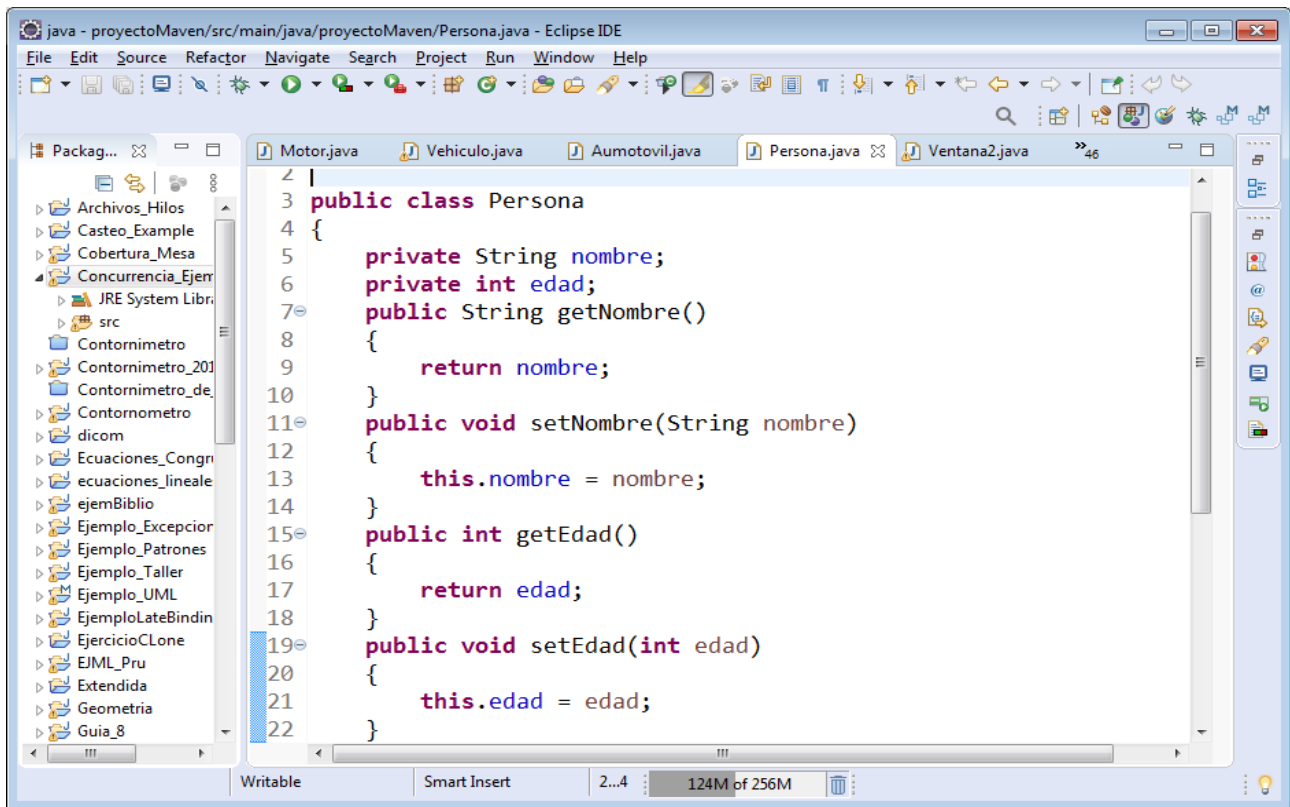
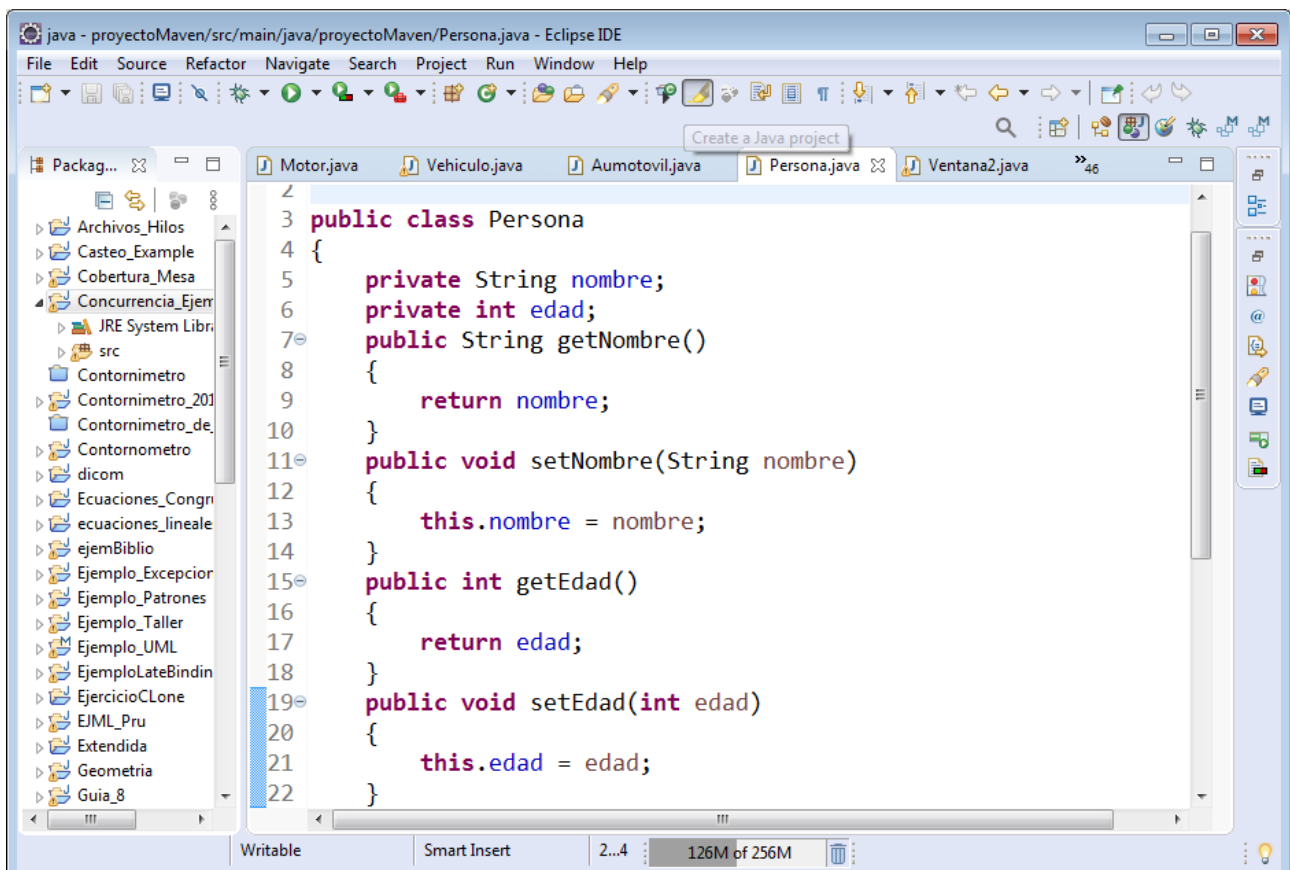


Primer proyecto en JAVA

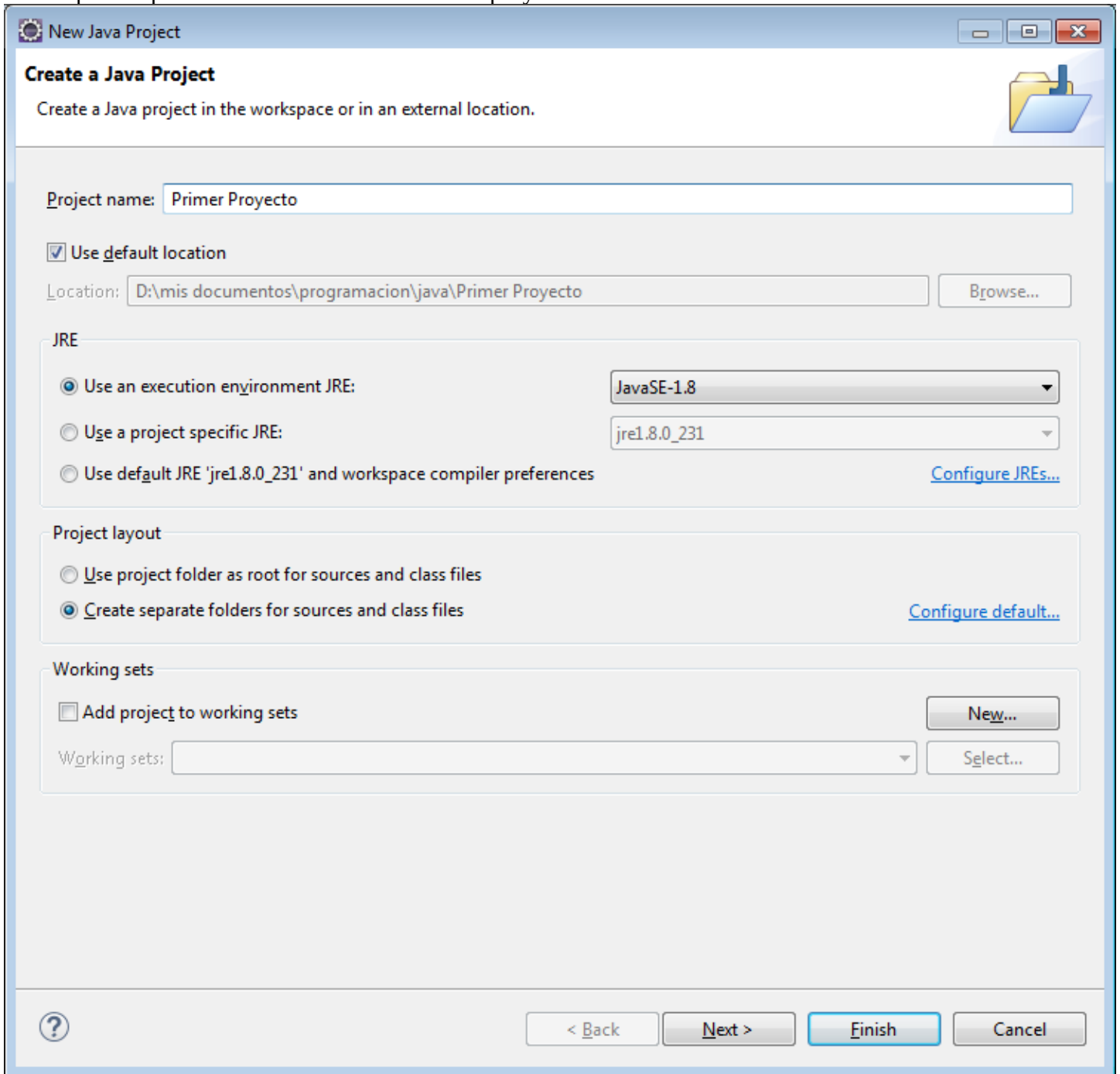
Entorno de trabajo Eclipse



Para crear un proyecto en Java se debe clicar en File → New → Java Project



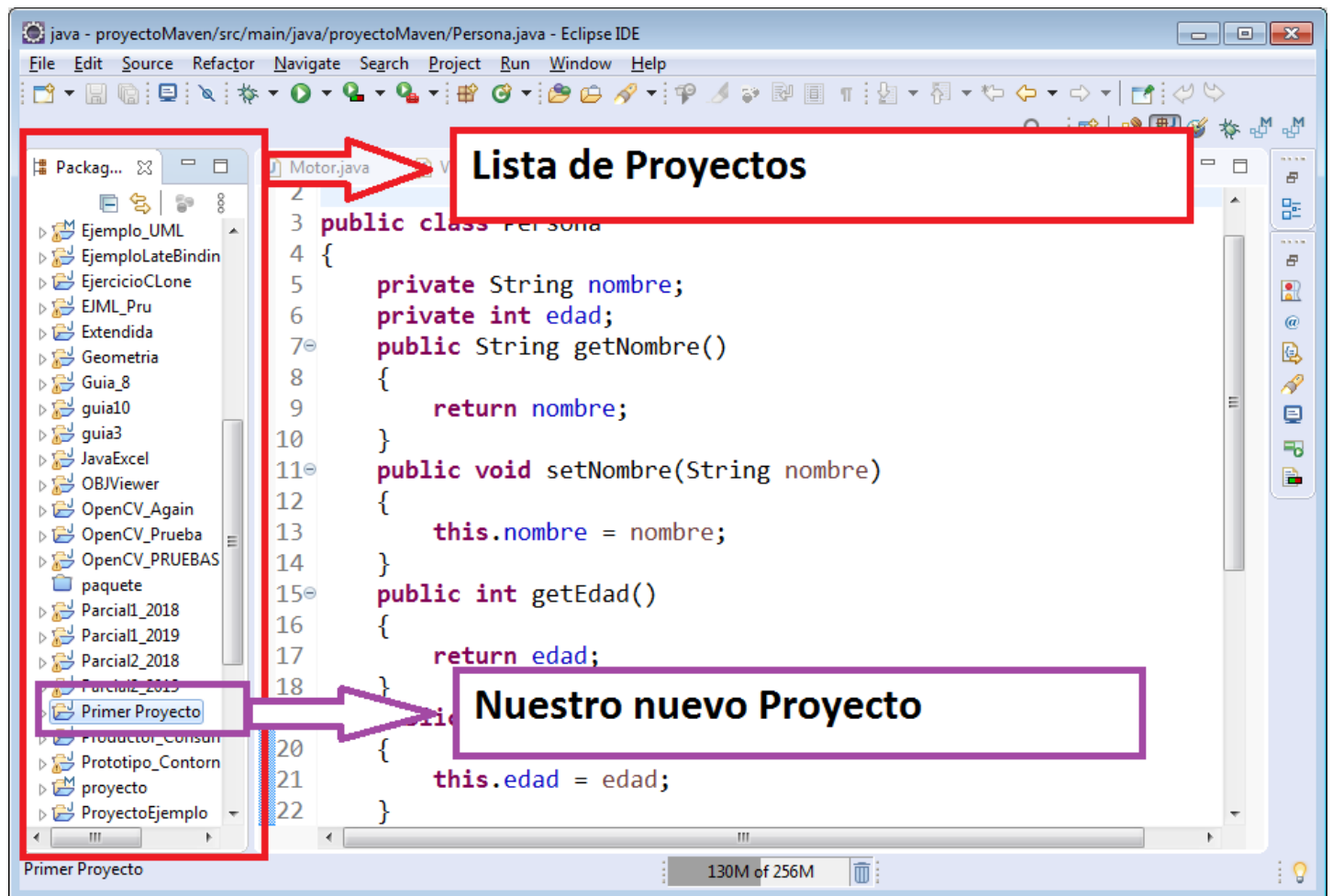
Se nos pedirá que le demos un nombre a nuestro proyecto:



The screenshot shows the 'New Java Project' dialog box in the Eclipse IDE. The title bar reads 'New Java Project'. The main heading is 'Create a Java Project' with a subtext 'Create a Java project in the workspace or in an external location.' and a folder icon. The 'Project name' field is filled with 'Primer Proyecto'. The 'Use default location' checkbox is checked, and the 'Location' field shows 'D:\mis documentos\programacion\java\Primer Proyecto' with a 'Browse...' button. Under the 'JRE' section, 'Use an execution environment JRE:' is selected, with 'JavaSE-1.8' chosen from the dropdown. Other options include 'Use a project specific JRE:' (with 'jre1.8.0_231' in the dropdown) and 'Use default JRE 'jre1.8.0_231' and workspace compiler preferences' (with a 'Configure JREs...' link). The 'Project layout' section has 'Create separate folders for sources and class files' selected, with a 'Configure default...' link. The 'Working sets' section has 'Add project to working sets' unchecked, a 'New...' button, and a 'Working sets:' dropdown with a 'Select...' button. At the bottom, there is a help icon, and navigation buttons: '< Back', 'Next >', 'Finish' (highlighted in blue), and 'Cancel'.

Luego de eso podemos hacer click en el botón **Finish**

En Elcipse tenemos una vista de todos los proyectos en los cuales hemos trabajo, usualmente esa lista de proyectos se visualiza del lado izquierdo de la pantalla (La distribución de todos los paneles es configurable). Para comenzar a trabajar debemos seleccionarlo.

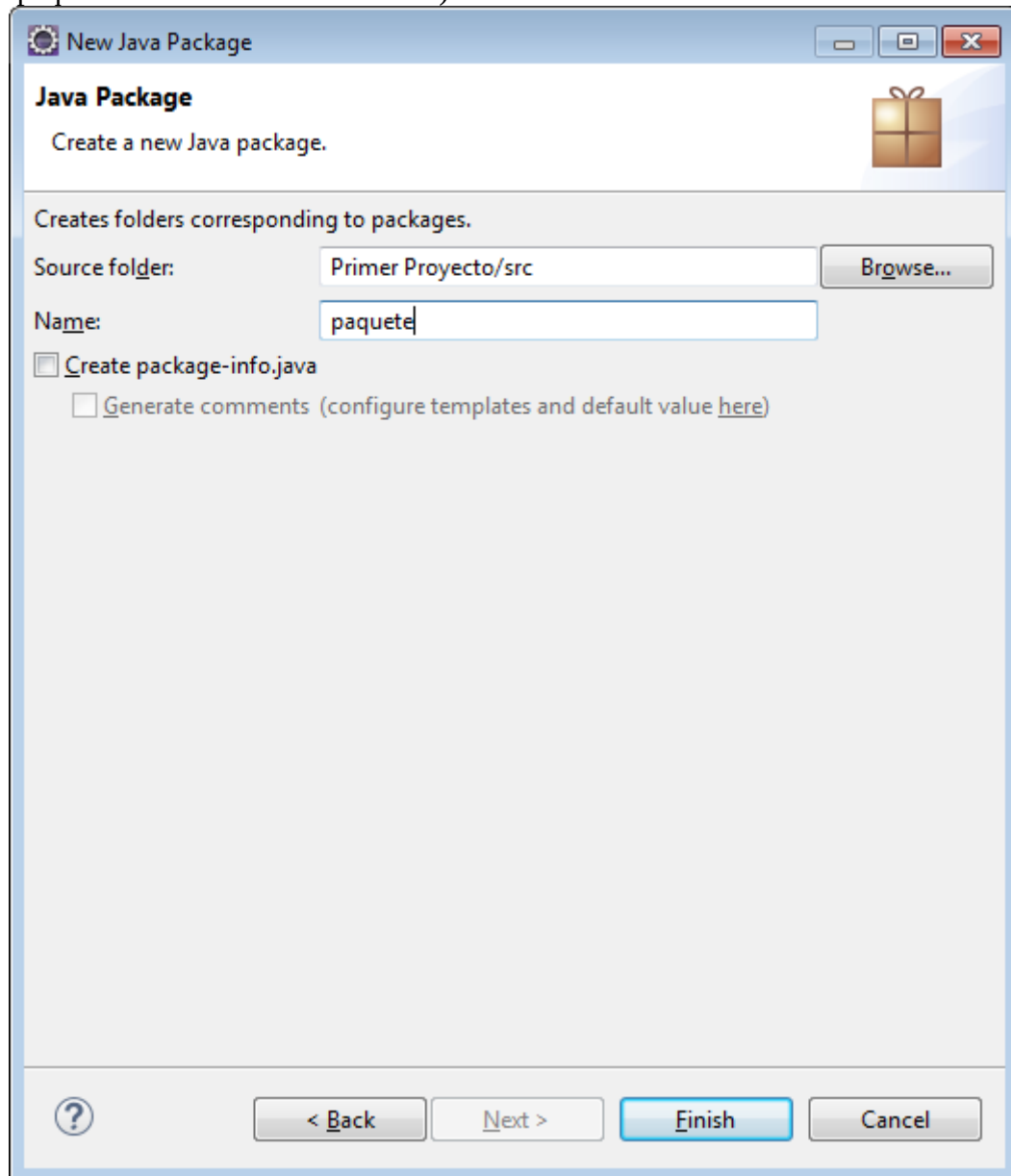


Desplegamos nuestro proyecto, y en haciendo click derecho sobre la carpeta **src** seleccionamos **new** → **package**

Cabe aclarar que las opciones que se nos presentan al seleccionar **new** son las más usadas habitualmente, por lo que es posible que en ocasiones no encontremos inmediatamente que la buscamos (en nuestro ejemplo, package). En ese caso, para acceder a la totalidad de las funcionalidades deberemos hacer click en **others** → **java**

Java nos permite mediante plugins agregar diferentes funcionalidades, como por ejemplo instalar paquetes para poder desarrollar proyectos en C++, PHP, Python y otros lenguajes. Es por eso que al hacer click en **new** la cantidad de opciones posibles varía en cada versión en la que estemos trabajando.

Al hacer click en new -> package (nuevo paquete) se nos pedirá que le demos un nombre al paquete (por convención los paquetes se escriben en minúscula)



En nuestro ejemplo le hemos puesto de nombre “*paquete*”

Para crear nuestras clases hacemos click derecho dentro del paquete y seleccionamos **new-Class**

En nuestro ejemplo a la nueva clase la nombramos **Persona** (por convención las clases de Java tienen su primer letra en mayúscula y el resto en minúscula). En el apartado Package indicamos en que paquete queremos que se cree la clase. Podríamos haber omitido el paso de crear el paquete, e indicarlo en este momento. Aquí podemos indicar muchas características de nuestra nueva clase, en particular destacaré la opción (que en este caso mantendremos desmarcada) que nos da la posibilidad de agregar el método estatico main.

New Java Class

Create a new Java class.

Source folder:

Package:

☐ Enclosing type:

Name:

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static

Superclass:

Interfaces:

Which method stubs would you like to create?

☐ public static void main(String[] args)

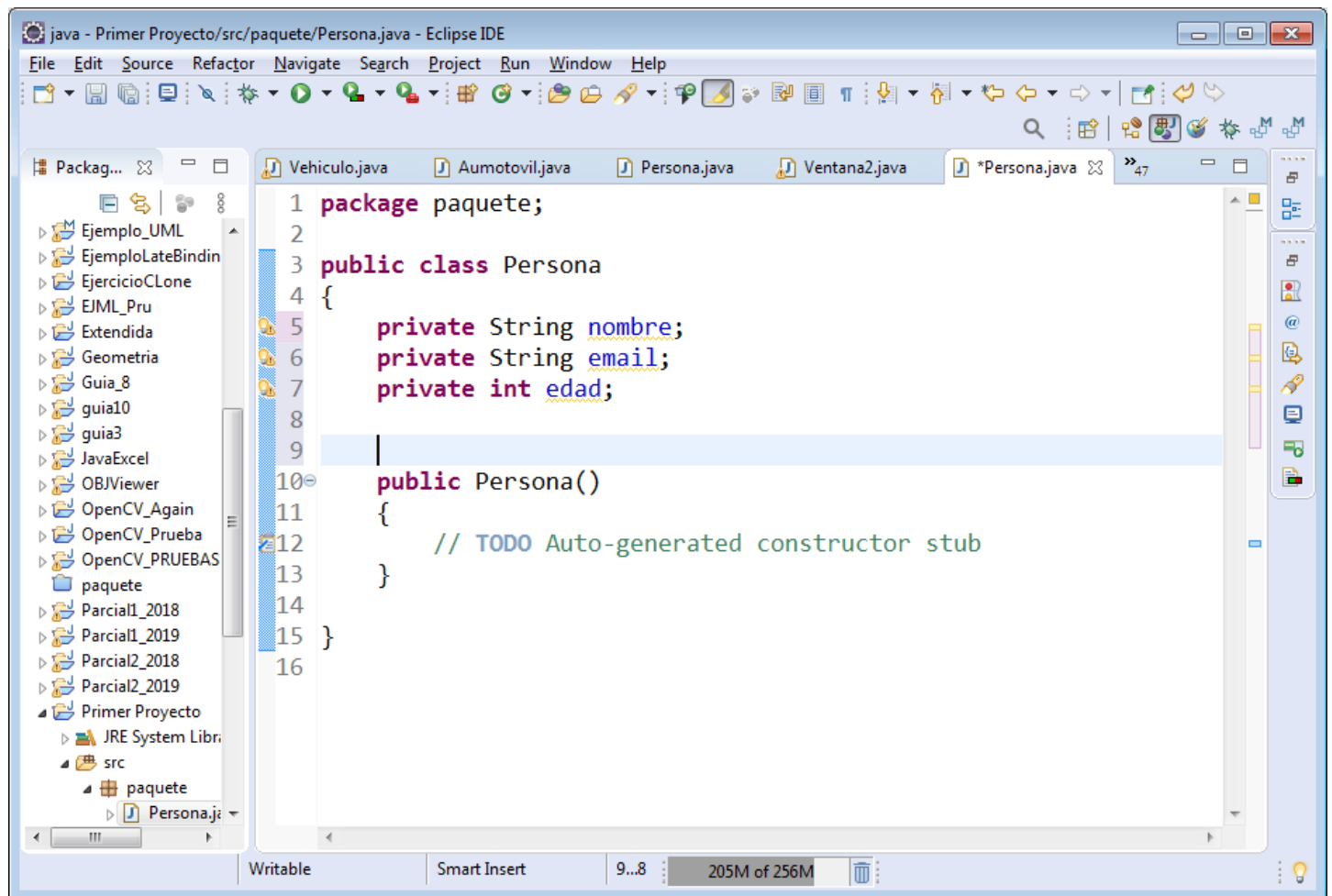
☒ Constructors from superclass

☐ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))

☐ Generate comments

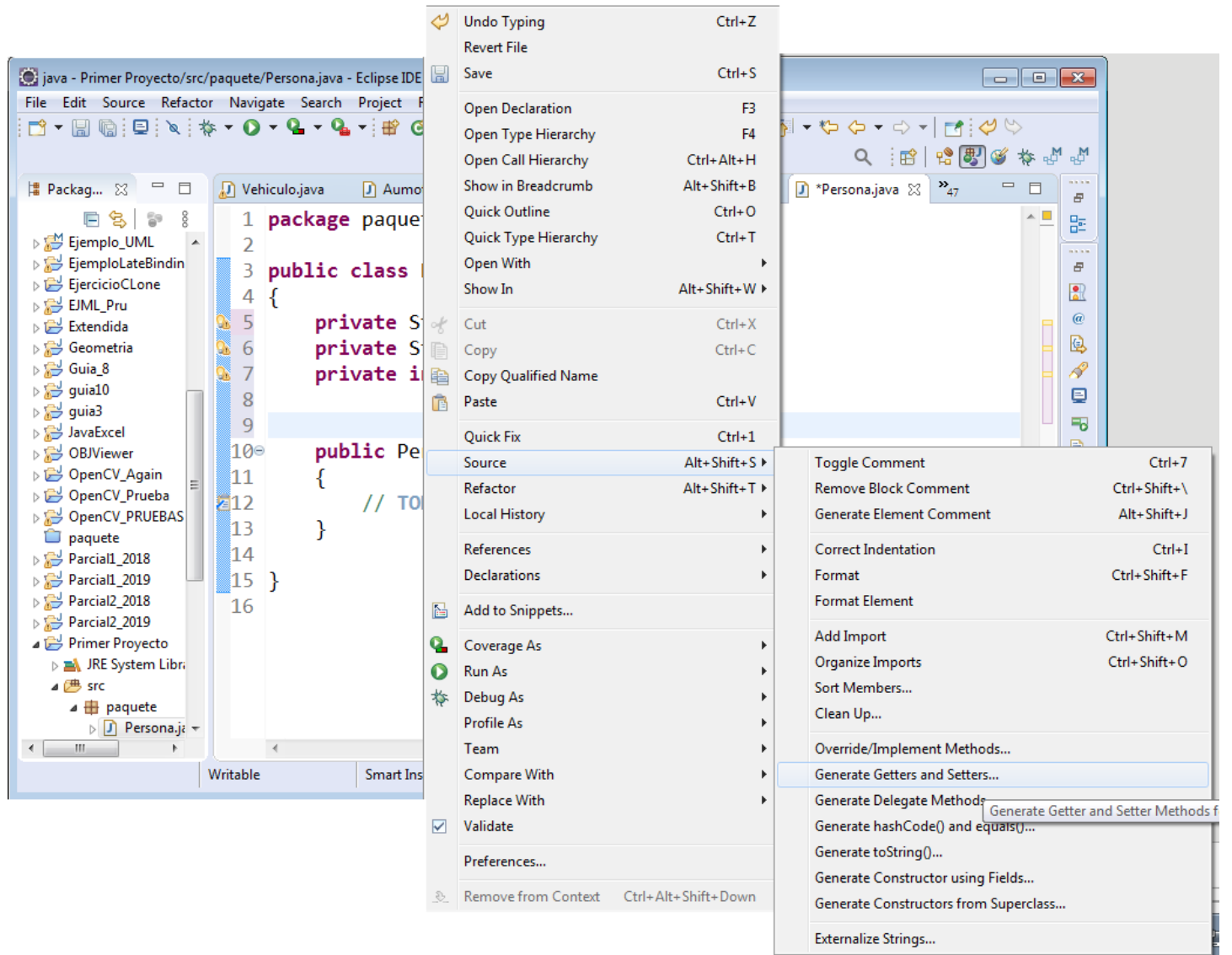
Luego de hacer en el boton **finish** ya podemos comenzar a escribir el código en nuestra clase.

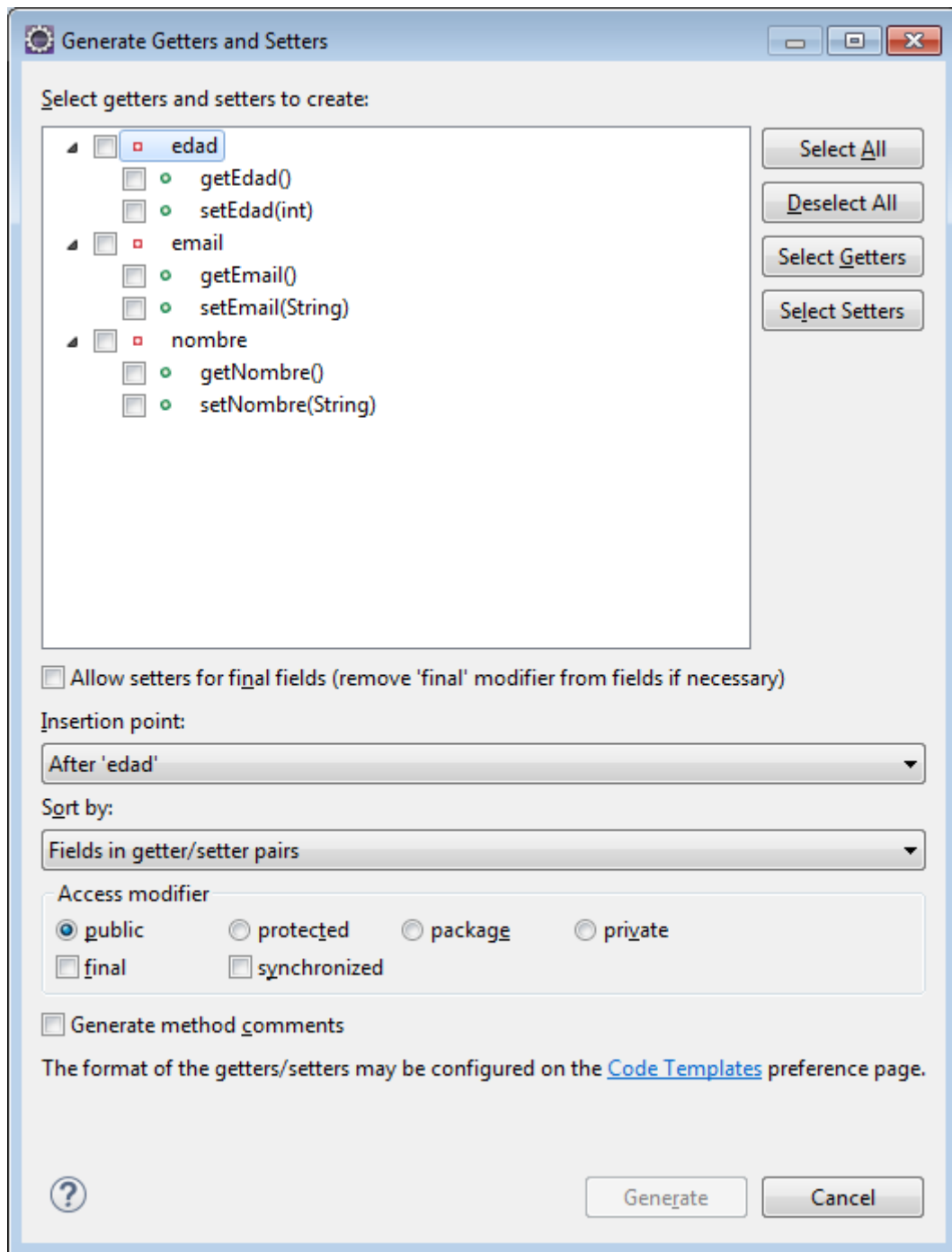


En todos los entornos de trabajo hay acciones comunes y rutinarias que se pueden realizar de forma automática. En este apunte destacar :

Crear getters y setters

Se hace click derecho sobre el lugar del código donde queremos generar los getters y setters y seleccionamos **source** → **Generate Getters and Setters**

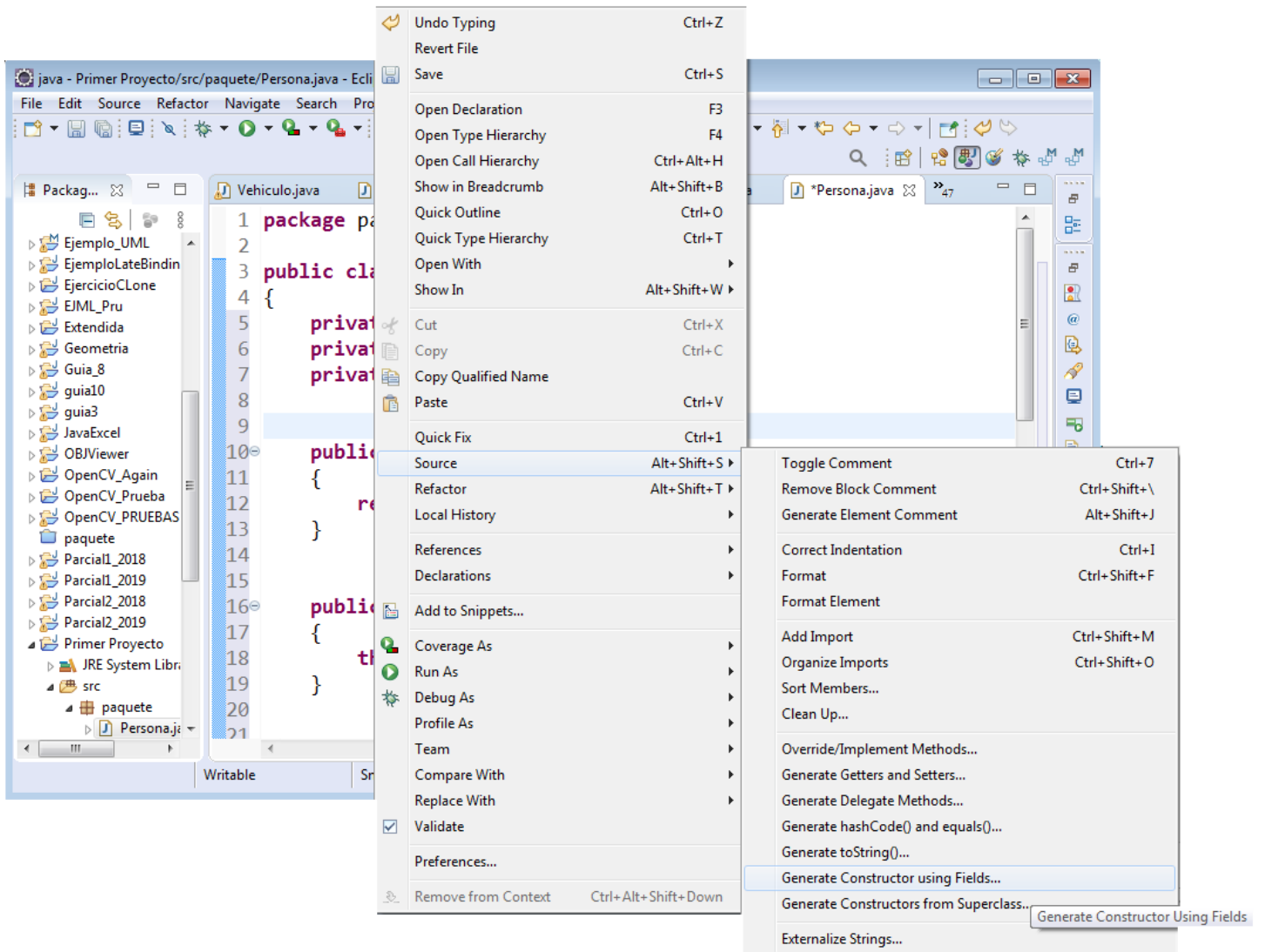


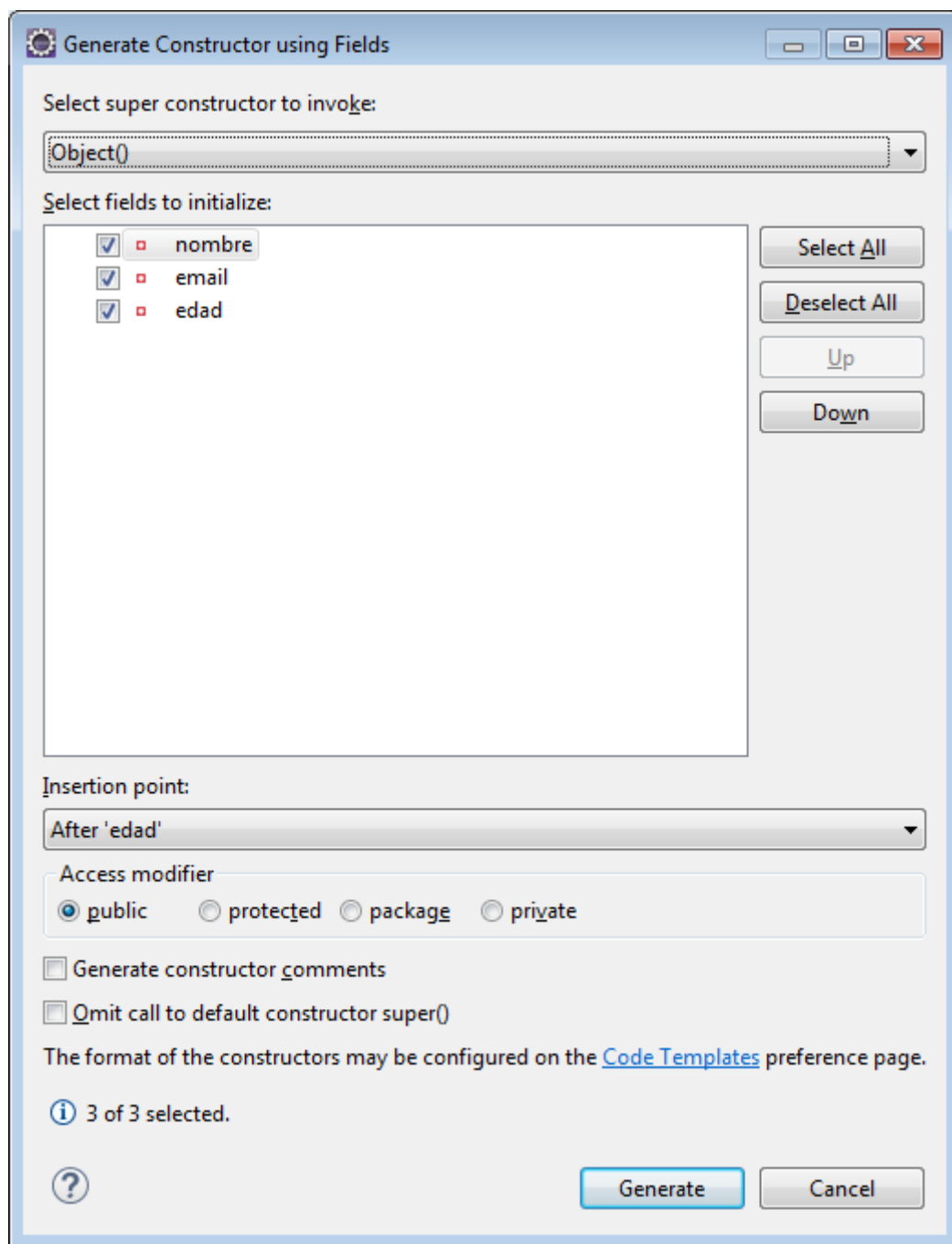


Ahora podemos seleccionar los atributos a los cuales queremos generarle sus correspondientes getters y setters, y también se nos permite seleccionar solo los getters, solo los setters, seleccionar o deseleccionar, cambiar los modificadores de acceso, etc.

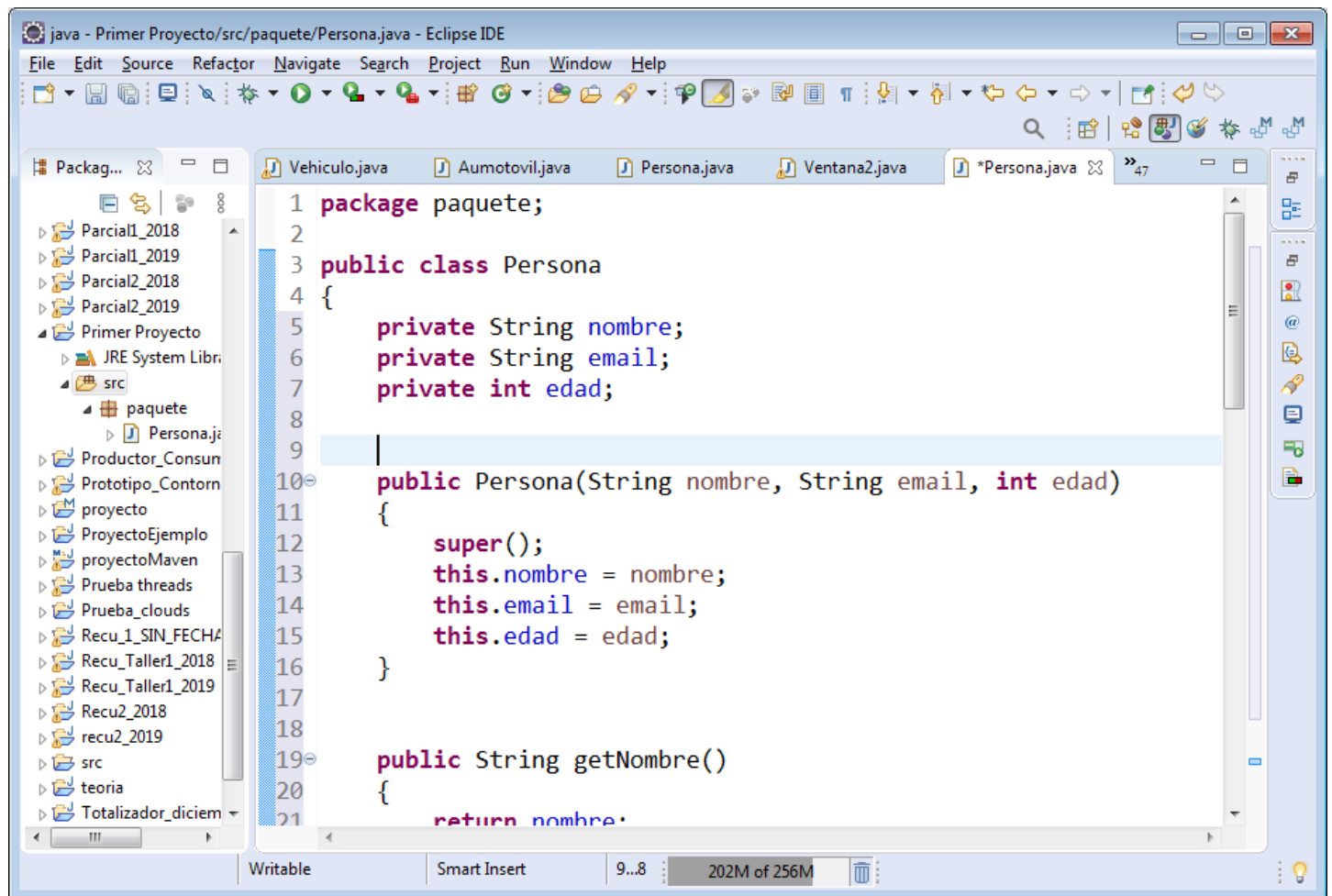
Generar Constructores

De la misma forma, hacemos click derecho en el lugar del código en donde queremos insertar los constructores (se recomienda ubicarlos al comienzo de la clase, abajo de los atributos) y seleccionamos **source** → **Generate constructor using fields**



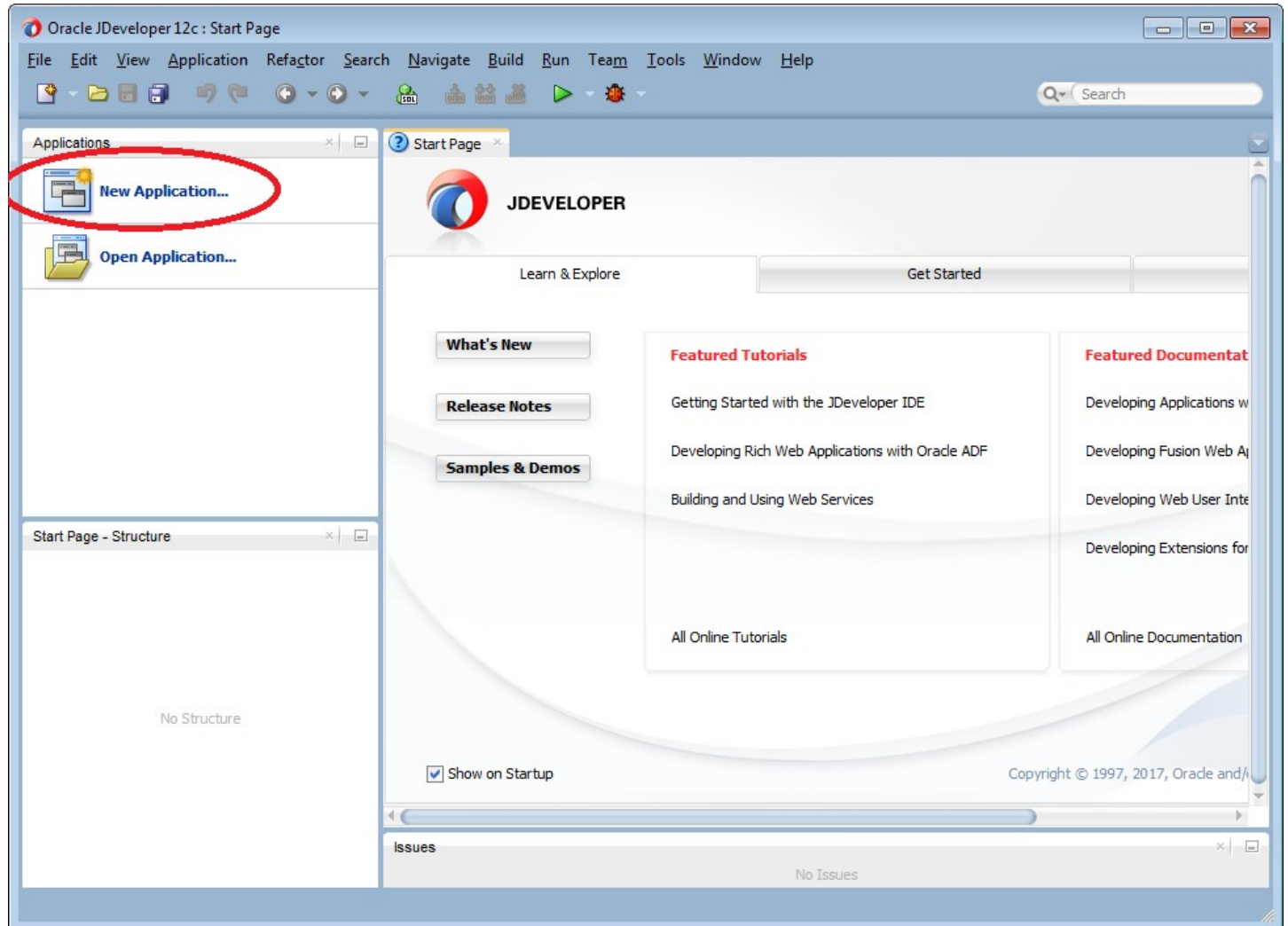


Podemos seleccionar que atributos se agregarán en el constructor así como cambiar los modificadores de acceso y otras opciones. Por defecto selecciona todos los atributos (en el orden en el que fueron declarados) y el acceso será público.

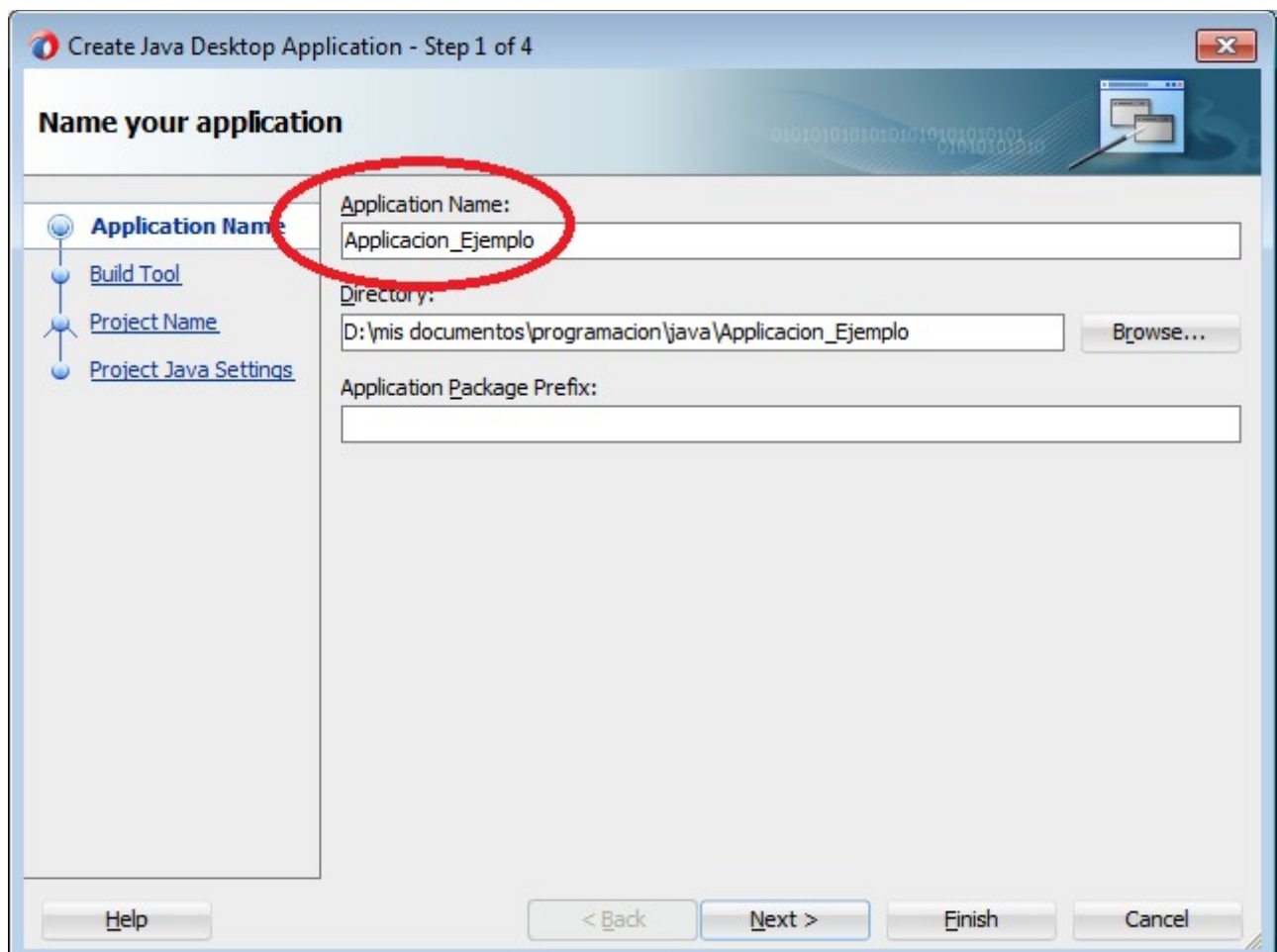
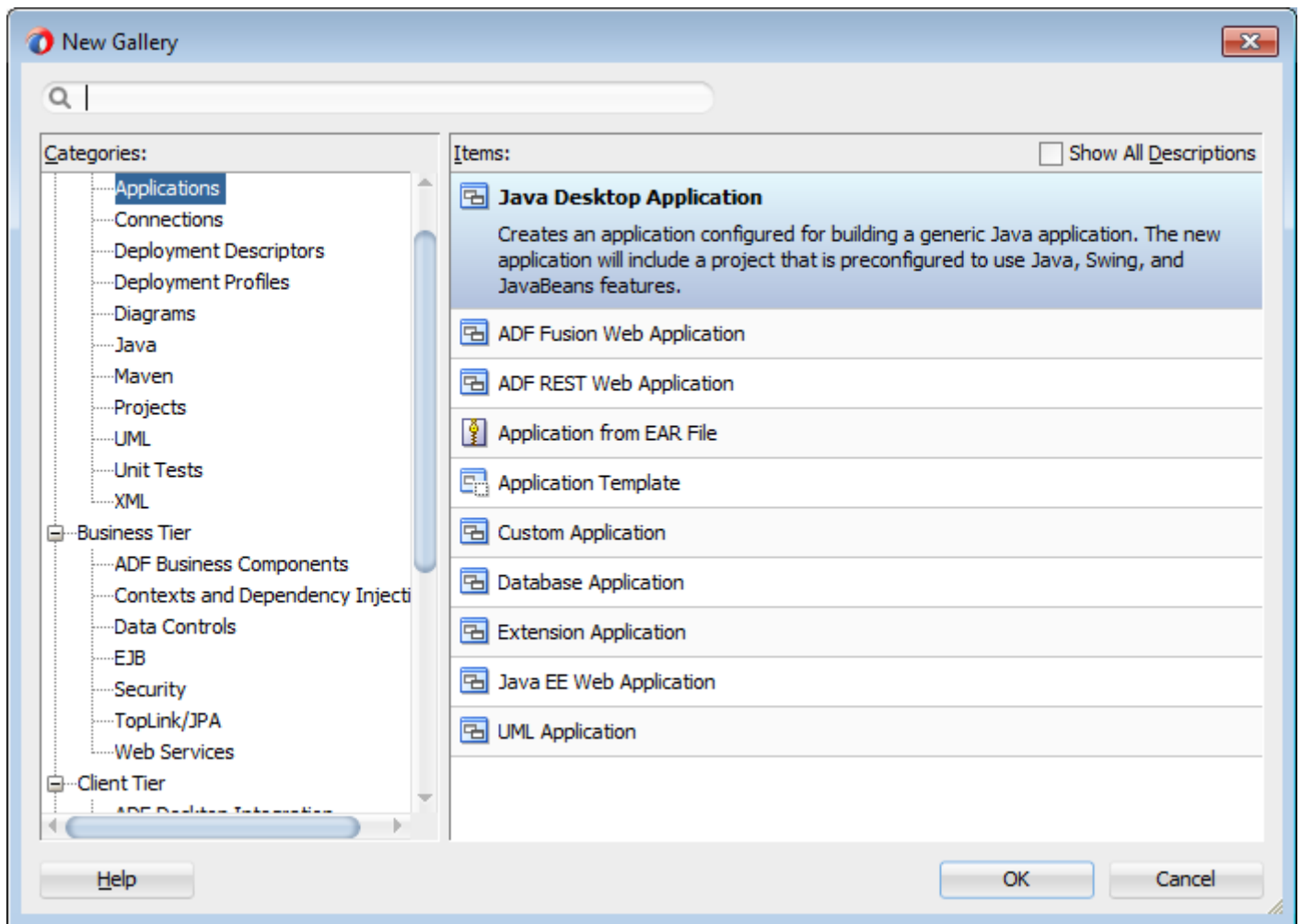


Entorno de trabajo JDeveloper

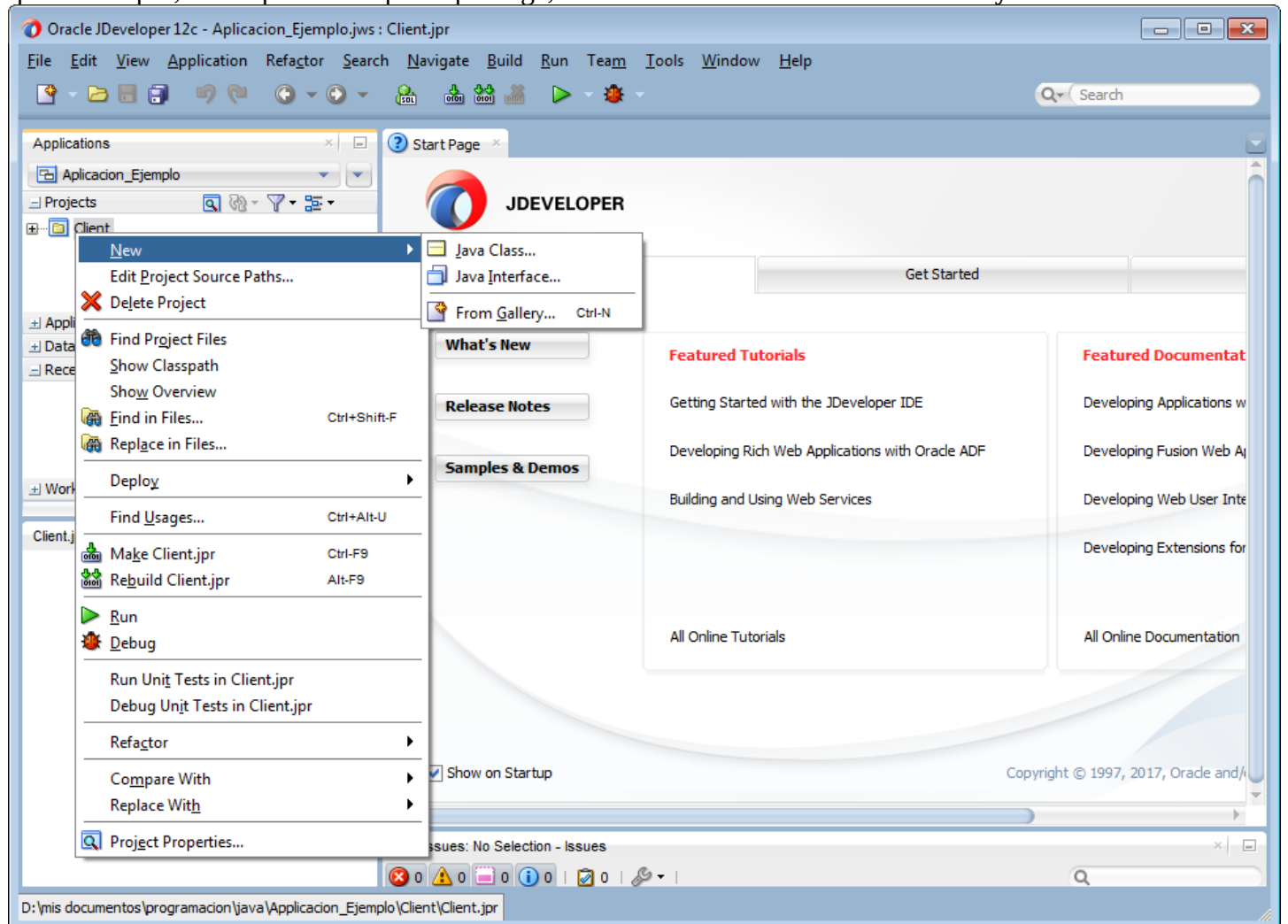
Para crear un nuevo proyecto en Jdeveloper primero deberemos crear una nueva aplicación

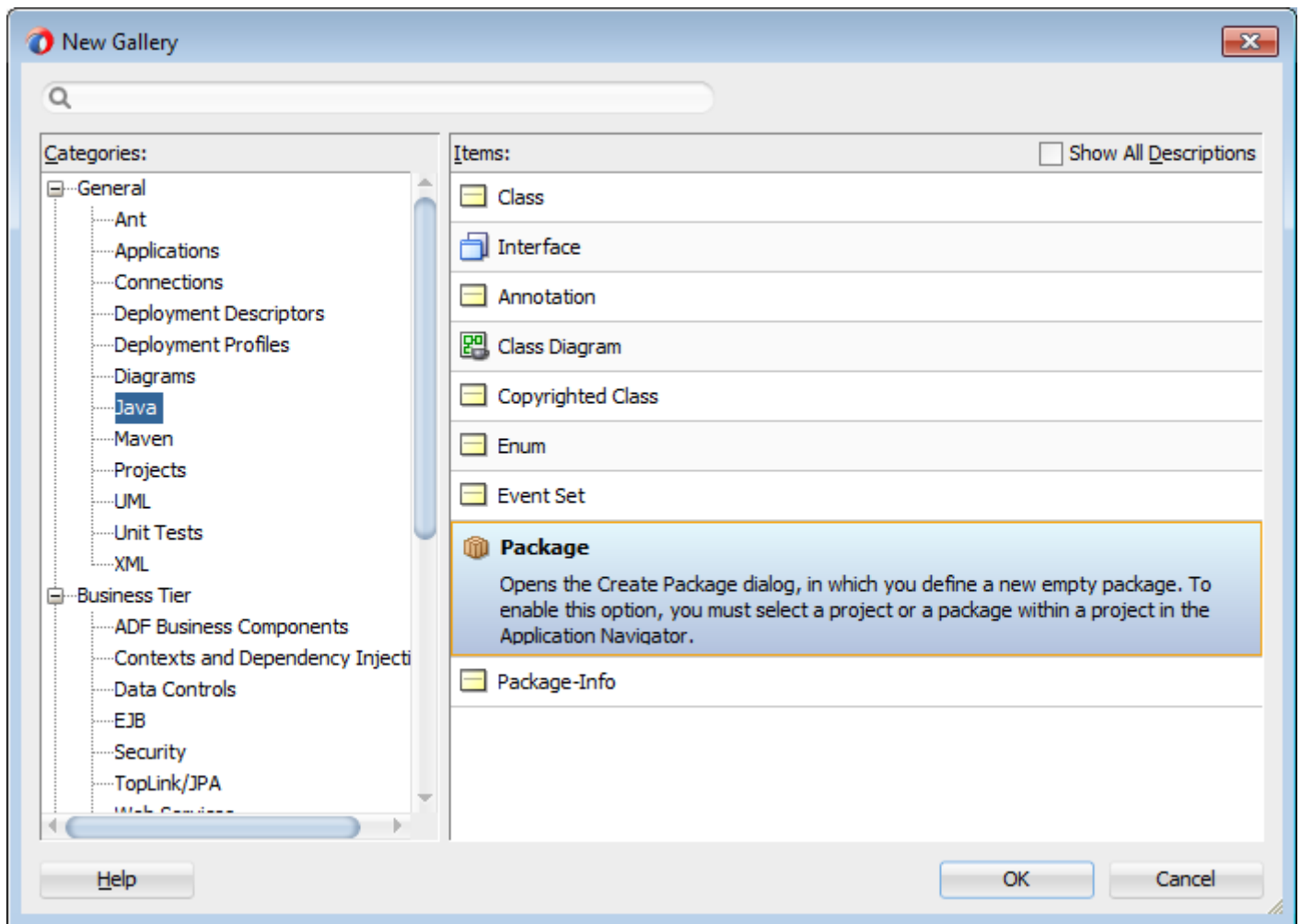


Seleccionamos Java Desktop Application

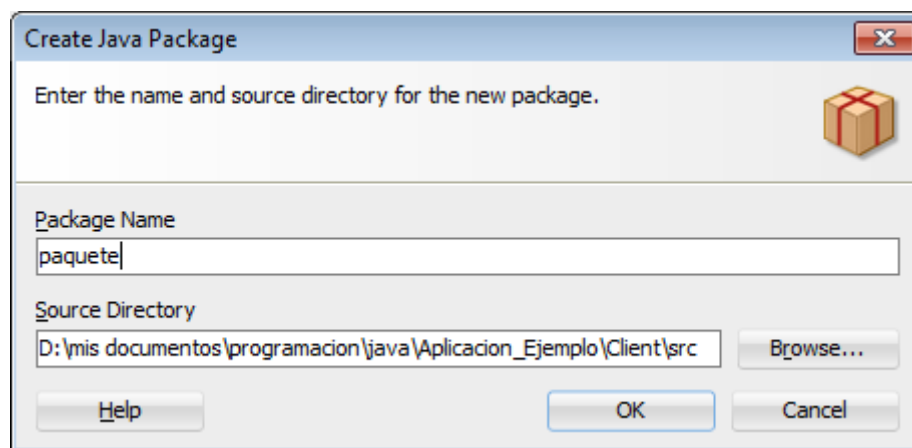


Le damos un nombre a la aplicación (en nuestro caso Aplicacion_Ejemplo) y hacemos click en **Finish**. Una vez creada la aplicación, por defecto tenemos un proyecto llamado Client en el cual podremos comenzar a crear nuestros paquetes y clases. También podemos crear nuevos proyectos dentro de la misma aplicación. Para crear nuestro paquete hacemos click derecho sobre nuestro proyecto Client y seleccionamos **New**, al igual que en Eclipse, si no aparece la opción package, la buscaremos dentro de **From Gallery**





Dentro de Gallery seleccionamos **General** → **Java** → **Package**



Le damos un nombre al paquete, en nuestro caso “paquete” y hacemos click en **ok**

Una vez creado el paquete podemos crear nuestras clases haciendo click derecho dentro del paquete y seleccionamos **new** → **Java Class**

Le asignamos un nombre (en nuestro ejemplo “Persona”) y podemos acceder a las diferentes opciones como por ejemplo agregar el método main (en nuestro caso lo dejaremos desmarcado)

Create Java Class

Enter the details of your new class.

Name:

Package:

Extends:

Optional Attributes

Implements:

Access Modifiers

☒ public

☐ package private

Other Modifiers

☒ <None>

☐ abstract

☐ final

☒ Constructors from Superclass

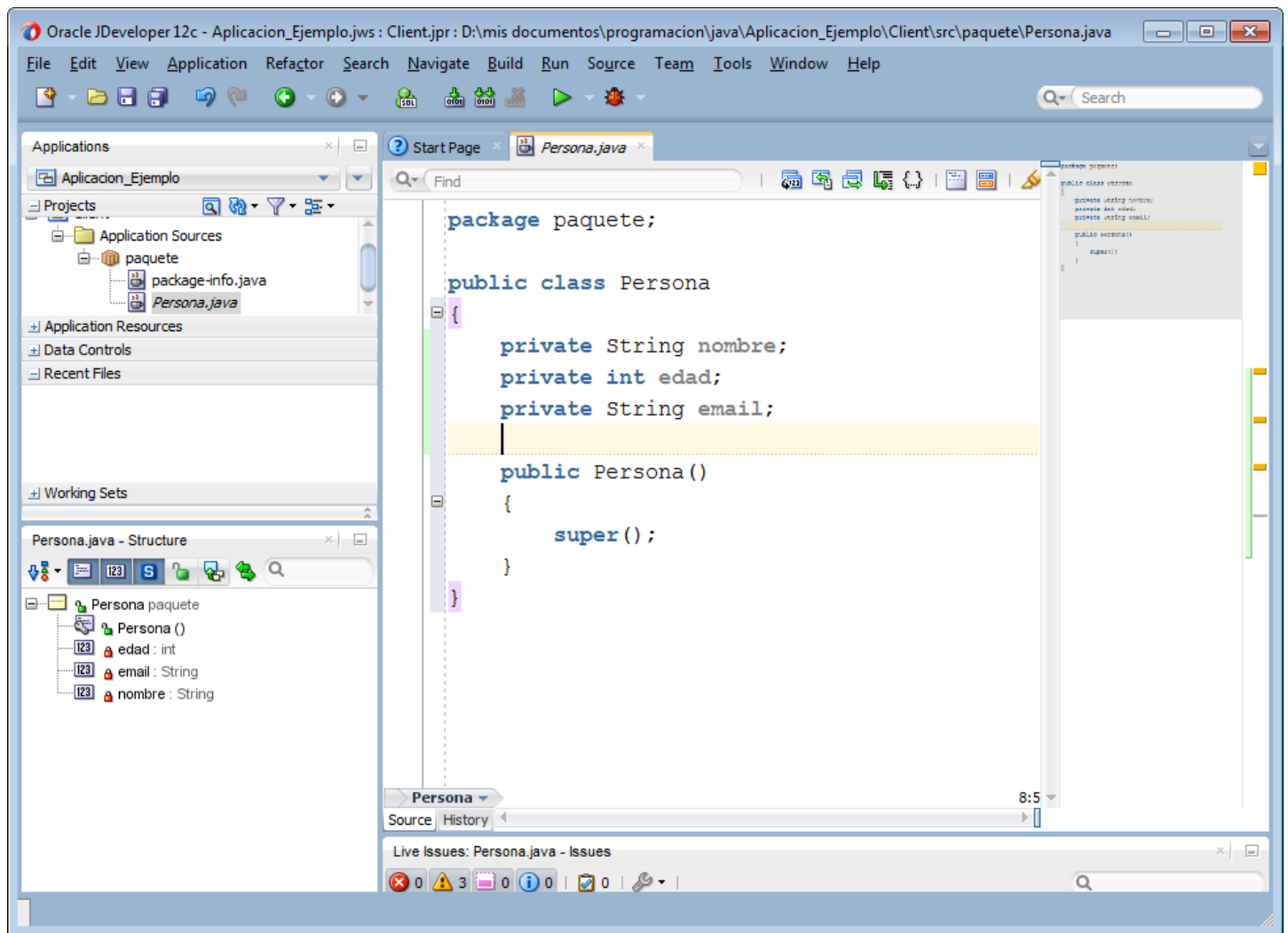
☒ Implement Abstract Methods

☐ Main Method

Messages:

Help OK Cancel

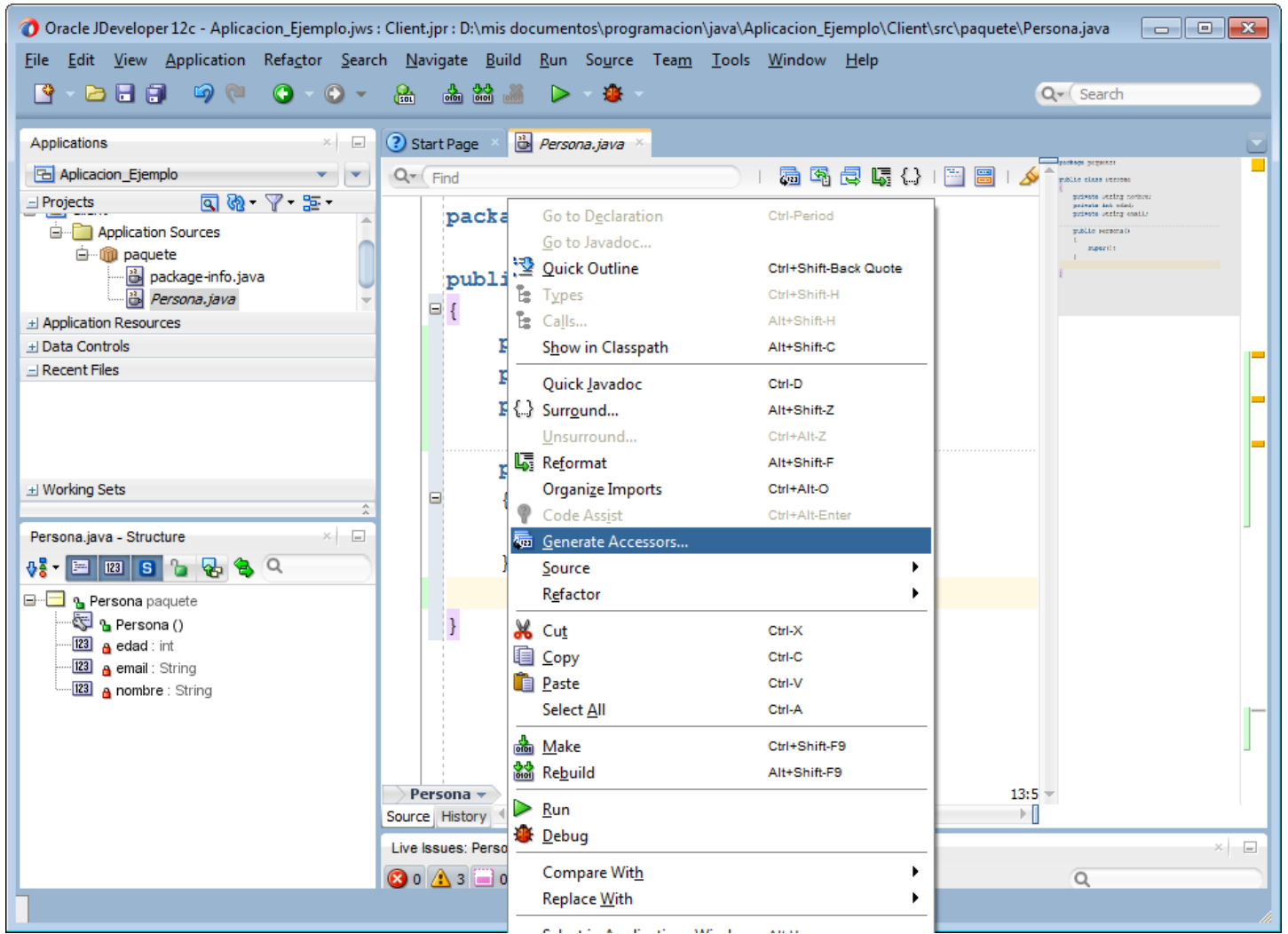
Una vez aceptado, nuestra clase estará creada y lista para que escribamos el código:

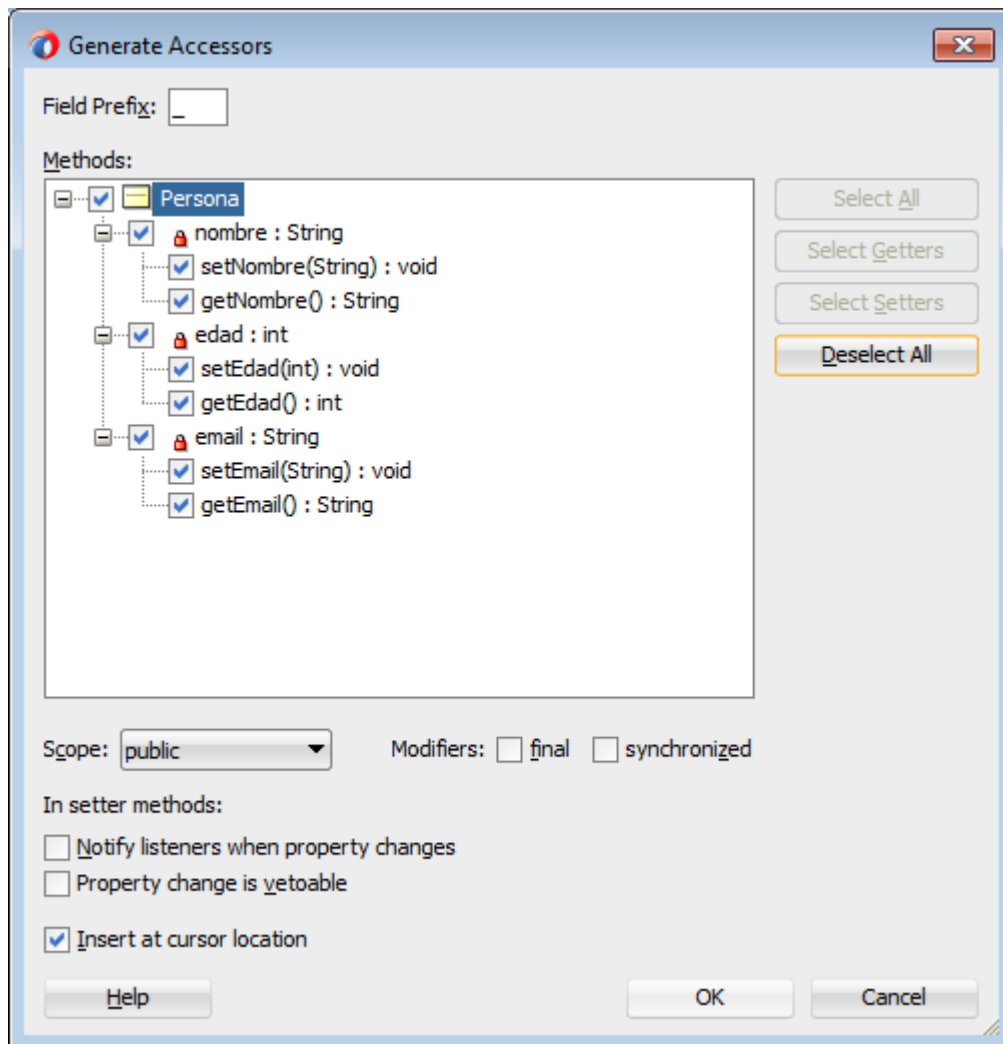


Al igual que en Eclipse, podremos crear automáticamente getters, setters y constructores.

Creacion de Getters y Setters.

Para generar automáticamente los getters y setters se hace click derecho en el lugar del código donde queremos insertarlos y seleccionamos **Generate Accesors**

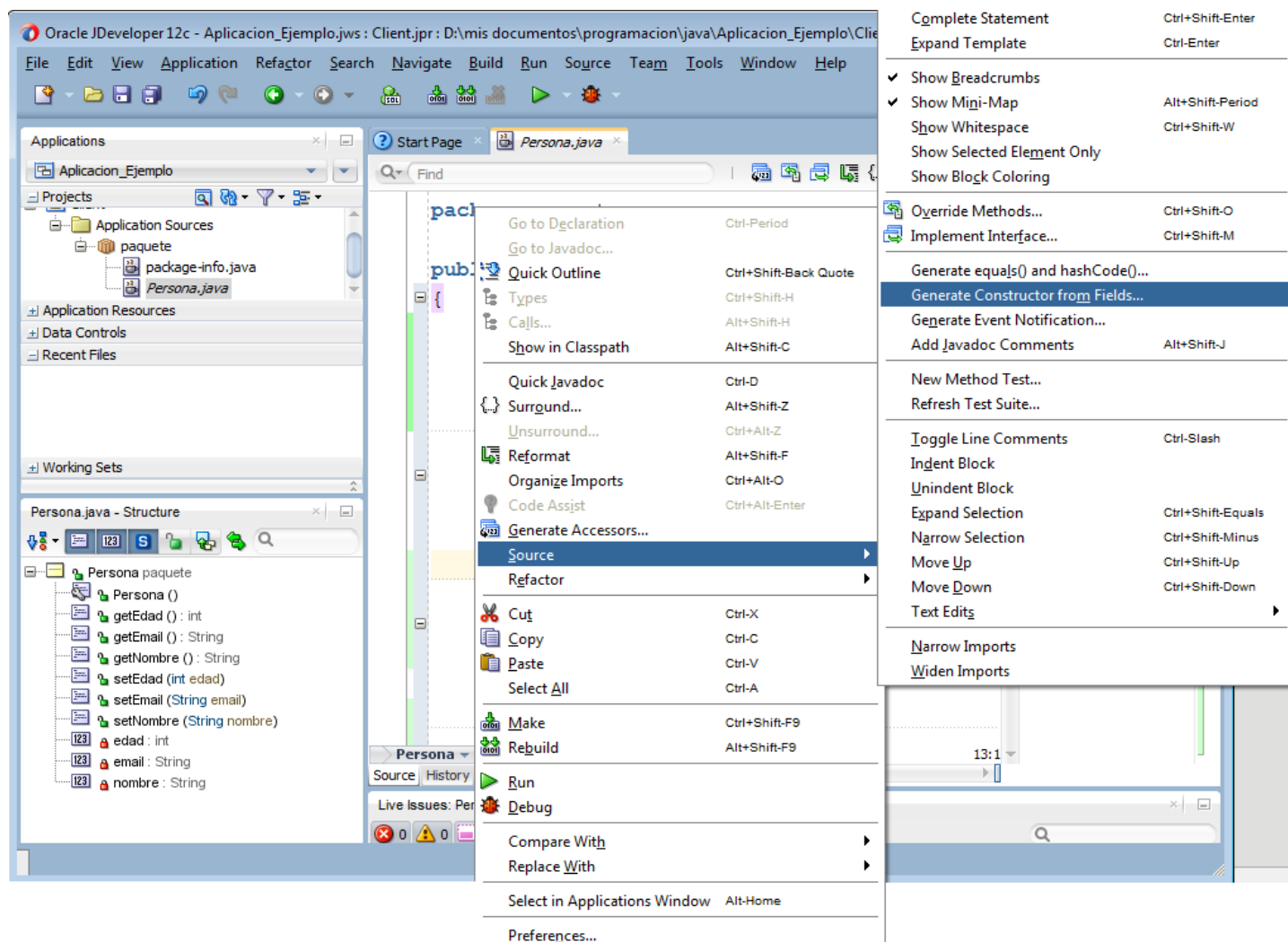




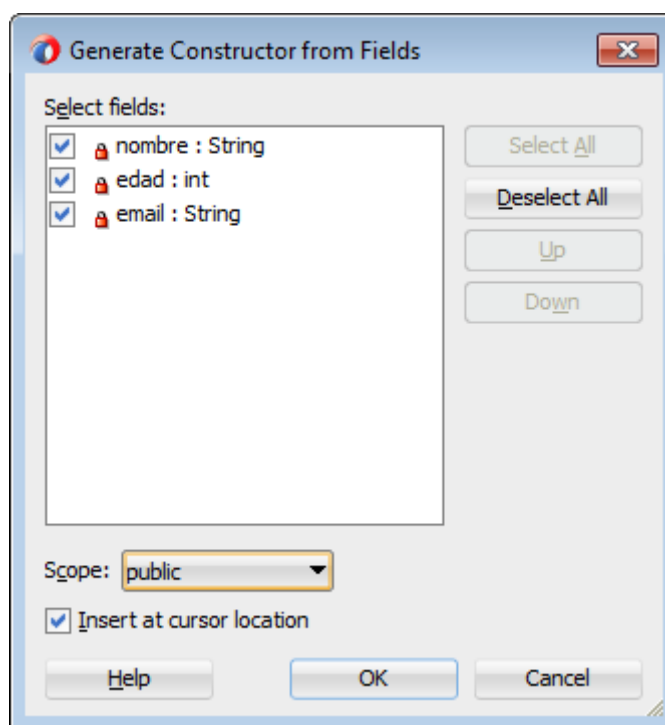
Podemos entonces seleccionar los getters y setters deseados.

Generar Constructores

De forma análoga, hacemos click derecho en el lugar del código donde queremos insertar los constructores y seleccionamos **source** → **Generate Constructor from Fields**



Podemos seleccionar los atributos que deseamos incluir en nuestro constructor (en el orden en el que fueron declarados)



Cabe aclarar que casi todas las funcionalidades descritas anteriormente, tanto en Eclipse como en Jdeveloper pueden accederse desde diferentes caminos y teclas rápidas. En el presente apunte solo se describe una forma de llegar a ellas.