

# Introducción a Netbeans

Programación 2 – Curso 2010/2011

Juan Manuel Gimeno  
José Luis González



# ¿Qué es Netbeans?

- NB es un entorno integrado de desarrollo o IDE (Integrated Development Environment)
- En él podemos realizar todas las tareas asociadas a la programación
  - Editar el código
  - Compilarlo
  - Ejecutarlo
  - Depurarlo

# ¿Por qué usarlo?

- Simplifica alguna de las tareas que, sobretodo en proyectos grandes, son tediosas
- Nos asiste (parcialmente) en la escritura de código, aunque no nos libera de aprender el lenguaje de programación
- Nos ayuda en la navegación de las clases predefinidas en la plataforma (miles)
- Aunque puede ser costoso su aprendizaje, los beneficios superan las dificultades

# La plataforma Java

Java Language											
JDK	Tools & Tool APIs	Java Language									
	java	javac	javadoc	apt	jar	javap	JPDA	JConsole	Java VisualVM		
	Security	Int'l	RMI	IDL	Deploy	Monitoring	Troubleshoot	Scripting	JVM TI		
	Deployment			Java Web Start				Java Plug-in			
	AWT			Swing			Java 2D				
	Accessibility		Drag n Drop		Input Methods		Image I/O	Print Service	Sound		
	IDL		JDBC™		JNDI™		RMI	RMI-IIOP			
	Beans		Intl Support		I/O	JMX		JNI			
	Networking		Override Mechanism		Security	Serialization		Extension Mechanism			
	lang and util Base Libraries		Collections		Concurrency Utilities		JAR		Logging		
JRE	lang and util Base Libraries		Ref Objects		Reflection		Regular Expressions		Management		
	Java Hotspot™ Client VM					Java Hotspot™ Server VM					
	Solaris™			Linux		Windows			Other		
Java SE API											

# Paso de compilación

```
import acm.program.ConsoleProgram;  
  
class HelloProgram extends ConsoleProgram {  
    public void run() {  
        println("Hello World");  
    }  
}
```

HelloProgram.java



acm.jar



Bibliotecas  
predefinidas

**javac**



```
ca fe ba be 00 00 00 32 00 14 0a 00 05  
00 0e 0a 00 04 00 0f 07 00 10 07 00 11  
3c 69 6e 69 74 3e 01 00 03 28 29 56 01  
6f 64 65 01 00 0f 4c 69 6e 65 4e 75 6d  
54 61 62 6c 65 01 00 03 72 75 6e 01 00  
75 72 63 65 46 69 6c 65 01 00 11 48 65  
.....
```

HelloProgram.class

# Paso de ejecución

```
ca fe ba be 00 00 00 32 00 14 0a 00 05  
00 0e 0a 00 04 00 0f 07 00 10 07 00 11  
3c 69 6e 69 74 3e 01 00 03 28 29 56 01  
6f 64 65 01 00 0f 4c 69 6e 65 4e 75 6d  
54 61 62 6c 65 01 00 03 72 75 6e 01 00  
75 72 63 65 46 69 6c 65 01 00 11 48 65  
.....
```

HelloProgram.class



acm.jar



Bibliotecas  
predefinidas

**HelloProgram**

**java**

Nombre de la  
clase principal  
(Main Class)

Ejecución en la máquina  
virtual de Java

# Hello World

The image shows a Mac OS X desktop environment. On the left, a terminal window titled "Terminal — bash — 64x17" displays a session where a user named "Sheldon" is building and running a Java program. The terminal output is as follows:

```
Sheldon >: ls
HelloProgram.java acm.jar
Sheldon >: javac -classpath .:acm.jar HelloProgram.java
Sheldon >: ls
HelloProgram.class HelloProgram.java acm.jar
Sheldon >: java -classpath .:acm.jar HelloProgram
Sheldon >: 
```

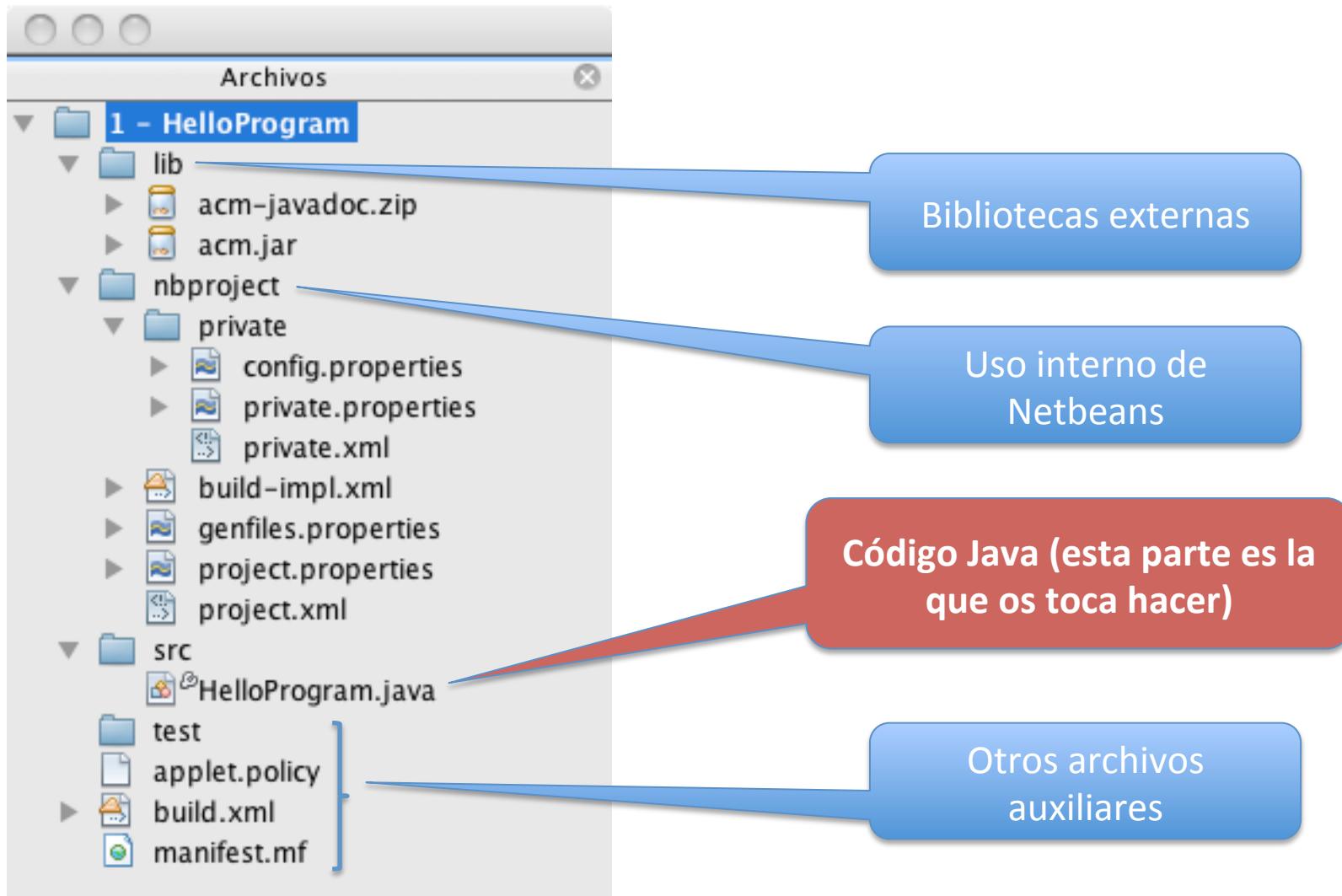
On the right, a Java application window titled "HelloProgram" is open, displaying the text "hello, world" in its main pane.

# **CARGA DE UN PROYECTO EXISTENTE**

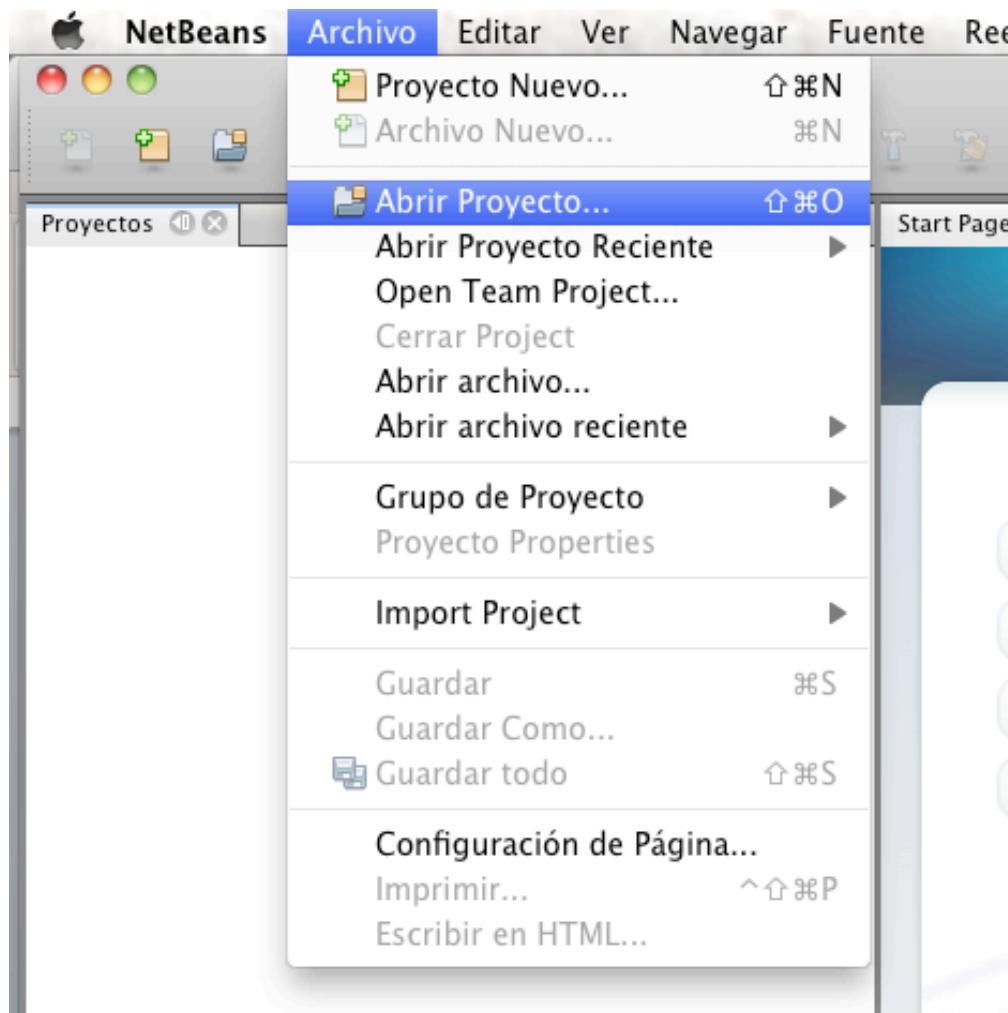
# Concepto de proyecto

- Netbeans no trabaja a nivel de archivo sino a nivel de proyecto
- Un proyecto incluye todos los recursos necesarios para construir un programa:
  - Archivos con el código
  - Bibliotecas externas (p.e. ACM Task Force)
  - Imágenes, sonidos, etc.
- *Físicamente* un proyecto Netbeans no es más que un directorio con una organización especial.

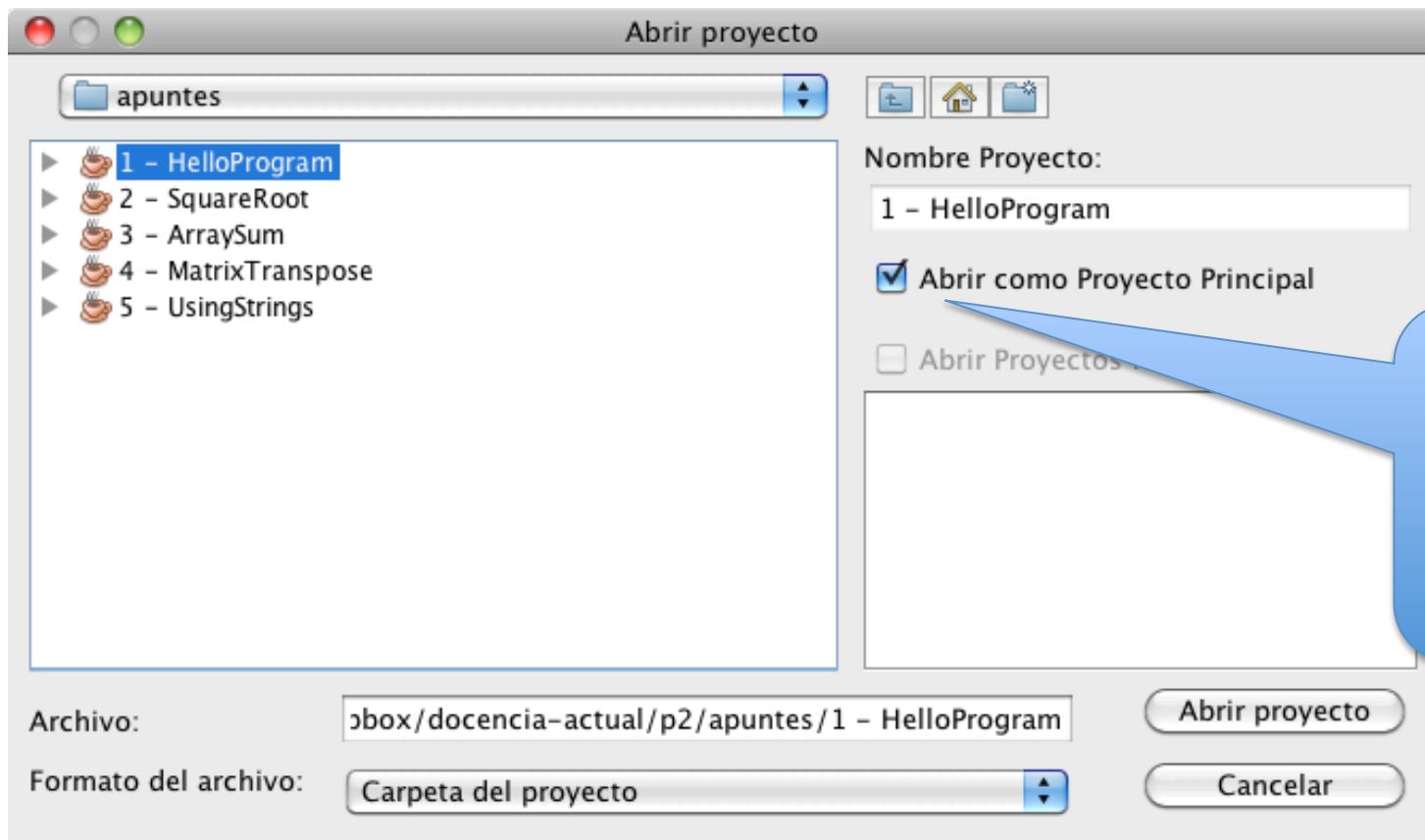
# Ejemplo: HelloProgram



# Carga de un proyecto



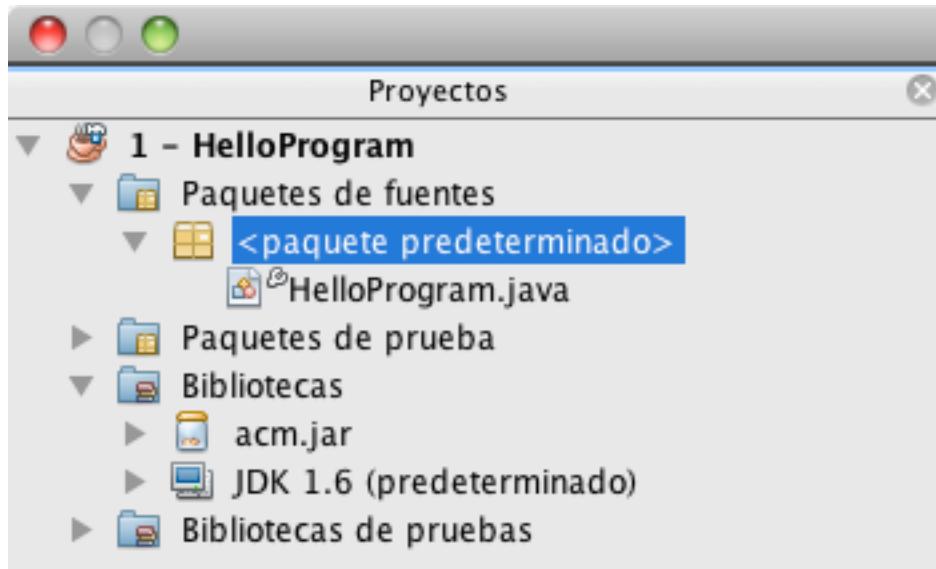
# Carga de un proyecto



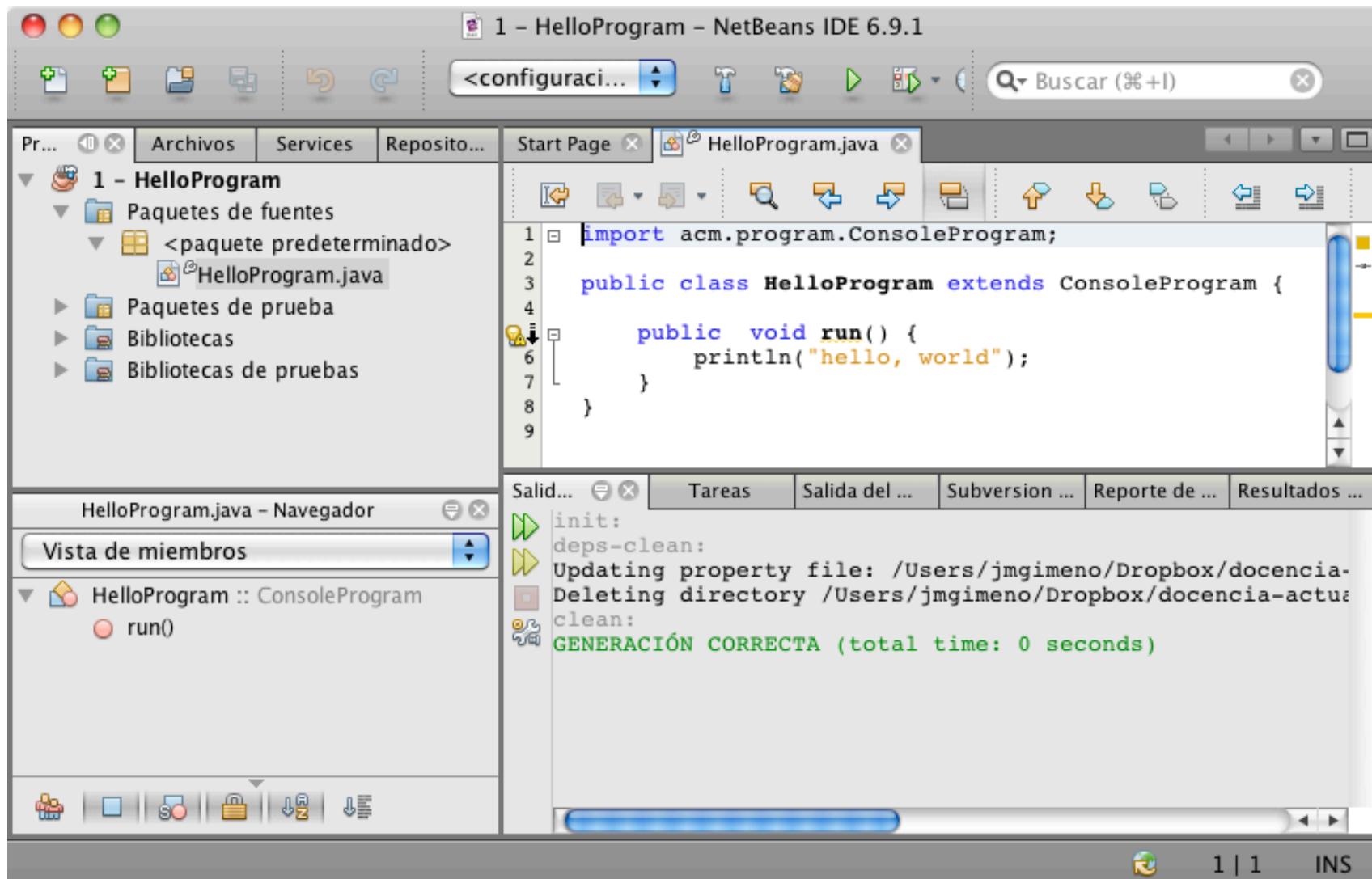
El proyecto principal es aquél en el que estás trabajando actualmente

# Vista de Proyectos

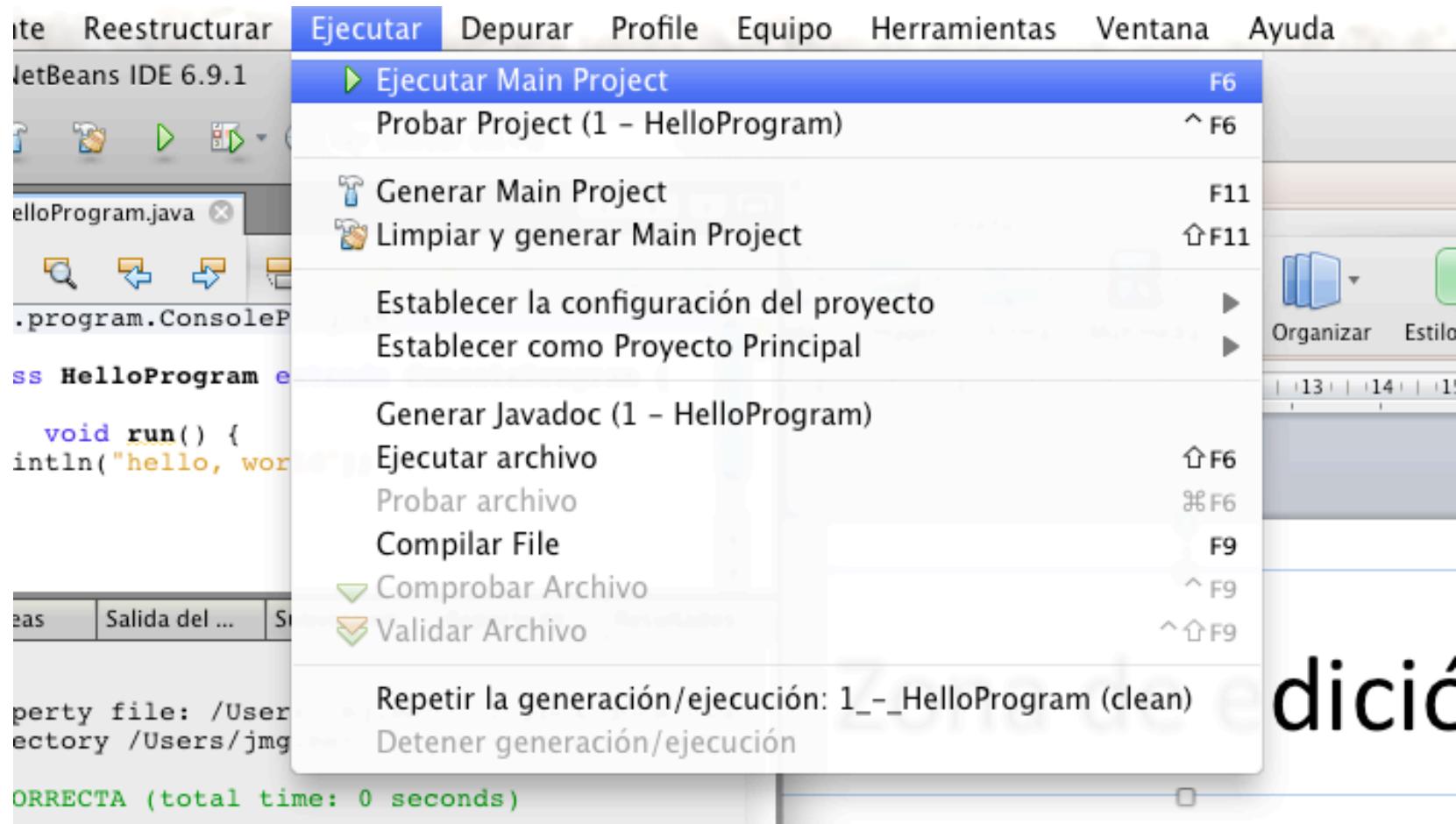
- Normalmente usamos la vista lógica de los **Proyectos**
- Si seleccionamos HelloProgram.java éste se abre en el editor



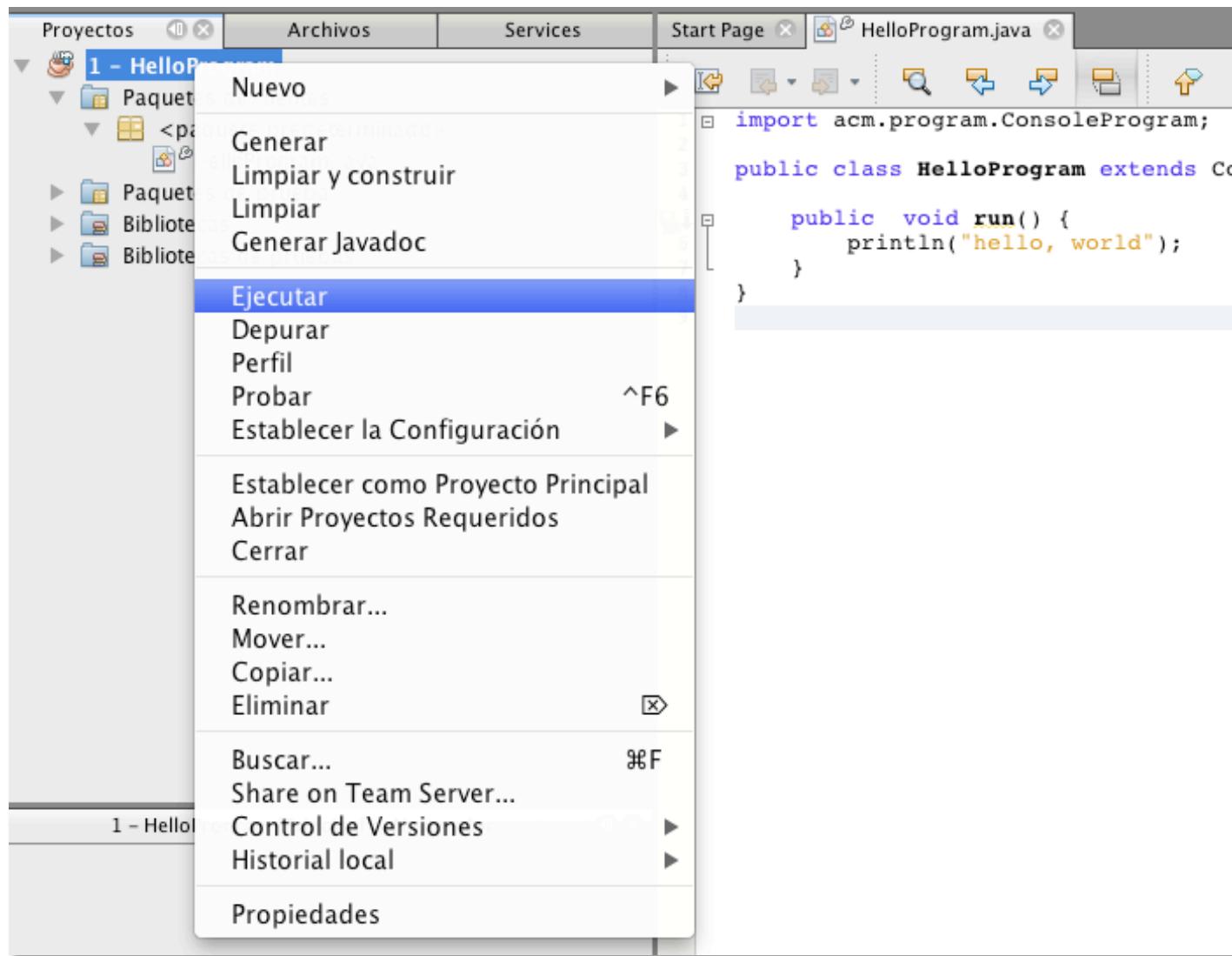
# Zona de edición



# Ejecutar el proyecto



# Ejecutar el proyecto



# **INSTALACIÓN DE LA BIBLIOTECA ACM**

# ACM Task Force

- Las bibliotecas predefinidas de la plataforma Java no están pensadas para su uso en cursos de programación básica
- Por ello se crearon las bibliotecas de la ACM Task Force, para simplificar algunos aspectos
- El problema es que los entornos de programación normalmente no las incluyen
- Por tanto hemos de configurarlas

# Primer paso

- Abrimos el Netbeans (debería estar instalado tanto en Windows como en Linux)
  - En el menú **Herramientas** seleccionamos **Bibliotecas**



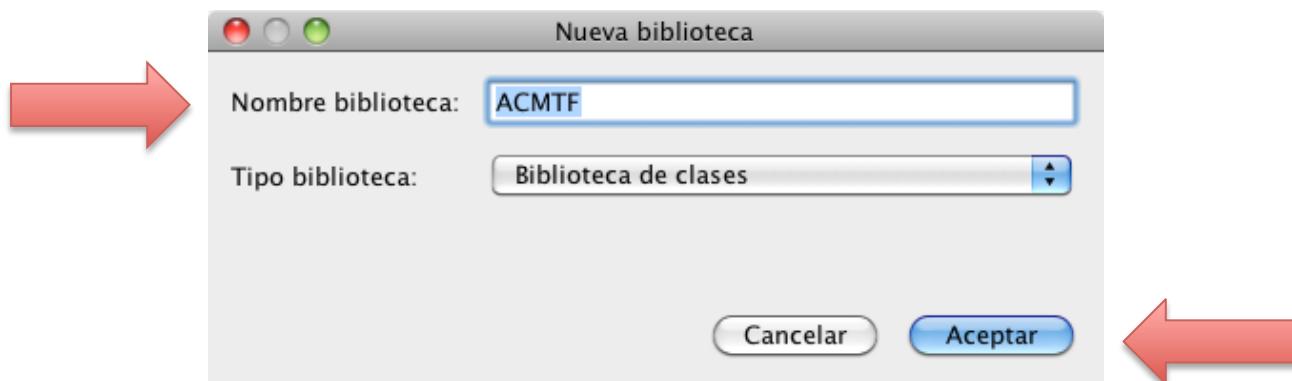
# Segundo paso

- Una vez abierto el **Administrador de Bibliotecas**, pulsamos en **Nueva Biblioteca**



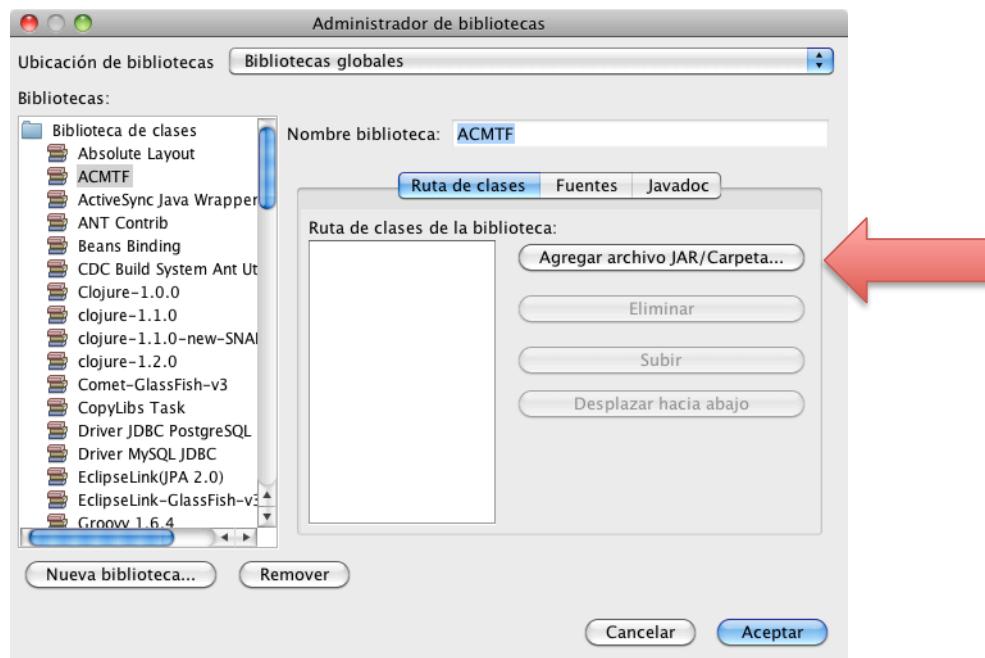
# Tercer paso

- En la pantalla **Nueva Biblioteca** le ponemos el nombre, **ACMTF**
- Pulsáis **Aceptar**



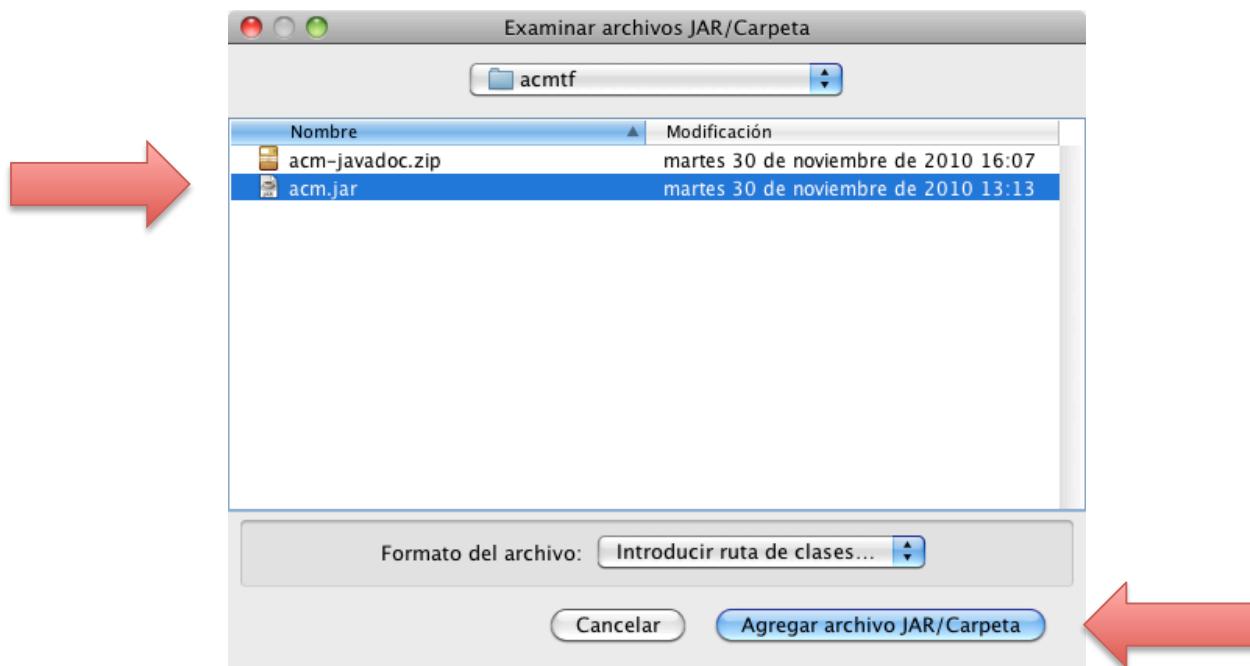
# Cuarto paso

- Seleccionamos **Agregar archivo JAR/Carpeta** desde el **Administrador de bibliotecas**



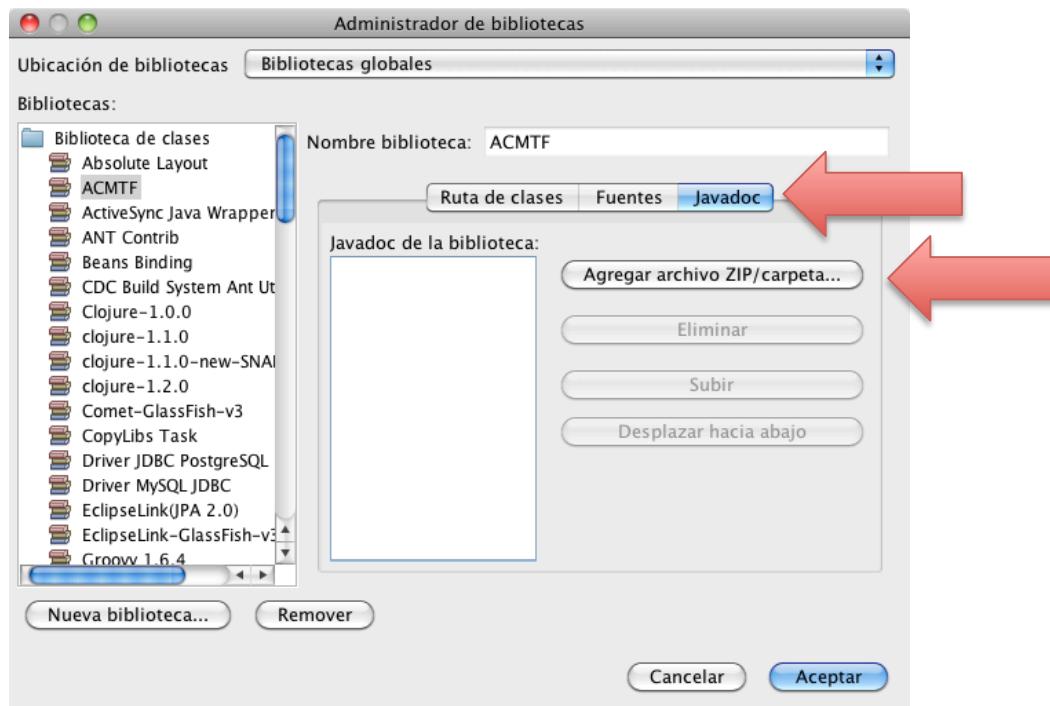
# Quinto paso

- Seleccionamos el archivo **acm.jar** mediante el selector de ficheros (OJO: **no** se copiará en los directorios de Netbeans)



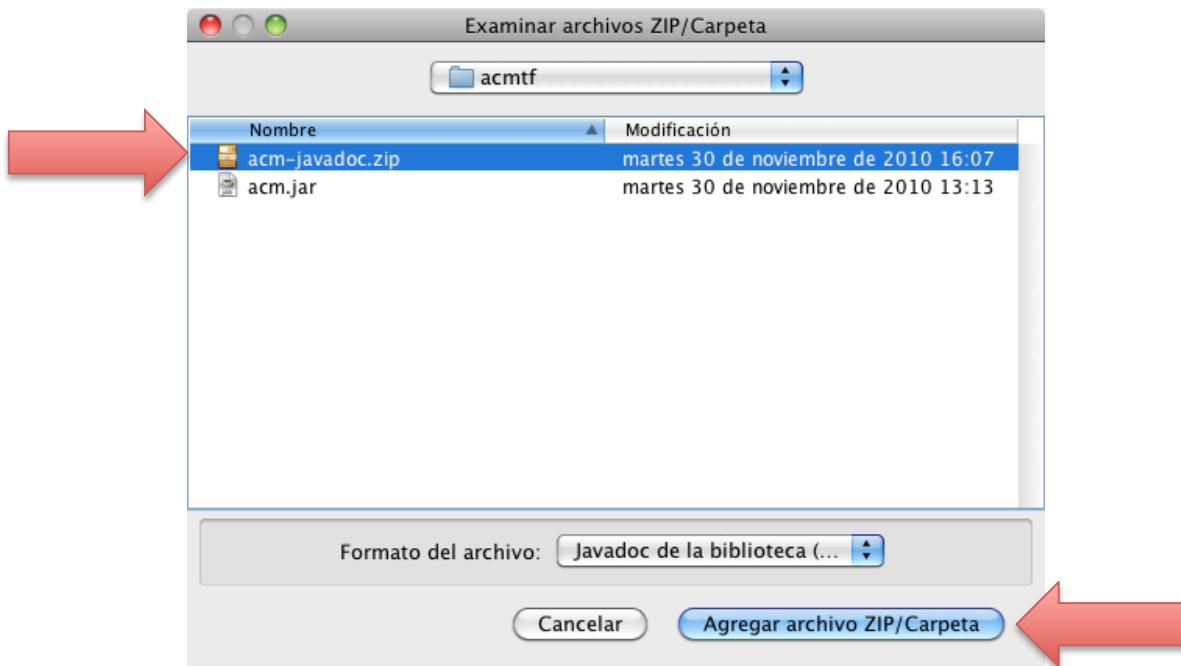
# Sexto paso

- Ahora hemos de añadir la documentación de la biblioteca, por ello seleccionamos **Javadoc** y pulsamos **Agregar archivo ZIP/Carpeta**



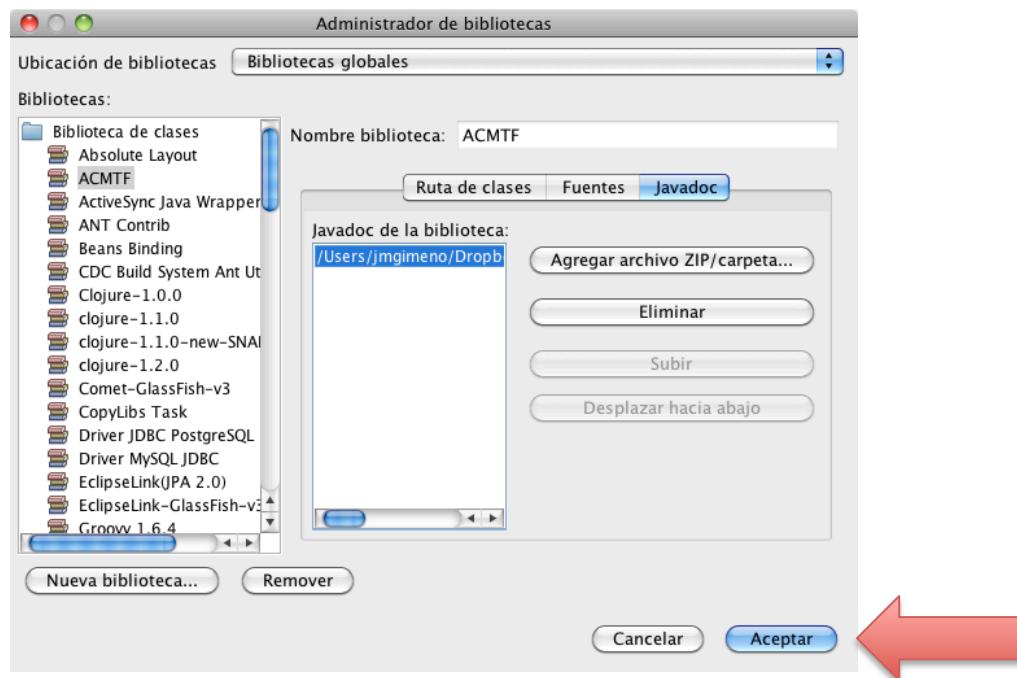
# Séptimo paso

- Seleccionamos el archivo **acm-javadoc.zip** en el selector de ficheros y pulsamos **Agregar archivo ZIP/Carpeta**

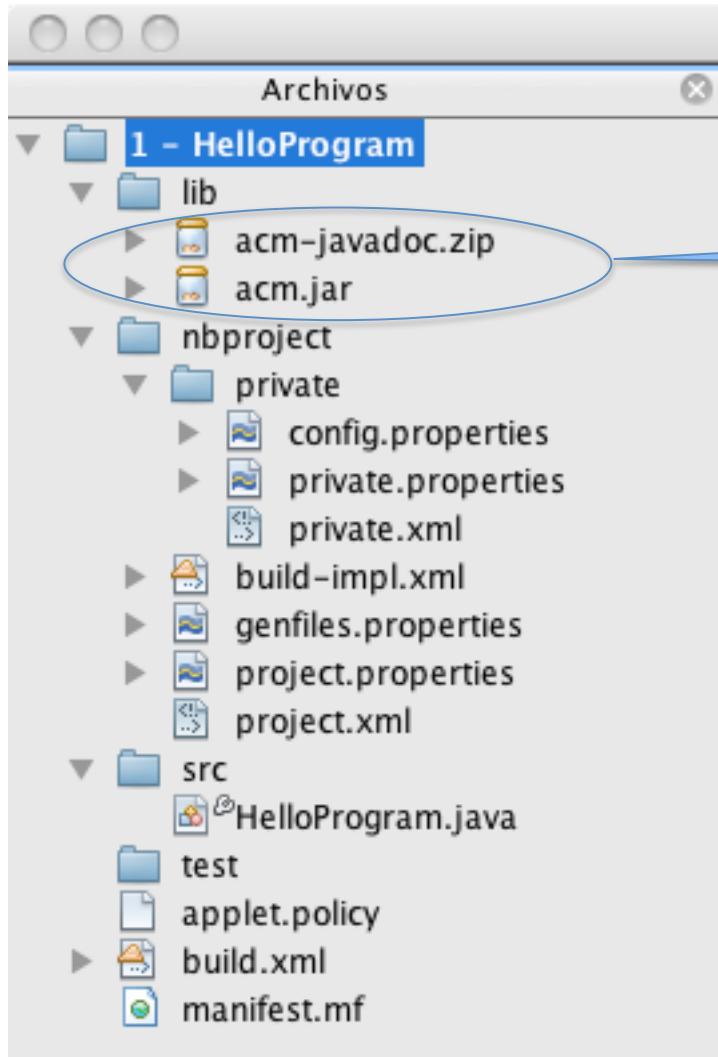


# Paso final

- Por fin pulsamos aceptar para agregar la nueva biblioteca y su documentación en el entorno Netbeans.



# Configuración antes de empezar



Bibliotecas ACMTF

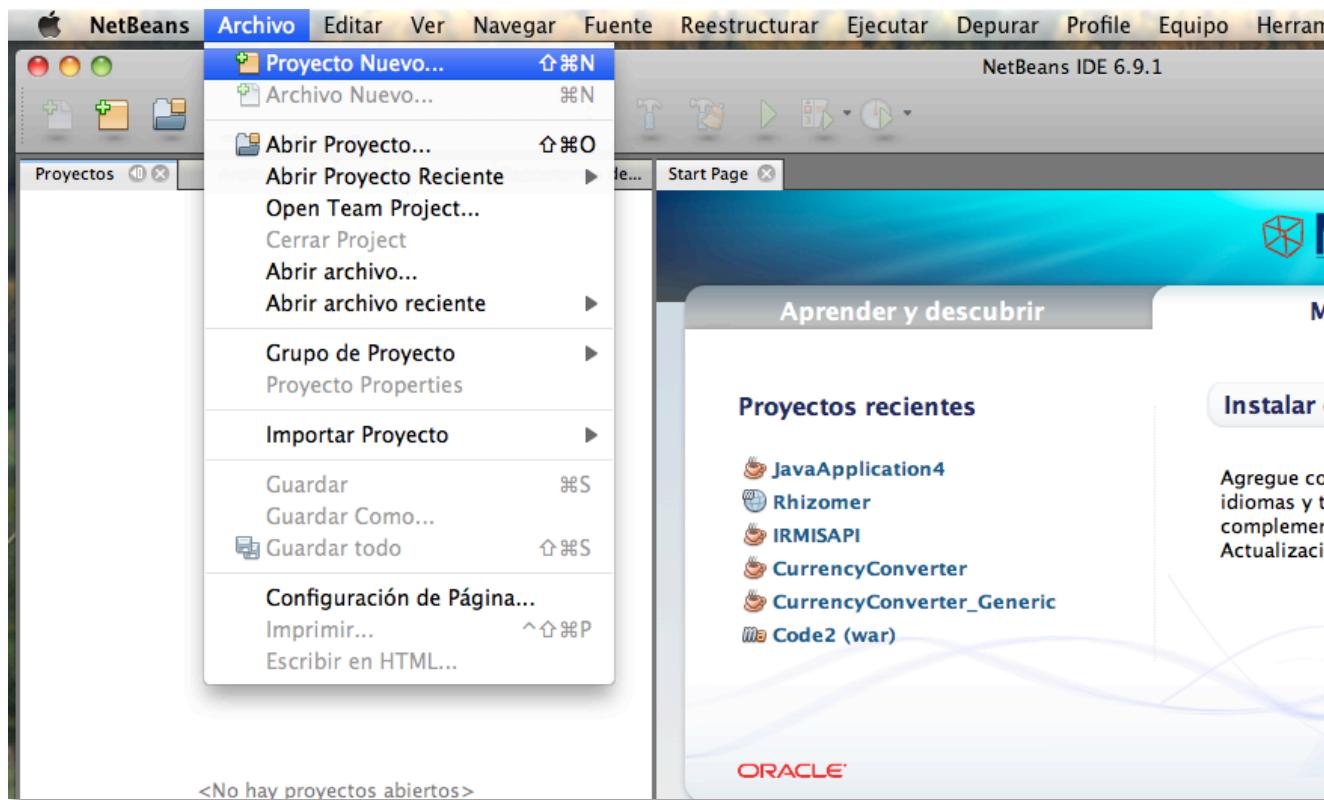
# **CREACIÓN DE UN NUEVO PROYECTO**

# Consideraciones previas

- En NB existen muchas maneras de conseguir lo mismo
- Lo que mostraremos es una de ellas, que he seleccionado por varias razones:
  - Los proyectos resultantes **no copian** los archivos de la biblioteca de ACM, ya que ocupan bastante espacio
  - Aprendemos funcionalidades de NB que después podemos usar para otros usos

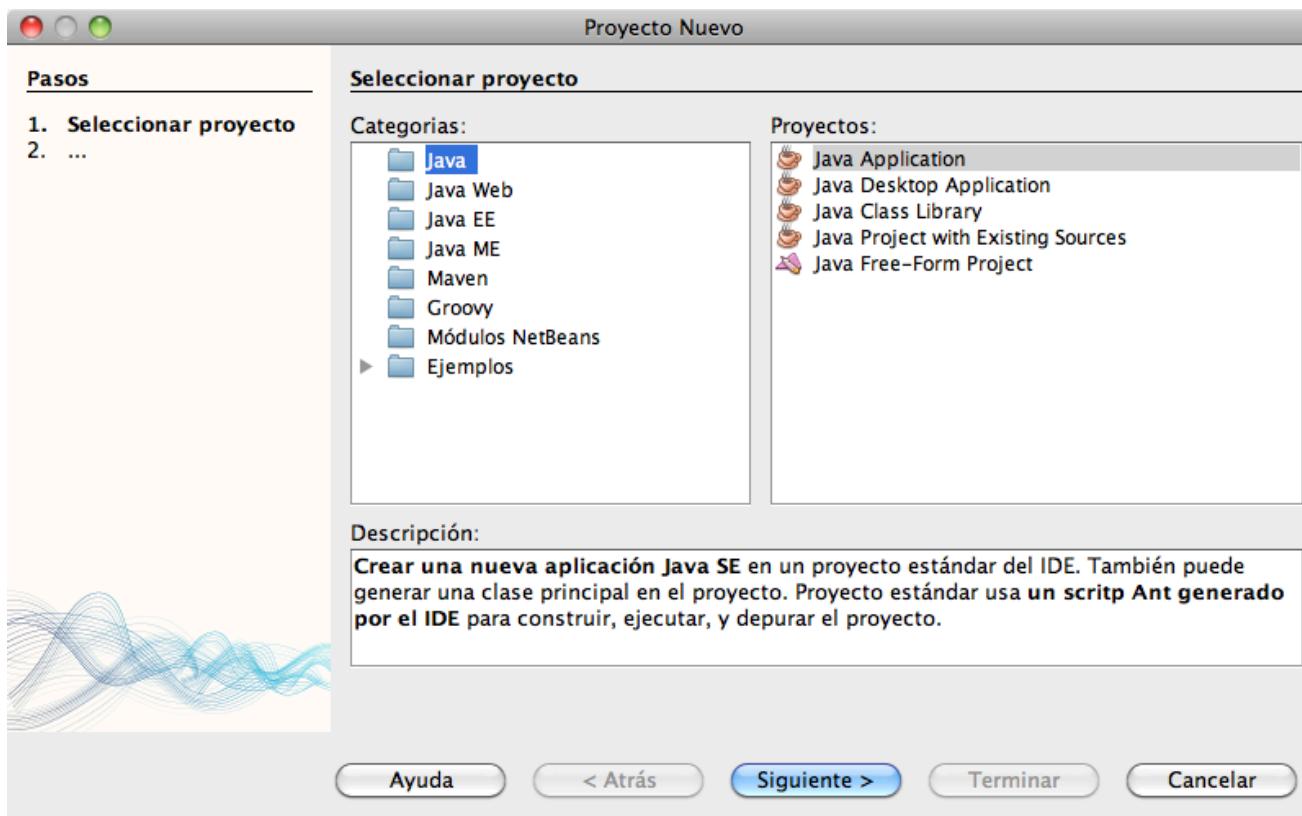
# Proyecto Nuevo

- Seleccionamos **Proyecto Nuevo** del menú **Archivo**

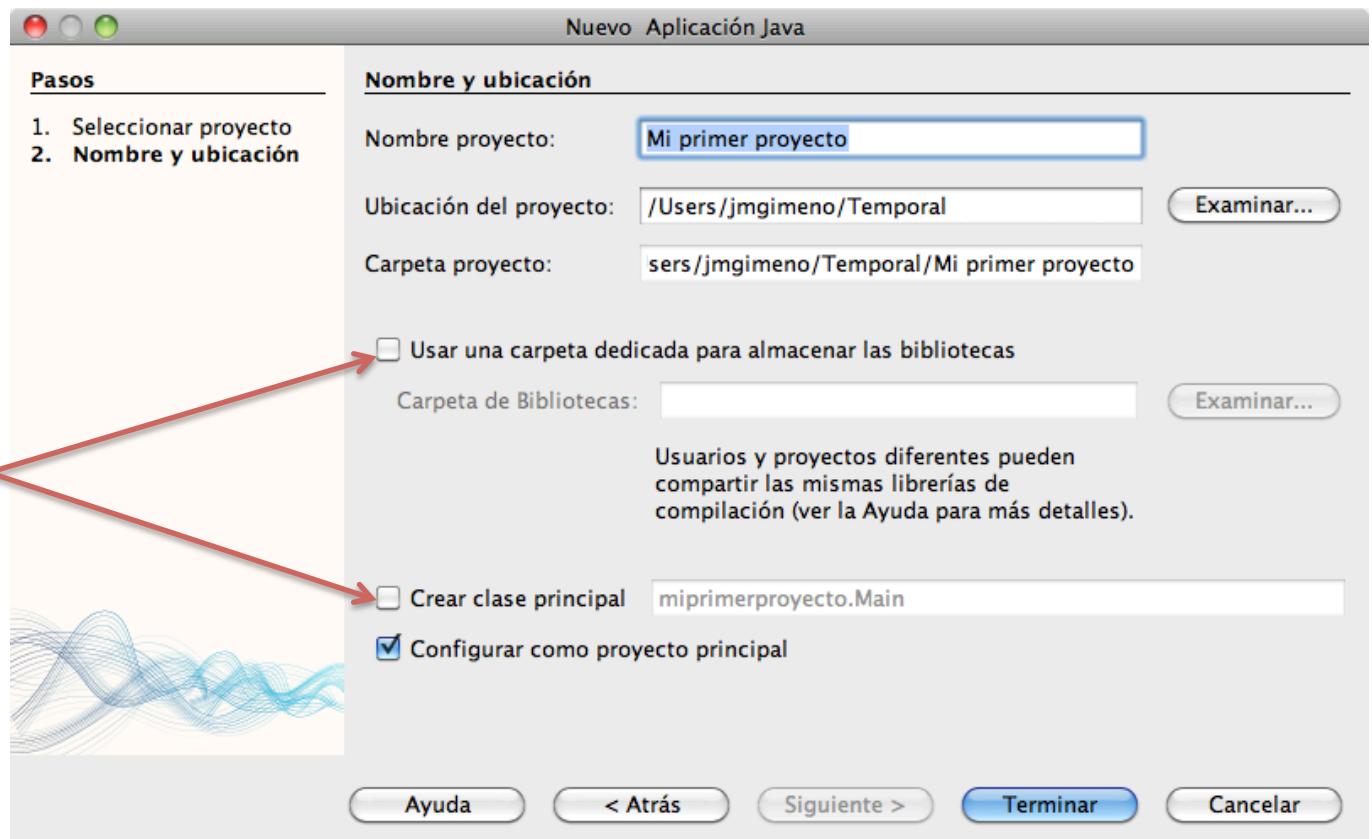


# Proyecto Nuevo

- Si no lo estuvieran, seleccionamos la Categoría Java y el Proyecto Java Application



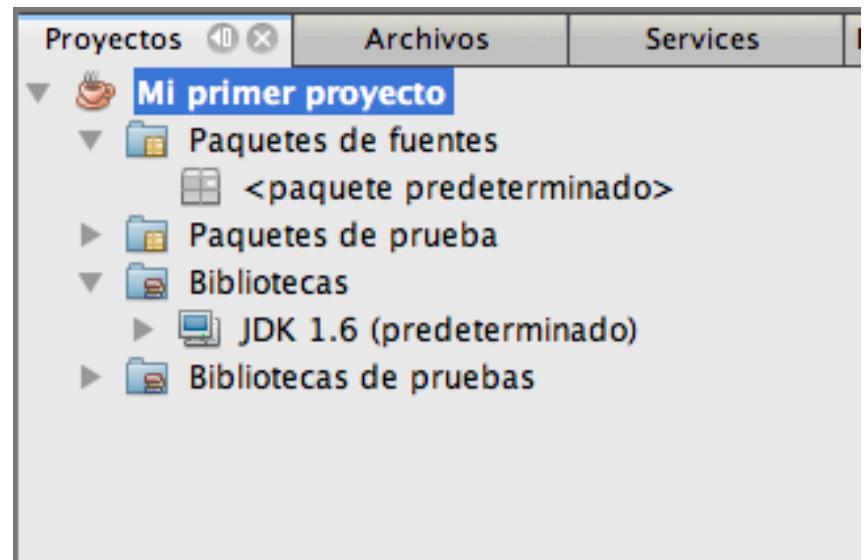
# Proyecto Nuevo



Mejor dejarlos  
desmarcados

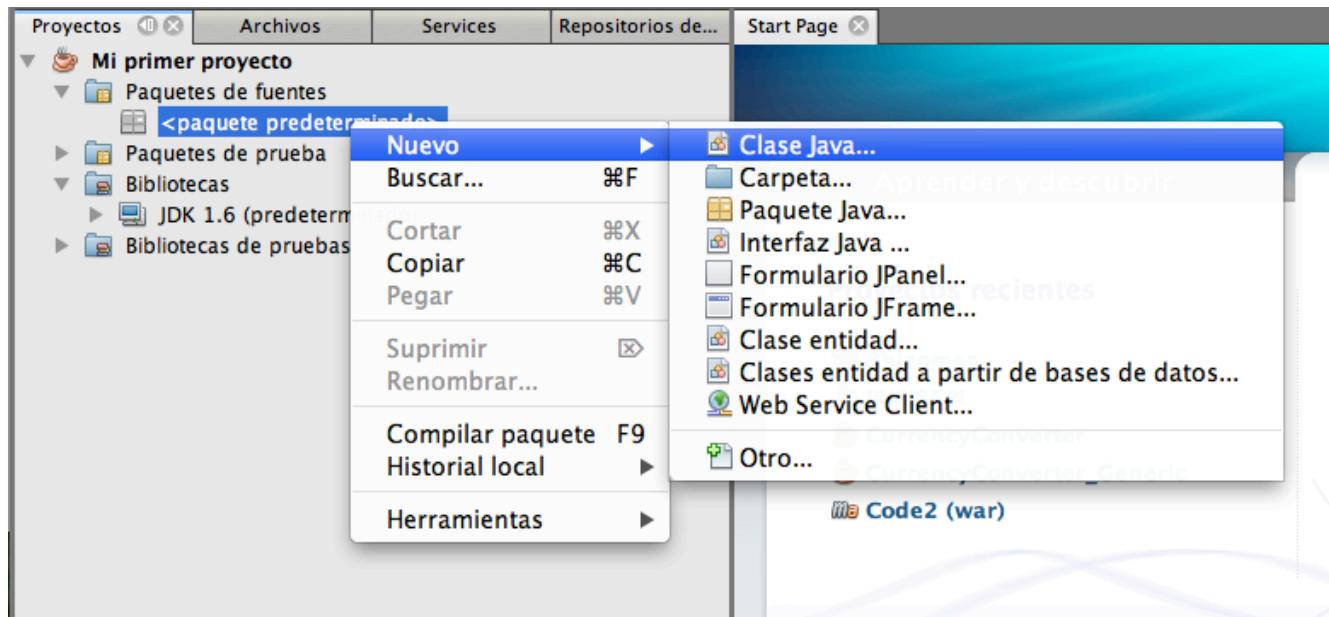
# Estructura del proyecto

- Después de pulsar **Terminar** se crea un nuevo proyecto en el área de proyecto
- Este proyecto no incluye aún
  - Ninguna clase
  - Biblioteca ACM
- Empecemos creando una clase

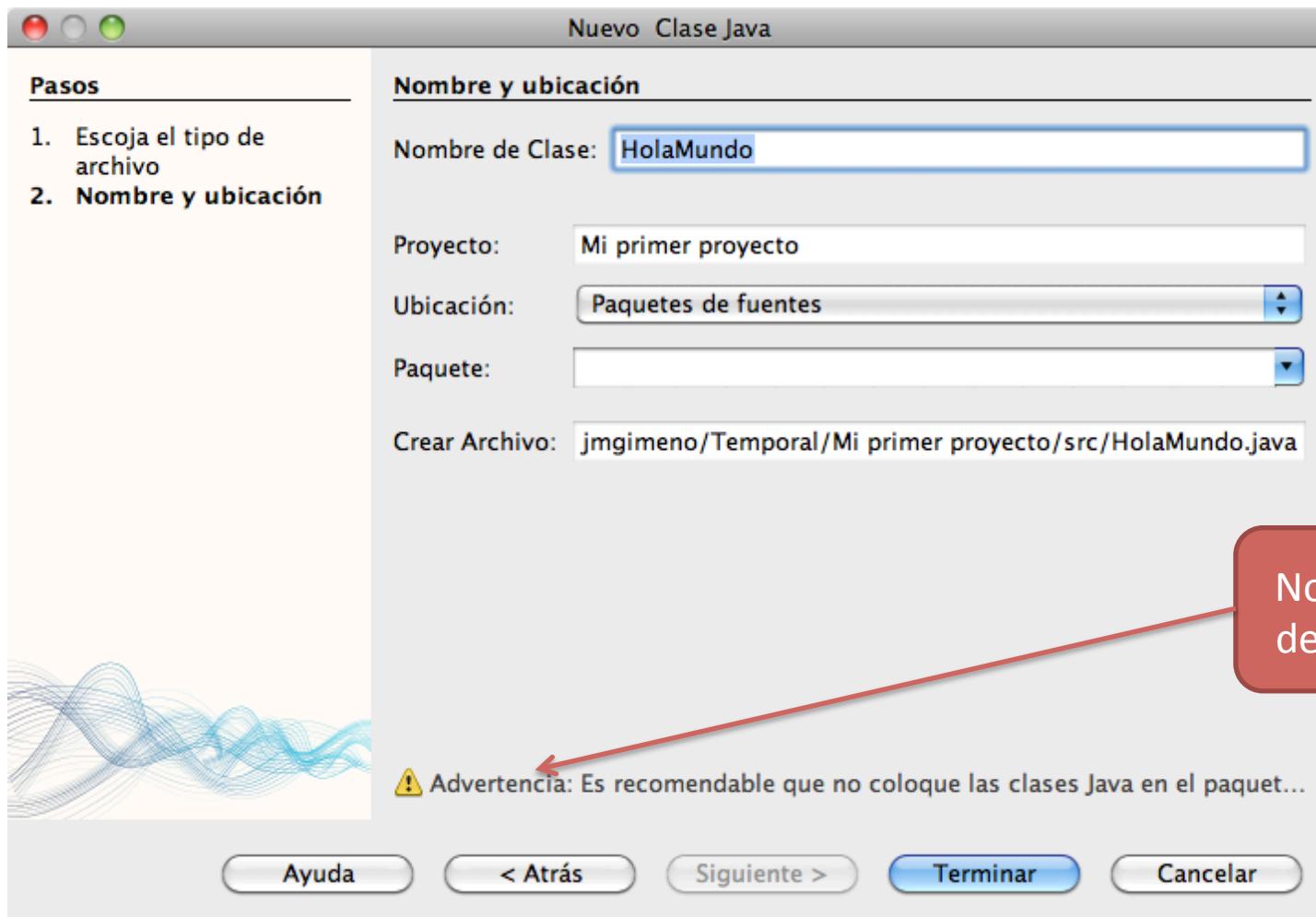


# Añadiendo una clase

- Abrimos el menú contextual sobre el <paquete predeterminado>
- Seleccionamos Nuevo y Clase Java

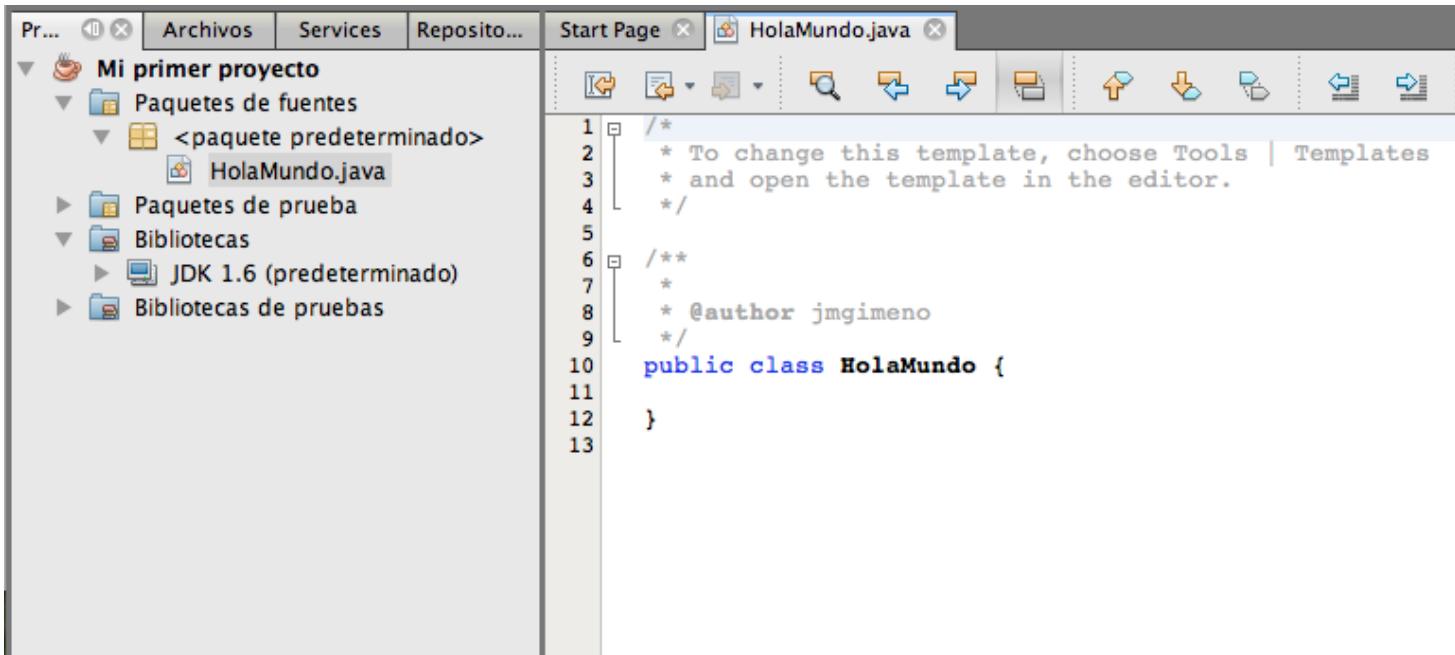


# Añadiendo una clase



# Estructura de la clase creada

- Vemos que la clase se ha añadido al proyecto y se ha abierto en el editor
- Ahora ya podemos editarla y modificarla

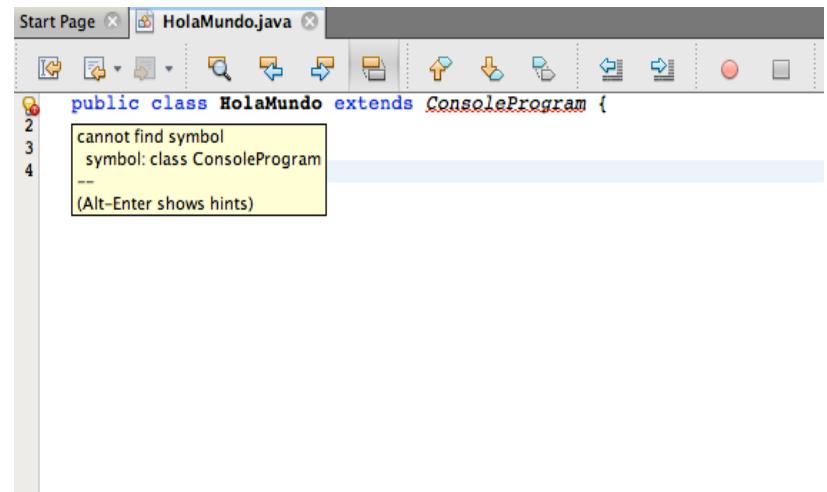


The screenshot shows the interface of an Integrated Development Environment (IDE). On the left, there is a project tree titled "Mi primer proyecto". It contains a "Paquetes de fuentes" folder with a "paquete predeterminado" folder containing a file named "HolaMundo.java". Below this are "Paquetes de prueba", "Bibliotecas" (with "JDK 1.6 (predeterminado)" selected), and "Bibliotecas de pruebas". On the right, the main window displays the code editor for "HolaMundo.java". The code is as follows:

```
1  /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5
6 /**
7 *
8 * @author jmgimeno
9 */
10 public class HolaMundo {
11
12 }
13
```

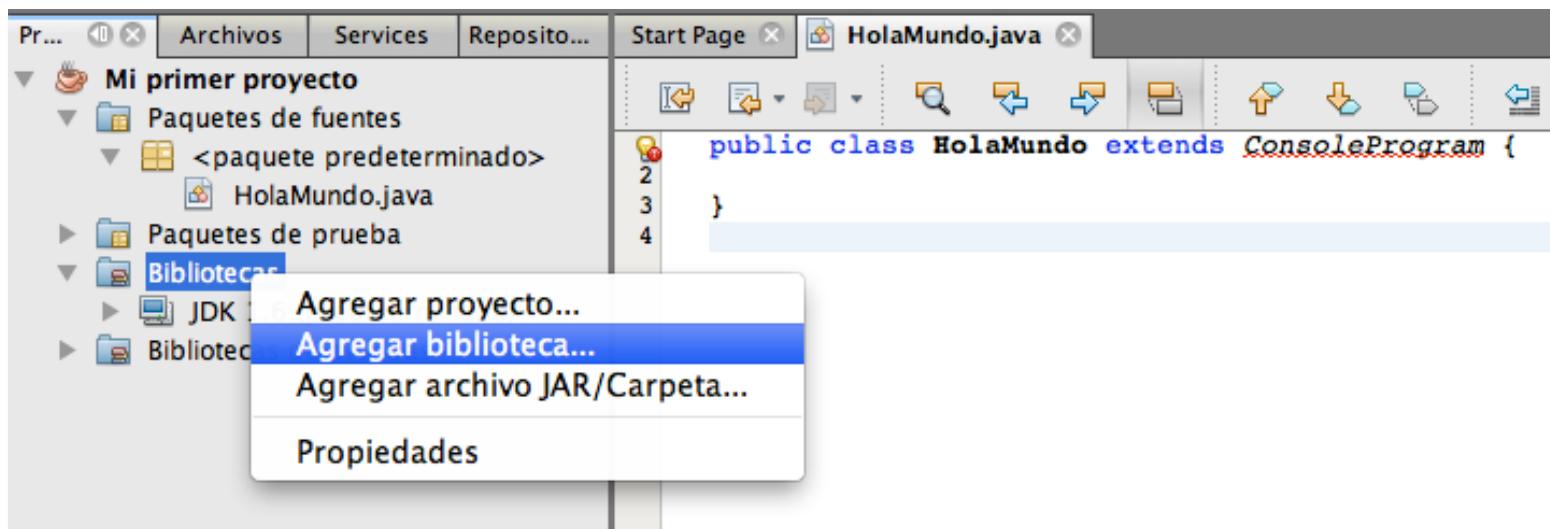
# Editando la clase

- Añadimos que la clase extiende ConsoleProgram, pero ...
- ... el entorno nos indica que no conoce la clase ConsoleProgram
- Es normal, ya que no hemos incluido la biblioteca ACMTF



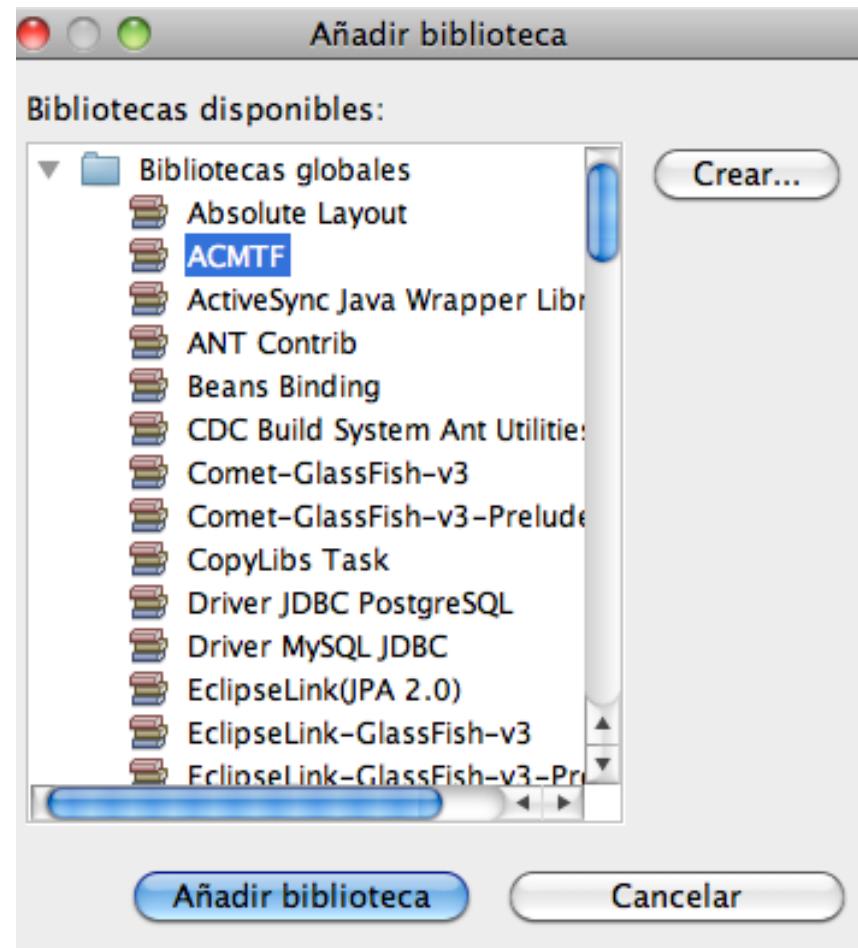
# Añadimos la biblioteca ACMTF

- Sobre el apartado **Bibliotecas** del proyecto seleccionamos **Agregar biblioteca**



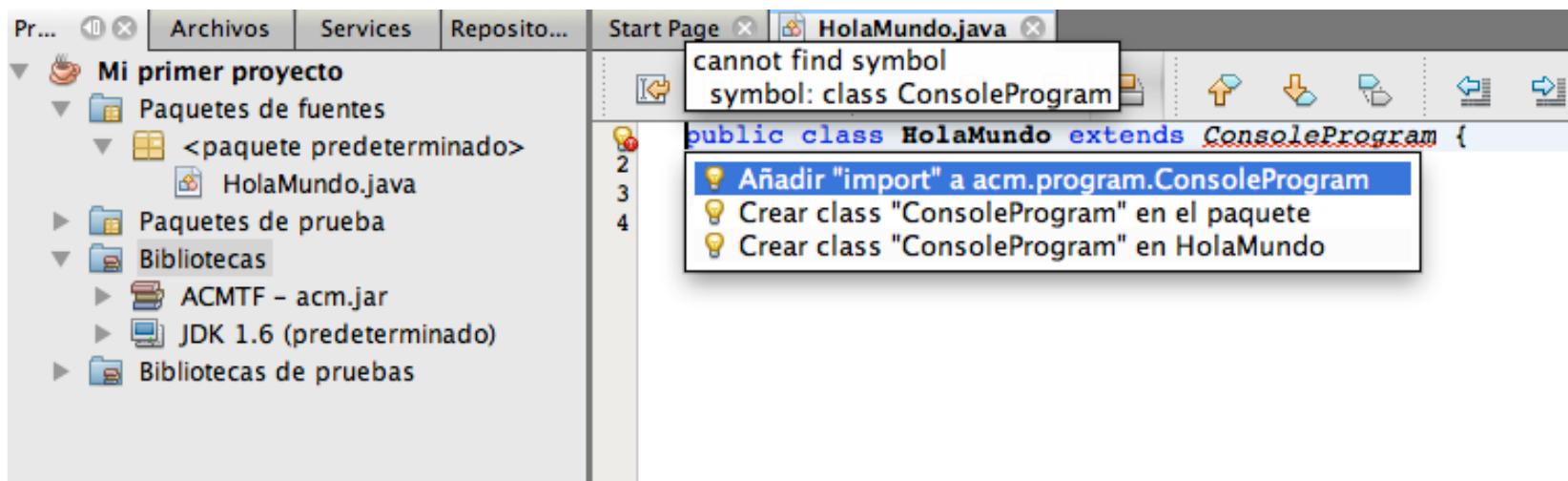
# Añadimos la biblioteca ACMTF

- Seleccionamos la biblioteca ACMTF que hemos definido anteriormente



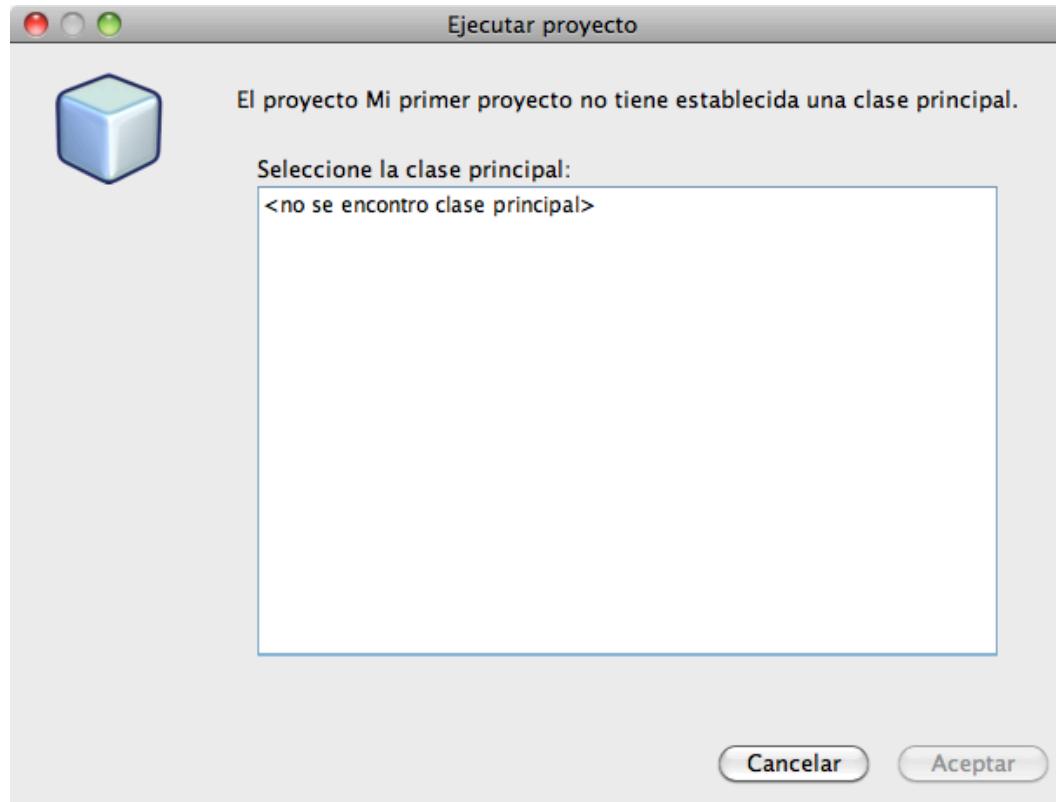
# Ahora ya podemos importarla

- Podemos añadir el import directamente o usando las sugerencias de corrección del Netbeans



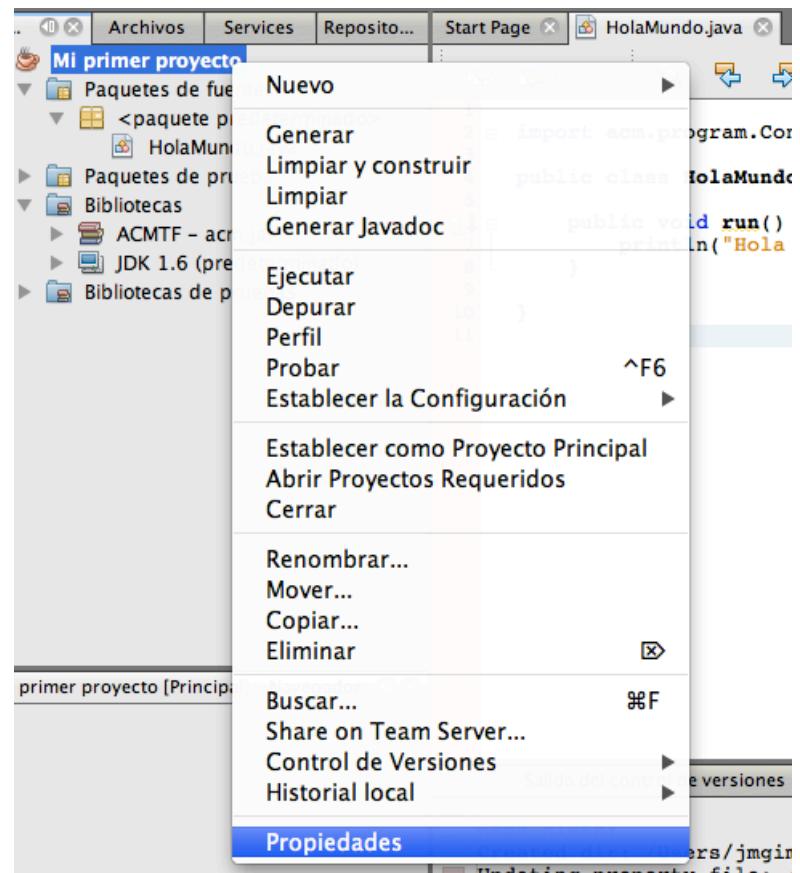
# Ejecutemos el proyecto

- Si ejecutamos el proyecto nos encontramos con una sorpresa: **no existe una clase principal**



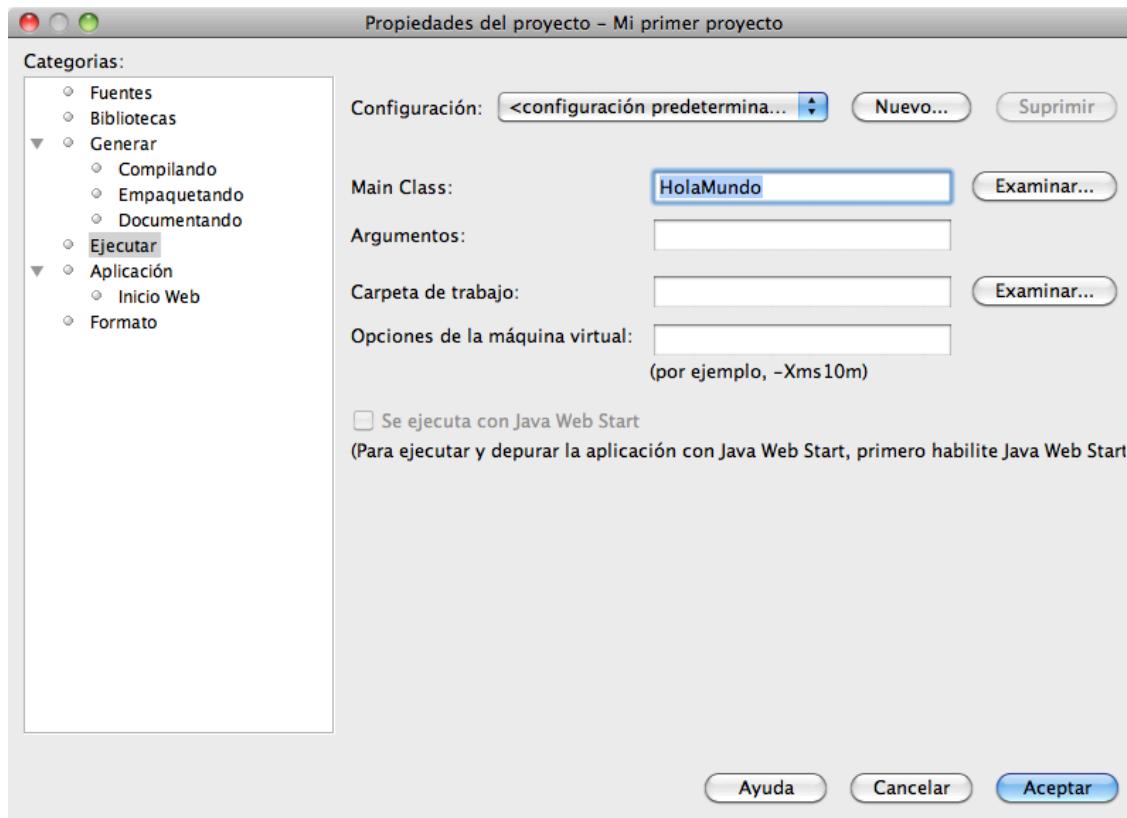
# Accediendo a las propiedades de un proyecto

- Abrimos el menú contextual asociado al proyecto y seleccionamos el elemento **Propiedades**



# Modificamos las propiedades

- Seleccionamos la categoría **Ejecutar** e indicamos el nombre de la clase principal.



# Volvemos a ejecutar ...

iii Y por fin todo  
funciona tal y  
como  
esperábamos !!!

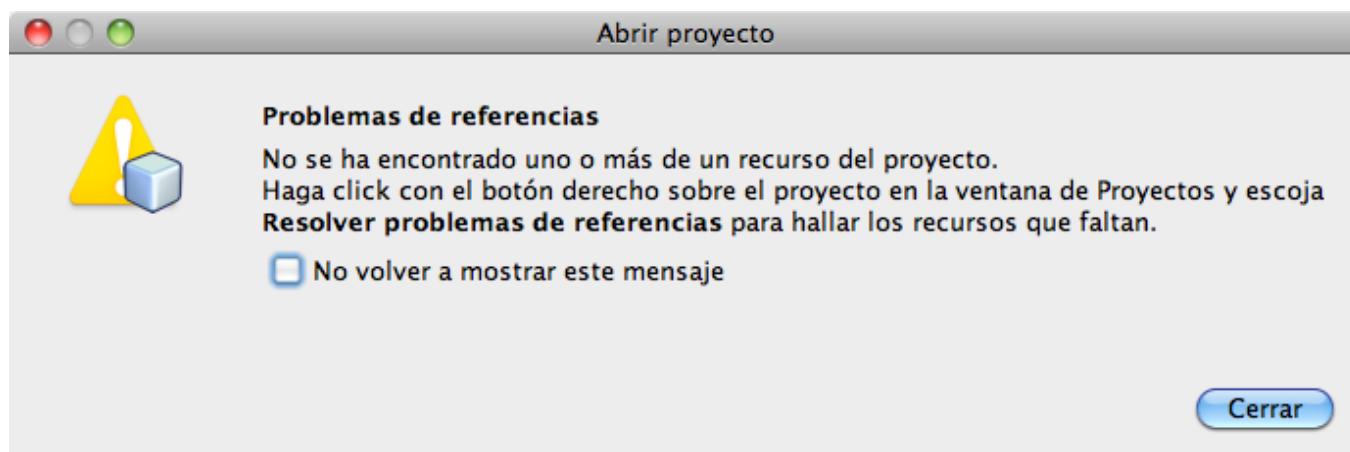


# **SOLUCIONES A PROBLEMAS COMUNES**

Abrimos un proyecto pero la biblioteca no está configurada

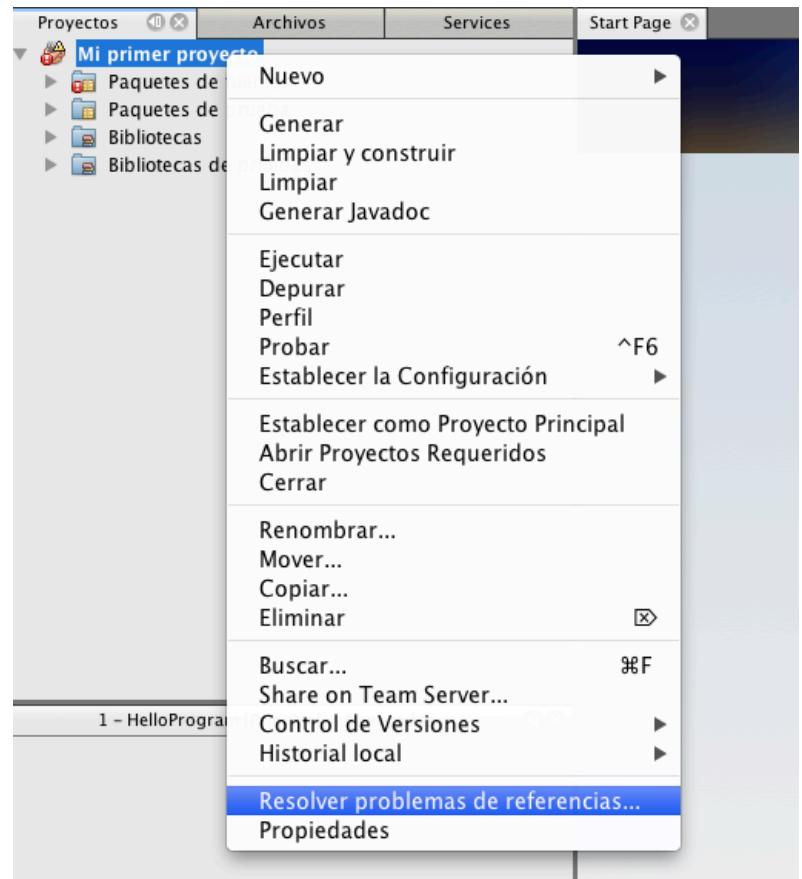
# Biblioteca no definida

- Si la biblioteca no está definida (o lo está con un nombre diferente)



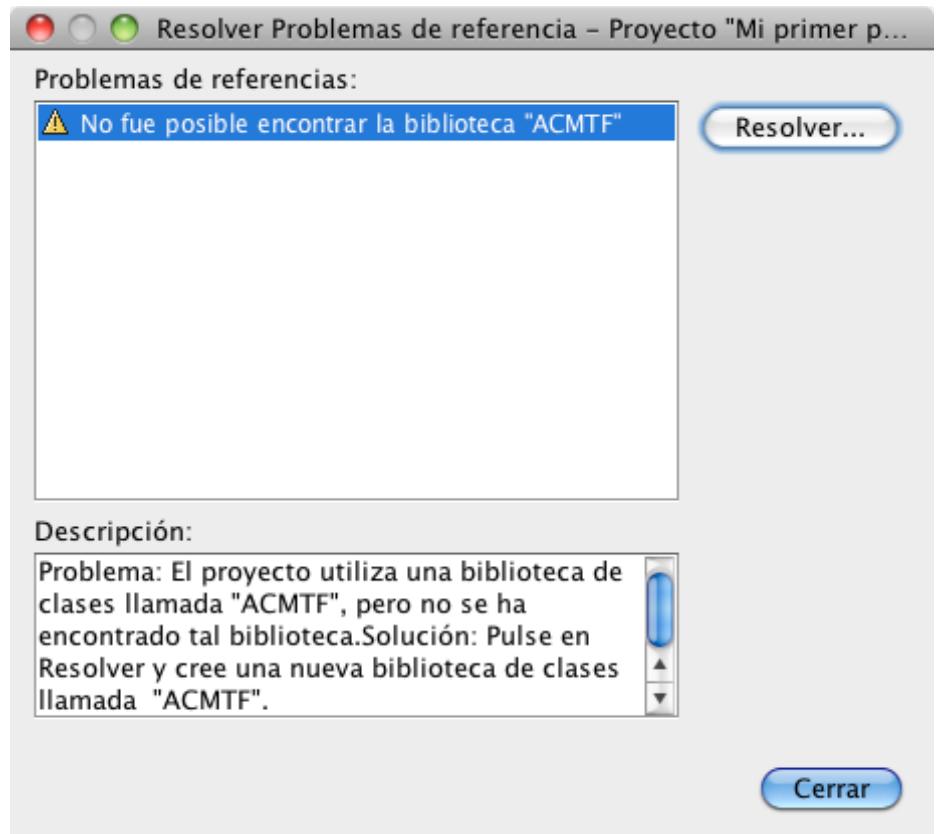
# Seguimos las indicaciones

- Abrimos el menú contextual del proyecto y seleccionamos **Resolver problemas de referencias...**

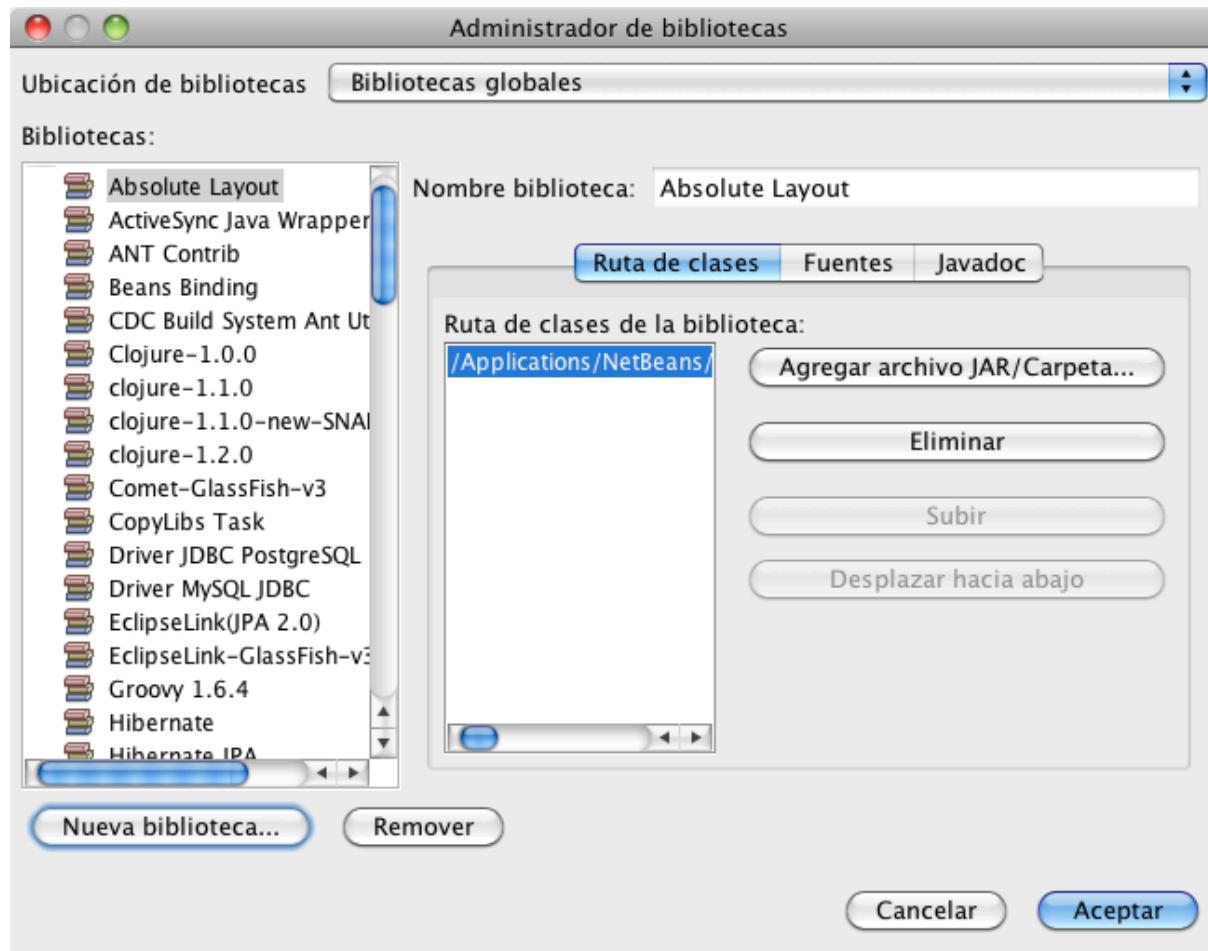


# Se nos indican los detalles

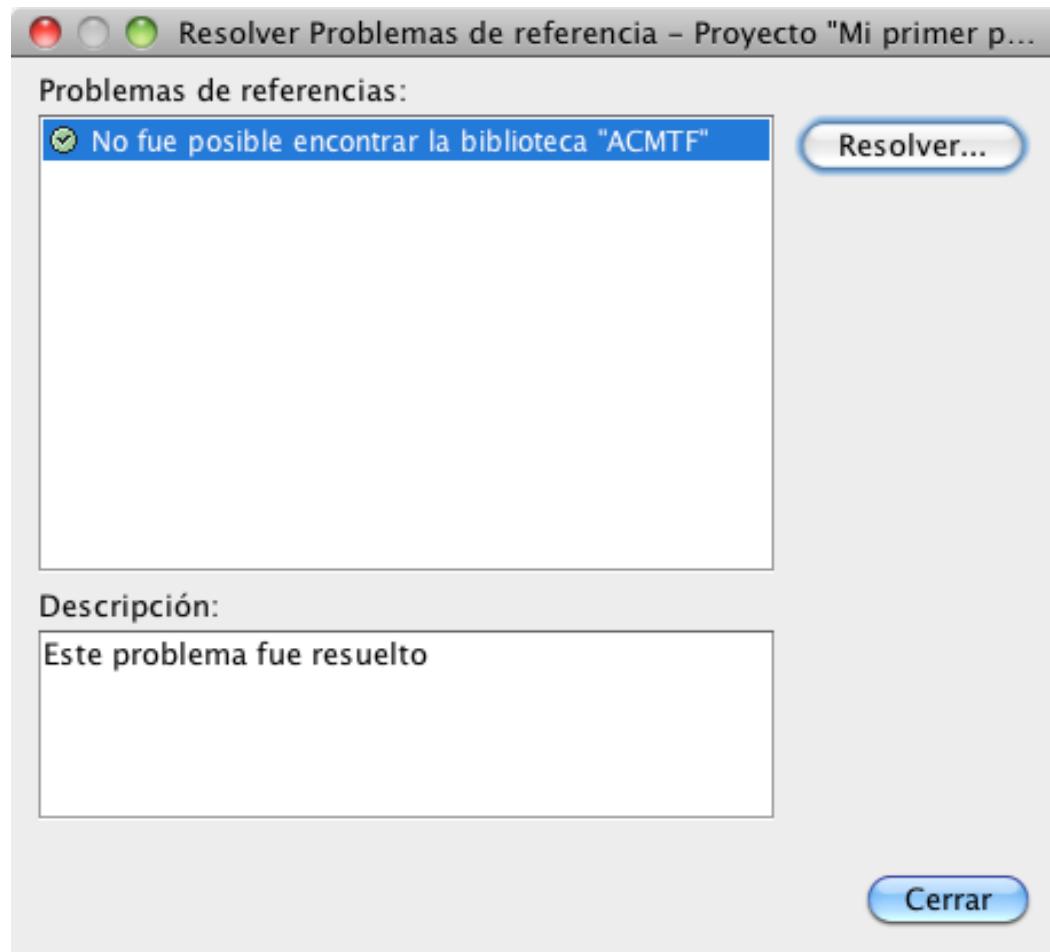
- Siempre hay que leer la información sobre el error
- Pulsamos **Resolver**



# Damos de alta la biblioteca



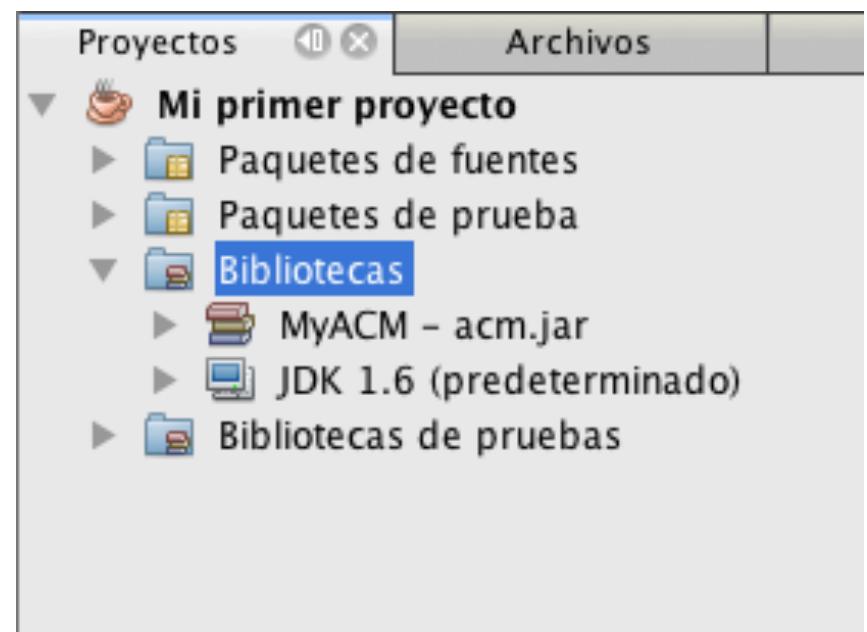
# Y problema resuelto



En el proyecto la biblioteca tiene otro nombre

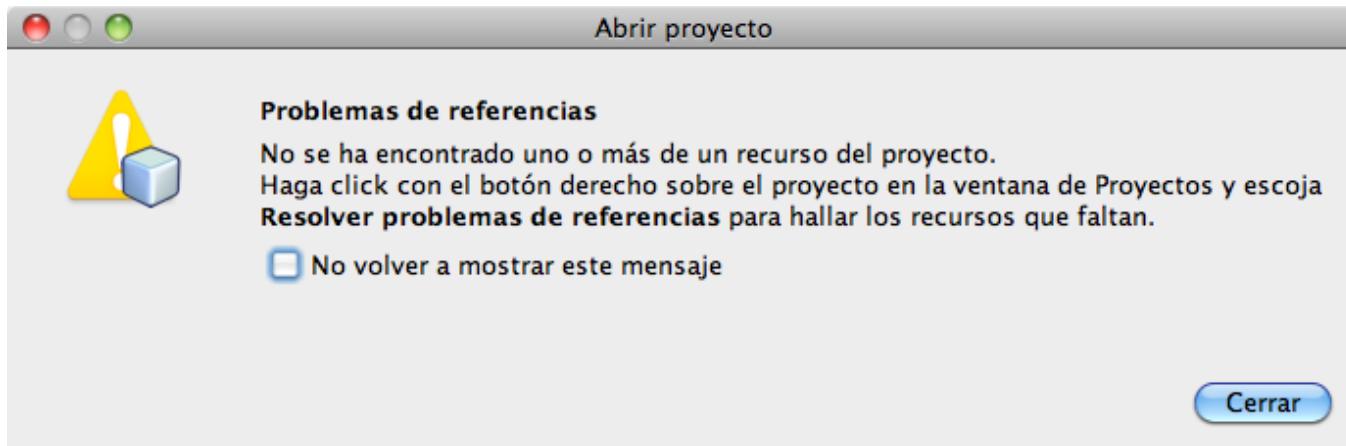
# La biblioteca tiene otro nombre

- Es posible que quien hizo el proyecto usó un nombre diferente para referirse a la biblioteca (p.e. **MyACM**)
- Lo que tendremos que hacer es lo siguiente:
  - Eliminar la referencia a la biblioteca con el nombre de quién hizo el proyecto
  - Añadir la biblioteca con el nombre que le hemos dado nosotros



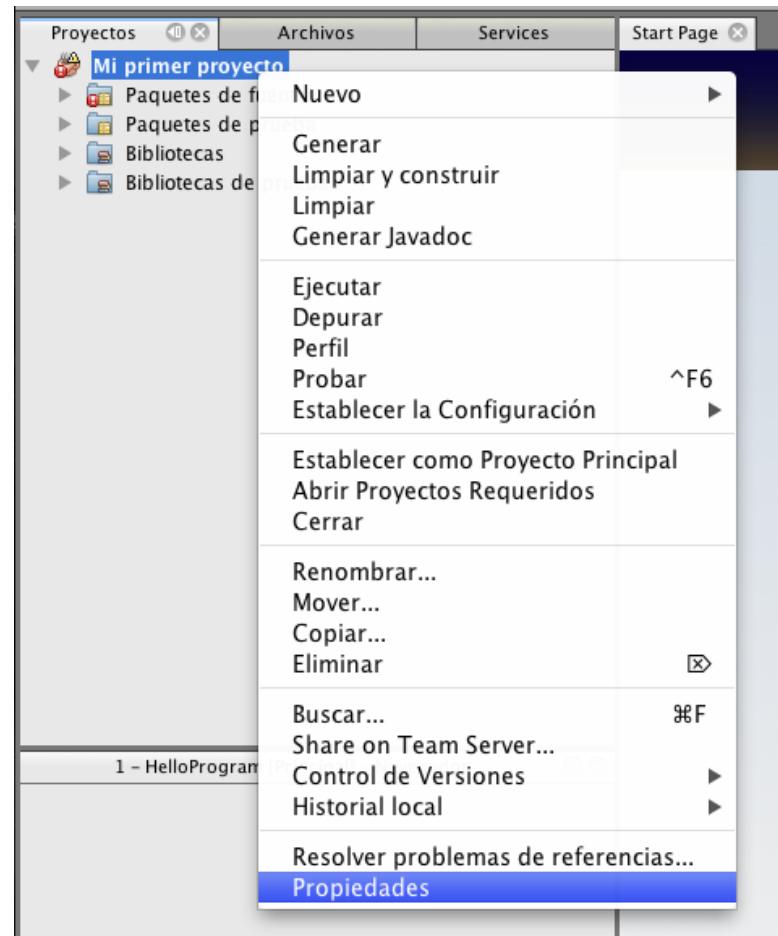
# Error al cargar el proyecto

- Como ya hemos visto si intentamos cargar el proyecto, dará **Problema de referencias**
- No tiene ahora sentido definir la biblioteca **MyACM** ya que tenemos la **ACMTF**



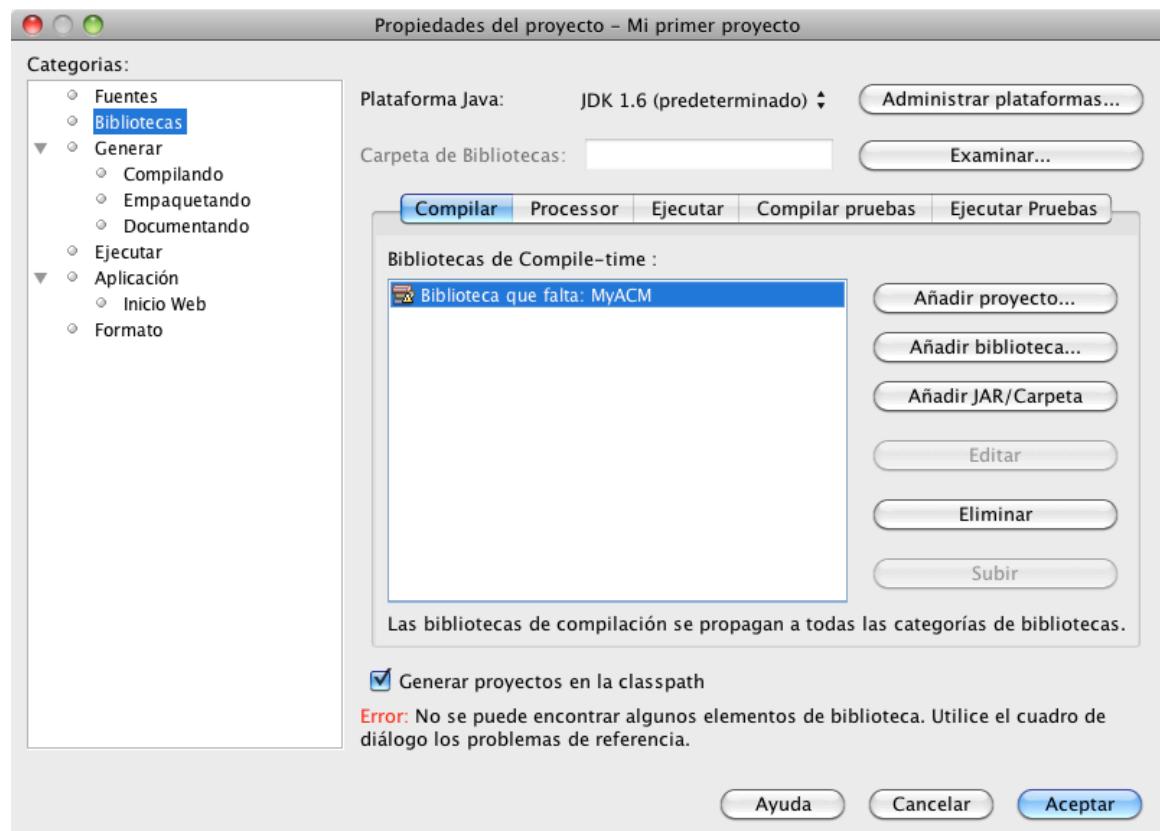
# Modificación de las propiedades

- Abrimos el menú contextual del proyecto y seleccionamos **Propiedades**



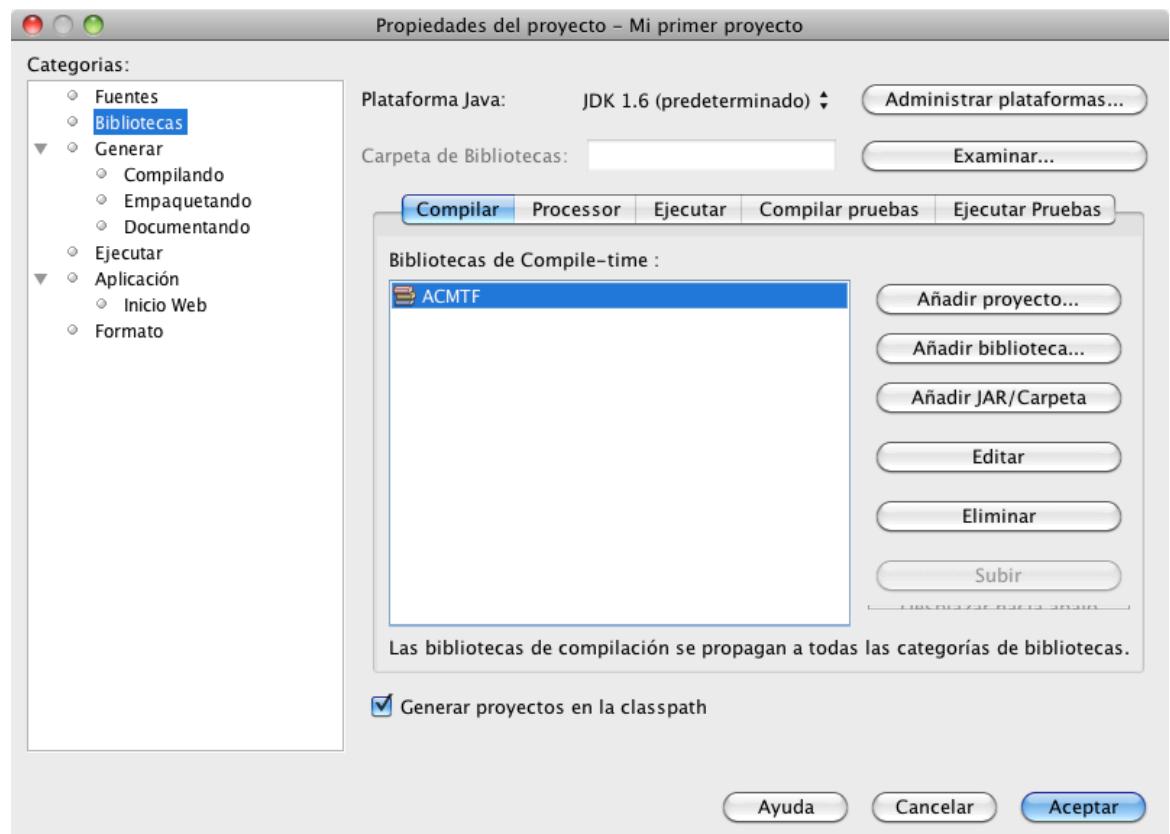
# Eliminamos la dependencia a MyACM

- Seleccionamos **Bibliotecas**
- Seleccionamos **MyACM**
- Pulsamos **Eliminar**



# Añadimos la dependencia a ACMTF

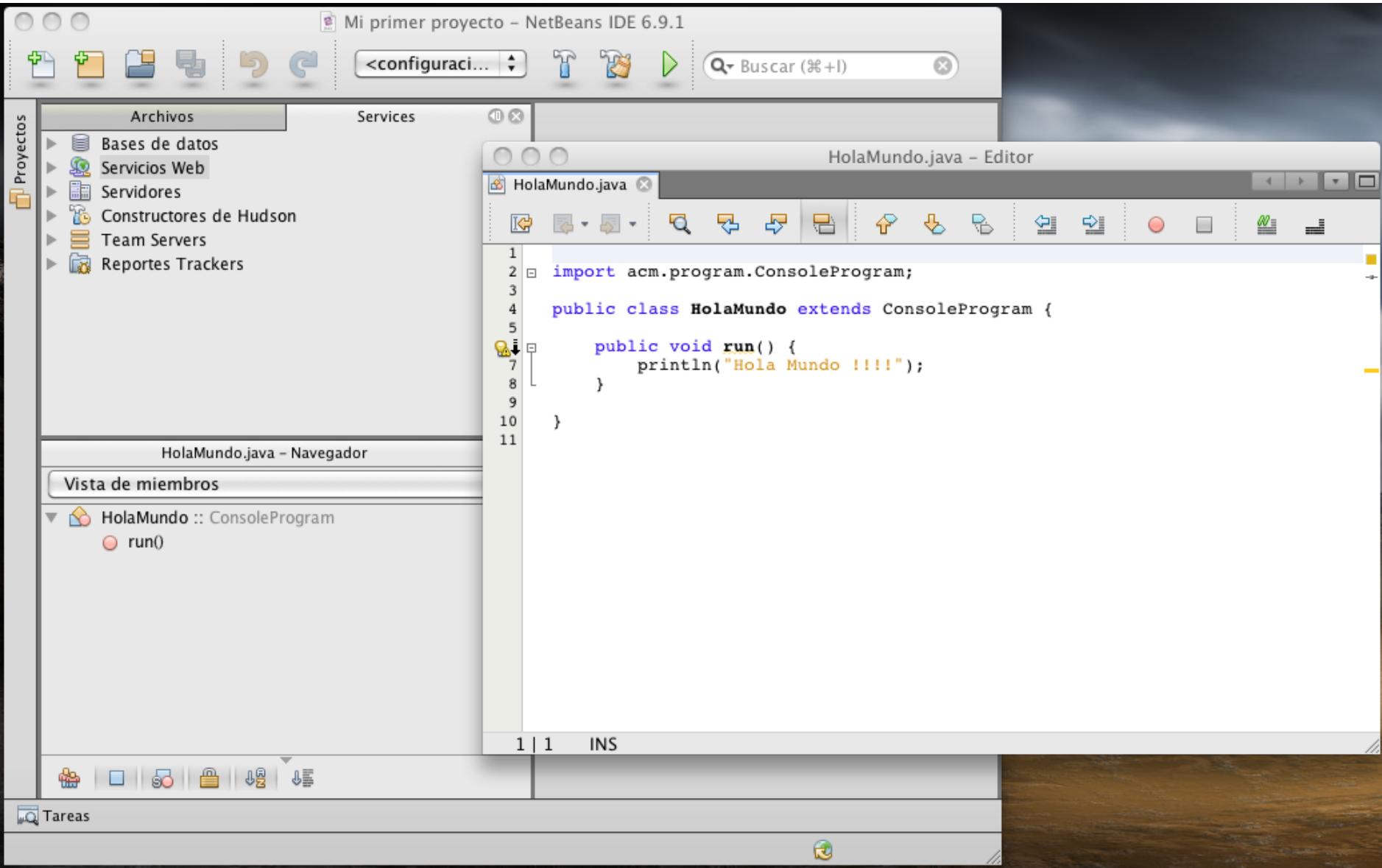
- Pulsamos **Añadir biblioteca**
- Seleccionamos **ACMTF**
- Pulsamos **Aceptar**



No encuentro las ventanas

# He perdido la ventana 😞

- Los paneles y ventanas del Netbeans son completamente configurables
  - Se pueden extraer como ventanas independientes
  - Se pueden iconificar en los laterales
  - Las pestañas se pueden reordenar
  - Se pueden eliminar
- Esta flexibilidad es buena pero a veces pulsamos algo por error y no sabemos qué hacer



# El botón del pánico 😊

- En el menú **Ventana**, seleccionamos **Restaurar ventana** y se reconfiguran las ventanas a su configuración estándar.



# **INFORMACIÓN ADICIONAL**

- Guía de instalación de Netbeans ([Catalán](#), [Castellano](#), [Inglés](#))
- Proyecto Netbeans: <http://netbeans.org>