

Java — Guía esencial para empezar

1) Preparación rápida

- 1) Instalá el JDK (Java 21).
- 2) Verificá en consola que tenés Java y el compilador disponibles:

```
java -version  
javac -version
```

2) Buenas prácticas básicas (antes de escribir código)

- Nombres en camelCase para variables y métodos: precioFinal, cantidadAlumnos.
- Tipos correctos: int para contadores/índices; long para valores grandes; double para decimales.
- Si usás Scanner recordá limpiar el salto con nextLine() al mezclar lecturas.

3) Tu primer programa (línea de comandos)

Estructura mínima de archivos:

```
hola/  
└─ Hola.java
```

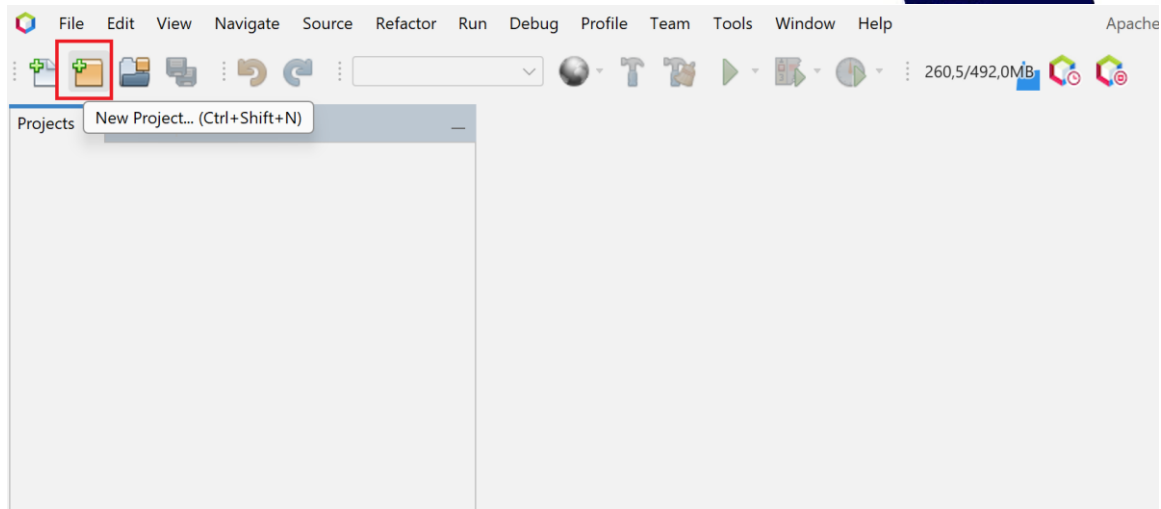
Contenido del archivo Hola.java:

```
public class Hola {  
    public static void main(String[] args) {  
        System.out.println("¡Hola, Java!");  
    }  
}
```

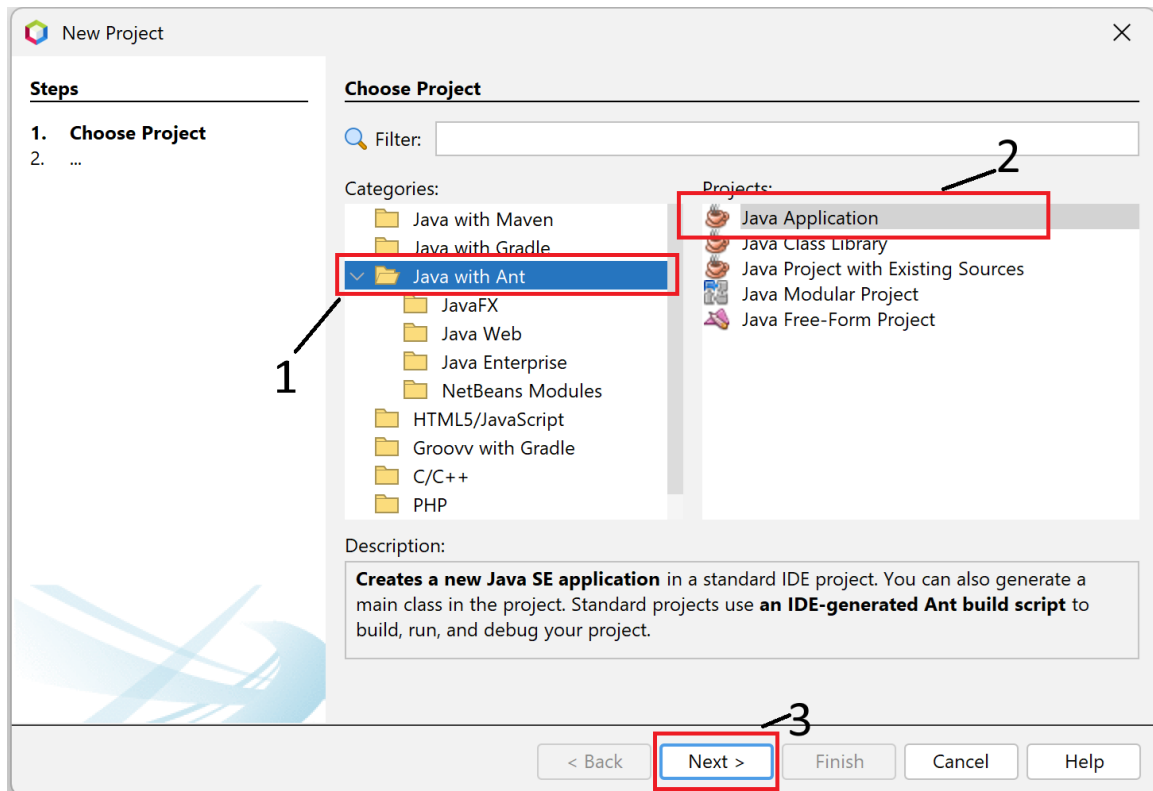
Compilar y ejecutar.

Paso a Paso en Netbeans

Paso 1: Generar un nuevo proyecto



En la ventana emergente seleccionamos Java with Ant->Java Application-> apretamos botón “next”.

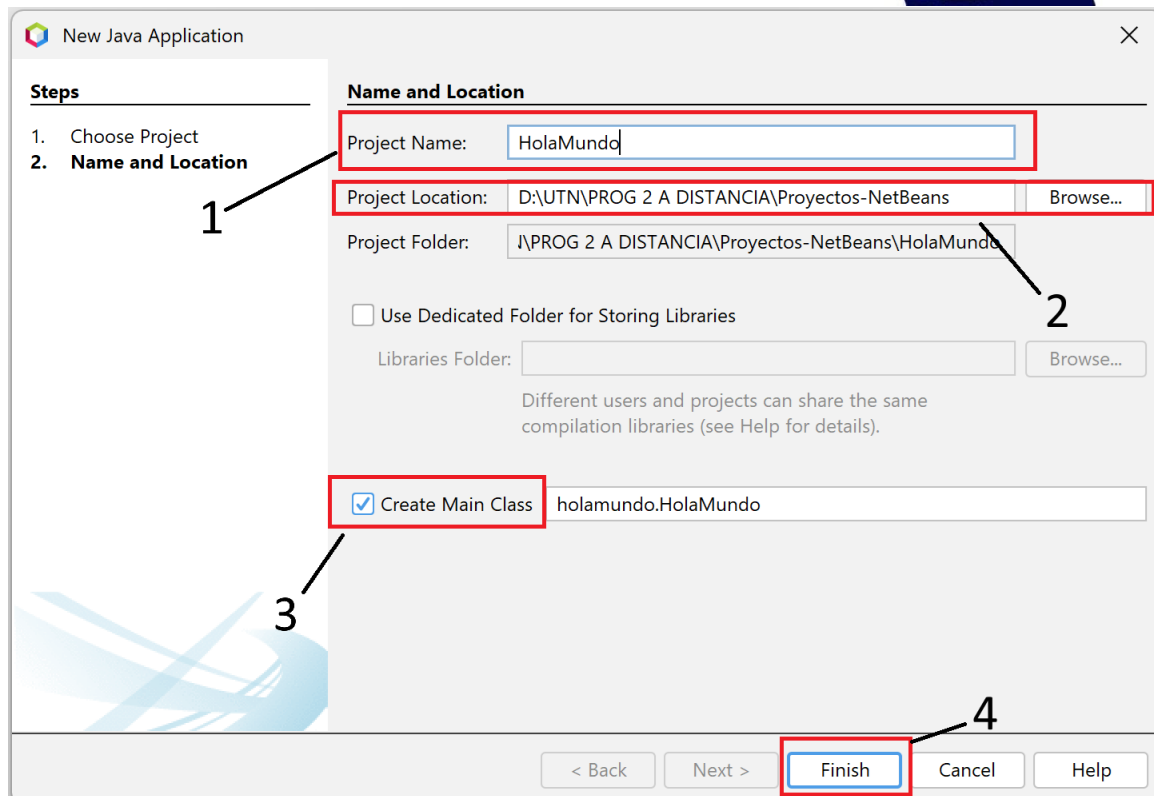


Paso 1: Definimos el nombre del proyecto, por convención se inicia en Mayuscula y se sigue con mayuscula cuando cambia de palabra.

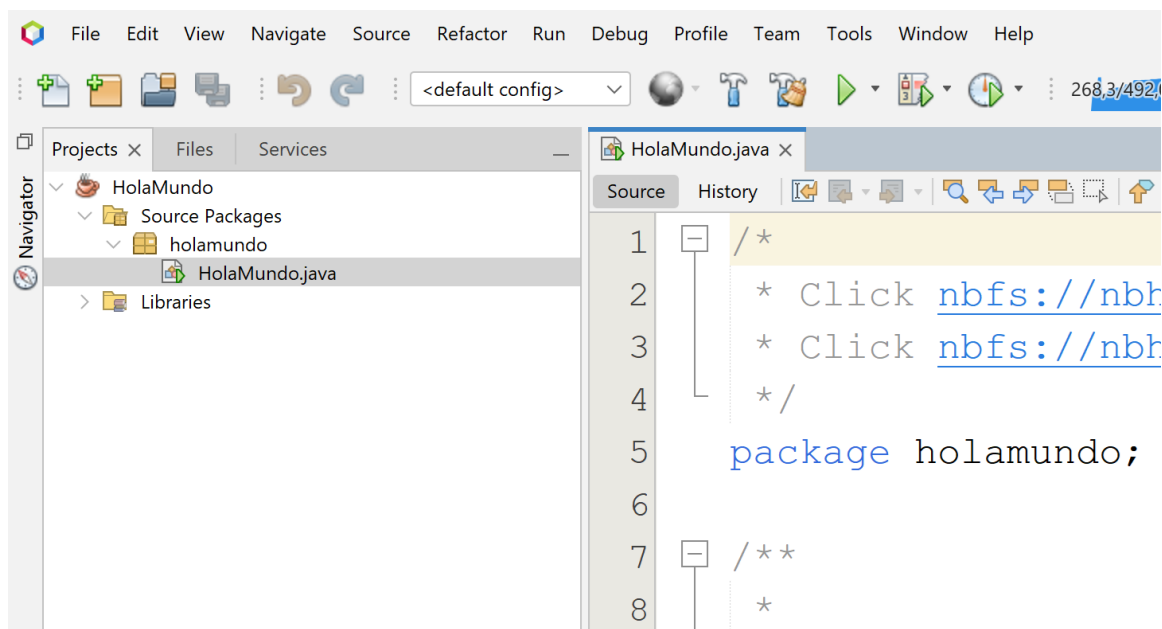
Paso 2: Verificamos en que carpeta se creara nuestro proyecto

Paso 3: verificamos que este marcada la creación de la clase Main que será el punto de entrada de nuestra app.

Paso 4: finalizamos la creación del proyecto.



Así se ve la estructura de nuestro proyecto iniciado.



Así se genera nuestra clase main:

El package indica donde se encuentra nuestra clase main que es la clase de inicio de nuestro programa.

Luego esta la clase Principal que se llama HolaMundo, y dentro de la misma esta nuestro método main que es el punto de entrada para que nuestro programa se ejecute.

```
package holamundo;

/**
 *
 * @author enfer
 */
public class HolaMundo {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    }

}
```

Ahora estamos listos para programar.

4) Variables y tipos primitivos (con rangos y valores por defecto)

Los valores por defecto aplican a campos (atributos) de clase/instancia. Las variables locales deben inicializarse.

Tipo	Bits	Rango / valores	Valor por defecto	Notas / Literales
byte	8	-128 a 127	0	Byte.MIN_VALUE / MAX_VALUE
short	16	-32 768 a 32 767	0	Short.MIN_VALUE / MAX_VALUE
int	32	-2 147 483 648 a 2 147 483 647	0	Literal: 10, 0b1010, 0xFF; separadores: 1_000_000
long	64	-9 223 372 036 854 775 808 a 9 223 372 036 854 775 807	0L	Sufijo L: 3_000_000_000L
float	32	$\approx \pm 1.4E-45$ a $\pm 3.4028235E38$	0.0f	Sufijo f; precisión ~7 dígitos; Float.MIN_VALUE = mínimo positivo
double	64	$\approx \pm 4.9E-324$ a $\pm 1.7976931348623157E308$	0.0d	Por defecto en decimales; precisión ~15-16;

				Double.MIN_VALUE = mínimo positivo
boolean	—	true / false	false	Sin rango numérico
char	16	'\u0000' (0) a '\uffff' (65535)	'\u0000'	Unicode; literales: 'A', '\n', '\u00F1'

Ejemplos rápidos:

```
int edad = 18;
long poblacion = 47_000_000L;
double precio = 99.99; // double por defecto
float tasa = 3.5f; // float requiere sufijo f
boolean activo = true;
char inicial = 'L';
```

5) Imprimir por consola — explicativo y práctico

La salida estándar es System.out (PrintStream). Opciones más usadas: print, println y printf (formateado).

```
System.out.print("sin salto");
System.out.println("con salto");
int edad = 18;
double precio = 99.987;
System.out.printf("Edad: %d, precio: %.2f\n", edad, precio);
// %d entero, %f decimal (%.2f → 2 decimales), %s texto, %n salto
portable
// Ejemplo de alineación:
System.out.printf("%-10s %8.2f\n", "Total:", 1234.5);
```

Cuándo usar cada uno: print/println para mensajes simples; printf para columnas, dinero y reportes con formato.

6) Leer desde consola — con Scanner y buenas prácticas

System.in es la entrada estándar (bytes). Scanner la convierte en tokens (palabras/números) o líneas completas, recordá limpiar el salto cuando combines lecturas.

```
import java.util.Scanner;
import java.util.Locale;

public class LeerDemo {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        sc.useLocale(Locale.US); // 12.5 con punto

        System.out.print("Edad (int): ");
```

```
int edad = sc.nextInt(); // lee token numérico
sc.nextLine();           // limpia el salto pendiente

System.out.print("Nombre (línea completa): ");
String nombre = sc.nextLine();

System.out.print("Precio (double, 12.5): ");
double precio = sc.nextDouble();

System.out.printf("Hola %s. Edad: %d, Precio: %.2f%n", nombre,
edad, precio);
    }
}
```

Patrón robusto: leer como texto y parsear (mejor manejo de errores).

```
String linea = sc.nextLine();

int n = Integer.parseInt(linea.trim());
```

Leer un carácter (no hay nextChar):

```
char opcion = sc.nextLine().trim().charAt(0);
```

7) Casteos y conversiones

Ampliación (widening) es implícita y segura; reducción (narrowing) requiere cast y puede perder información.

```
int i = 42;
double d = i;           // widening implícito

double x = 42.9;
int j = (int) x;        // narrowing: j = 42 (trunca)

long grande = 3_000_000_000L;
int chico = (int) grande; // peligro: overflow

char c = 'A';
int codigo = c;          // 65
char otra = (char) 66;   // 'B'
```

Conversiones entre String y números:

```
int n = Integer.parseInt("250");
double z = Double.parseDouble("3.14");
```

```
String s1 = String.valueOf(n);    // "250"  
String s2 = Double.toString(2.5); // "2.5"
```

8) Programa ejemplo integrador (aplicando buenas prácticas)

Usa constantes, nombres claros, Scanner, printf y un casteo controlado.

```
import java.util.Scanner;  
import java.util.Locale;  
  
public class EjemploIntegrador {  
    private static final double CIEN = 100.0;  
  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in).useLocale(Locale.US);  
  
        System.out.print("Precio base (double, ej. 199.99): ");  
        double precioBase = Double.parseDouble(sc.nextLine().trim());  
  
        System.out.print("Descuento en % (int, ej. 15): ");  
        int descuento = Integer.parseInt(sc.nextLine().trim());  
  
        double precioConDesc = precioBase * (1 - descuento / CIEN);  
  
        int precioEntero = (int) precioConDesc; // demostrativo: trunca  
        decimales  
  
        System.out.printf("Precio final: %.2f (entero: %d)%n",  
            precioConDesc, precioEntero);  
    }  
}
```