

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
 Ese apoyo que siempre quiso tener...

1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
 HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
 Gestor de contenidos (Alfresco)
 Aplicaciones híbridas

Tareas programadas (Quartz)
 Gestor documental (Alfresco)
 Inversión de control (Spring)

Control de autenticación y
 acceso (Spring Security)
 UDDI
 Web Services
 Rest Services
 Social SSO
 SSO (Cas)

JPA-Hibernate, MyBatis
 Motor de búsqueda empresarial (Solr)
 ETL (Talend)

Dirección de Proyectos Informáticos.
 Metodologías ágiles
 Patrones de diseño
 TDD

BPM (jBPM o Bonita)
 Generación de informes (JasperReport)
 ESB (Open ESB)



Hosting Patrocinado por
enREDados.com


[Home](#) | [Quienes Somos](#) | [Empleo](#) | [Foros](#) | [Tutoriales](#) | [Servicios Gratuitos](#) | [Contacte](#)

	<p>Tutorial desarrollado por: Roberto Canales Mora 2003-2005 Creador de AdictosAlTrabajo.com y</p> <p>Director General de Autentia S.L.</p> <p>Recuerda que me puedes contratar para echarte una mano:</p> <p>Desarrollo y arquitectura Java/J2EE Asesoramiento tecnológico Web Formación / consultoría integrados en tu proyecto</p> <p>No te cortes y contacta: 655 99 11 72 rcanales@autentia.com.</p>	
---	--	---

Descargar este documento en formato PDF [editor.pdf](#)

Nuevo negocio

Nuevos contactos profesionales para sus negocios. Prueba gratis.

JSP Editor

Edit JSP, XML, DTD, Schema, XSLT & SOAP. Easy-to-Use! Free Trial.

Negocio en Casa

Obtenga el negocio #1 de Internet en casa u oficina. Apoyo completo.

Formación Empresas

Consultoría de Formación Tecnologías Web

Anuncios Goooooogle

Anunciarse en este sitio

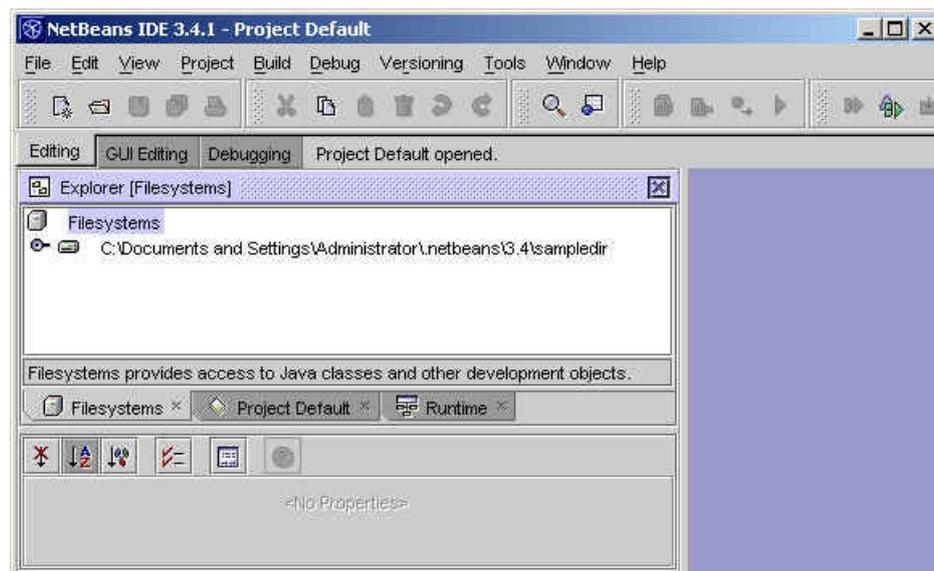
Aplicación gráfica con NetBeans

En este tutorial vamos a mostrar varias cosas a la vez:

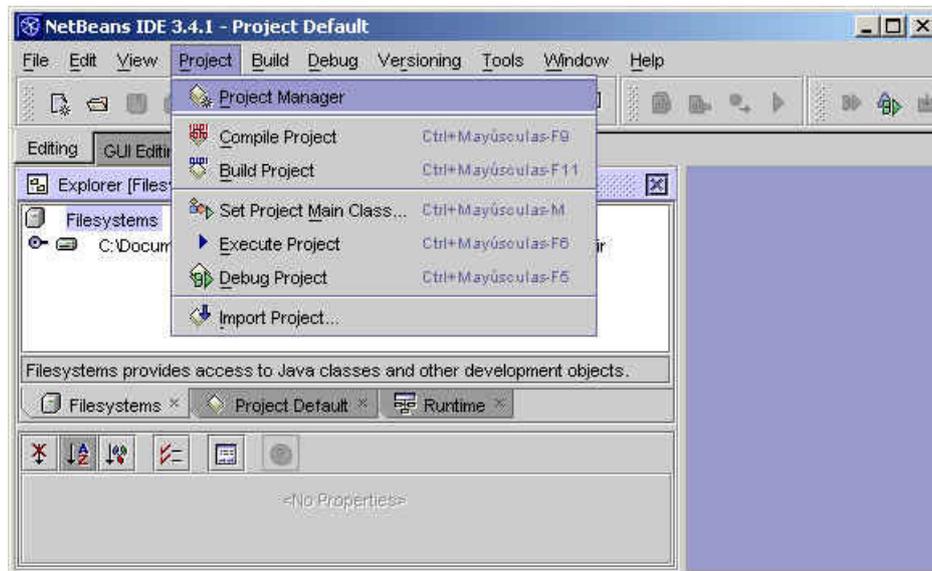
- Como crear una aplicación gráfica con NetBeans
- Como hacer una pequeña herramienta de dibujo
- Como contruir ... a través de clases abstractas programas ampliables

NetBeans es un entorno de desarrollo gratuito, muy sencillo y potenteVamos a capturar todas las pantallas para mostrarnos como crear el esqueleto de nuestro programa.

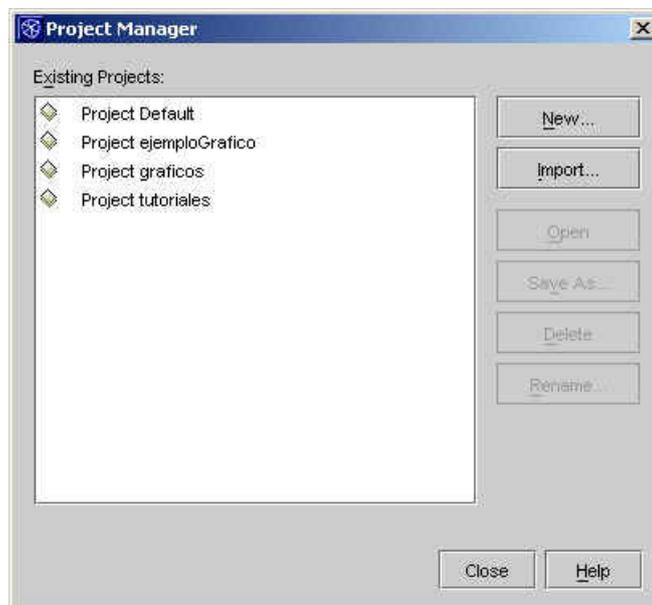
Este es el aspecto de NetBeans al arrancarlo:



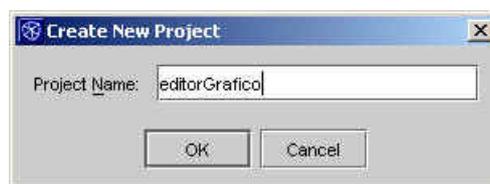
Creamos un nuevo proyecto



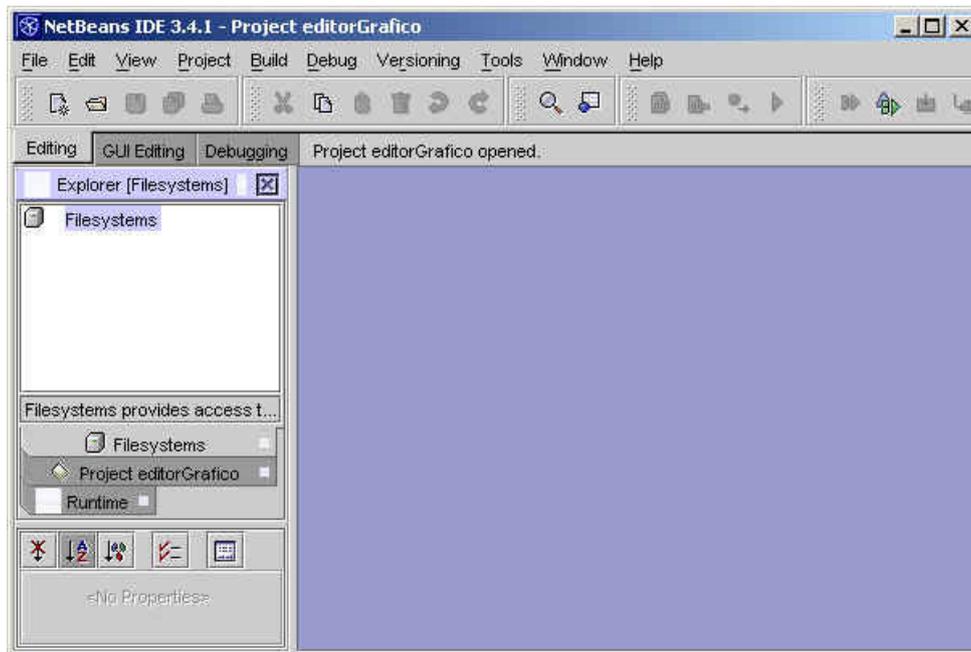
Pulsamos el botón NEW



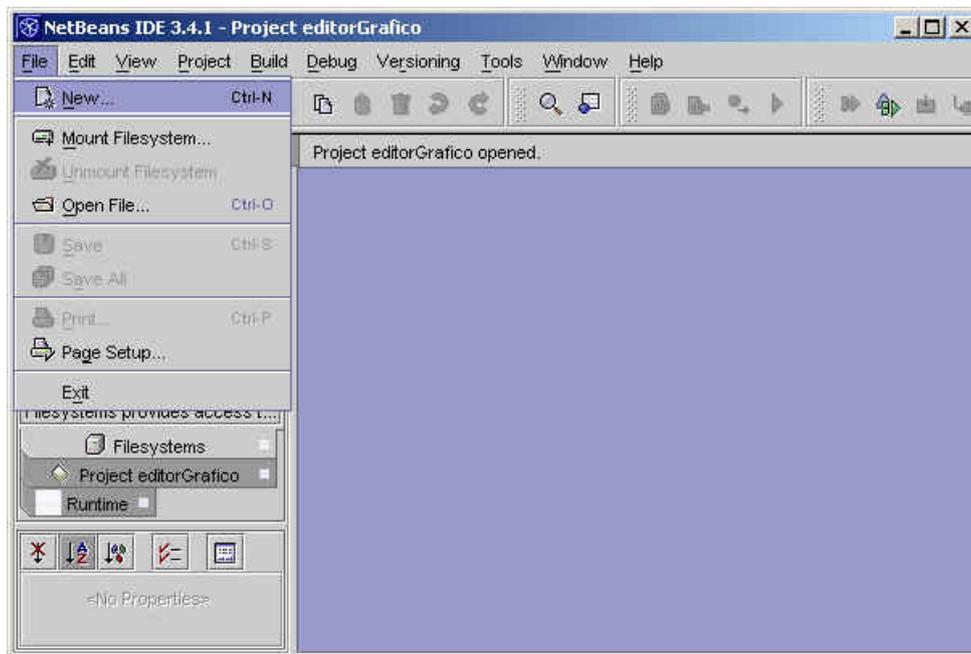
Seleccionamos el título del proyecto



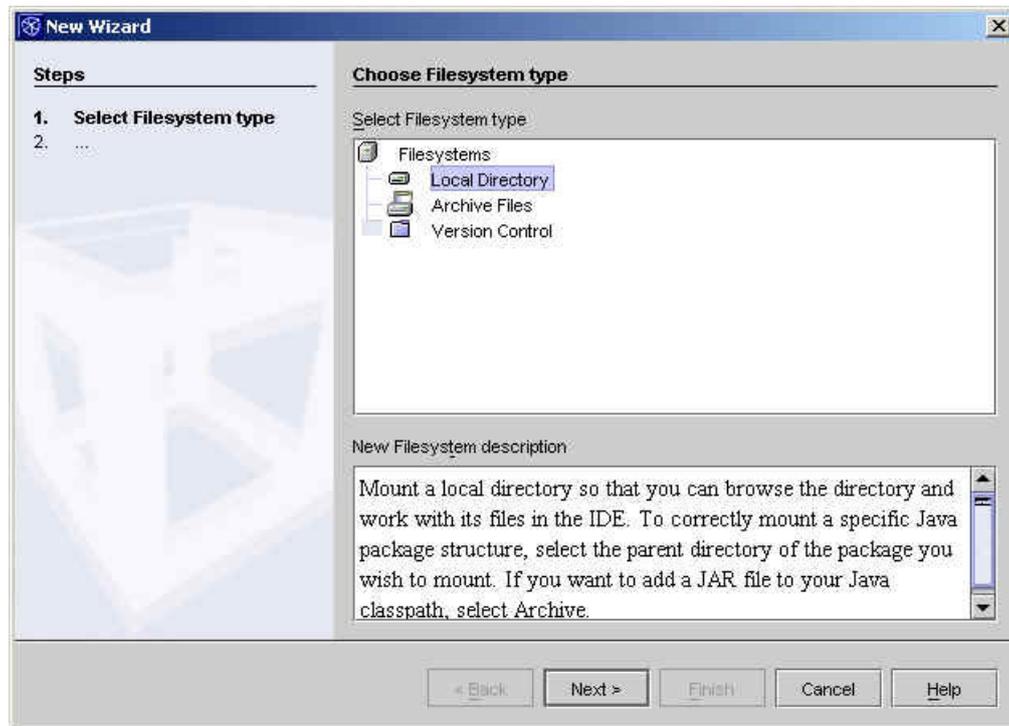
Ya tenemos el proyecto donde meter nuestros ficheros



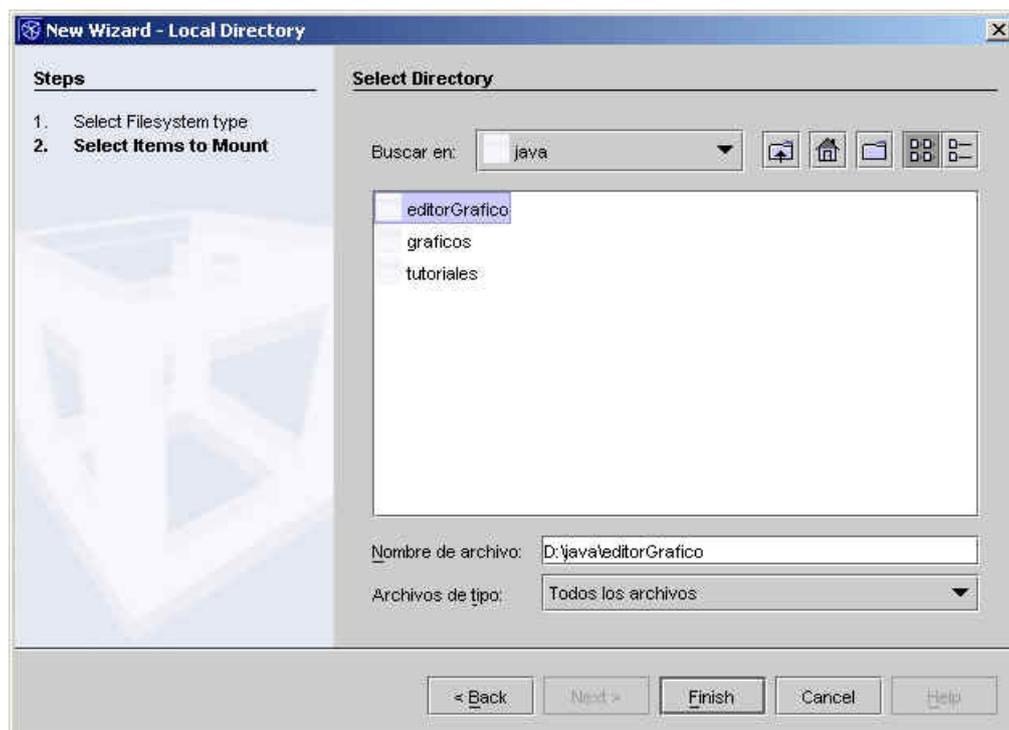
Vamos a crear un directorio



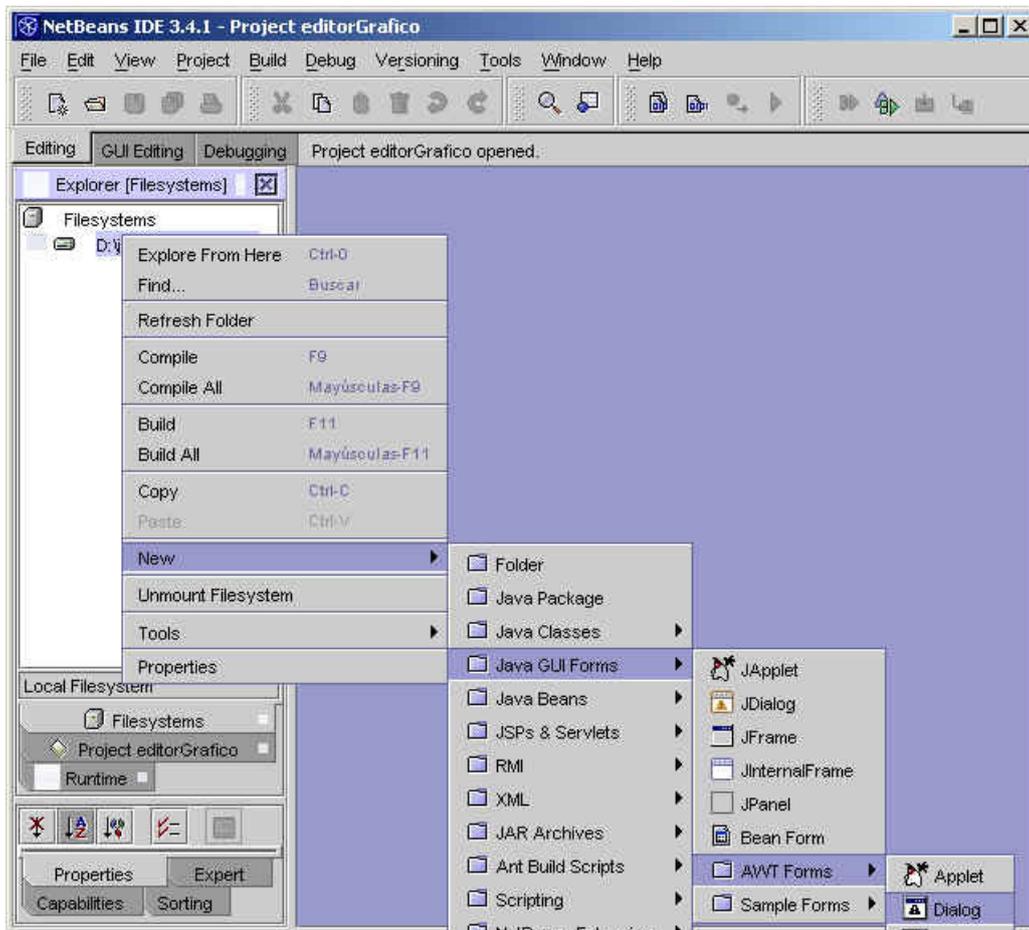
Seleccionamos el sistema de ficheros



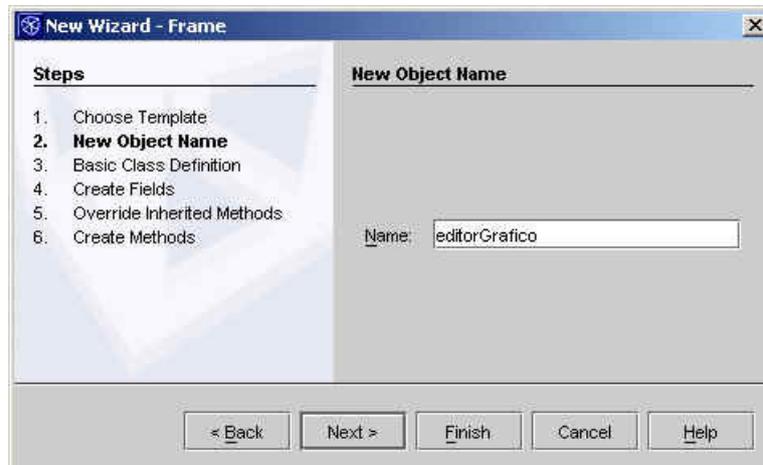
Creamos un nuevo directorio ... editor gráfico

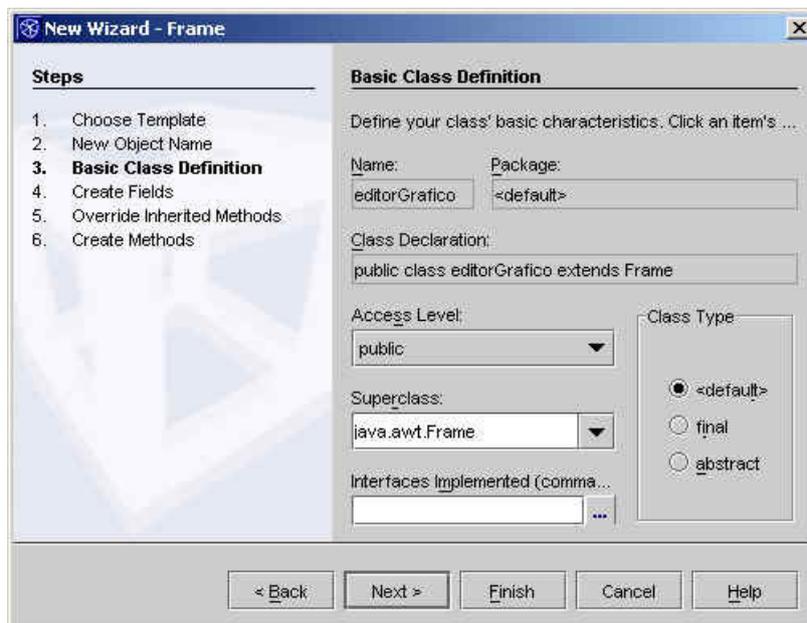


Ahora sobre nuestro proyecto pulsamos el botón derecho

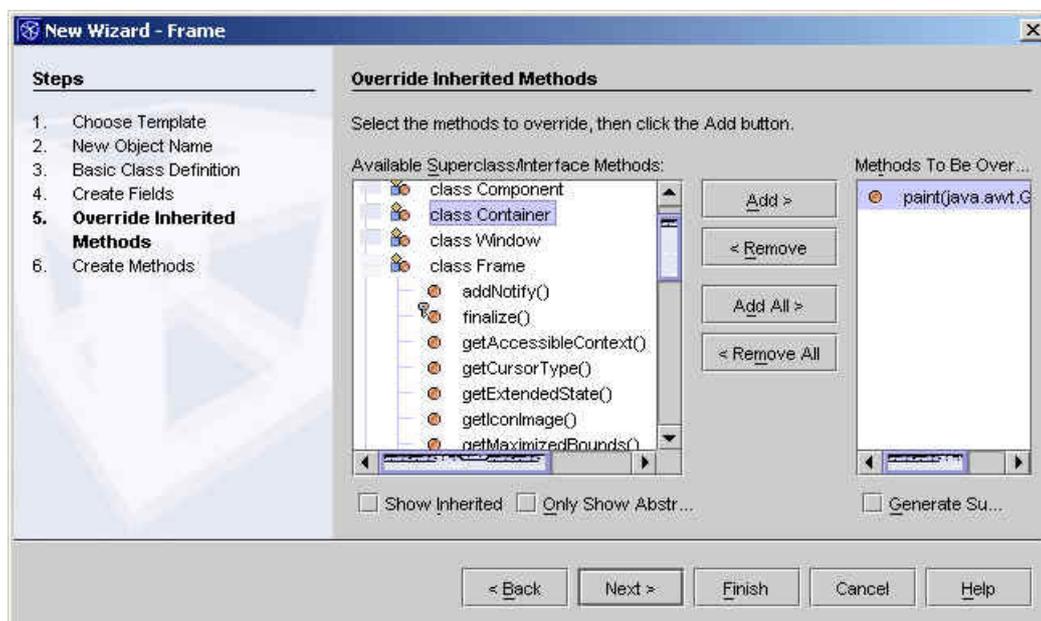


Hemos elegido tipo Frame

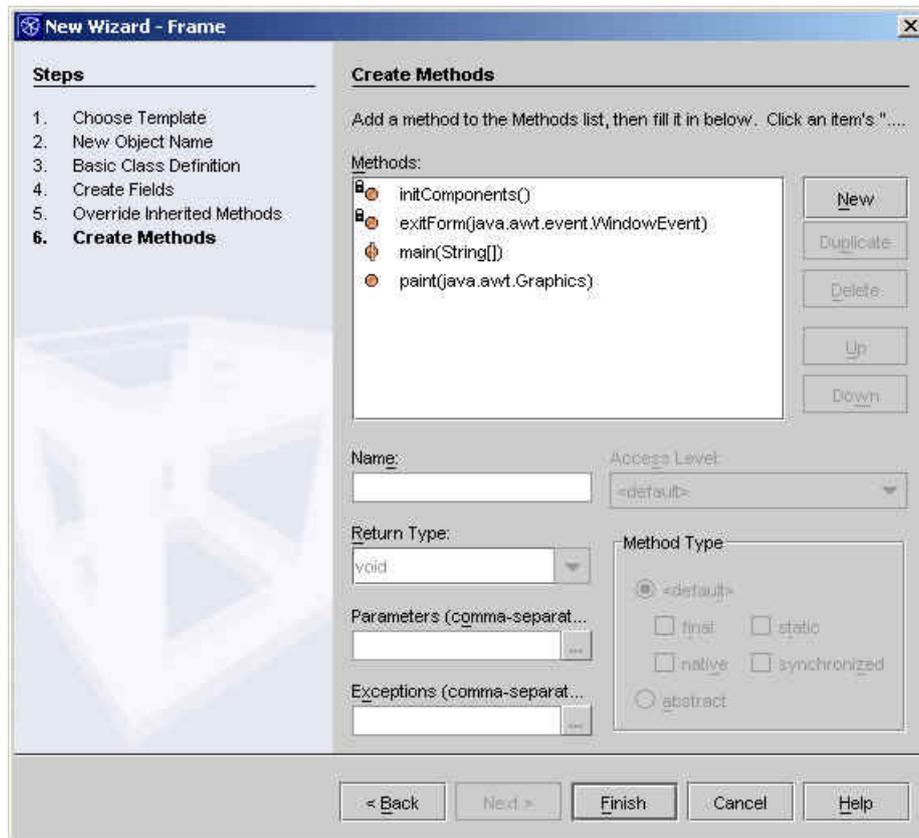




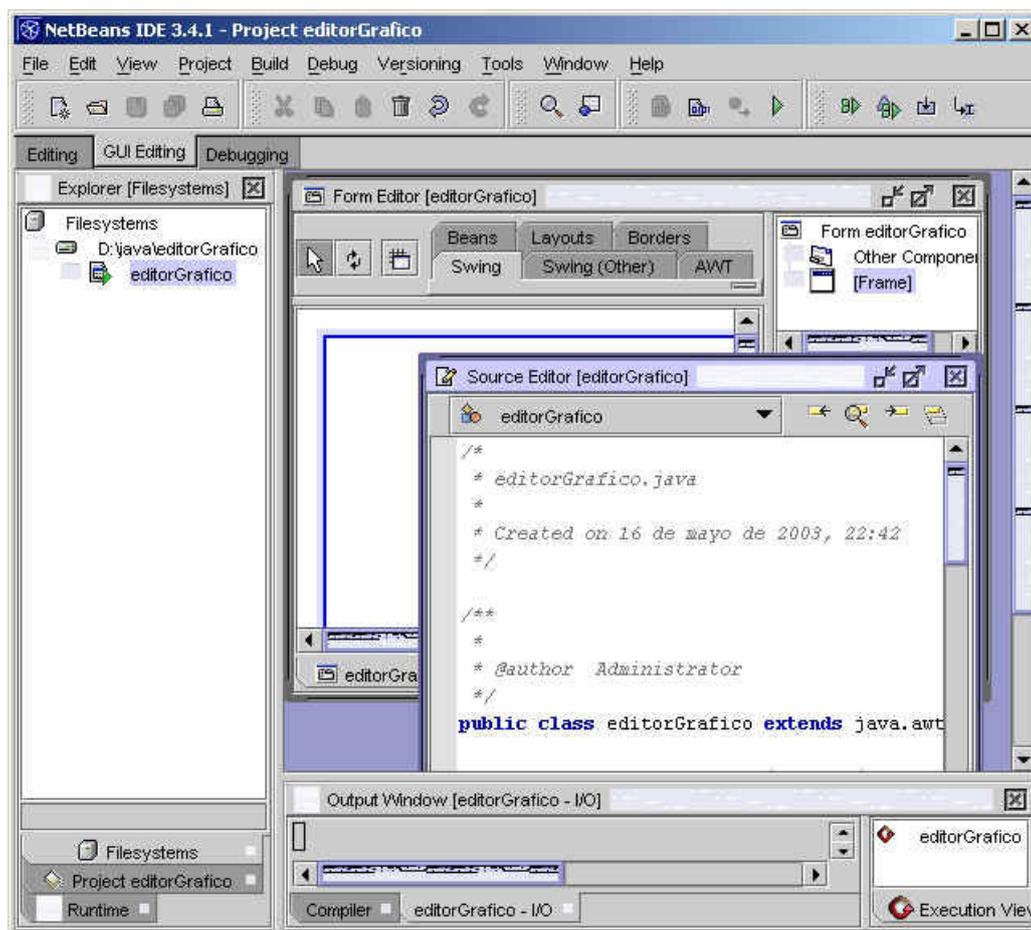
Elegimos los metodos a sobre escribir



Vemos el resumen



El sistema ha escrito el código por nosotros



```

/*
 * editorGrafico.java

```

```
*
* Created on 16 de mayo de 2003, 22:42
*/
/**
 *
 * @author Administrator
 */
public class editorGrafico extends java.awt.Frame {

    /** Creates new form editorGrafico */
    public editorGrafico() {
        initComponents();
    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    private void initComponents() {

        addWindowListener(new java.awt.event.WindowAdapter() {
            public void windowClosing(java.awt.event.WindowEvent evt) {
                exitForm(evt);
            }
        });
        pack();
    }

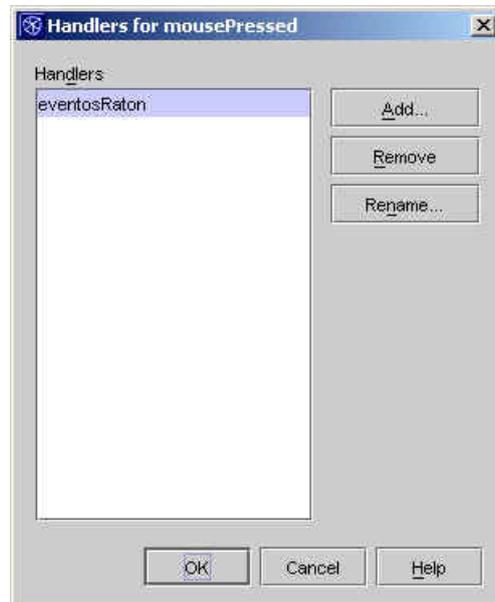
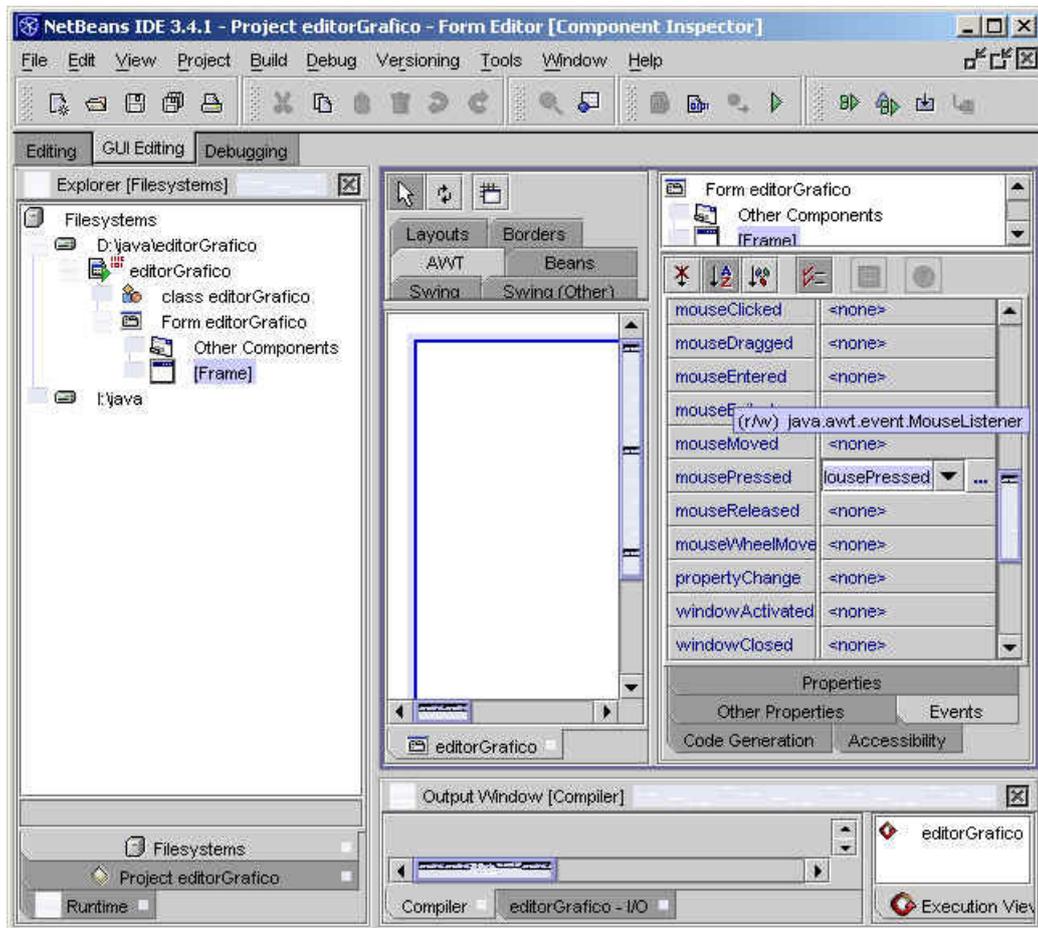
    /** Exit the Application */
    private void exitForm(java.awt.event.WindowEvent evt) {
        System.exit(0);
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        new editorGrafico().show();
    }

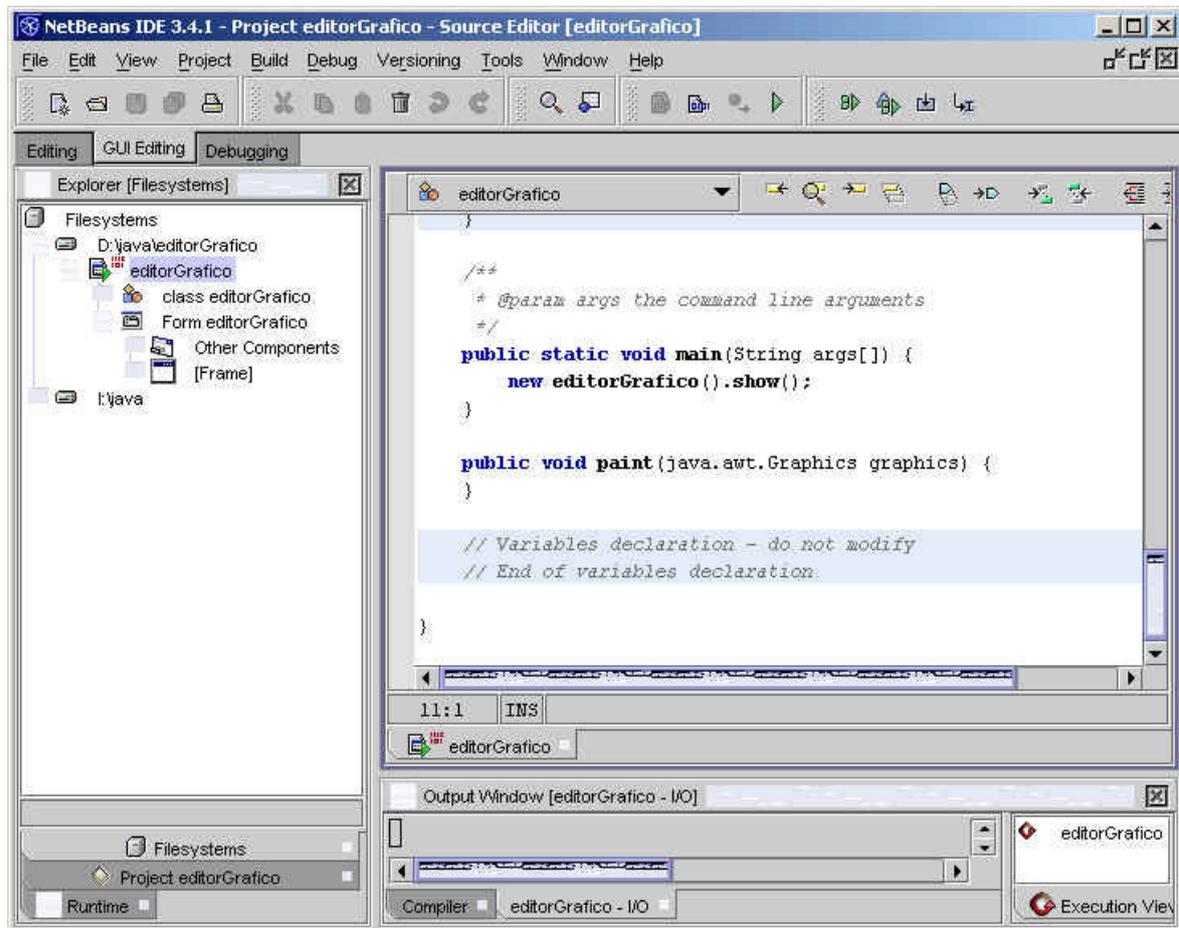
    public void paint(java.awt.Graphics graphics) {
    }

    // Variables declaration - do not modify
    // End of variables declaration
}
}
```

Queremos atender los eventos del ratón... porque pintaremos cuando el usuario pulse y suelte en ratón

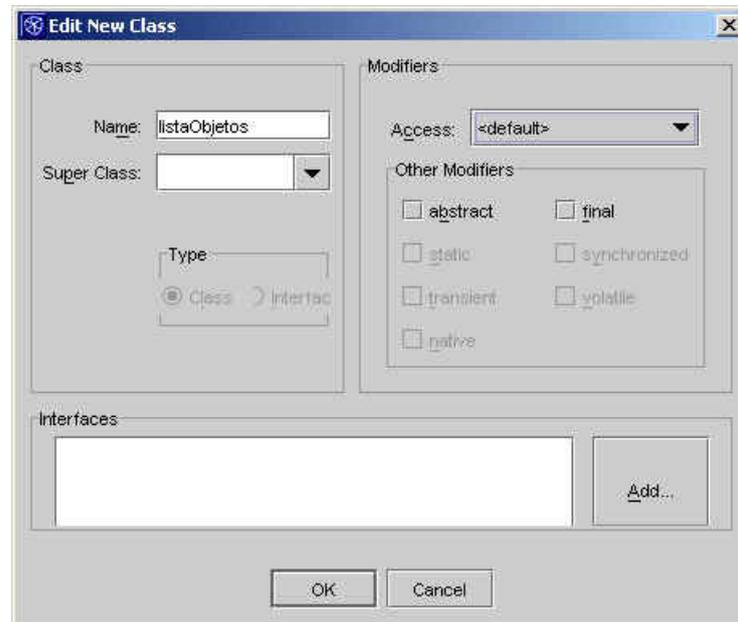
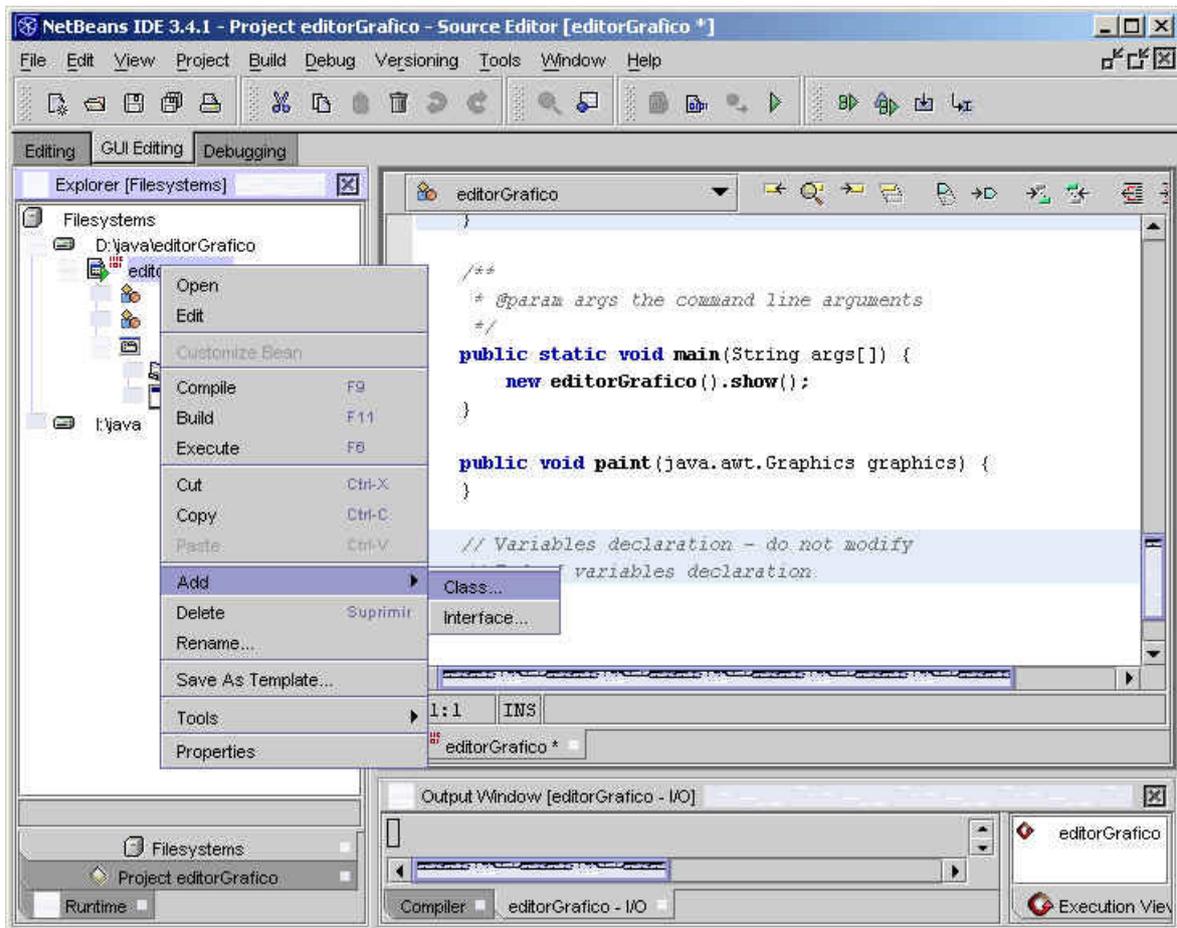


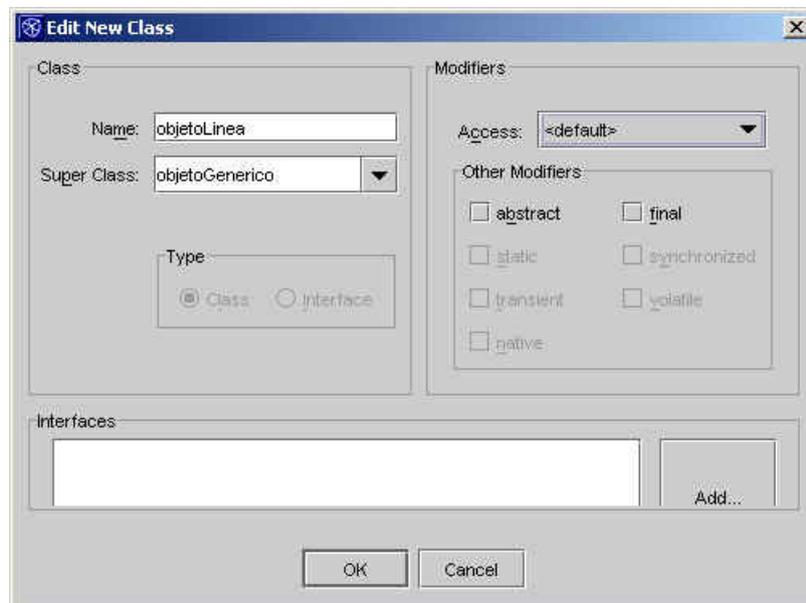
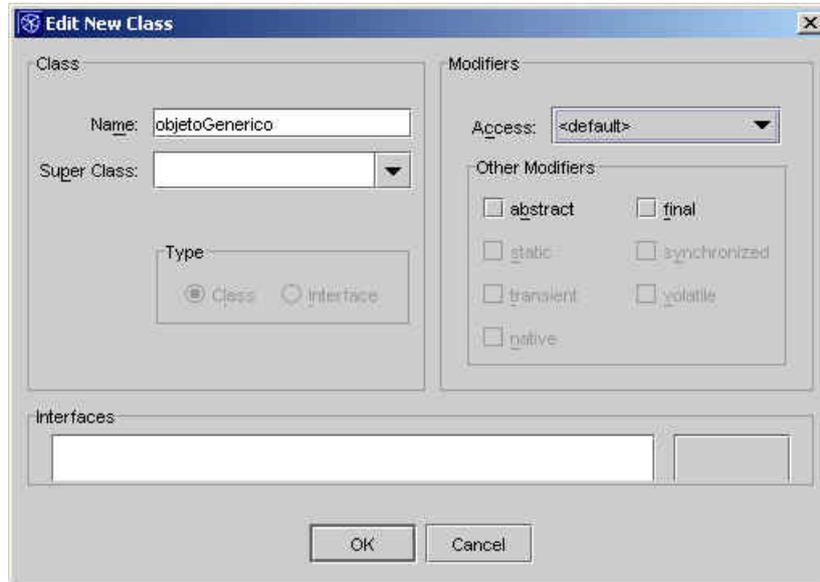
Vemos que ha generado el código necesario de un modo sencillo



Nuestra aplicación pretende ser un pequeño programa de dibujo simple que nos permita dibujar distintos tipos de objetos y repintarlos cuando sea necesario, aunque lo vamos a dejar preparado para que haga más cosas ...

Vamos a crear una clase que represente todos los objetos pintados, otra que represente cada uno de los objetos a pintar (una clase abstracta) y otra que sea una implementación del primer tipo de elemento una línea





Bueno ... aunque tengamos muchas pantallas el código por ahora generado es poco

```

/*
 * editorGrafico.java
 *
 * Created on 16 de mayo de 2003, 22:42
 */
/**
 *
 * @author Administrator
 */
public class editorGrafico extends java.awt.Frame {

    /** Creates new form editorGrafico */
    public editorGrafico() {
        initComponents();
    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    private void initComponents() {

        addMouseListener(new java.awt.event.MouseAdapter() {
            public void mousePressed(java.awt.event.MouseEvent evt) {
                botonPulsado(evt);
            }
            public void mouseReleased(java.awt.event.MouseEvent evt) {
                botonSoltado(evt);
            }
        });
    }
}

```

```

    }
  });

  addWindowListener(new java.awt.event.WindowAdapter() {
    public void windowClosing(java.awt.event.WindowEvent evt) {
      exitForm(evt);
    }
  });

  pack();
}

private void botonSoltado(java.awt.event.MouseEvent evt) {
  // Add your handling code here:
}

private void botonPulsado(java.awt.event.MouseEvent evt) {
  // Add your handling code here:
}

/** Exit the Application */
private void exitForm(java.awt.event.WindowEvent evt) {
  System.exit(0);
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
  new editorGrafico().show();
}

public void paint(java.awt.Graphics graphics) {
}

// Variables declaration - do not modify
// End of variables declaration
}

class listaObjetos {
}

class objetoGenerico {
}

class objetoLinea extends objetoGenerico {
}

```

Bueno .. creo que lo hemos captado ...

por lo que vemos a seguir escribiendo a mano el código

La verdad es que tampoco hay que escribir demasiado os mostramos el programa final y luego explicamos un poco la lógica

```

import java.awt.*;
import java.util.*;
import java.lang.*;
import java.io.*;

/*
 * editorGrafico.java
 *
 * Created on 16 de mayo de 2003, 22:42
 */

/**
 *
 * @author Administrator
 */
public class editorGrafico extends java.awt.Frame
{
  // objeto que contiene todo lo que pintamos
  private listaObjetos nuestrosObjetos = new listaObjetos();

  // variables temporales donde almacenar los puntos
  int xIniTemp = 0;
  int yIniTemp = 0;
  int xFinTemp = 0;
  int yFinTemp = 0;

  // metodo para centrar mensajes de depuracion
  void depura(String mensaje)
  {
    System.out.println("El mensaje es " + mensaje);
  }
}

```

```

/** Creates new form editorGrafico */
public editorGrafico() {

    initComponents();
}

/** This method is called from within the constructor to
 * initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is
 * always regenerated by the Form Editor.
 */
private void initComponents() {

    addMouseListener(new java.awt.event.MouseAdapter() {
        public void mousePressed(java.awt.event.MouseEvent evt) {
            botonPulsado(evt);
        }
        public void mouseReleased(java.awt.event.MouseEvent evt) {
            botonSoltado(evt);
        }
    });

    addWindowListener(new java.awt.event.WindowAdapter() {
        public void windowClosing(java.awt.event.WindowEvent evt) {
            exitForm(evt);
        }
    });

    pack();
}

private void botonSoltado(java.awt.event.MouseEvent evt) {

    depura("Boton soltado");

    Point punto = evt.getPoint();
    xFinTemp = punto.x;
    yFinTemp = punto.y;

    objetoLinea nElemento = new objetoLinea(xIniTemp,yIniTemp,xFinTemp,yFinTemp);

    Graphics g = this.getGraphics();
    nElemento.pinta(g);
    g.dispose();

    nuestrosObjetos.insertaElemento(nElemento);

    // Add your handling code here:
}

private void botonPulsado(java.awt.event.MouseEvent evt) {
    // Add your handling code here:
    depura("Boton pulsado");
    Point punto = evt.getPoint();
    xIniTemp = punto.x;
    yIniTemp = punto.y;
}

/** Exit the Application */
private void exitForm(java.awt.event.WindowEvent evt) {
    System.exit(0);
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    editorGrafico miapp = new editorGrafico();
    miapp.resize(600,400);
    miapp.show();
}

public void paint(java.awt.Graphics pContexto)
{
    nuestrosObjetos.pintaTodo(pContexto);
}

// Variables declaration - do not modify
// End of variables declaration
}

class listaObjetos implements Serializable
{
    private Vector arrayObjetos = new Vector();

    public void insertaElemento(objetoGenerico pElemento)
    {
        arrayObjetos.add(pElemento);
    }

    public void pintaTodo(Graphics pContexto)

```

```

    {
        for (int i = 0 ; i< arrayObjetos.size(); i++)
        {
            objetoGenerico elemento = (objetoGenerico) arrayObjetos.get(i);
            elemento.pinta(pContexto);
        }
    }
}

abstract class objetoGenerico implements Serializable
{
    abstract public void pinta(Graphics pContexto);
}

class objetoLinea extends objetoGenerico
{
    int xorigen = 0;
    int yorigen = 0;
    int xdestino = 0;
    int ydestino = 0;

    objetoLinea(int pXorigen, int pYorigen,int pXdestino, int pYdestino)
    {
        xorigen = pXorigen;
        yorigen = pYorigen;
        xdestino = pXdestino;
        ydestino = pYdestino;
    }

    public void pinta(Graphics pContexto)
    {
        pContexto.drawLine(xorigen,yorigen,xdestino,ydestino);
    }
}

```

Lo que hemos hecho es crear un objeto que representa todo lo que vamos pintando

```
private listaObjetos nuestrosObjetos = new listaObjetos();
```

Este objeto tiene dentro un vector, que es quien irá agragando los puntos

```
private Vector arrayObjetos = new Vector();

public void insertaElemento(objetoGenerico pElemento)
{
    arrayObjetos.add(pElemento);
}

```

De tal modo que nuestra aplicación, cuando el usuario pulsa el boton, captura las coordenadas iniciales y cuando lo suelta, captura las finales, creando un objeto (en este caso una línea) y añadiendolo al Vector

```
private void botonSoltado(java.awt.event.MouseEvent evt) {
    depura("Boton soltado");

    Point punto = evt.getPoint();
    xFinTemp = punto.x;
    yFinTemp = punto.y;

    objetoLinea nElemento = new objetoLinea(xIniTemp,yIniTemp,xFinTemp,yFinTemp);

    Graphics g = this.getGraphics();
    nElemento.pinta(g);
    g.dispose();

    nuestrosObjetos.insertaElemento(nElemento);

    // Add your handling code here:
}

private void botonPulsado(java.awt.event.MouseEvent evt) {
    // Add your handling code here:
    depura("Boton pulsado");
    Point punto = evt.getPoint();
    xIniTemp = punto.x;
    yIniTemp = punto.y;
}

```

Cuando hay que repintar la pantalla, los que hacemos es decir que se repinten todos los objetos

```
public void paint(java.awt.Graphics pContexto)
{
    nuestrosObjetos.pintaTodo(pContexto);
}

```

Esta función itera por todos los elementos y los manda pintar

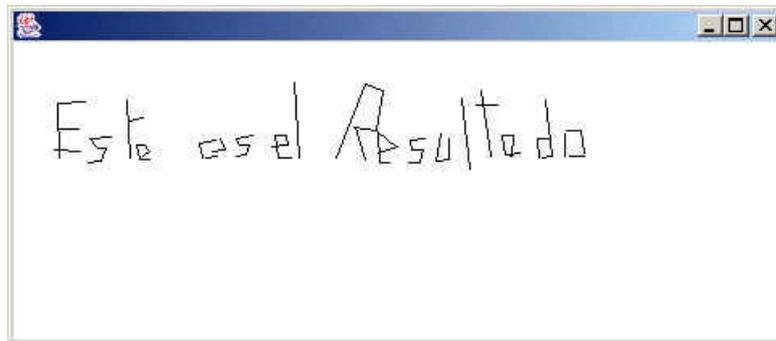
```
public void pintaTodo(Graphics pContexto)
{
    for (int i = 0 ; i< arrayObjetos.size(); i++)
    {
        objetoGenerico elemento = (objetoGenerico) arrayObjetos.get(i);
        elemento.pinta(pContexto);
    }
}
```

Si veis ... lo hemos dejado preparado para poder leerlo y escribirlo en disco

```
class listaObjetos implements Serializable
```

Y para poder .. modificando 4 lineas pintar distintos tipos de objetos (uso de clase abstracta)

```
abstract class objetoGenerico implements Serializable
{
    abstract public void pinta(Graphics pContexto);
}
```



[Descargarse el código](#)

[Sobre el Autor..](#)

Si desea contratar formación, consultoría o desarrollo de piezas a medida puede contactar con

J2EE, EJBs, Struts...

[Autentia S.L.](#) Somos expertos en:
J2EE, C++, OOP, UML, Vignette, Creatividad ..
 y muchas otras cosas

Nuevo servicio de notificaciones

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales, inserta tu dirección de correo en el siguiente formulario.

Subscribirse a Novedades	
e-mail	<input type="text"/>
	<input type="button" value="Enviar"/>

Otros Tutoriales Recomendados ([También ver todos](#))

Nombre Corto

[Primeros pasos con Swing](#)

[SAX, DOM y NetBeans](#)

Descripción

Os mostramos como crear una aplicación básica utilizando las librerías de Java Swing

En este tutorial os vamos a mostrar como manipular documentos XML desde Java y NetBeans

Os mostramos como crear librerías de etiquetas para vuestros JSP y así simplificar su

[TagLibs y JSPs](#)

construcción.

[Decompilar Java](#)

Os mostramos como recuperar el fuente de vuestro código a partir de los ficheros compilados .class

Nota: Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento.

Los contenidos y comentarios de los tutoriales son responsabilidad de sus respectivos autores.

En algún caso se puede hacer referencia a marcas o nombres cuya propiedad y derechos es de sus respectivos dueños. Si algún afectado desea que incorporemos alguna reseña específica, no tiene más que solicitarlo.

Si alguien encuentra algún problema con la información publicada en este Web, rogamos que informe al administrador rcanales@adictosaltrabajo.com para su resolución.

[Patrocinados por enredados.com Hosting en Castellano con soporte Java/J2EE](#)



www.AdictosAlTrabajo.com Optimizado 800X600