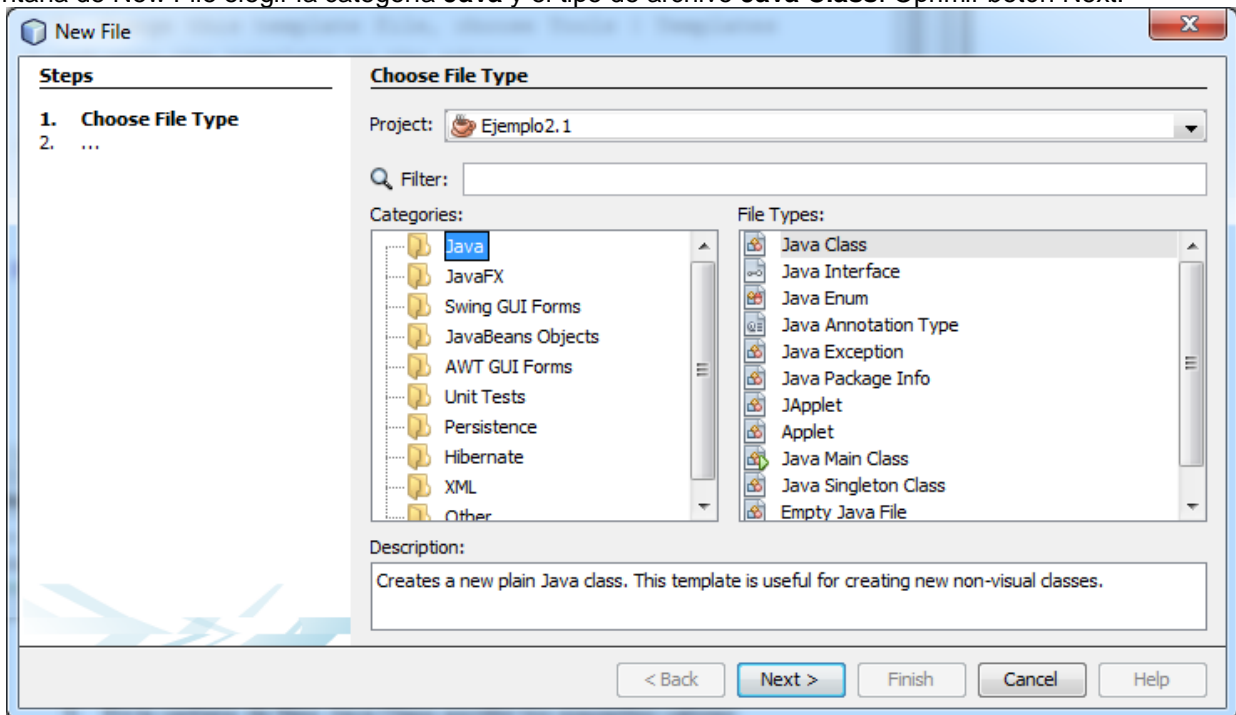
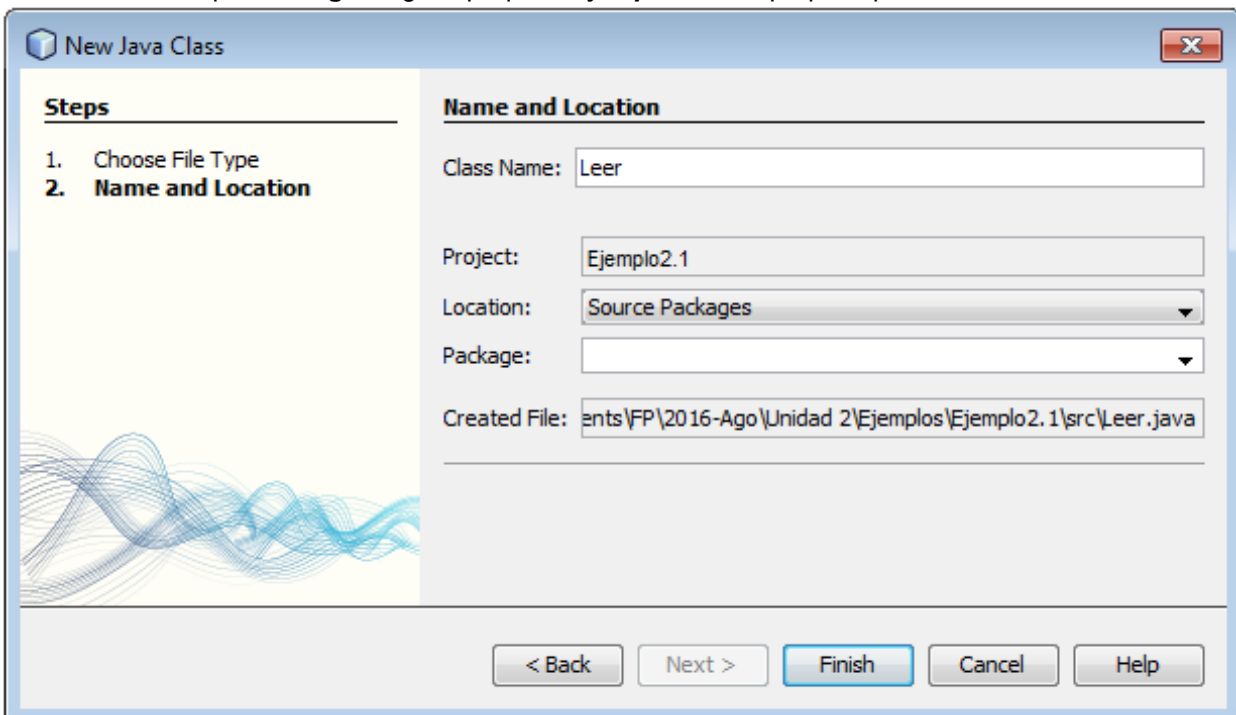


Cómo utilizar la clase Leer en Java

1. Abrir el programa NetBeans
2. Abrir el proyecto “Ejemplo2.1” en **File/ Open Project**
3. Crear un nuevo archivo en **File/ New File**
4. En la ventana de New File elegir la categoría **Java** y el tipo de archivo **Java Class**. Oprimir botón Next.



5. En la ventana de New Java Class escribir los siguientes valores:
 - En el campo **Class Name** escribir **Leer** (que será el nombre de la clase)
 - En el campo **Package** elegir el paquete **Ejemplo2.1** o el paquete por default





6. Al abrirse la ventana para escribir el código de la clase Leer, borrar el código que aparece después de la zona del autor
7. Abre el archivo `Clase_Leer.txt`, entonces copia todo el código que viene dentro del archivo y pégalo como código de la clase Leer.
Si no tienes el archivo `Clase_Leer.txt`, copia el siguiente código (desde import hasta el final) y págalo en el código de la clase Leer.

Código en JAVA de la clase Leer

```
import java.io.*;
public class Leer {

    public static String datoString()
    {
        String sdato = "";
        try
        {
            // Definir un flujo de caracteres de entrada: flujoE
            InputStreamReader isr = new InputStreamReader(System.in);
            BufferedReader flujoE = new BufferedReader(isr);
            // Leer. La entrada finaliza al pulsar la tecla Entrar
            sdato = flujoE.readLine();
        }
        catch(IOException e)
        {
            System.err.println("Error: " + e.getMessage());
        }
        return sdato; // devolver el dato tecleado
    }

    public static short datoShort()
    {
        try
        {
            return Short.parseShort(datoString());
        }
        catch(NumberFormatException e)
        {
            return Short.MIN_VALUE; // valor más pequeño
        }
    }

    public static int datoInt()
    {
        try
        {
            return Integer.parseInt(datoString());
        }
        catch(NumberFormatException e)
        {
            return Integer.MIN_VALUE; // valor más pequeño
        }
    }

    public static long datoLong()
    {
        try
        {
            return Long.parseLong(datoString());
        }
    }
}
```



```

    }
    catch(NumberFormatException e)
    {
        return Long.MIN_VALUE; // valor más pequeño
    }
}

public static float datoFloat()
{
    try
    {
        Float f = new Float(datoString());
        return f.floatValue();
    }
    catch(NumberFormatException e)
    {
        return Float.NaN; // No es un Número; valor float.
    }
}

public static double datoDouble()
{
    try
    {
        Double d = new Double(datoString());
        return d.doubleValue();
    }
    catch(NumberFormatException e)
    {
        return Double.NaN; // No es un Número; valor double.
    }
}
}

```

8. Ahora cambia a la ventana del programa principal del Ejemplo2.1 y borra el código donde se le asigna el valor a la variable base
9. A la variable **base** se le asignará la clase **Leer** . y se elegirá el método **datoInt ()**

```

14  * @param args the command line arguments
15  */
16  public static void main(String[] args) {
17      // TODO code application logic here
18      int base, altura, perimetro, superficie;
19      System.out.println("Introduce la base: ");
20      base = Leer.
21          * datoDouble() double
22          * datoFloat() float
23          * datoInt() int
24          * datoLong() long
25          * datoShort() short
26          * datoString() String
27          * class
28  }

```

Observa que al escribir **Leer.** se abre la lista de los métodos declarados en la clase Leer

Los métodos deben coincidir con los tipos de datos declarados para las variables

10. Asegúrate que el código del programa sea el siguiente:

```
public class Ejemplo21 {  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
        // TODO code application logic here  
        int base, altura, perimetro, superficie;  
  
        System.out.println("Introduce la base: ");  
        base = Leer.datoInt();  
        System.out.println("Introduce la altura: ");  
        altura = Leer.datoInt();  
  
        perimetro = 2 * (base + altura);  
        superficie = base * altura;  
  
        System.out.println("El perímetro es: " + perimetro);  
        System.out.println("La superficie es: " + superficie);  
    }  
}
```

11. Para ejecutar el programa oprime las teclas **Shift + F6**

Como se observa, solo es necesario hacer uso de la función de acuerdo al tipo de dato utilizado. La sintaxis es:

`Variable = nombre_de_la_clase.nombre_del_método();`

Por ejemplo: `altura = Leer.datoInt();`

Donde: `altura` es la variable

`Leer` es el nombre de la clase

`datoInt()` es el nombre del método, que está dentro de la clase

Los métodos que definimos en la clase Leer son:

Métodos para datos enteros:

- **datoShort()** cuando se usan variables de tipo **short** (-32,768 a 32,767)
- **datoInt()** cuando se usan variables de tipo **int** (-2,147,483,648 a 2,147,483,647)
- **datoLong()** cuando se usan variables de tipo **long** (-9,223,372,036,854,775,808 a 9,223,372,036,854,775,807)

Métodos para datos reales:

- **datoFloat()** cuando se usan variables de tipo **float** (32 bits de long. y precisión aprox. de 6 dígitos)
- **datoDouble()** cuando se usan variables de tipo **Double** (64 bits de long. y precisión aprox. de 16 dígitos)

Métodos para datos alfanuméricos:

- **datoString()** cuando se usan variables de tipo **String** (cadena de caracteres)