

BlueJ 3.0.4

Tutorial Academico

Parte I



Programacion Orientada a objetos
FaCENA-UNNE

Tabla de contenido

| | | |
|----------|--|-----------|
| 1 | Introducción | 2 |
| 1.1 | Alcance del tutorial | 2 |
| 1.2 | Licencia y Distribución | 2 |
| 1.3 | Sistemas Operativos Compatibles..... | 2 |
| 1.3.1 | Requisitos:..... | 2 |
| 2 | Instalación en Windows | 3 |
| 3 | Instalación en Ubuntu..... | 6 |
| 4 | Primeros Pasos con BlueJ..... | 9 |
| 4.1 | Creación de un proyecto | 9 |
| 4.2 | Creación de Clases | 10 |
| 4.3 | Creación de Clases de Prueba – Ingreso de datos | 13 |
| 4.3.1 | Como constantes | 13 |
| 4.3.2 | Como argumentos del método main..... | 14 |
| 4.3.3 | Por Teclado – Usando la clase Scanner | 15 |
| 5 | Errores de compilación | 17 |
| 6 | Visor de navegación (Navigation View) | 17 |
| 7 | Como generar la documentación de un proyecto..... | 18 |

1 Introducción

Este tutorial es una introducción al entorno de desarrollo BlueJ 3.0.4 basado en una revisión práctica del uso del mismo.

BlueJ es un entorno de desarrollo Java, diseñado **específicamente para la enseñanza** a un nivel introductorio, por el equipo BlueJ en la Universidad de Kent, en Canterbury, Reino Unido y en la Universidad de Deakin, Melbourne, Australia.

En <http://www.bluej.org> hay disponible información sobre versiones anteriores de BlueJ.

1.1 Alcance del tutorial

Este tutorial esta orientado a alumnos de la materia Programación Orientada a Objetos de la Licenciatura en Sistemas de Información (FaCENA-UNNE), con el propósito de familiarizarse con el entorno, el lenguaje y, fundamentalmente, con el paradigma de objetos. No está pensado para enseñar Java. Se recomienda a los alumnos para profundizar su aprendizaje del lenguaje la lectura del material Anexo proporcionado por la cátedra, y la consulta a libros (Por ej: “Java: Como Programar”, Deitel&Deitel).

Este no es un manual de referencia completo del entorno ya que solamente queremos dar una introducción breve y concisa. Para obtener una referencia más detallada, ver el “*The BlueJ Environment Reference Manual*”, disponible en el sitio Web de BlueJ (www.bluej.org).

1.2 Licencia y Distribución

BlueJ esta disponible gratuitamente en la pagina www.bluej.org, para cualquier sistema y para cualquier tipo de uso y redistribución no comercial.

BlueJ no persigue fines de lucro, ya que su ideología principal es el de usarlo con fines educativos. El Copyright (derecho de autor) es de M. Kölling y J. Rosenberg, sus autores.

1.3 Sistemas Operativos Compatibles

BlueJ se distribuye en cuatro tipos de formatos:

- Para Windows (2000, XP, Vista o Seven),
- Para MacOS,
- Para Debian, Ubuntu;
- Y otro para los demás sistemas.

La instalación es sencilla. En este tutorial mostraremos para Windows y Ubuntu.

1.3.1 Requisitos:

En general se debe tener J2SE v1.6 (JDK 1.6) o posterior instalado en el sistema para utilizar BlueJ. En general, es recomendable actualizar a la última versión estable (no beta) de Java. Si no tiene el JDK instalado puede descargarlo del sitio Web de Sun en <http://java.sun.com> Seleccionar el SO deseado (en primer lugar elegiremos Windows):

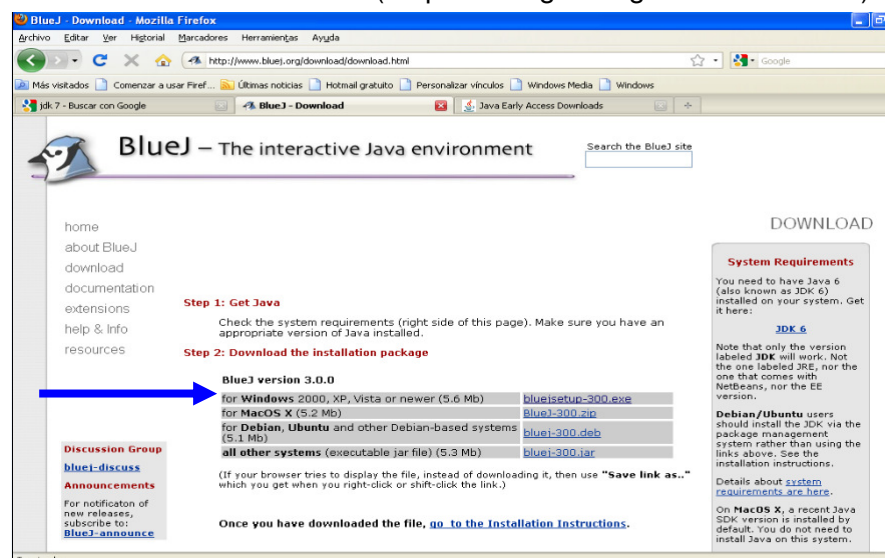


Imagen 1. Ventana de descarga del entorno BlueJ.

Una vez descargado el entorno BlueJ y el correspondiente JDK, procederemos a explicar paso a paso la instalación, manejo y uso del entorno BlueJ, en primer lugar para el Sistema Operativo Windows (2000, XP, Vista o Seven).

2 Instalación en Windows

Hacemos doble click en *bluejsetup-300.exe* y aparecerá una ventana similar a la siguiente:

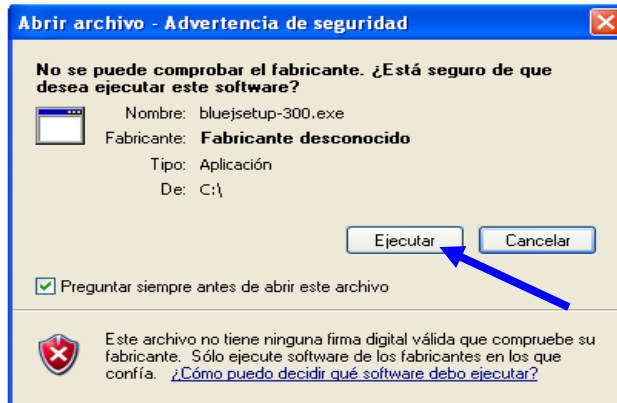


Imagen 2. ventana de ejecución de *bluejsetup-3.0.0.exe*

Hacer click en **Ejecutar** y nos abrirá una ventana similar al siguiente:

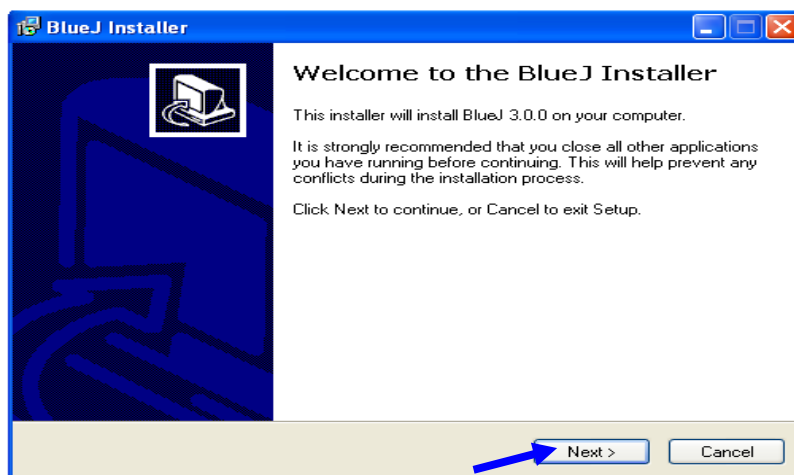


Imagen 3. Primeros pasos en la instalación.

Presionar **Next>** y permitirá seleccionar el lugar de instalación. Por defecto **C:\BlueJ**

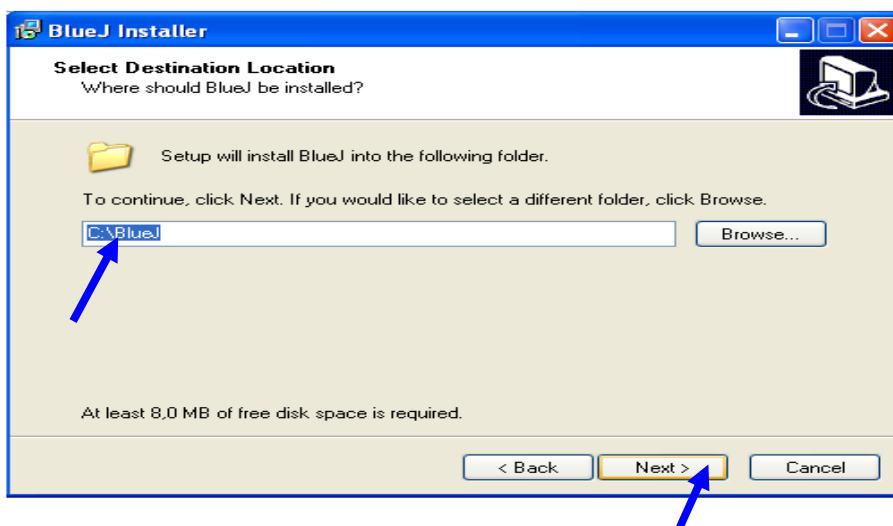


Imagen 4. Selección de la carpeta donde se instalará BlueJ (por defecto es en C).

Seleccionar la opción de agregar acceso y establecer la asociación con la extensión *.BlueJ.

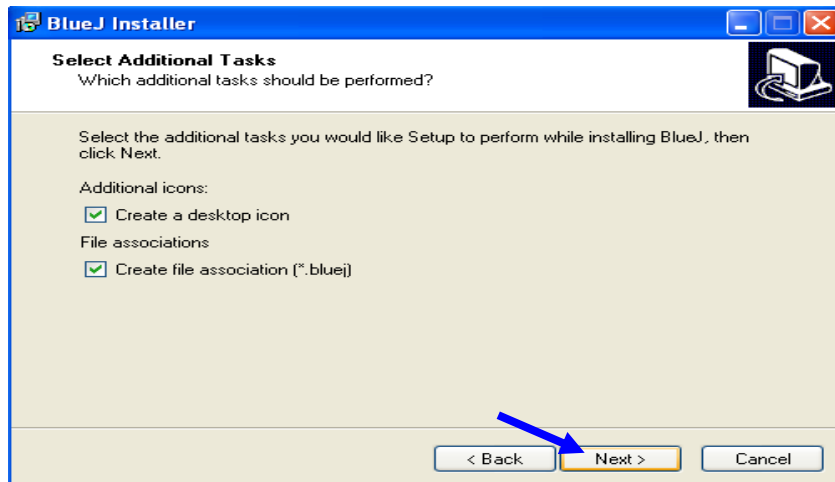


Imagen 5. Creación de icono de inicio y asociación con extensión *.bluej

La siguiente imagen indica que fue creado el icono de inicio que aparecerá en el escritorio y también brinda la localización de la carpeta en la cual se encuentra instalado BlueJ.

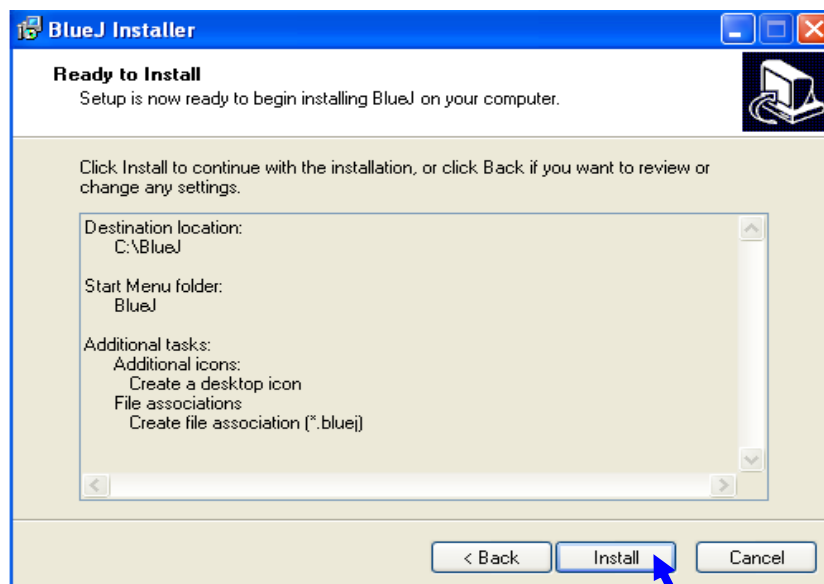
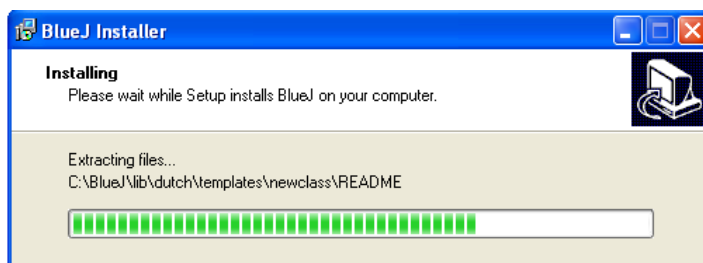


Imagen 6. Información sobre la instalación.

Click en **Install** y la barra indicará el avance de la instalación.



Completada la instalación en forma satisfactoria, la siguiente ventana presenta dos opciones, las cuales pueden o no ser seleccionadas. La primera de ellas abre un texto que brinda información sobre el entorno instalado recientemente.

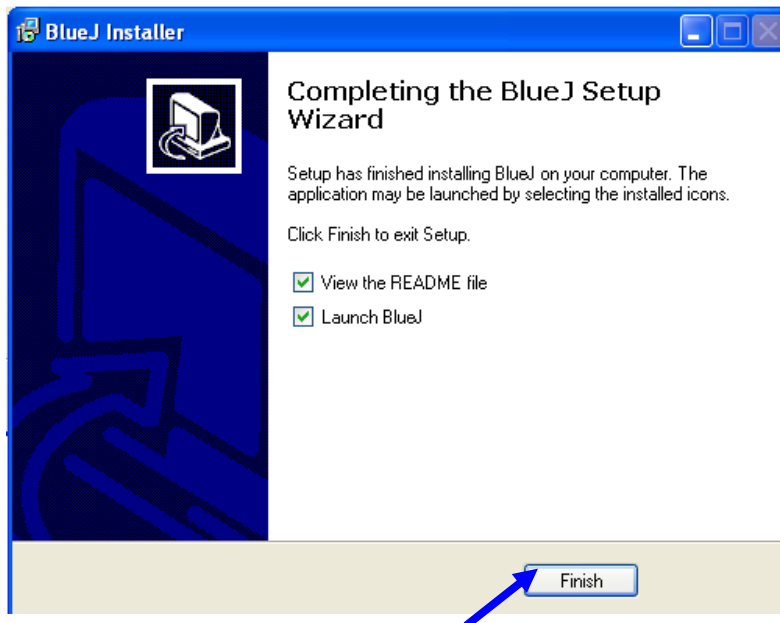
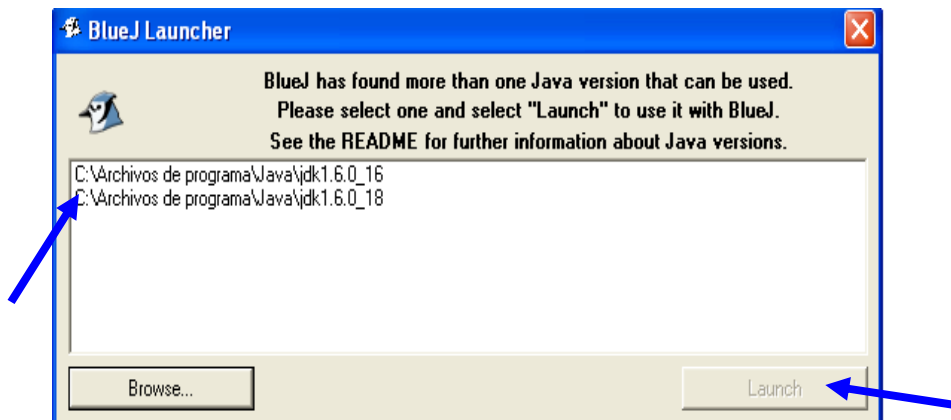


Imagen 7. Instalación realizada en forma satisfactoria.

La segunda opción permite, una vez presionado **Finish**, la aparición de la siguiente ventana:



Donde se puede elegir alguna de las opciones del jdk que tengamos o hayamos instalado. Luego habilitará el botón **Launch**, **que permite** empezar a usar el entorno de BlueJ 3.0.0



Luego de esta presentación aparecerá el entorno de uso, que será explicado en detalle más adelante.

3 Instalación en Ubuntu

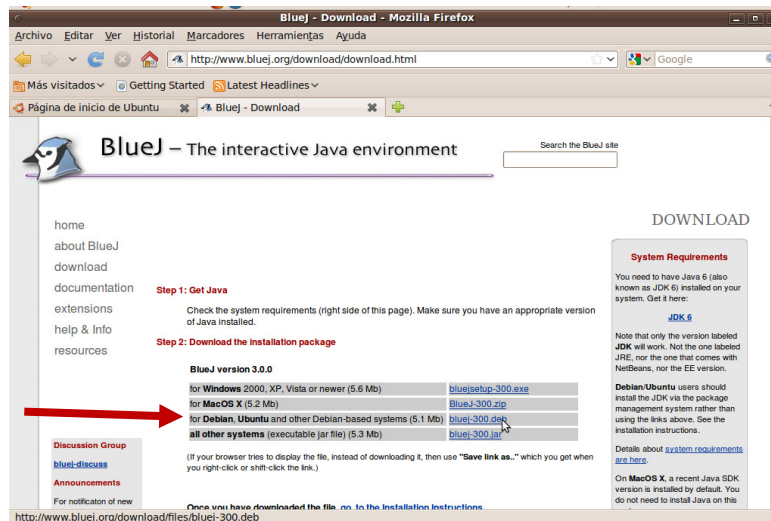
Para instalar BlueJ en Ubuntu en primer lugar se debe verificar si está instalado el *jdk* (*Java Development Kit*). Para ello, ir a **Sistema/Administración/Gestión de paquetes synaptic/** y buscar **sun-javaX-jdk**, donde **X** es la versión de Java (en nuestro caso la versión 6). Si no está instalado, ir a **Aplicaciones - Accesorios - Terminal**; abrir un Terminal y colocar la siguiente línea:

sudo apt-get install sun-javaX-jdk

Donde **X** es la versión de java (en nuestro caso 6).

Nota: Para aplicar estos pasos se debe tener conexión a internet

Una vez instalado el JDK, procedemos a la instalación de BlueJ. En la página de descargas de BlueJ bajar el **bluej-300.deb**.

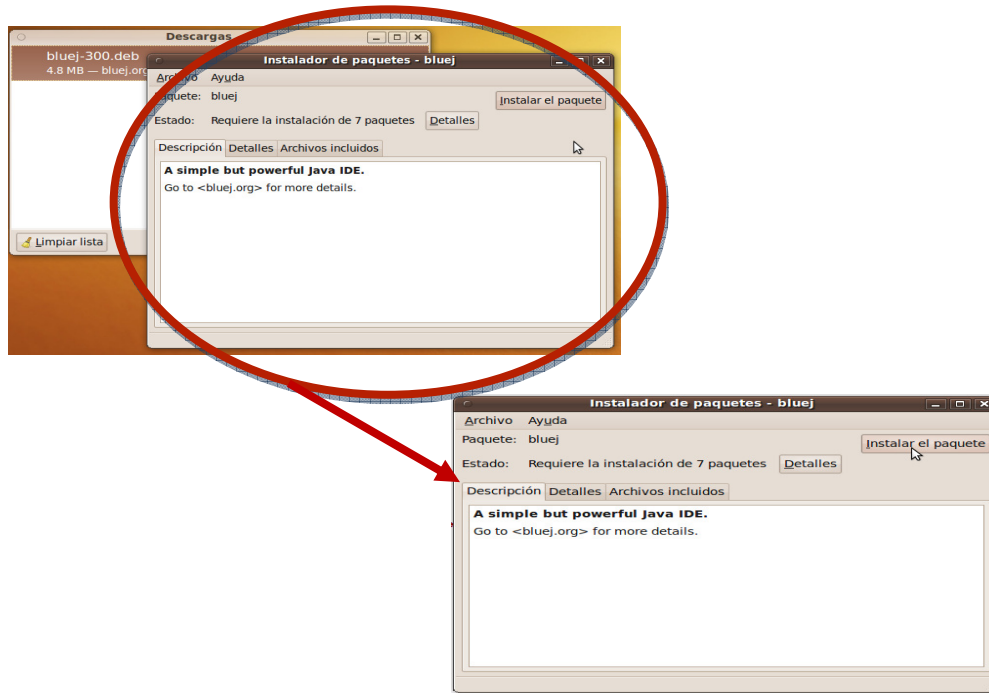


Y luego abre una ventana similar a la siguiente:

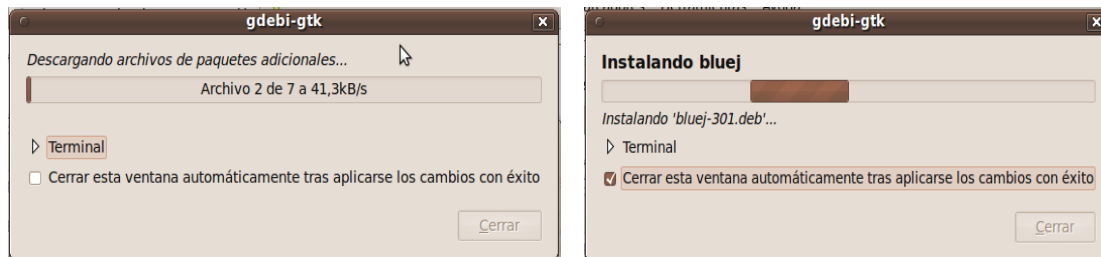


En caso de tener instalado el correspondiente *jdk*, aceptamos abrir con el instalador de paquetes GDebi. Se abrirá una ventana que muestra el estado de la descarga de BlueJ.

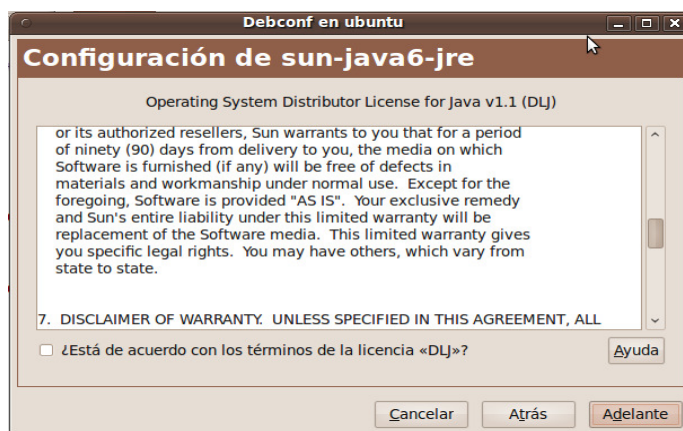
Se abre otra ventana para instalar todo el paquete de BlueJ. Hacer click en *“Instalar el paquete”*.



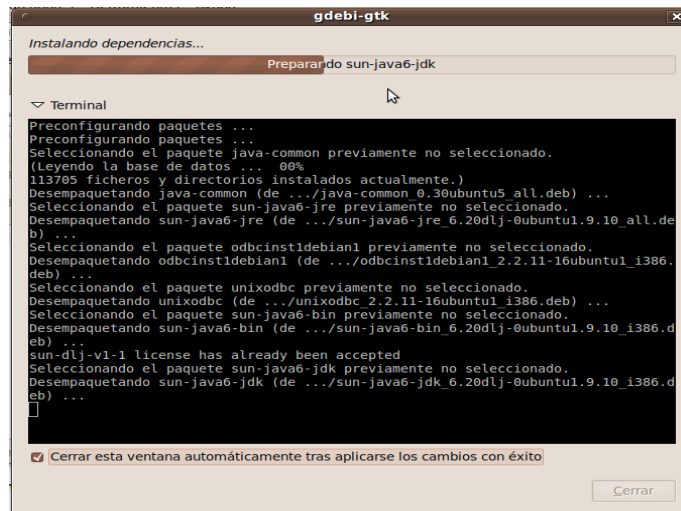
Esto habilitará otras dos ventanas, que muestran la descarga de paquetes adicionales, y posterior instalación de BlueJ, propiamente dicha.



Aceptamos los términos de uso de jre en ubuntu con respecto a la licencia "DLJ".



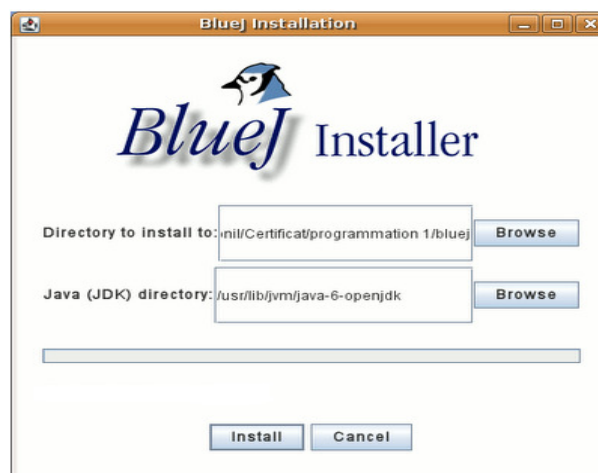
Luego el Gdebi preparara el jdk para su posterior uso; preconfigurando, seleccionando y desempaquetando paquetes necesarios para que podamos utilizar el BlueJ.



Nota: Trabajando en Ubuntu 9.10 todas las ventanas mostradas se abren en forma automática y son para instalación y reconfiguración de paquetes necesarios.

Para hacer uso de BlueJ vamos a Aplicaciones-Programacion-BlueJ y empezamos a usar BlueJ en forma normal como ya veremos más adelante.

Otra forma de abrir el BlueJ consiste en ir a la carpeta donde fuera descargado, hacer click en el icono llamado bluej y por último en *lanzar*. Pero es más fácil crear un lanzador en el escritorio haciendo click derecho, seleccionar "Lanzador" y luego colocar los datos como muestra la siguiente imagen:



Para mayor información al respecto recomendamos visitar los siguientes links:

www.guia-ubuntu.org/

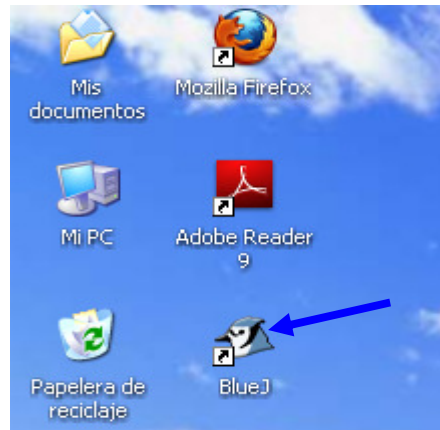
www.ubuntu-es.org/

www.ubuntu-ar.org/

4 Primeros Pasos con BlueJ

4.1 Creación de un proyecto

Hacemos doble click en el icono del BlueJ que se encuentra en el escritorio Windows. En el caso de Ubuntu, seguir el procedimiento explicado: Aplicaciones-Programacion-BlueJ.



Aparecerá la pantalla principal del entorno BlueJ 3.0.0 y desde allí accedemos a la opción Project, hacemos click en New Project. (**Imagen 8**)

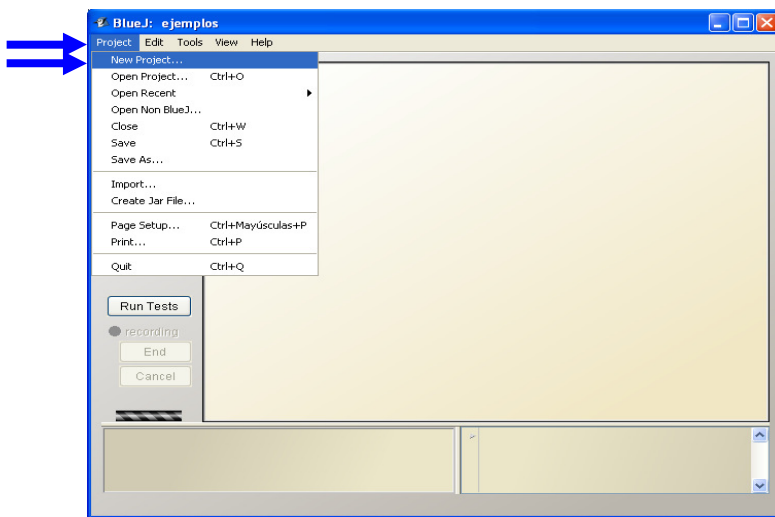


Imagen 8. Elaboración de un nuevo proyecto usando el entorno bluej.

En la opción **“Buscar en:”** elegir la carpeta donde se guardará el proyecto. En la opción **“Nombre de archivo:”** colocar el nombre del proyecto. En este ejemplo es **“ProyectoEjemplo”** y luego presionar el botón Create. Aparece la ventana de la **Imagen 10**.

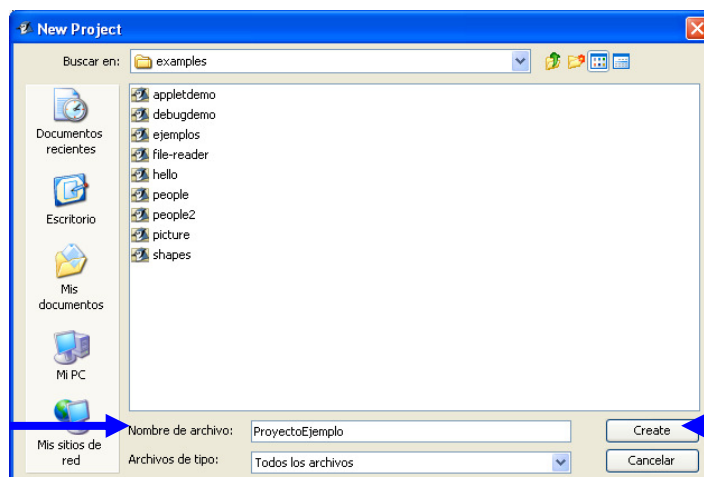


Imagen 9. Creacion del nuevo proyecto de nombre **“ProyectoEjemplo”**.

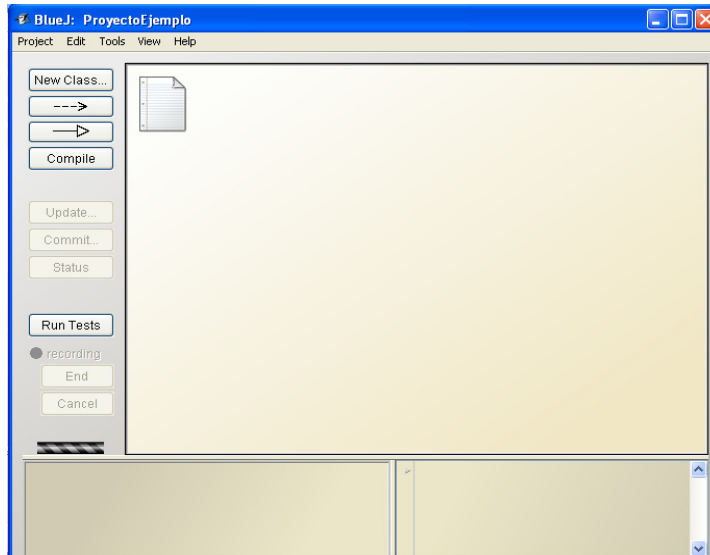


Imagen 10. Ventana de proyecto vacío.

4.2 Creación de Clases

Una vez creado el proyecto, presionar el botón **New Class (1)** y escribir en **Class Name (2)** el nombre de la clase, que en nuestro caso es Artículo. Los nombres de las clases por convención se escriben con la primera letra en mayúscula (Imagen 11). En **Class Type (3)** seleccionamos la opción **Class** y presionamos **Ok (4)**.

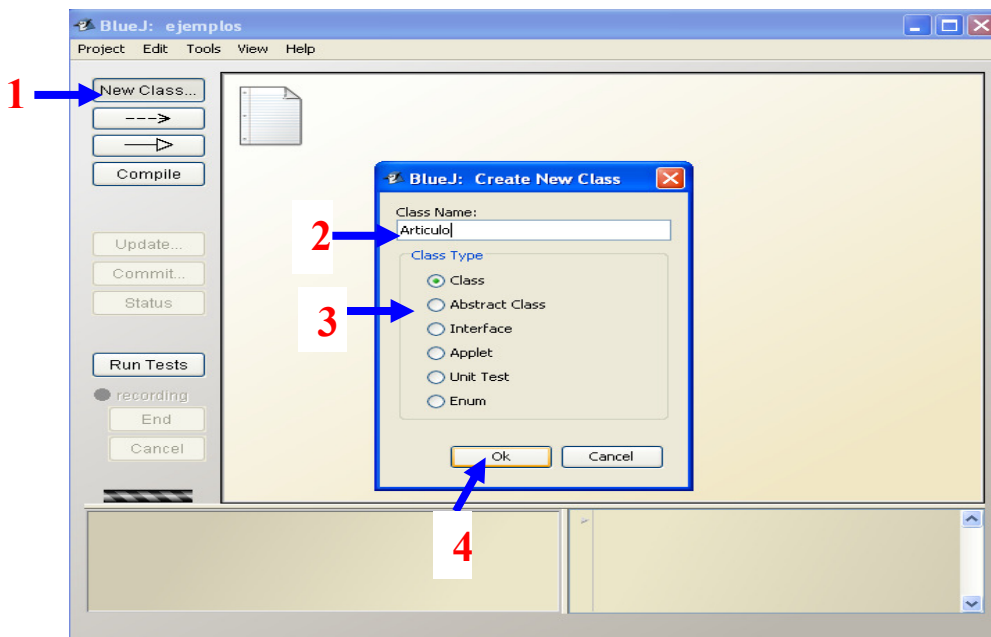
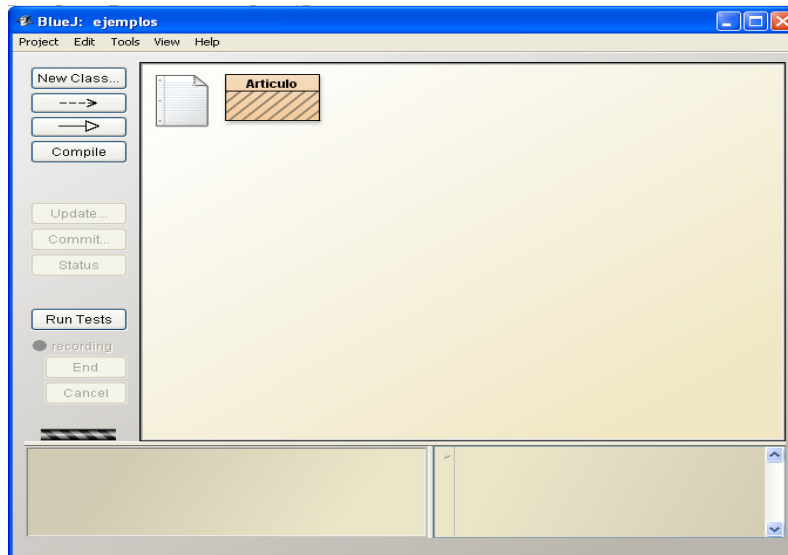


Imagen 11. Creación de la clase Artículo.

Aparecerá un icono que representa a la nueva clase creada.



Al hacer doble click en el icono de la clase creada aparece un editor mostrando el código por defecto. Eliminamos ese código y empezamos a implementar nuestra clase.

```
/**
 * Escribir la descripción de la Clase.
 *
 * @author (Oscar Fernández, Romero Carlos)
 * @version (1.1)
 */

public class Articulo{
    private int codigo;
    private String descripcion;
    private float precio;

    /**
     * Constructor público de la clase Artículo con parámetros.
     */

    public Articulo(int p_codigo,String p_desc,float p_precio){
        this.setCodigo(p_codigo);
        this.setDescripcion(p_desc);
        this.setPrecio(p_precio);
    }

    /**
     * Declaramos los Setters de cada uno de los Atributos de la Clase
     */

    public void setCodigo(int p_codigo){
        this.codigo = p_codigo;
    }
    public void setDescripcion(String p_desc){
        this.descripcion = p_desc;
    }
    public void setPrecio(float p_precio){
        this.precio = p_precio;
    }
}

/**
 * Declaramos los Getters de cada uno de los Atributos de la Clase
 */
```

```

    public int getCodigo(){
        return this.codigo;
    }
    public String getDescripcion(){
        return this.descripcion;
    }
    public float getPrecio(){
        return this.precio;
    }

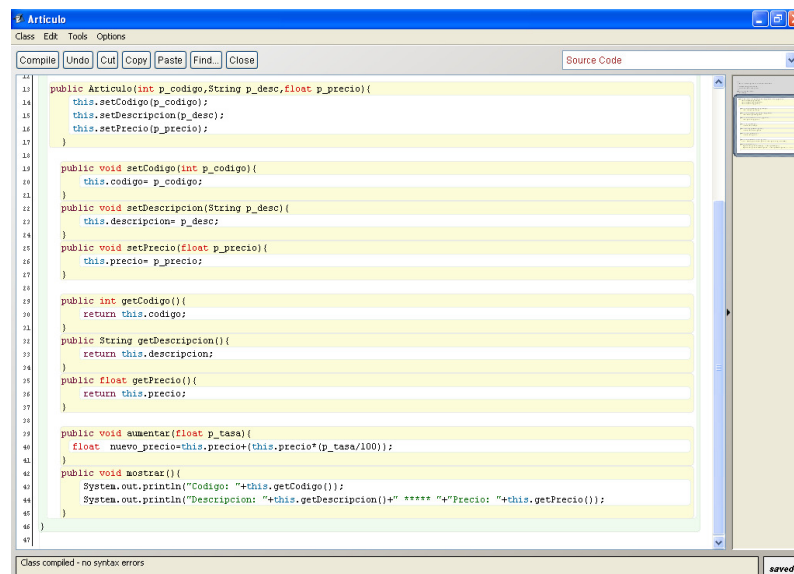
/**
 * Método Aumentar que realiza el incremento del precio de un
 * determinado Artículo.
 */

    public float aumentar(float p_tasa){
        float nuevo_precio = this.precio+(this.precio*(p_tasa/100));
        return nuevo_precio;
    }

/**
 * Método Mostrar por el cual podemos visualizar los datos del
 * Artículo.
 */

    public void mostrar(){
        System.out.println("Codigo: "+this.getCodigo());
        System.out.println("Descripcion: "+this.getDescripcion()+
            "***** "+"Precio: "+this.getPrecio());
    }
}

```



Presionamos el botón **Compile** para realizar la compilación. En el panel inferior se muestran los errores de compilación, tema que desarrollaremos mas adelante.

4.3 Creación de Clases de Prueba – Ingreso de datos

Siguiendo las pautas anteriores, crearemos la clase **PruebaArticulo**.

Observamos los iconos de las clases creadas unidas por una flecha discontinua, lo que indica la relación “**usa**” entre las mismas.

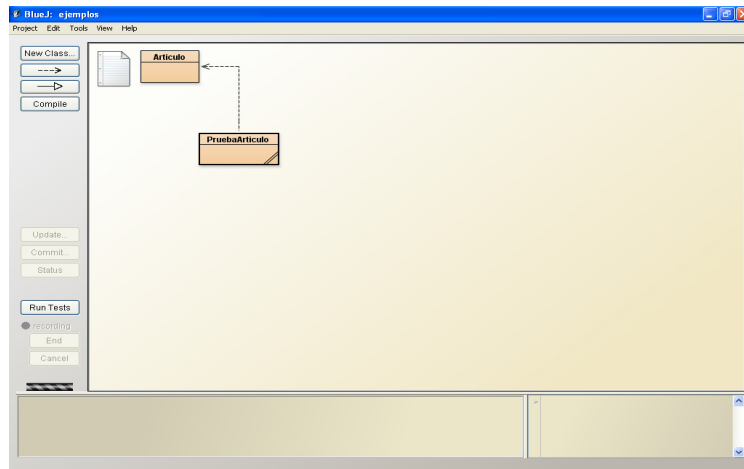


Imagen 12. clase *PruebaArticulo* “**usa**” clase *Articulo*.

El ingreso de datos se puede realizar de tres formas diferentes:

- 1- Como constantes.
- 2- Como argumentos del método main.
- 3- Por teclado.

4.3.1 Como constantes

Los objetos son creados pasando **constantes** como argumentos del constructor:

```
public class PruebaArticulo{
    public static void main(String []args){
        Articulo miArticulo= new Articulo(1,"pen drive",75f);
        System.out.println("el articulo original es: ");
        miArticulo.mostrar();
        System.out.println("el precio aumentado en 10% es:" +
            miArticulo.aumentar(10));
    }
}
```

Para ejecutar, click con el botón derecho del mouse sobre la clase que contiene el método principal (*main*), que en nuestro caso es la clase *PruebaArticulo*. Seleccionamos la opción `void main(String[])`, que abrirá una ventana silmilar a la Imagen 13.

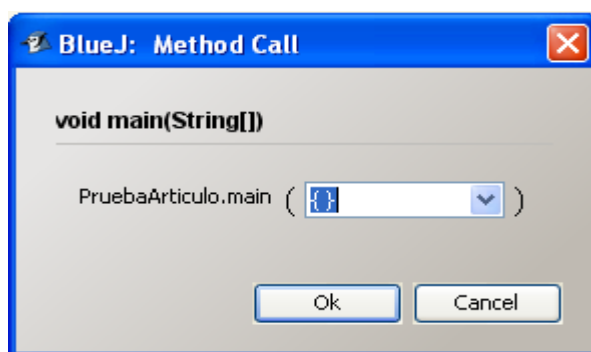


Imagen 13. Ejecución de clase *PruebaArticulo* con datos pasados como constantes.

Presionamos *Ok*. El programa se ejecuta y muestra la salida en una ventana llamada "Terminal Window". La salida muestra los resultados obtenidos al operar con los datos ingresados como constantes (Imagen 14).

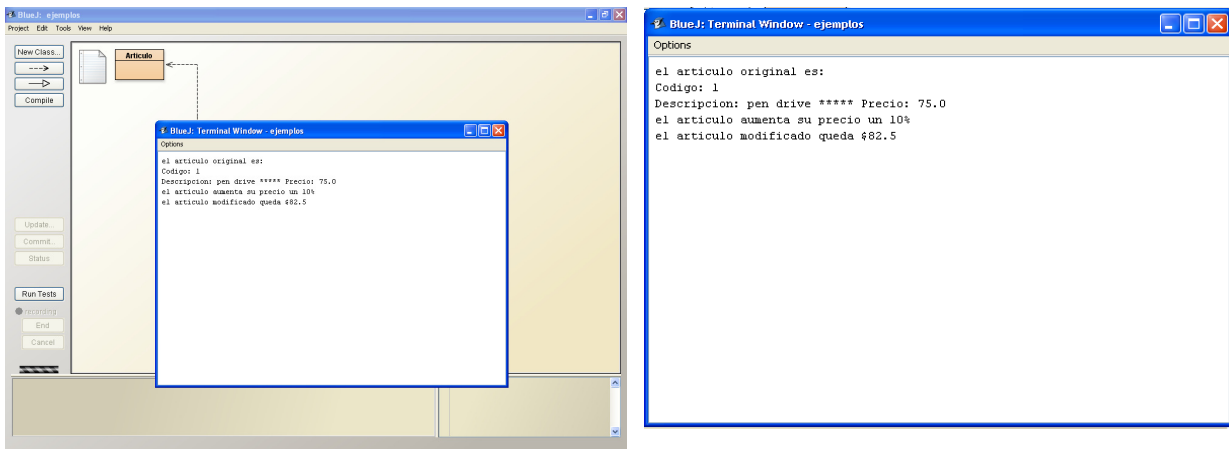


Imagen 14. Imagen del "Terminal Window".

La única forma de modificar los datos del artículo creado, es accediendo al código de prueba, modificarlos y volver a compilar.

4.3.2 Como argumentos del método main

Debido a que el array del main es de tipo String, es necesario convertir los datos ingresados al tipo deseado, según se observa en el siguiente ejemplo:

```
public class PruebaArgum{
    public static void main(String []args){
        int codigo = Integer.parseInt(args[0]);
        String descripcion = args[1];
        float precio = Float.parseFloat(args[2]);
        float tasa = Float.parseFloat(args[3]);
        Articulo miArticulo = new Articulo(codigo,descripcion,precio);
        miArticulo.mostrar();
        System.out.println("El precio aumentado en $: "+tasa + " es " +
            miArticulo.aumentar(tasa));
    }
}
```

Al ejecutar la clase PruebaArgum (botón derecho del mouse - void main(String[])), ingresamos los argumentos necesarios para la ejecución. Estos deben estar entre comillas y separados por comas, debido a que se trata de un arreglo de tipo String. (Imagen 15).

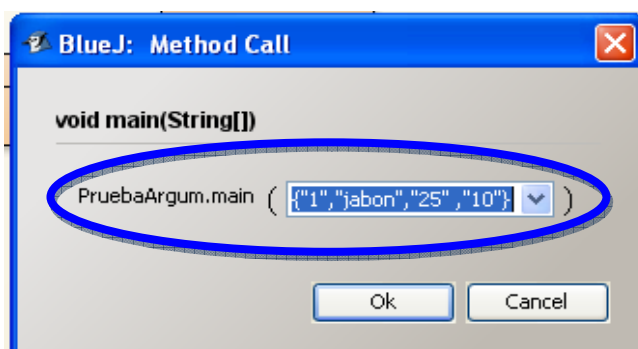


Imagen 15. Ingreso de datos como argumentos del main

Presionamos en Ok, el programa se ejecutará y en la ventana “Terminal Window” mostrará el resultado de la ejecución.



4.3.3 Por Teclado – Usando la clase Scanner

Se debe crear un objeto de tipo Scanner, que luego será utilizado para el acceso a los datos:

```
import java.util.Scanner;

public class ArticuloScan{
    public static void main(String []args) {
        Scanner teclado = new Scanner(System.in);
        teclado.useDelimiter("\n");

        try {
            System.out.println("Ingrese Codigo de Articulo");
            int a_codigo = teclado.nextInt();

            System.out.println("Ingrese Nombre del Producto");
            String a_desc = teclado.next();

            System.out.println("Ingrese Precio");
            float a_precio = teclado.nextFloat();

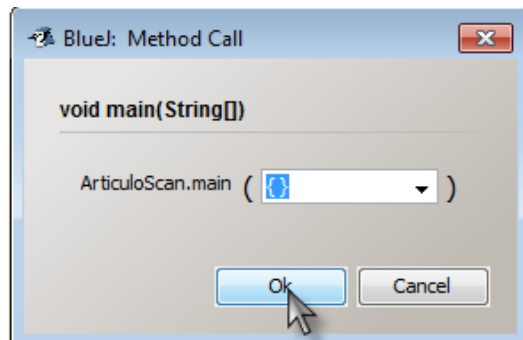
            System.out.println("Ingrese Porcentaje de Incremento");
            float inc = teclado.nextFloat();

            Articulo miArticulo= new Articulo(a_codigo, a_desc, a_precio);

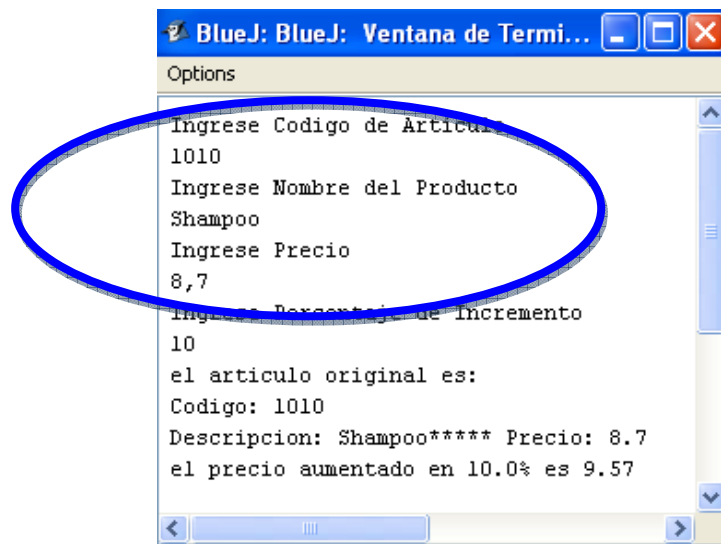
            System.out.println("el articulo original es: ");
            miArticulo.mostrar();
            System.out.println("el precio aumentado en "+ inc + "% es "
                               +miArticulo.aumentar(inc)) ;
        }
        catch (Exception error) {
            System.out.println("Debes Respetar el Formato Establecido");
        }
    }
}
```

La clase Scanner toma todos los datos ingresados como String, por lo que necesario hacer la conversión pertinente, usando los métodos provistos por esta clase (nextInt, nextFloat, etc).

Ejecutar esta clase (botón derecho - método *main*). Se abrirá la siguiente ventana. Presionar *Ok*.

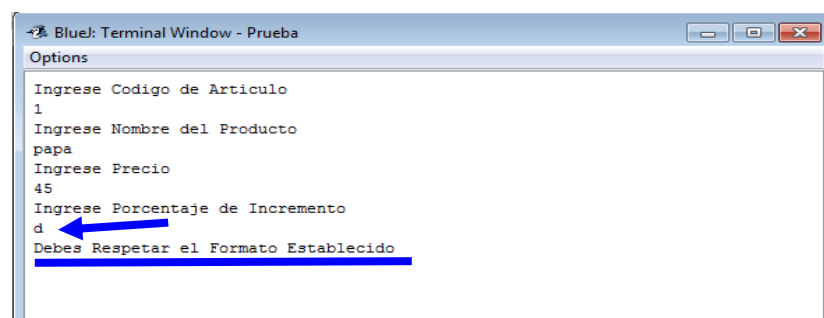


El programa se ejecuta y muestra la ventana "Terminal Window".



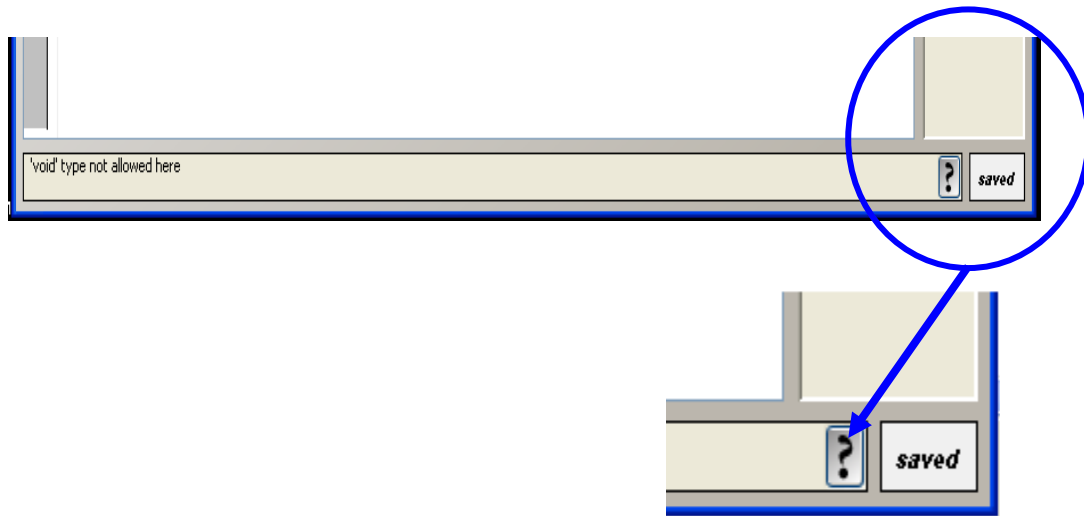
En esta terminal se ingresan los datos solicitados, debiendo respetar el tipo de datos. Luego se muestra el resultado de la evaluación.

En caso de ingresar un dato incorrecto, mostrará la leyenda definida en el manejo de excepciones (try – catch). En este caso la leyenda es **"Debes Respetar el Formato Establecido"**, como puede observarse en el siguiente ejemplo:

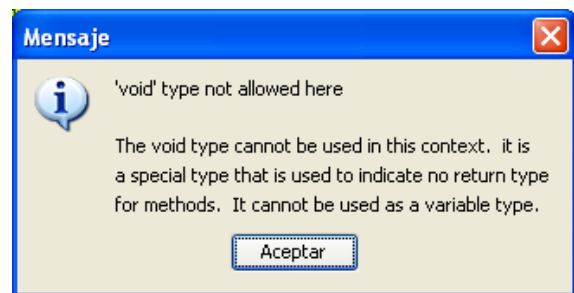


5 Errores de compilación

Al compilar un programa puede aparecer un mensaje en el área de información del editor (en la parte inferior del editor). A la derecha de dicha área aparece un icono de interrogación, que se puede pulsar para obtener más información sobre el tipo de error cometido.



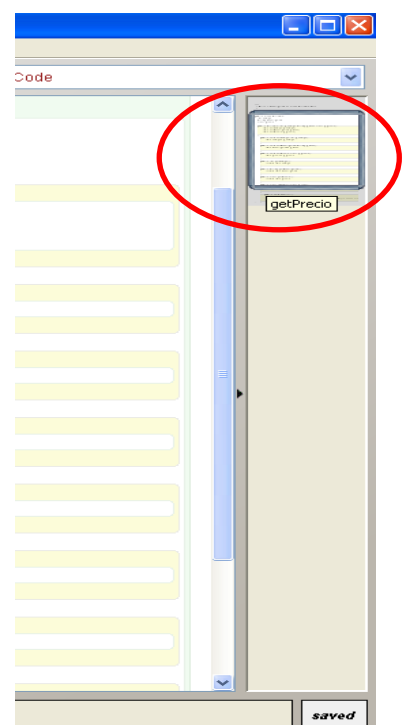
Se abrirá una ventana con un mensaje que brindará una breve explicación del error cometido.



6 Visor de navegación (Navigation View)

A diferencia de las anteriores versiones de BlueJ, a la derecha del editor se observa una “Vista de Navegación” o Navigation View, que muestra una versión en miniatura del texto completo de la clase, y en el que se destaca un visor de selección. Es útil como una barra de desplazamiento: la vista de navegación se puede arrastrar haciendo click sobre una parte del código, trasladando el visor ó moviendo la rueda del mouse.

Al pasar el mouse sobre la vista de navegación, brinda información acerca del código, y muestra los nombres de los métodos, lo cual permite acceder a métodos específicos más rápidamente.

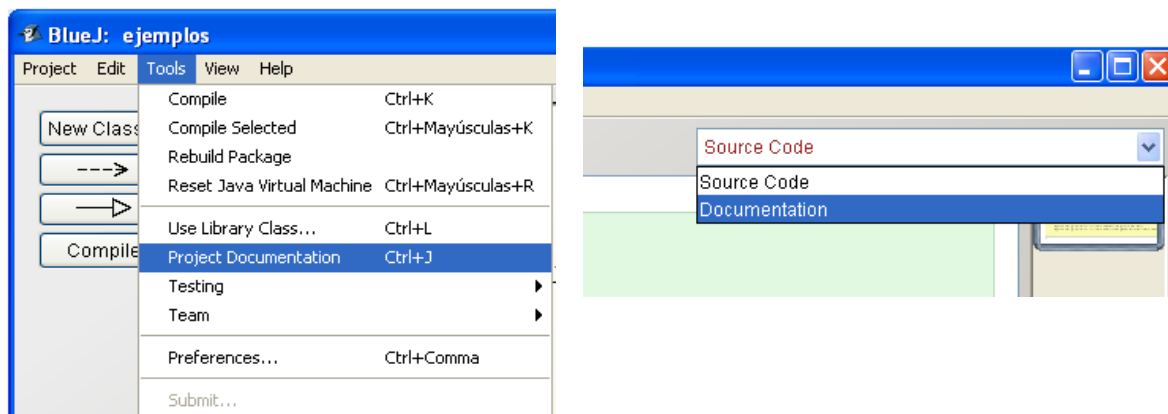


7 Como generar la documentación de un proyecto

Es posible generar la documentación de un proyecto en el formato estándar de javadoc (herramienta provista por Java), desde dentro de BlueJ. Para ello, es preciso que el cuerpo de los comentarios inicie con `/**` y se cierre con `*/`. Cada línea del comentario inicia con un `*`.

Para generar la documentación del proyecto, seleccionar menú *Tools\Project Documentation*. Se generará un documento con todos los comentarios colocados en el código, y se abrirá el navegador para mostrarlo.

También se puede generar y ver la documentación para una única clase en el editor de BlueJ. Para hacer esto, abrir el editor de la clase y usar el menú emergente en la barra de herramientas. Cambiar la selección de *Source Code* a *Documentation*. Esto mostrará la documentación al estilo de javadoc (el interfaz de la clase) en el editor.



En la imagen de la izquierda se observa cómo generar la documentación de un proyecto completo. En la imagen de la derecha sin embargo, se muestra cómo generar la documentación de una clase en particular.