

# Manual Netbeans



Encarnación Sánchez Gallego

# Contenido

- Introducción
- Creación de proyectos
- Compilación de proyectos/archivos
- Ejecución del proyecto





# 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**

# Creación de proyectos



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 Project

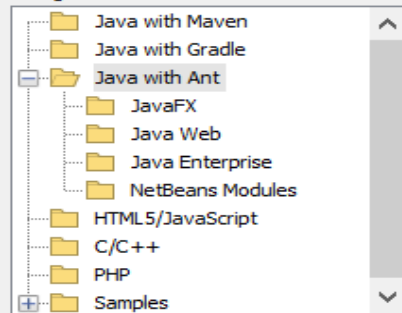
### Steps

1. Choose Project
2. ...

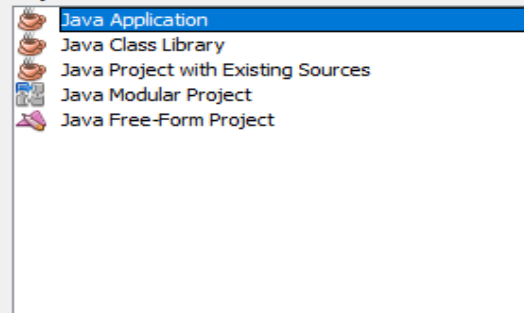
### Choose Project

Filter:

Categories:



Projects:



Description:

**Creates a new Java SE application** in a standard IDE project. You can also generate a main class in the project. Standard projects use **an IDE-generated Ant build script** to build, run, and debug your project.

< Back

Next >

Finish

Cancel

Help





New Java Application

**Steps**

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

**Name and Location**

Project Name:

Project Location:

Project Folder:

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:

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

☒ Create Main Class

< Back   Next >   **Finish**   Cancel   Help

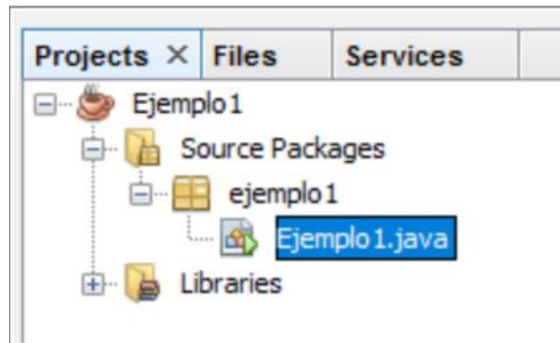


# Creación de proyectos



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).

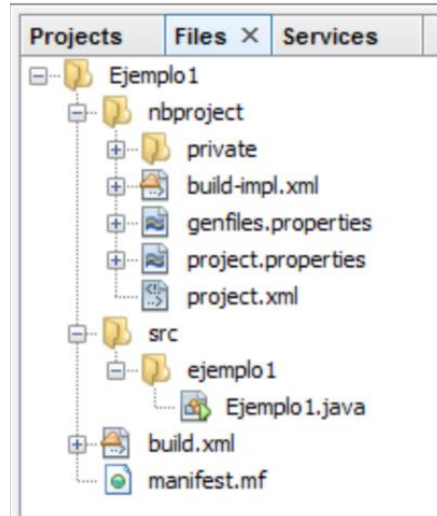
La pestaña de proyectos es el punto principal de entrada a los ficheros fuentes del proyecto. Muestra una vista lógica de la estructura del proyecto en cuanto a paquetes y ficheros que lo componen. Mediante el botón de la derecha del ratón y teniendo seleccionado el nombre del proyecto (en nuestro caso 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).

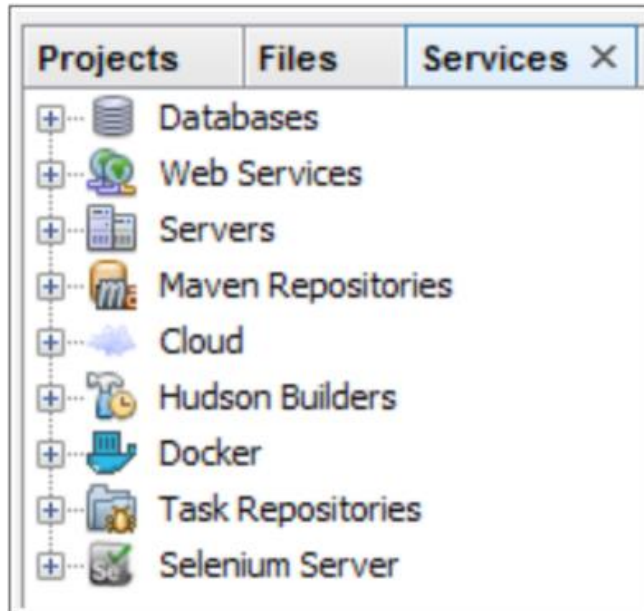
Podemos acceder al proyecto compilado y paquetizado en ficheros JAR, WAR, etc.







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.





# 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...**

New Java Class

**Steps**

1. Choose File Type
2. **Name and Location**

**Name and Location**

Class Name: OtroEjemplo

Project: Ejemplo1

Location: Source Packages

Package: ejemplo1

Created File: c:\pns\OneDrive\Documentos\NetBeansProjects\Ejemplo1\src\ejemplo1\OtroEjemplo.java

Superclass:  Browse...

Interfaces:  Browse...

< Back Next > Finish Cancel Help

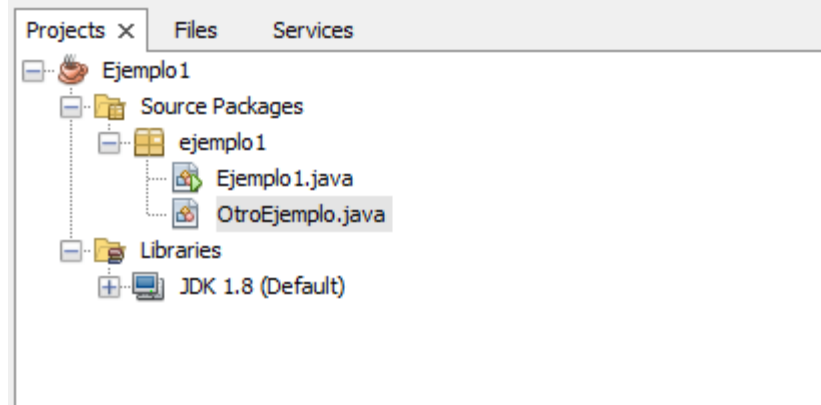
Añadimos el nombre de la clase y le asignamos el paquete





# Creación de ficheros

Así tendremos nuestro nuevo fichero.





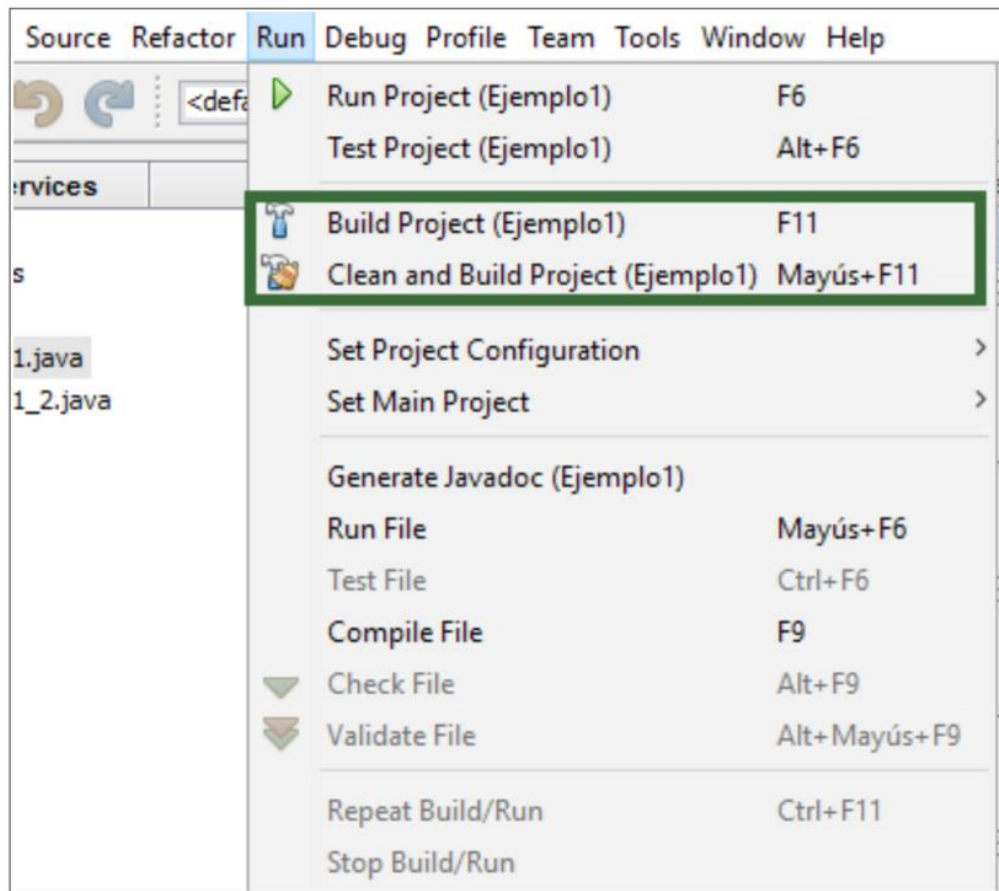
# Compilación de proyectos/archivos

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:

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





Apache  
**NetBeans** IDE

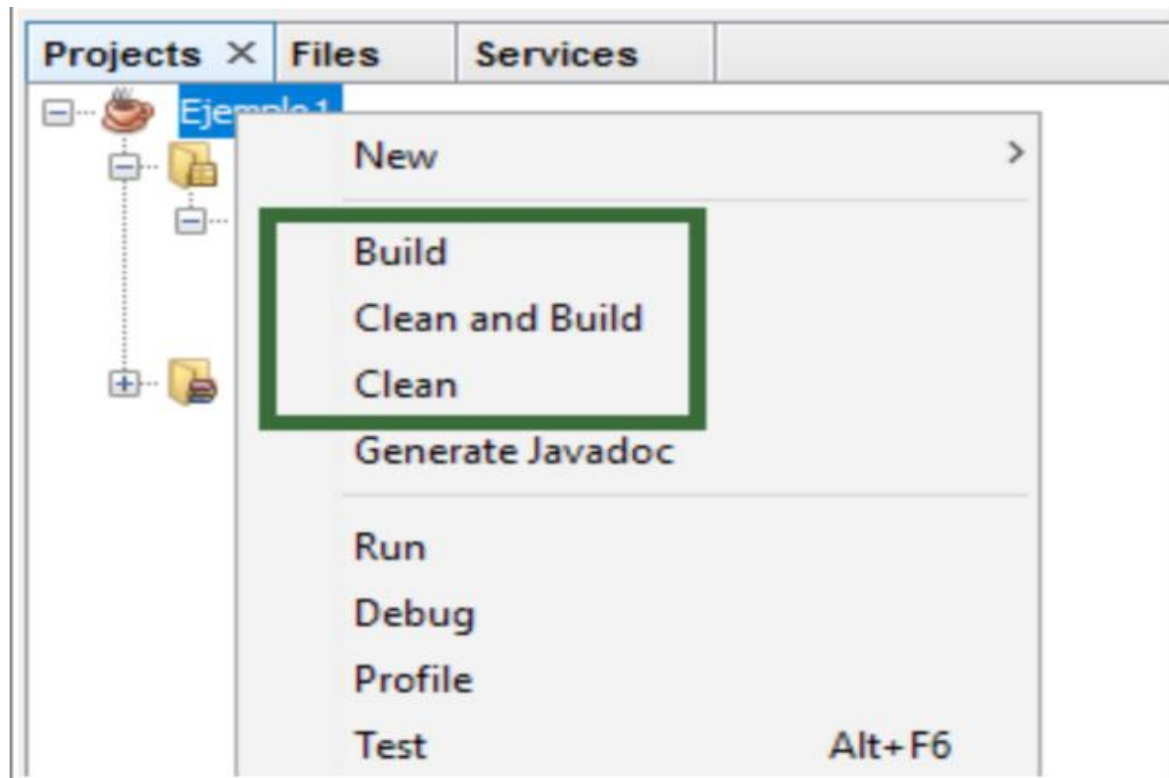


# Compilación de proyectos/archivos

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.

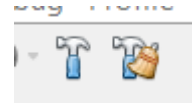






# Compilación de proyectos/archivos

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:





# Compilación de proyectos/archivos



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



Apache  
**NetBeans** IDE



# Compilación de proyectos/archivos

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.

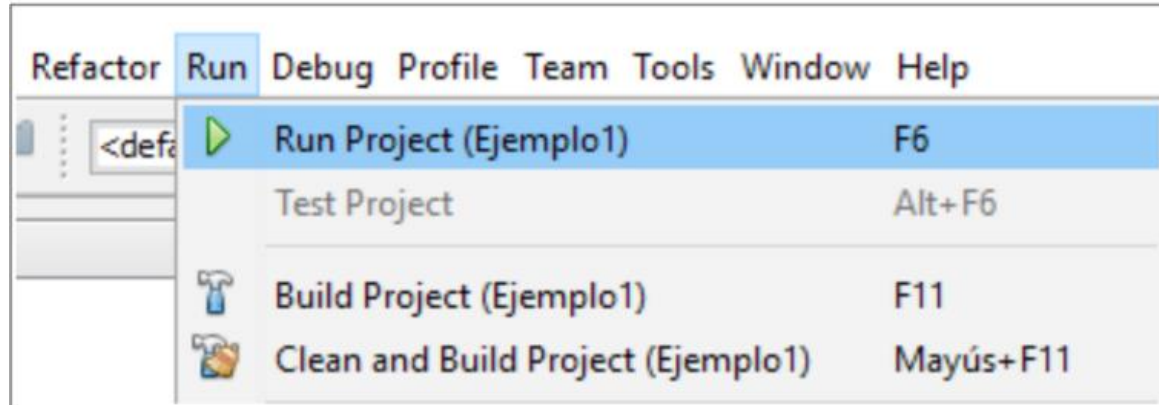




# Ejecución del proyecto

Para ejecutar una aplicación podemos:

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



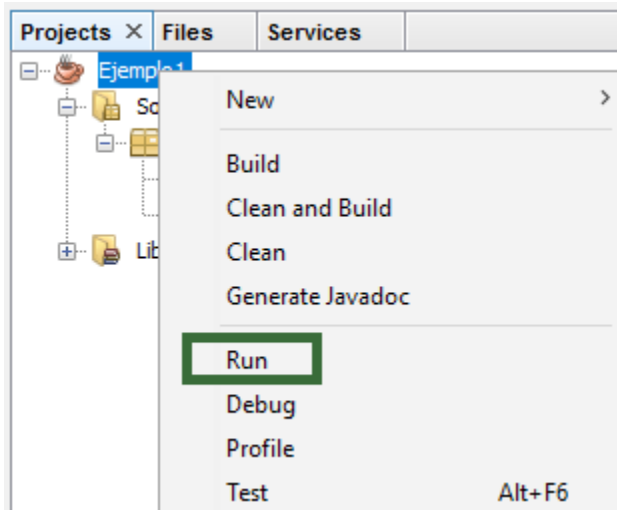
# Ejecución del proyecto



Hacer clic en los iconos de la barra de herramientas



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 }
```



Apache  
**NetBeans** IDE

# Ejecución del proyecto



Si ejecutamos nuestro programa saldrá el mensaje que habíamos añadido al principio:

