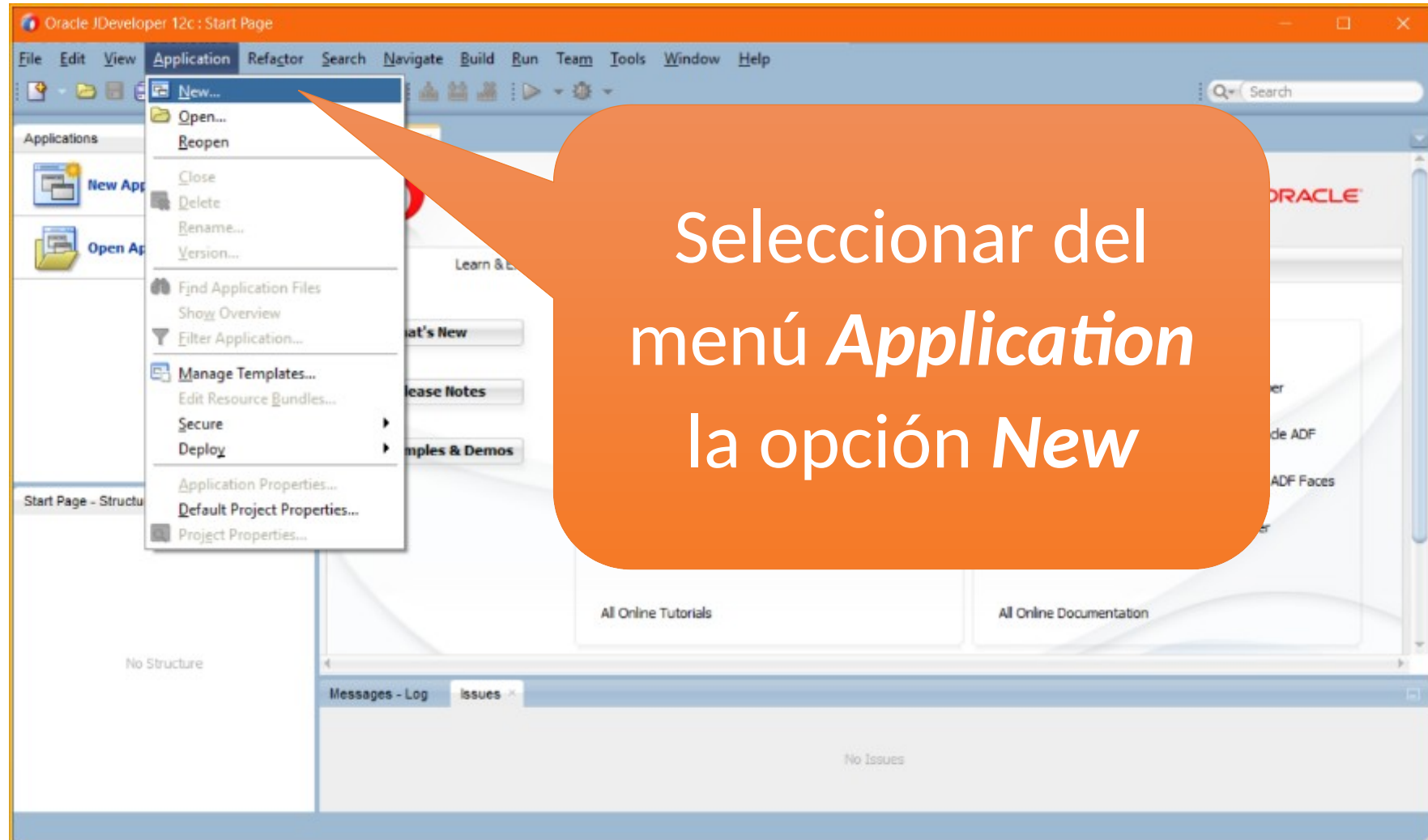
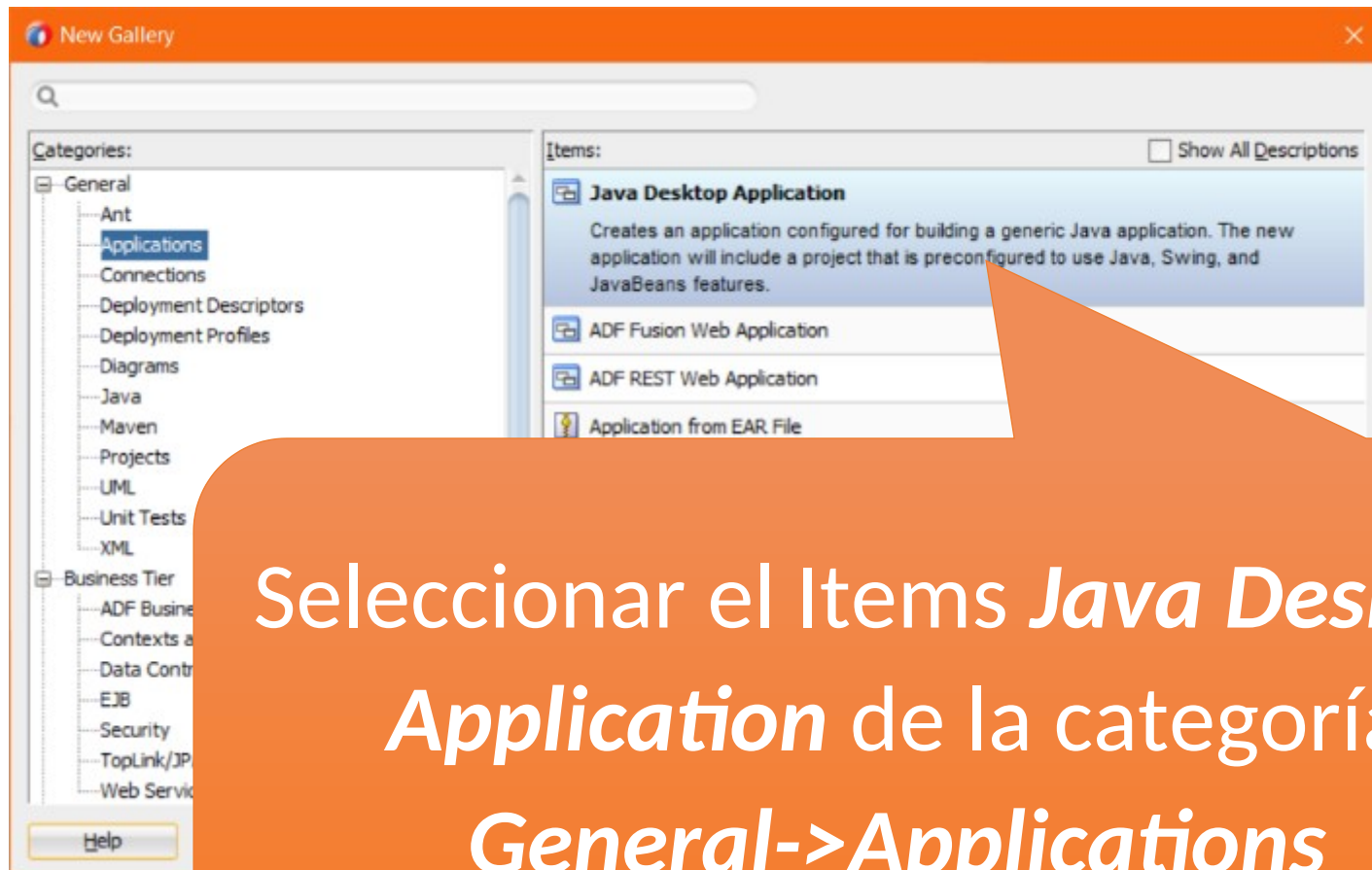


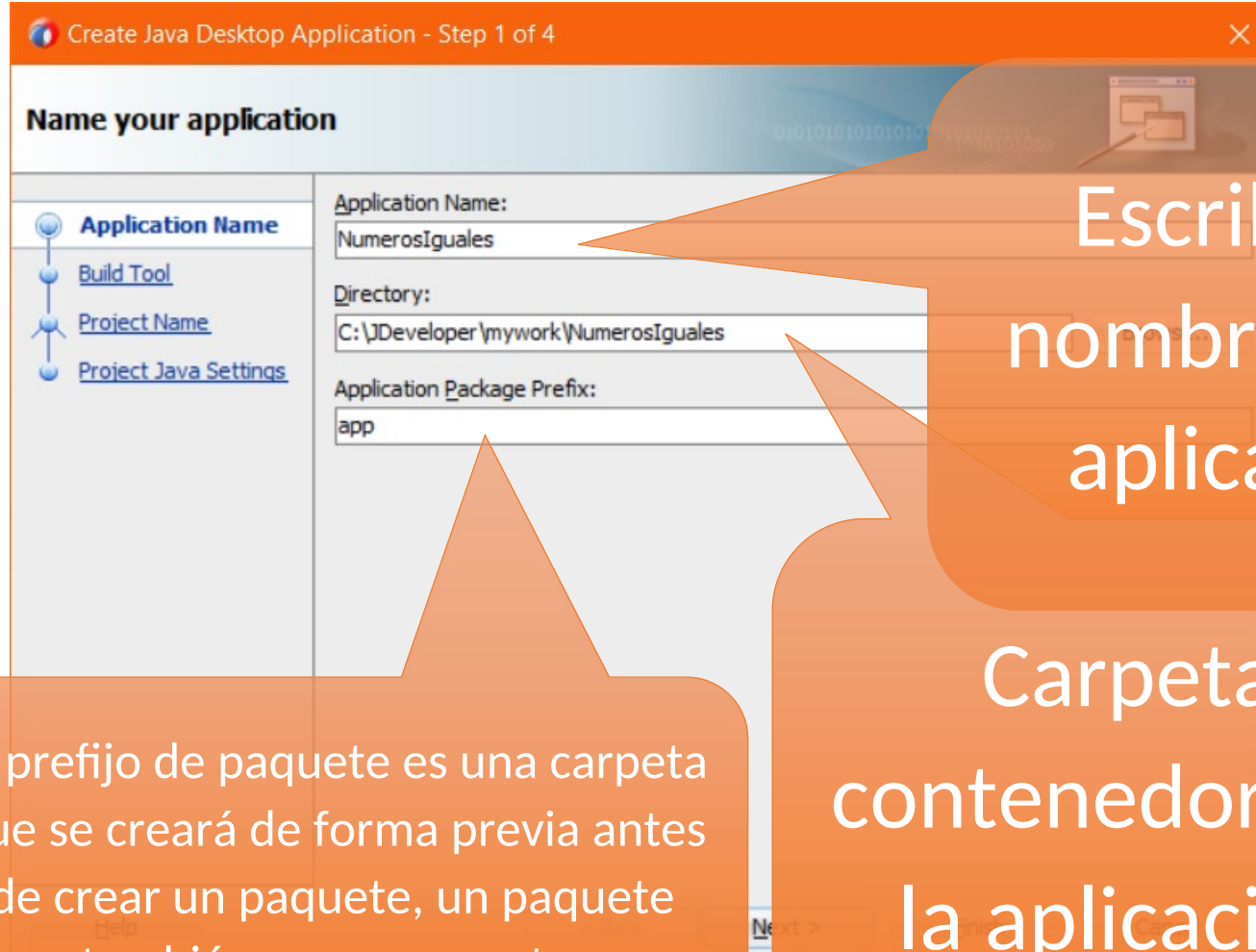
Crear una aplicación, una aplicación Java contendrá uno o más proyectos con los recursos que solucionan el problema planteado.

Requerimiento del usuario para crear una aplicación Java:

Dado dos números enteros, indicar cuál es el mayor.







Create Java Desktop Application - Step 1 of 4

Name your application

Application Name: NumerosIguales

Directory: C:\Developer\mywork\numerosiguales

Application Package Prefix: app

Application Name

Build Tool

Project Name

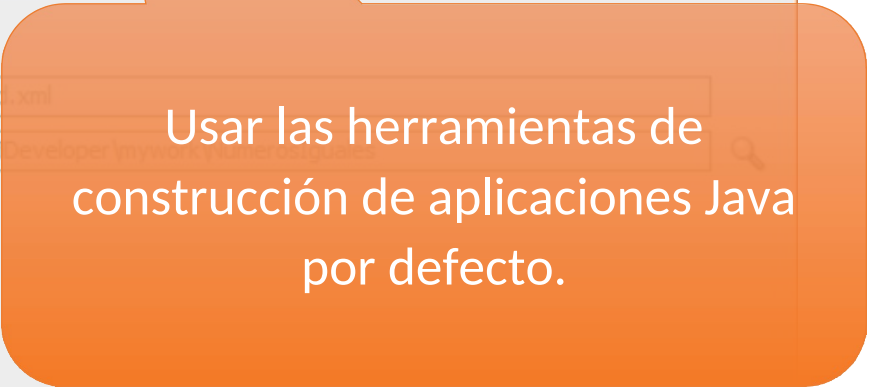
Project Java Settings

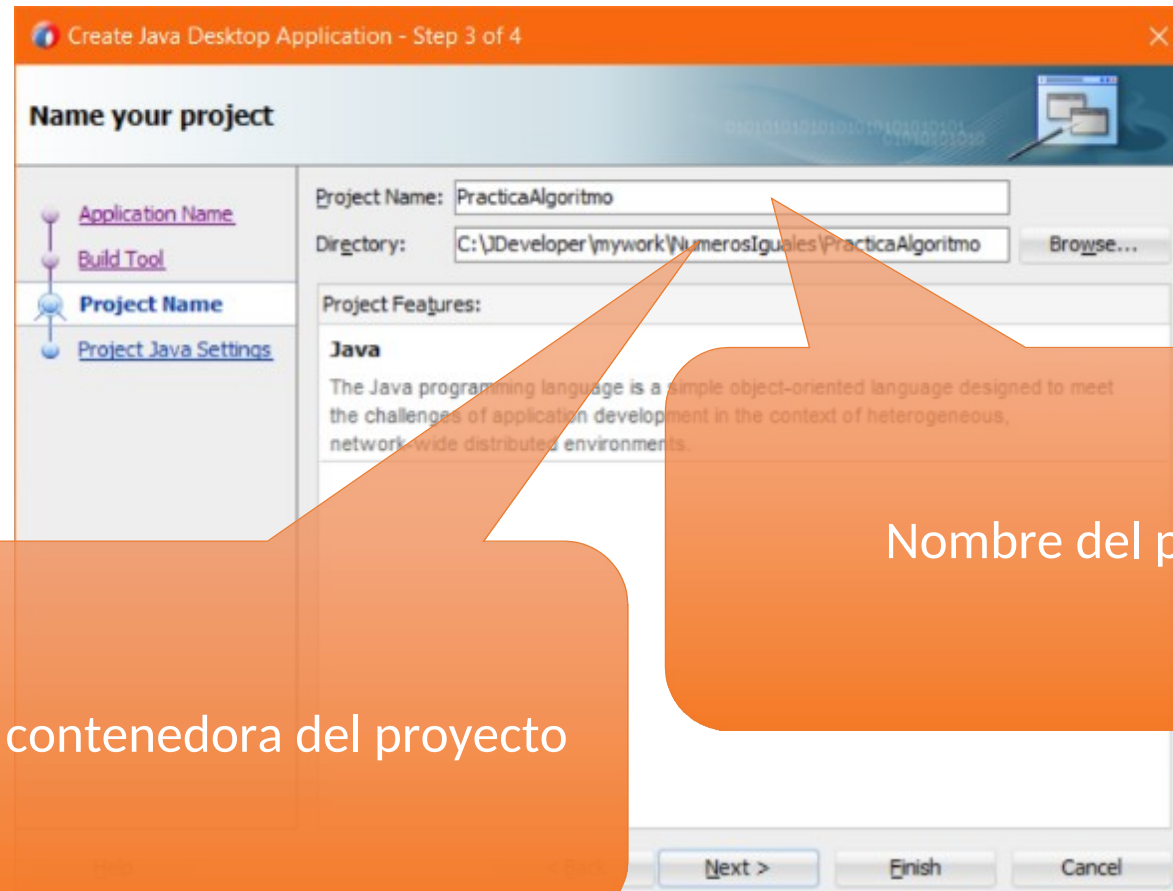
Next >

Escribir el
nombre de la
aplicación

Carpeta
contenedora de
la aplicación

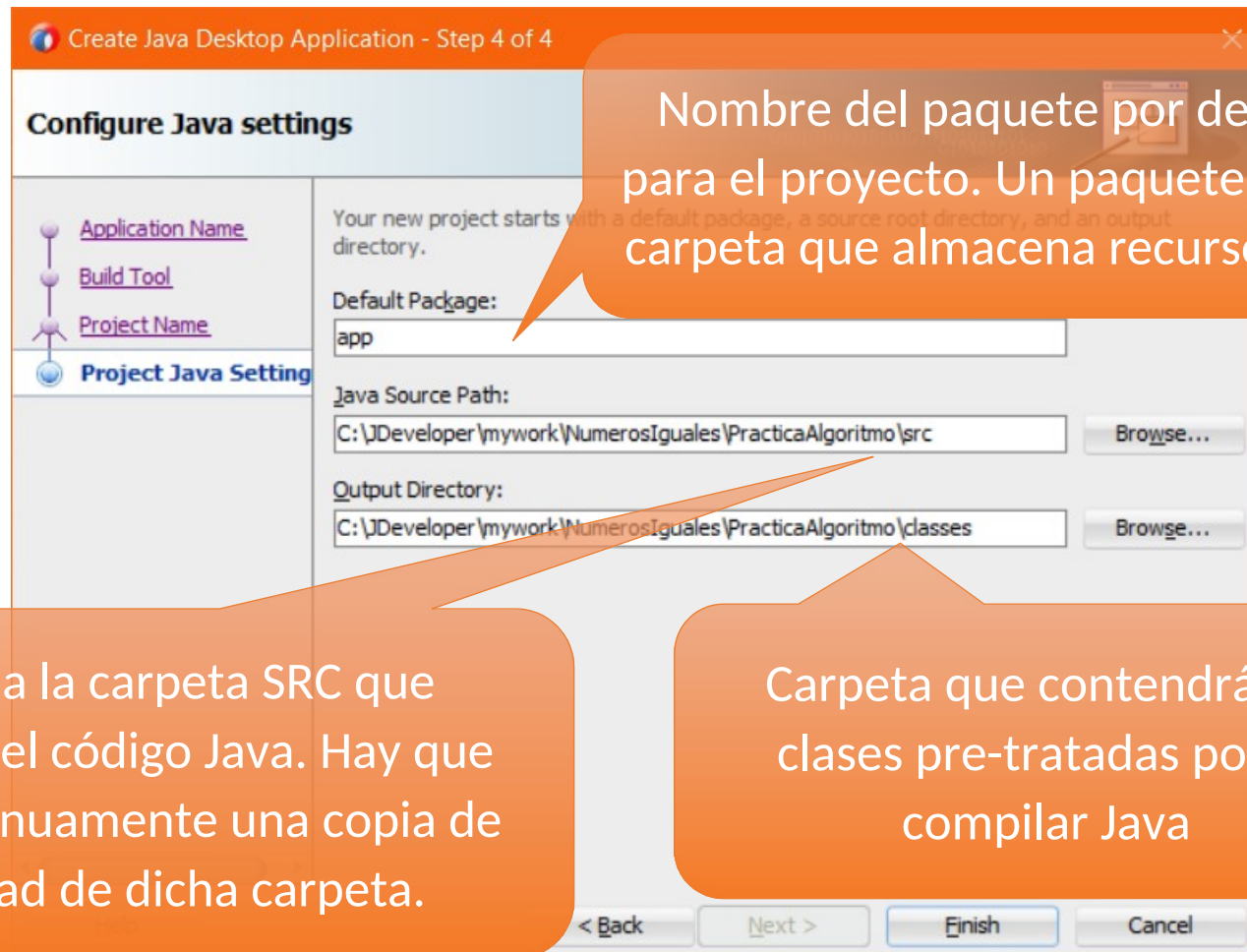
El prefijo de paquete es una carpeta
que se creará de forma previa antes
de crear un paquete, un paquete
también es una carpeta.

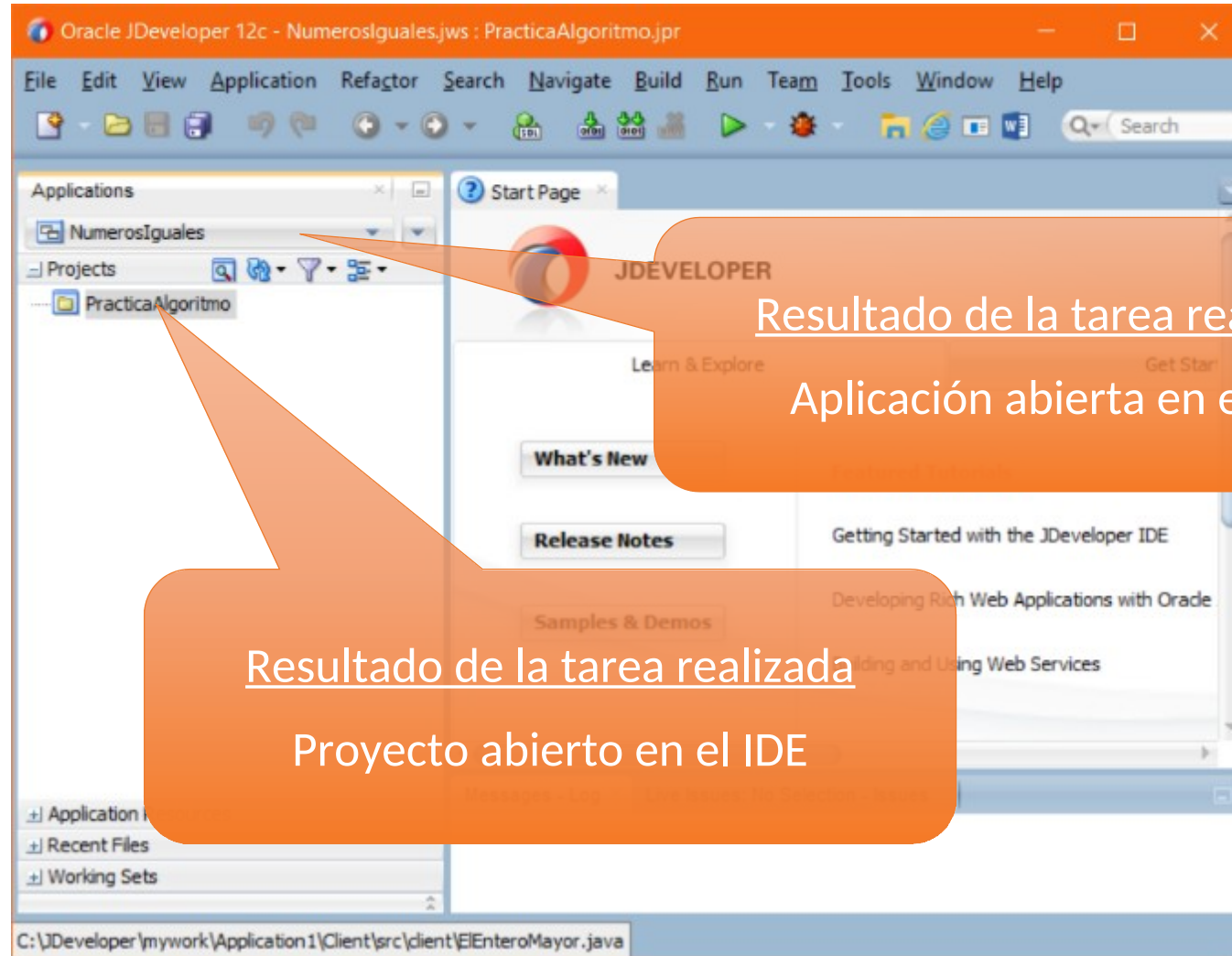


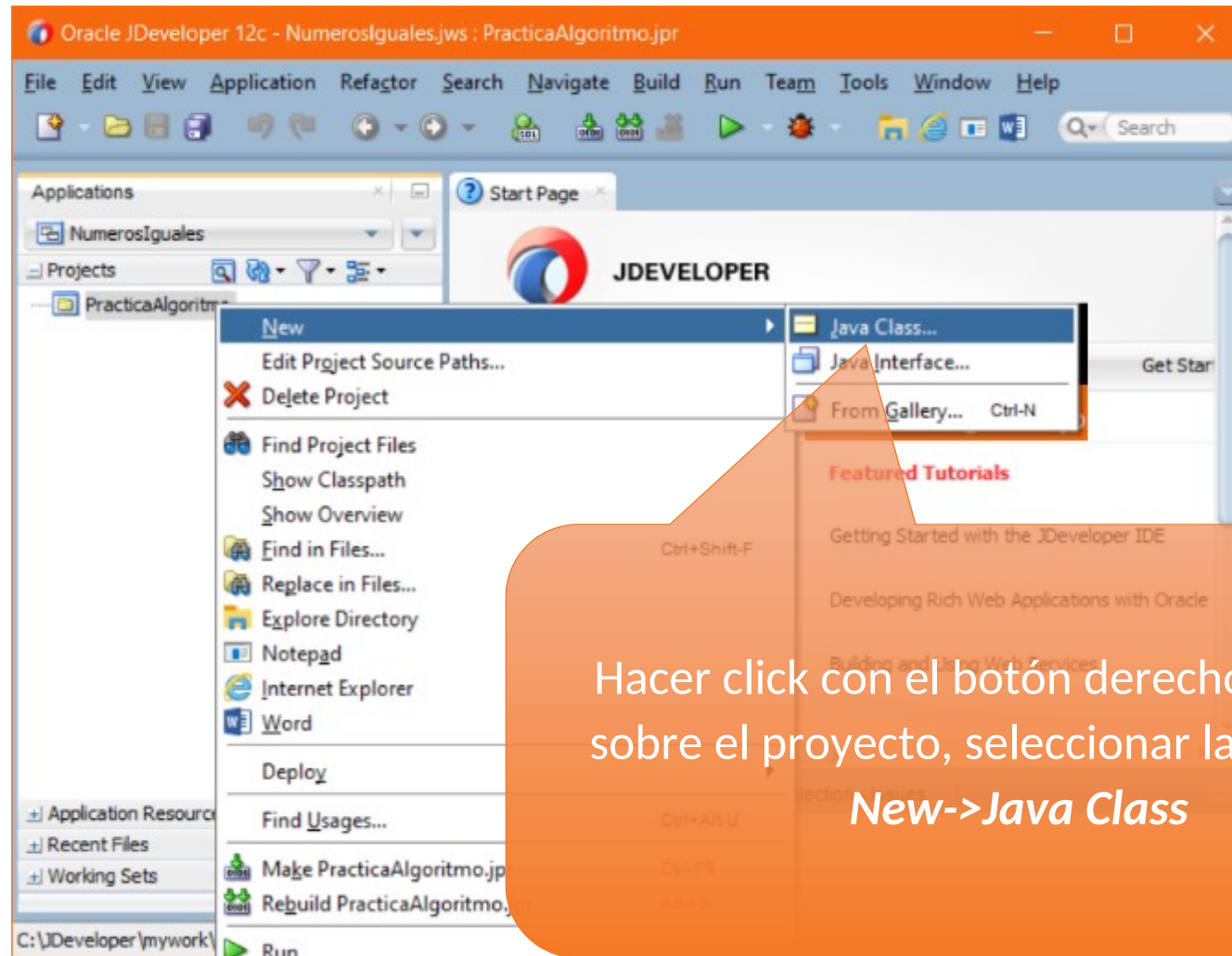


Carpeta contenedora del proyecto

Nombre del proyecto







Create Java Class

Enter the details of your new class.

Name: CualEsMayor

Package: app

Extends: java.lang.Object

Optional Attributes

Implements:

Access Modifiers

☒ public

☐ package private

Other Modifiers

☒ <None>

☐ abstract

☐ final

☒ Constructors from Superclass

☒ Implement Abstract Methods

☒ Main Method

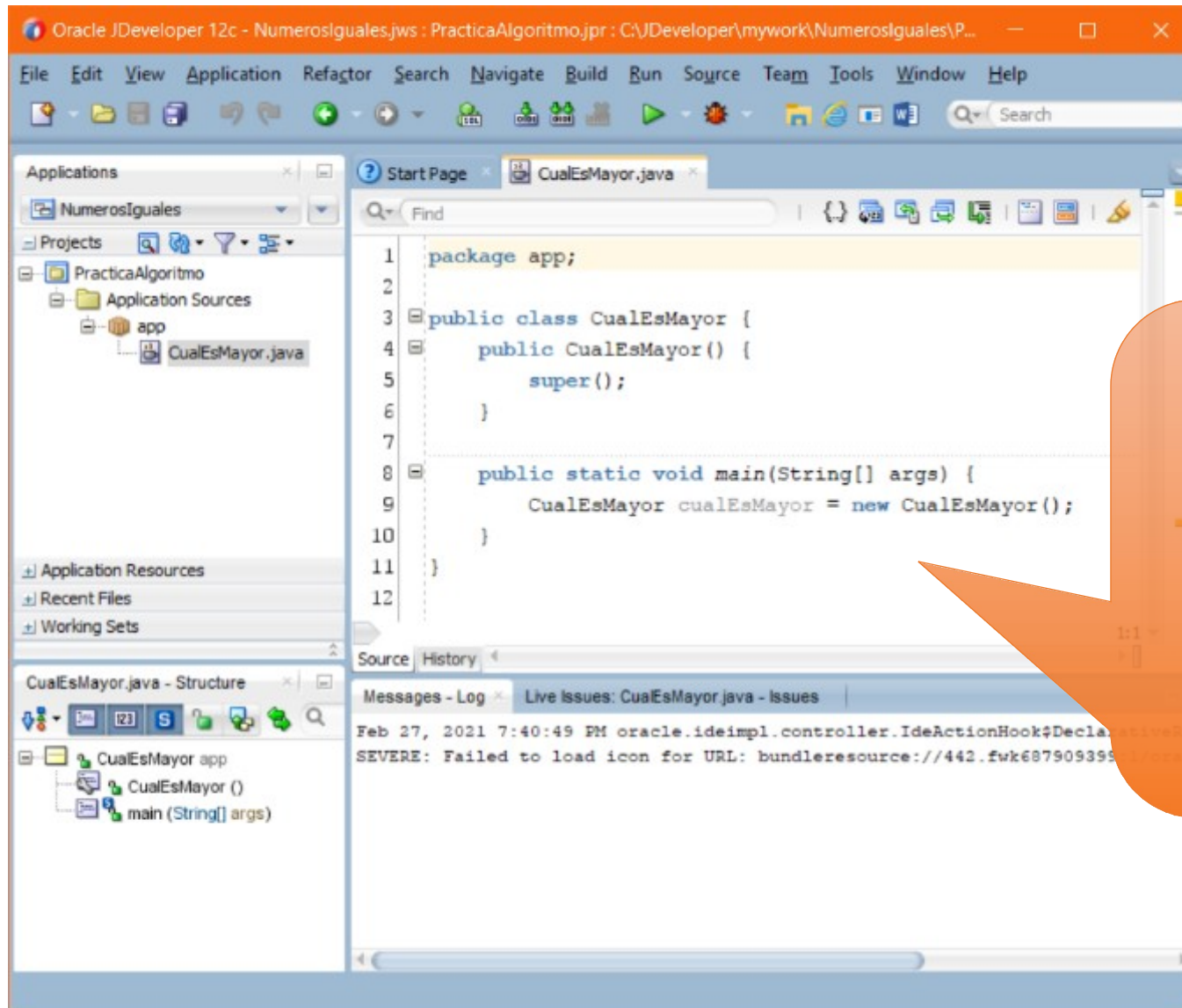
Messages:

Help OK Cancel

Escribir el nombre de la clase. La clase será un archivo en el disco fijo y contendrá el código Java que definirá: El paquete contenedor, la importación de otras clases, el nombre de la clase, atributos y métodos.

Seleccionar que se incluya el método main() al cuerpo de la clase: CualEsMayor{}. Es factible tener varias clase con un método main(). Los métodos públicos son la interfaz de acceso a los atributos privados de la clase.

Programar con Java – Práctica de algoritmos



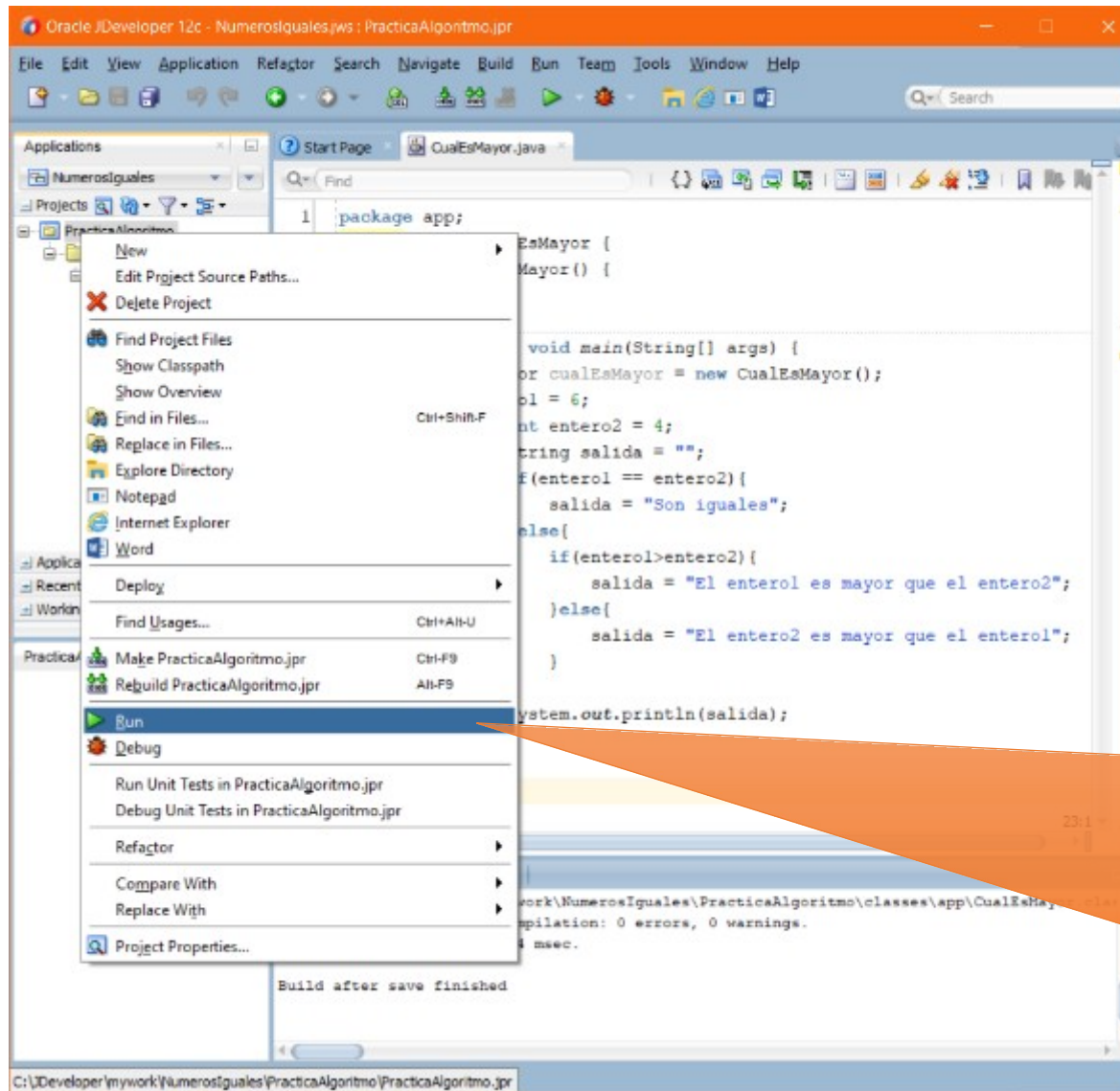
Tarea realizada

Agregar el siguiente código faltante al archivo CualEsMayor.java

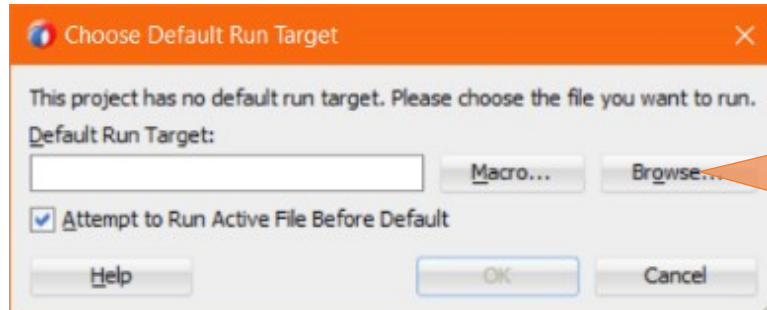
```
package app;

public class CualEsMayor {
    public CualEsMayor() {
        super();
    }

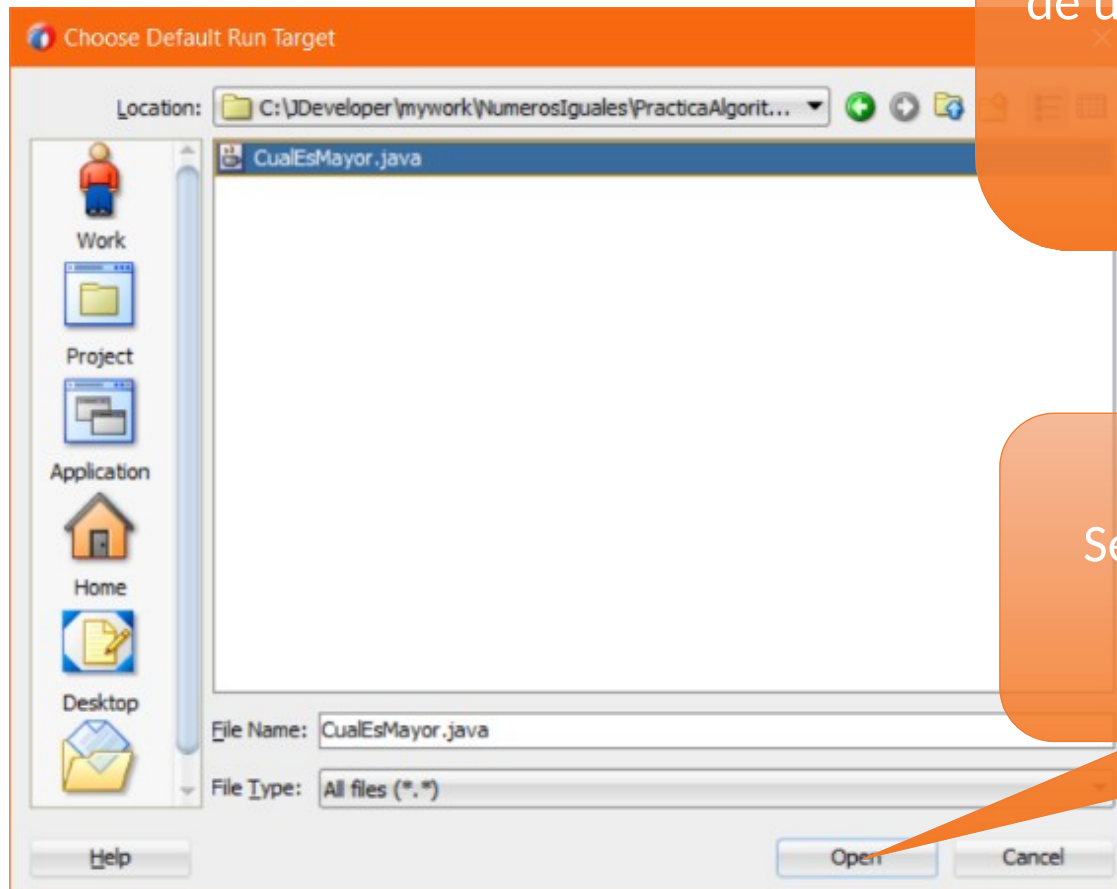
    public static void main(String[] args) {
        //CualEsMayor cualEsMayor = new CualEsMayor();
        int entero1 = 6;
        int entero2 = 4;
        String salida = "";
        if(entero1 == entero2){
            salida = "Son iguales";
        }else{
            if(entero1>entero2){
                salida = "El entero1 es mayor que el entero2";
            }else{
                salida = "El entero2 es mayor que el entero1";
            }
        }
        System.out.println(salida);
    }
}
```



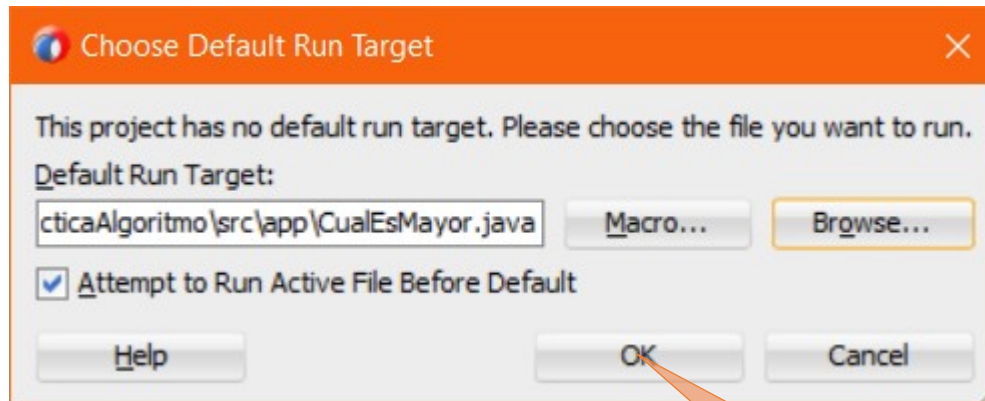
Hacer click con el botón derecho del ratón sobre el proyecto y seleccionar la opción **Run**



Buscar la clase que tiene el método `main()`. Puede haber más de un método `main()` por proyecto Java.

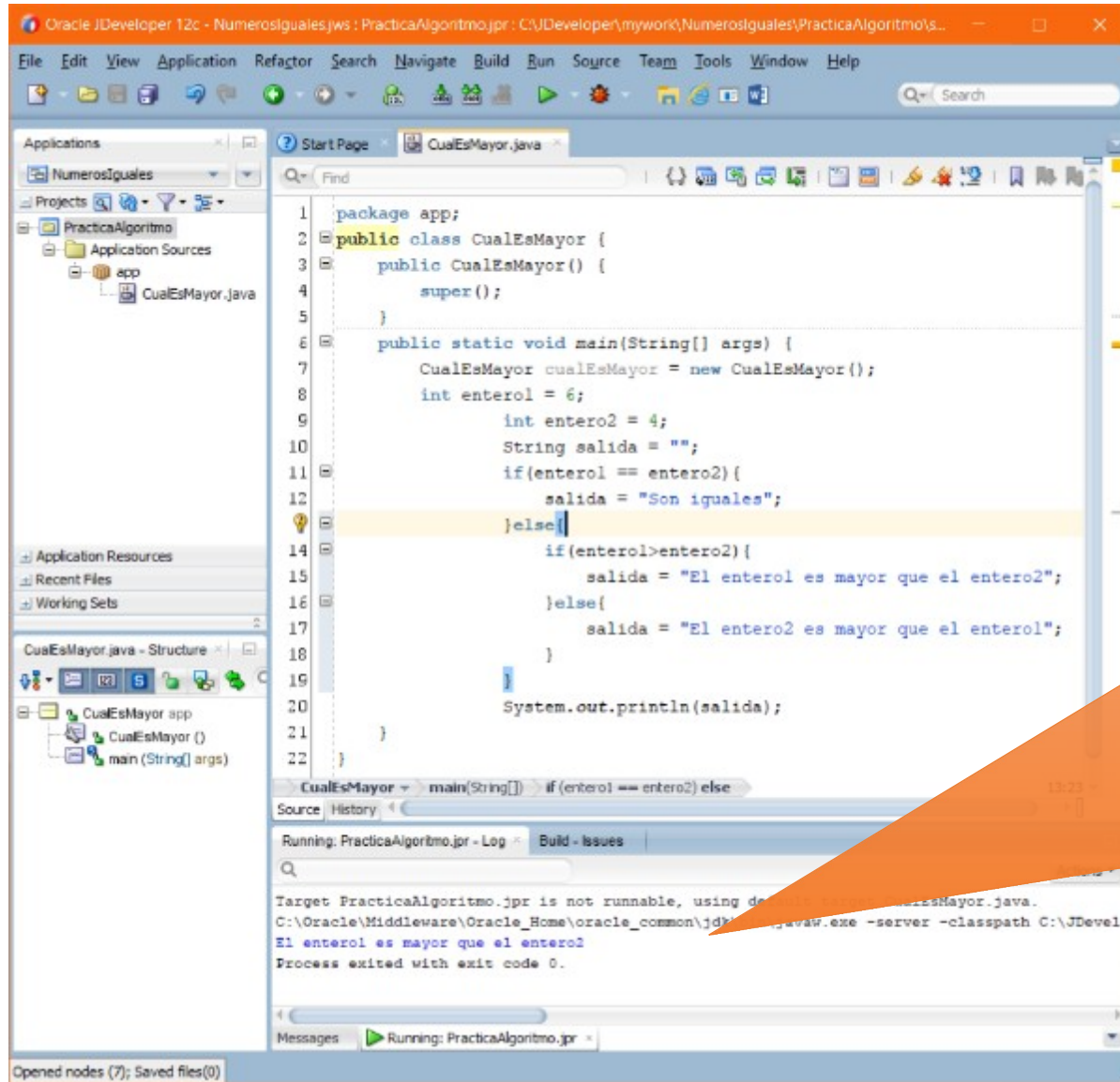


Seleccionar la clase que tiene el método `main()`.



Resultado de la tarea realizada.
Hacer click en OK.

Programar con Java – Práctica de algoritmos



The screenshot displays the Oracle JDeveloper 12c IDE. The main editor window shows the file `CualEsMayor.java` with the following Java code:

```
1 package app;
2 public class CualEsMayor {
3     public CualEsMayor() {
4         super();
5     }
6     public static void main(String[] args) {
7         CualEsMayor cualEsMayor = new CualEsMayor();
8         int entero1 = 6;
9         int entero2 = 4;
10        String salida = "";
11        if(entero1 == entero2){
12            salida = "Son iguales";
13        }else{
14            if(entero1 > entero2){
15                salida = "El entero1 es mayor que el entero2";
16            }else{
17                salida = "El entero2 es mayor que el entero1";
18            }
19        }
20        System.out.println(salida);
21    }
22 }
```

The left sidebar shows the project structure for `PracticaAlgoritmo`, including the `app` package and the `CualEsMayor.java` file. The bottom console window shows the output of the program:

```
Running: PracticaAlgoritmo.jpr - Log
Target PracticaAlgoritmo.jpr is not runnable, using default target.
C:\Oracle\Middleware\Oracle_Home\oracle_common\jdk\bin\java.exe -server -classpath C:\JDevelo
El entero1 es mayor que el entero2
Process exited with exit code 0.
```

Resultado de la tarea realizada.

La línea de código: `int entero1 = 6;` declara una referencia de memoria al valor entero 6. El valor 6 se lo conoce como un valor literal escrito en el código Java.



La etiqueta `entero1` representa un lugar de memoria de cuatro bytes y en ese lugar de memoria está almacenado el número 6.

Un lugar de memoria ocupa 8 bit, que es equivalente a un byte.

En el caso de los enteros en java ocupan cuatro lugares de memoria y se puede almacenar el siguiente rango de números: $2^{32} \Rightarrow 0 \dots 42.949.672.96$. En caso de considerar el signo el rango sería: Valor mínimo $-(2^{31})$, el valor máximo es $(2^{31} - 1)$

Tipos de datos primitivos en Java

Tipos de datos primitivos:

- byte
- short
- int
- long
- float
- double
- boolean
- char

Descripción de cada tipo de dato primitivo

- byte: Representa un tipo de dato de 8 bits con signo. De tal manera que puede almacenar los valores numéricos de -128 a 127 (ambos inclusive).
- short: Representa un tipo de dato de 16 bits con signo. De esta manera almacena valores numéricos de -32.768 a 32.767.
- int: Es un tipo de dato de 32 bits con signo para almacenar valores numéricos. Valor mínimo (-2^{31}), el valor máximo es ($2^{31} - 1$).
- long: Es un tipo de dato de 64 bits con signo que almacena valores numéricos entre -2^{63} a $2^{63} - 1$
- float: Es un tipo dato para almacenar números en coma flotante con precisión simple de 32 bits.
- double: Es un tipo de dato para almacenar números en coma flotante con doble precisión de 64 bits.
- boolean: Sirve para definir tipos de datos booleanos. Es decir, aquellos que tienen un valor de true o false. Ocupa 1 bit de información.
- char: Es un tipo de datos que representa a un carácter Unicode sencillo de 16 bits.

Nota:

Java no tiene un tipo de dato primitivo para cadenas de caracteres. Java tiene una clase especial llamada String para el manejo de cadenas.