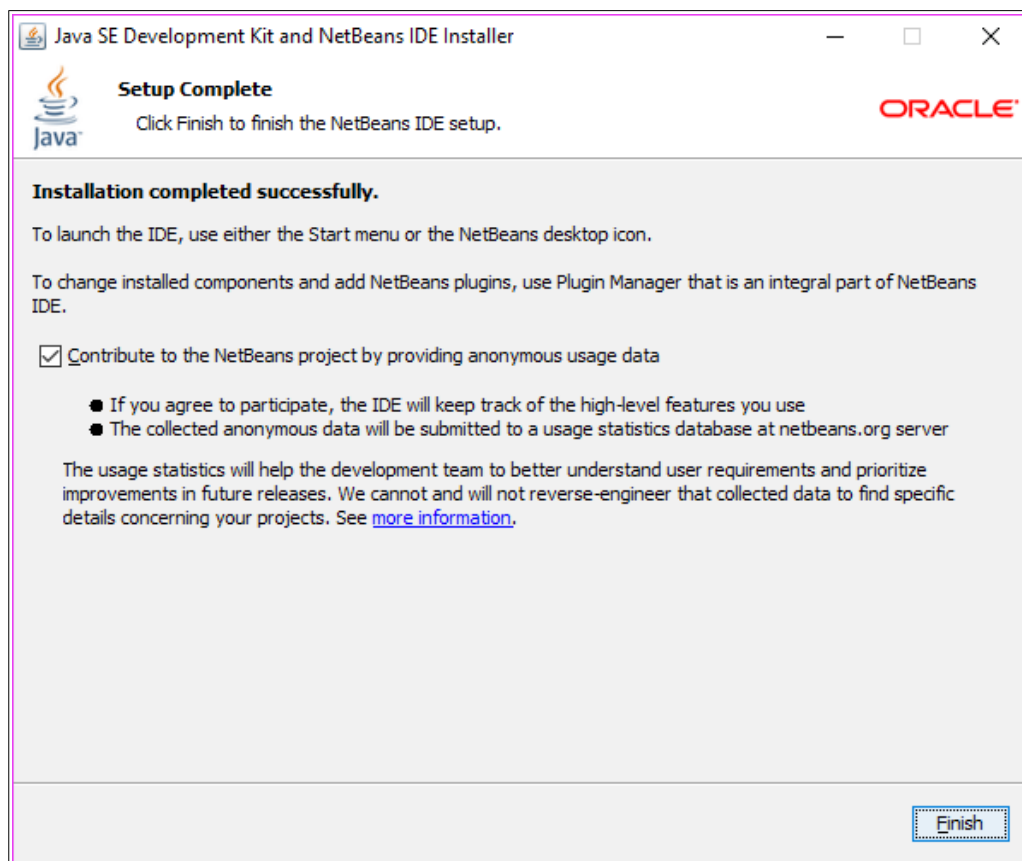
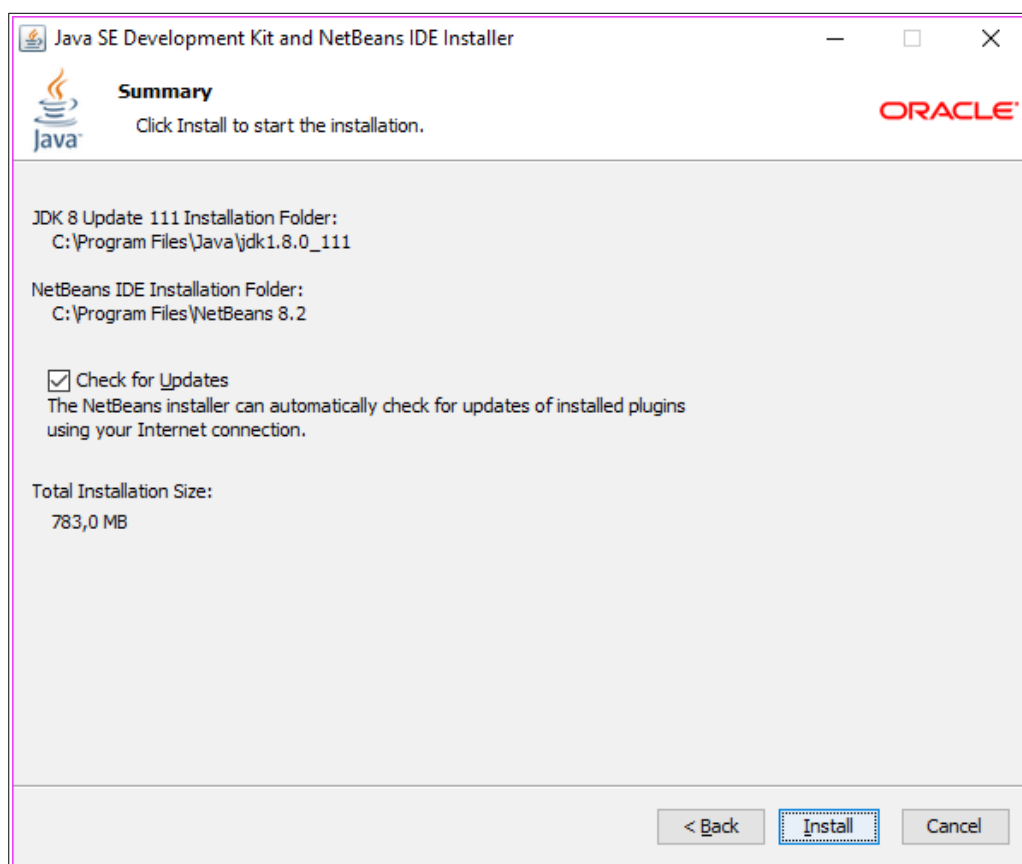
☐ Accept License Agreement ☒ Decline License Agreement

Java SE and NetBeans Cobundle (JDK 8u111 and NB 8.2)		
Product / File Description	File Size	Download
Linux x86	286.73 MB	 jdk-8u111-nb-8_2-linux-i586.sh
Linux x64	282.57 MB	 jdk-8u111-nb-8_2-linux-x64.sh
Mac OS X x64	342.99 MB	 jdk-8u111-nb-8_2-macosx-x64.dmg
Windows x86	317.21 MB	 jdk-8u111-nb-8_2-windows-i586.exe
Windows x64	326.03 MB	 jdk-8u111-nb-8_2-windows-x64.exe

2.2 Instalación de JDK & Netbeans IDE

Una vez descargado el paquete, la instalación es muy sencilla, simplemente hemos de seguir los pasos del instalador:





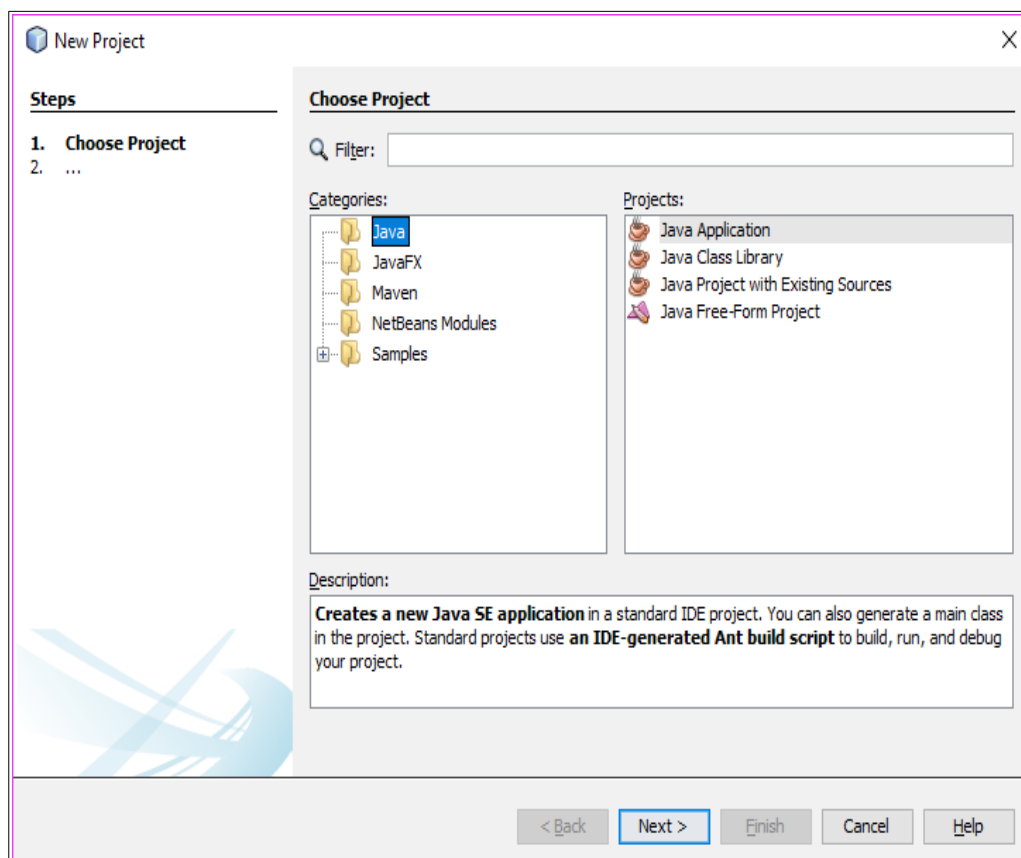
3. CREACIÓN DE PROYECTOS

3.1 Conceptos básicos

Un **proyecto** es un conjunto de archivos fuente en java que contiene información asociada a ese conjunto de archivos, como es el classpath, archivos de librería, estructura de directorios, etc. Toda esta información reside en una carpeta que normalmente se llama igual que el nombre del proyecto. En esta carpeta añade un fichero .xml que es el que se encarga de fijar los pasos para la compilación, depuración, ejecución y empaquetado del mismo, este mecanismo se conoce como Ant Project (<http://ant.apache.org>).

Cuando creamos un proyecto, el IDE genera de forma automática un script Ant para compilar el proyecto, o bien utilizar un script de Ant que tengamos predefinido.

Si accedemos a **File -> New Project**, vemos que por defecto el IDE contiene un conjunto de plantillas que podemos utilizar en función del proyecto que vayamos a crear.



Proyectos estándar (El IDE genera un script Ant para compilar):

- Java Application. Plantilla para crear un esqueleto de un proyecto J2SE con una clase principal.

- Java Class Library. Plantilla para crear un esqueleto en Java sin clase principal.
- Java Project with Existing Sources. Plantilla para crear un proyecto J2SE basado en un conjunto de ficheros java existentes.
- Java Free-Form Project. Plantilla para crear un proyecto importando una aplicación Java existente y su script de construcción en un proyecto de IDE.

3.2 Creación de un proyecto

Como hemos comentado, para crear un proyecto **File -> New Project (Ctrl. + Mayús + N)**, a continuación aparecerá el asistente para la creación del proyecto.

Para nuestro ejemplo escogeremos de la categoría general el proyecto de **Java Application** (aplicación java J2SE estándar).

Una vez seleccionada la plantilla hacemos clic en Next y rellenamos el nombre del proyecto, el nombre que le demos al proyecto servirá como nombre de paquete y creará una carpeta con ese mismo nombre.

Para nuestro ejemplo pondremos como nombre **Ejemplo1**. Guardaremos el proyecto en una carpeta de nuestro sistema, seleccionando esta carpeta en la segunda caja de texto.

New Java Application

Steps

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

Name and Location

Project Name: Ejemplo1

Project Location: C:\Users\user\Documents\NetBeansProjects Browse...

Project Folder: C:\Users\user\Documents\NetBeansProjects\Ejemplo1

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder: Browse...

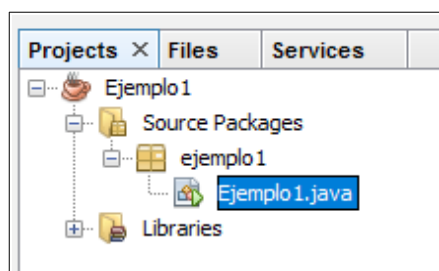
Different users and projects can share the same compilation libraries (see Help for details).

☒ Create Main Class ejemplo1.Ejemplo1

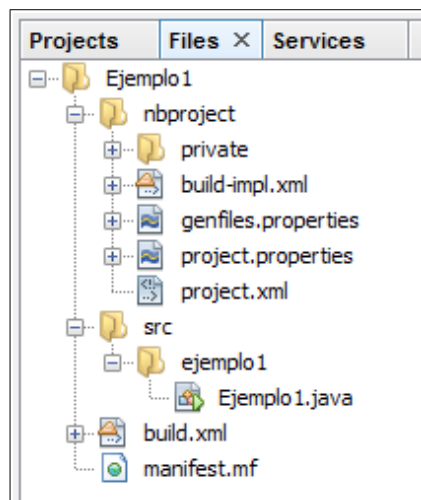
< Back Next > Finish Cancel Help

Una vez hemos creado el proyecto, en la parte izquierda podemos ver tres pestañas: la del proyecto (Projects), la de los archivos (Files) y la de Servicios (Services).

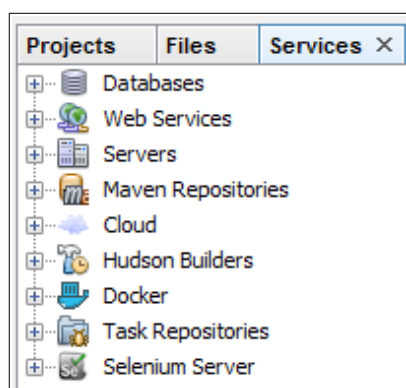
La pestaña de proyectos es el punto principal de entrada a los ficheros fuentes del proyecto. Muestra una vista lógica de la estructura del proyecto en cuanto a paquetes y ficheros que lo componen. Mediante el botón de la derecha del ratón y teniendo seleccionado el nombre del proyecto (en nuestro caso Ejemplo1), podemos acceder al menú contextual para modificar las opciones de compilación, debugger y ejecución del proyecto.



La pestaña de ficheros muestra la vista de ficheros basada en una estructura de directorios, incluyendo ficheros y carpetas que no se muestran en la ventana de proyectos. Desde aquí podemos editar, modificar tanto ficheros fuentes del proyecto como scripts de compilación (Ant). Podemos acceder al proyecto compilado y paquetizado en ficheros JAR, WAR, etc.



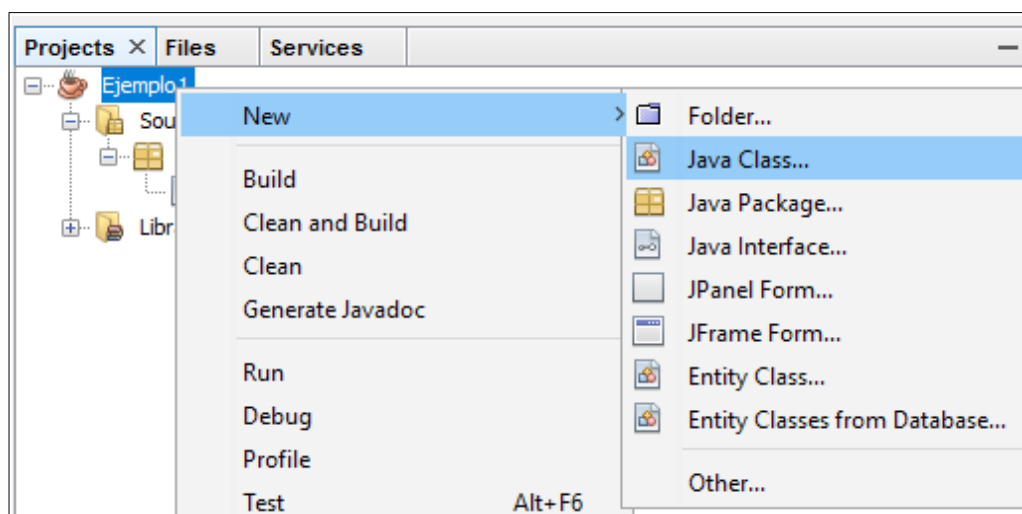
En la pestaña de servicios encontramos los diferentes servicios de los que podemos hacer uso desde nuestra aplicación, como la conexión a bases de datos o a un servidor web.



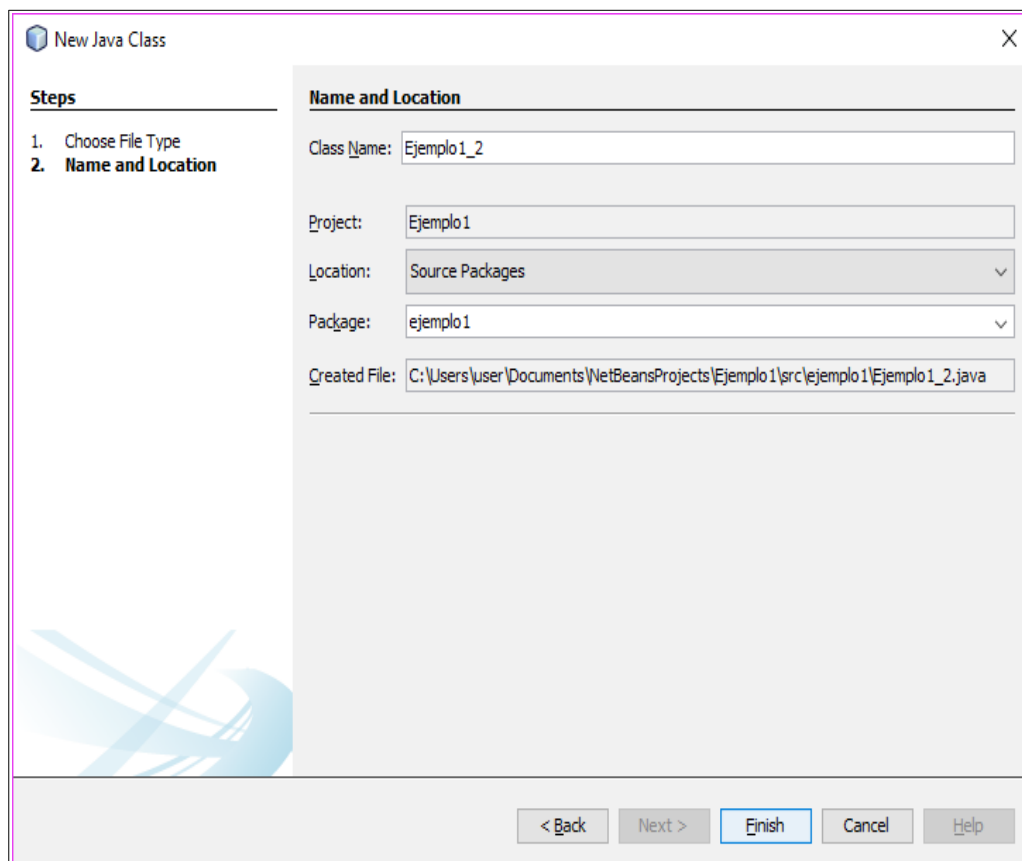
3.3 Creación de ficheros

Cuando creamos un proyecto se crea automáticamente un fichero, en este caso `Ejemplo1.java`. Si quisiéramos crear uno nuevo tendríamos que pinchar con el botón derecho sobre el nombre del proyecto.

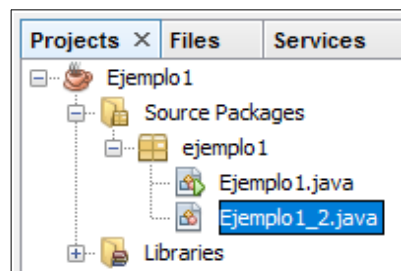
En nuestro caso accedemos al proyecto **Ejemplo1** y con el botón de la derecha accedemos al menú **New**. Dependiendo del tipo de plantilla seleccionado nos aparecerá un asistente que nos indicará los pasos necesarios para terminar la creación de la plantilla, seleccionamos **Java Class...**



Rellenamos el nombre de la clase y el nombre del package **ejemplo1**.



Así tendremos nuestro nuevo fichero.



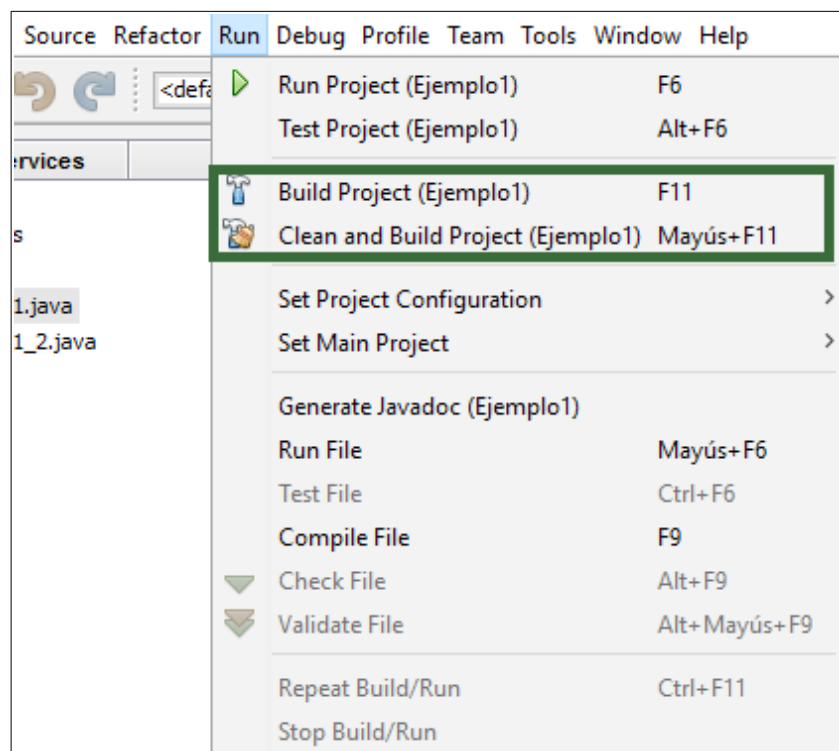
4. COMPILACIÓN Y EJECUCIÓN DE APLICACIONES

4.1 Compilación

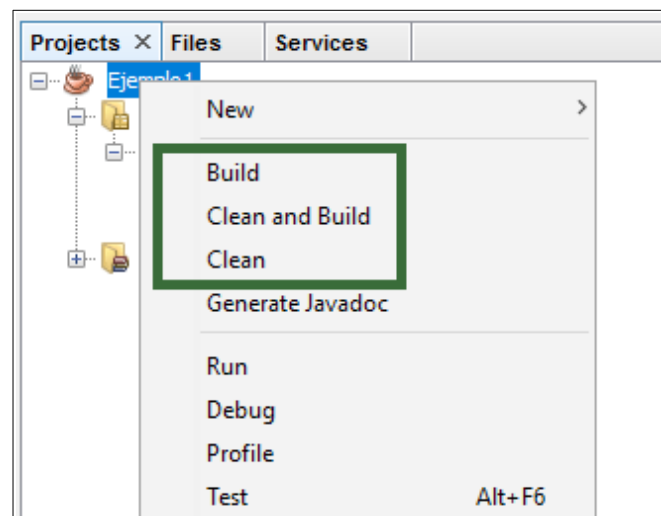
Para hacer una compilación con el IDE es bastante simple. Lo primero es seleccionar el paquete, proyecto o archivo que queramos compilar.

Para compilar un proyecto, paquete o archivo, nos situamos en la ventana de proyectos y elegimos una de las siguientes opciones de la barra de herramientas:

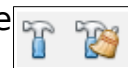
- En el menú principal, hacemos clic en **Run-> Build Project (F11)**, para compilar el proyecto.
- En el menú principal, hacemos clic en **Run-> Clean and Build Project (Shift + F11)**, para limpiar y compilar el proyecto.



- En la ventana de proyectos botón de la derecha sobre el nombre del proyecto **Build Project**, para compilar el proyecto.
- En la ventana de proyectos botón de la derecha sobre el nombre del proyecto **Clean and Build Project**, para eliminar los archivos objeto y compilar el proyecto.
- En la ventana de proyectos botón de la derecha sobre el nombre del proyecto **Clean**, para eliminar los archivos objeto del proyecto.



- También se pueden utilizar los botones de la barra de herramientas:



Siguiendo nuestro ejemplo probamos a compilar el proyecto principal Ejemplo1. La salida sería la siguiente:

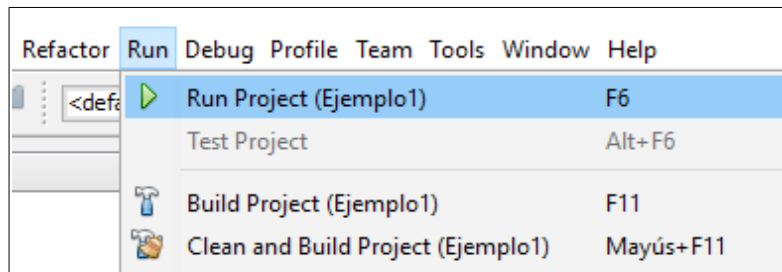
```
Output - Ejemplo1 (clean.jar) X
ant -f C:\Users\user\Documents\NetBeansProjects\Ejemplo1 -Dnb.internal.action.name=rebuild clean jar
init:
deps-clean:
Updating property file: C:\Users\user\Documents\NetBeansProjects\Ejemplo1\build\build-clean.properties
Deleting directory C:\Users\user\Documents\NetBeansProjects\Ejemplo1\build
clean:
init:
deps-jar:
Created dir: C:\Users\user\Documents\NetBeansProjects\Ejemplo1\build
Updating property file: C:\Users\user\Documents\NetBeansProjects\Ejemplo1\build\build-jar.properties
Created dir: C:\Users\user\Documents\NetBeansProjects\Ejemplo1\build\classes
Created dir: C:\Users\user\Documents\NetBeansProjects\Ejemplo1\build\empty
Created dir: C:\Users\user\Documents\NetBeansProjects\Ejemplo1\build\generated-sources\ap-source-output
Compiling 2 source files to C:\Users\user\Documents\NetBeansProjects\Ejemplo1\build\classes
compile:
Created dir: C:\Users\user\Documents\NetBeansProjects\Ejemplo1\dist
Copying 1 file to C:\Users\user\Documents\NetBeansProjects\Ejemplo1\build
Nothing to copy.
Building jar: C:\Users\user\Documents\NetBeansProjects\Ejemplo1\dist\Ejemplo1.jar
To run this application from the command line without Ant, try:
java -jar "C:\Users\user\Documents\NetBeansProjects\Ejemplo1\dist\Ejemplo1.jar"
jar:
BUILD SUCCESSFUL (total time: 0 seconds)
```

Cualquier error que haya en compilación aparece en la ventana de salida Output (Ctrl. + 4). La ventana de salida es una ventana multi-tabulada que muestra todos los mensajes que haya habido en la compilación. El mensaje de error aparece subrayado en azul y es un hipervínculo a la línea de código que ha causado dicho error.

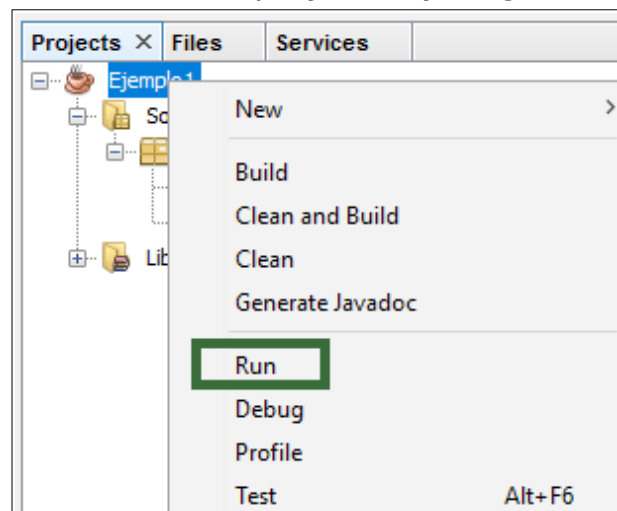
4.2 Ejecución

Para ejecutar una aplicación podemos:

- Hacer clic en **Run -> Run Main Project (F6)**



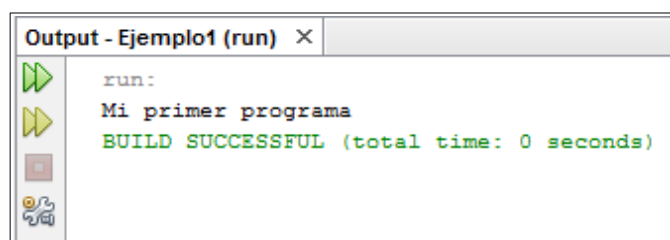
- Hacer clic en los iconos de la barra de herramientas del
- Hacer clic con el botón derecho sobre su nodo del proyecto en la ventana de los proyectos y elegir **Run Project**



Si modificamos nuestro fichero java e insertamos una línea para mostrar por pantalla el mensaje “Mi primer programa”,

```
6 package ejemplo1;
7
8 /**
9  *
10  * @author user
11  */
12 public class Ejemplo1 {
13
14     /**
15      * @param args the command line arguments
16      */
17     public static void main(String[] args) {
18         // TODO code application logic here
19         System.out.println("Mi primer programa");
20     }
21
22 }
```

compilamos y ejecutamos la salida sería la siguiente:



5. AGRADECIMIENTOS

Apuntes actualizados y adaptados al CEEDCV a partir de la siguiente documentación:

- [1] Apuntes Programación de José Antonio Díaz-Alejo. IES Camp de Morvedre.