

# Tutorial Javafx + Inkscape + Netbeans

Primer tutorial Javafx en [adictosalainformatica.com](http://adictosalainformatica.com).

JavaFX es un lenguaje de programación de scripts desarrollado por SUN , que nos permite hacer aplicaciones para móviles , televisores , aplicaciones de escritorio (ventana) , y Applets (incrustados en páginas web).

[Inkscape](#) (Programa de diseño gráfico vectorial) , nos permite crear un gráfico y exportarlo como código javafx .

Este será el primero de una serie de tutoriales JavaFX , y el primero también de Inkscape.

Utilizaremos [Netbeans 6.9 con JavaFX instalado](#) .

Primero crearemos un proyecto en Netbeans en File/New Project/JavaFX/JavaFX Script Application.

Le ponemos el nombre que queramos

Esto nos genera un proyecto con una carpeta src , un fichero main.fx , lo modificamos para que quede así:

```
Stage {
title: "Adictos a la informática" //cabecera de la aplicación
scene: Scene {
width: 705
height:500
content: [
Text { //caja de texto , opcional
font : Font {
size : 16
}
effect: Reflection { //Opcional , para aplicarle reflejo al
texto
fraction: 0.75
```

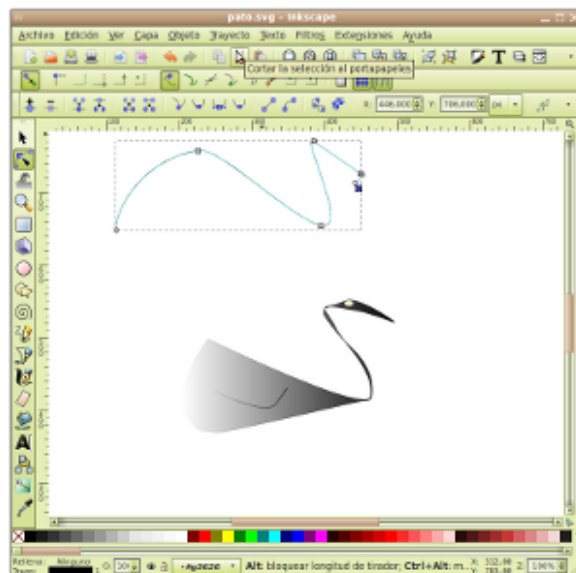
```

topOffset: 0.0
topOpacity: 0.5
bottomOpacity: 0.0
}
x: 10
y: 10
content: "Scuraki" //texto Poned lo que querais , opcional
}
]
}
}

```

Podemos probar como se ve pulsando el botón de run o el de debug.

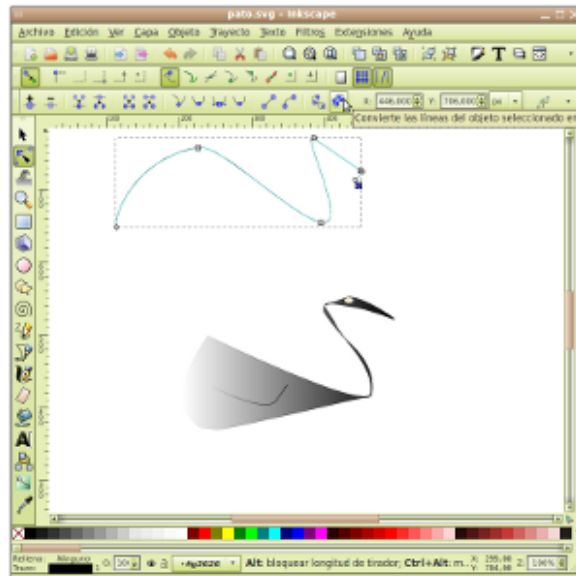
Pasamos a inkscape , y realizamos un gráfico .  
 En mi caso he dibujado un pato en 1,5 minutos , he trazado una curva,  
 después pulsando el botón de convertir las líneas del objeto seleccionado en trayectos ,  
 esto nos permite seleccionar los nodo y crear una figura .Es posible que tengamos demasiados nodos y nos sea difícil , si dejamos pocos nodos nos será más fácil , seleccionamos los que queremos borrar y pulsamos la tecla Supr.



## Pantallazo-pato.svg - Inkscape

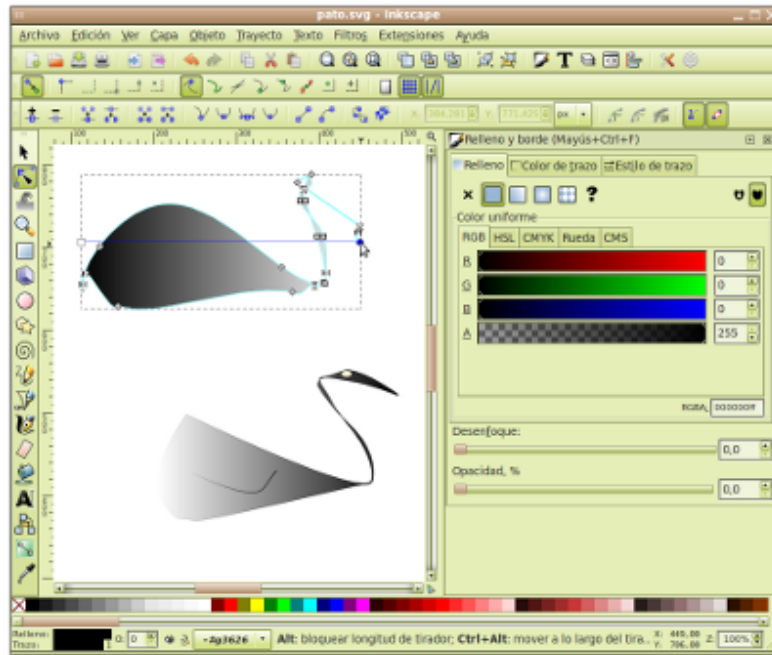
Para colorear nuestro dibujo en el menú objeto abrimos la opción relleno y borde.

Esto nos abre el cuadro de colores , donde nos permite cambiar el color , utilizar un degradado lineal , un degradado circular , o un patrón .Seleccionamos el degradado lineal .



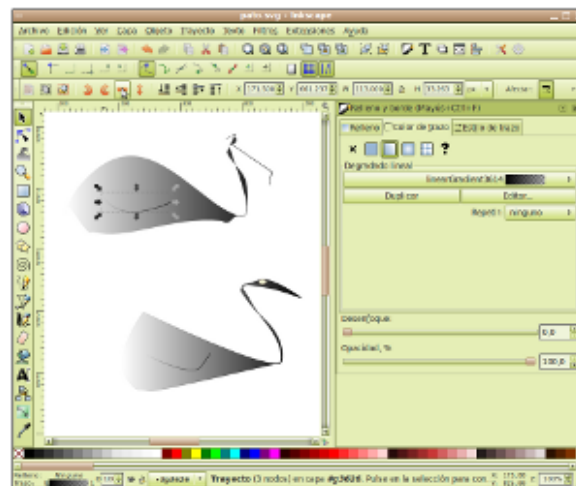
## Pantallazo-pato.svg - Inkscape-1

Si queremos podemos girar el degradado como mejor nos quede , en el modo de edición de nodos, en el degradado se ven 2 nodos enlazados con una línea , si movemos los nodos movemos el degradado.



Pantallazo-pato.svg - Inkscape-3

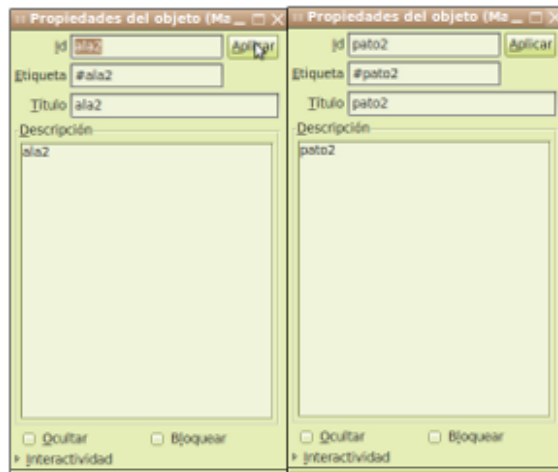
Bueno , tenemos una figura en mi caso es como un pato , le pondré un ala , para que de mejor impresión, dibujo una curva y pongo el degradado que he utilizado con el pato , para ahorrar y la situo en el pato , si es necesario se puede voltear girar , rotar .



Pantallazo-pato.svg - Inkscape-6

Para un mejor seguimiento de las partes de nuestro diseño , con botón derecho en el pato le damos a propiedades , y le llamamos pato , hacemos lo propio con el ala , y la llamamos

ala.



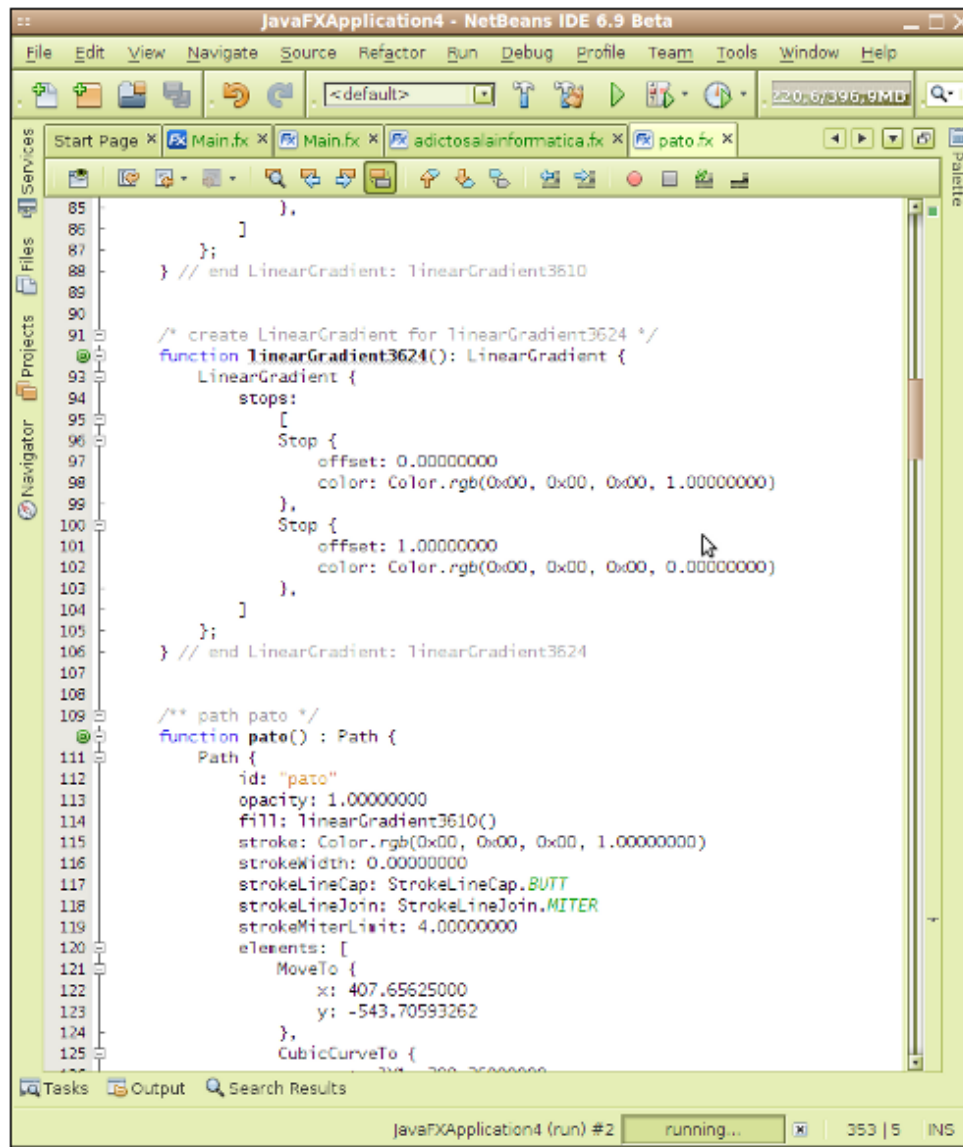
Pantallazo-Propiedades del objeto

Seleccionamos los 2 objetos y en el menú objeto seleccionamos agrupar (Alt + G también agrupa).

Bueno , ya tenemos un pato agrupado , ahora lo vamos a convertir en código JavaFX , guardamos el documento como svg de inkscape para no perderlo, y volvemos a guardar , pero esta vez lo guardaremos como fichero .fx (desplegar el combo y seleccionar .fx de javafx) , en la carpeta src de nuestro proyecto de Netbeans que hemos creado antes.

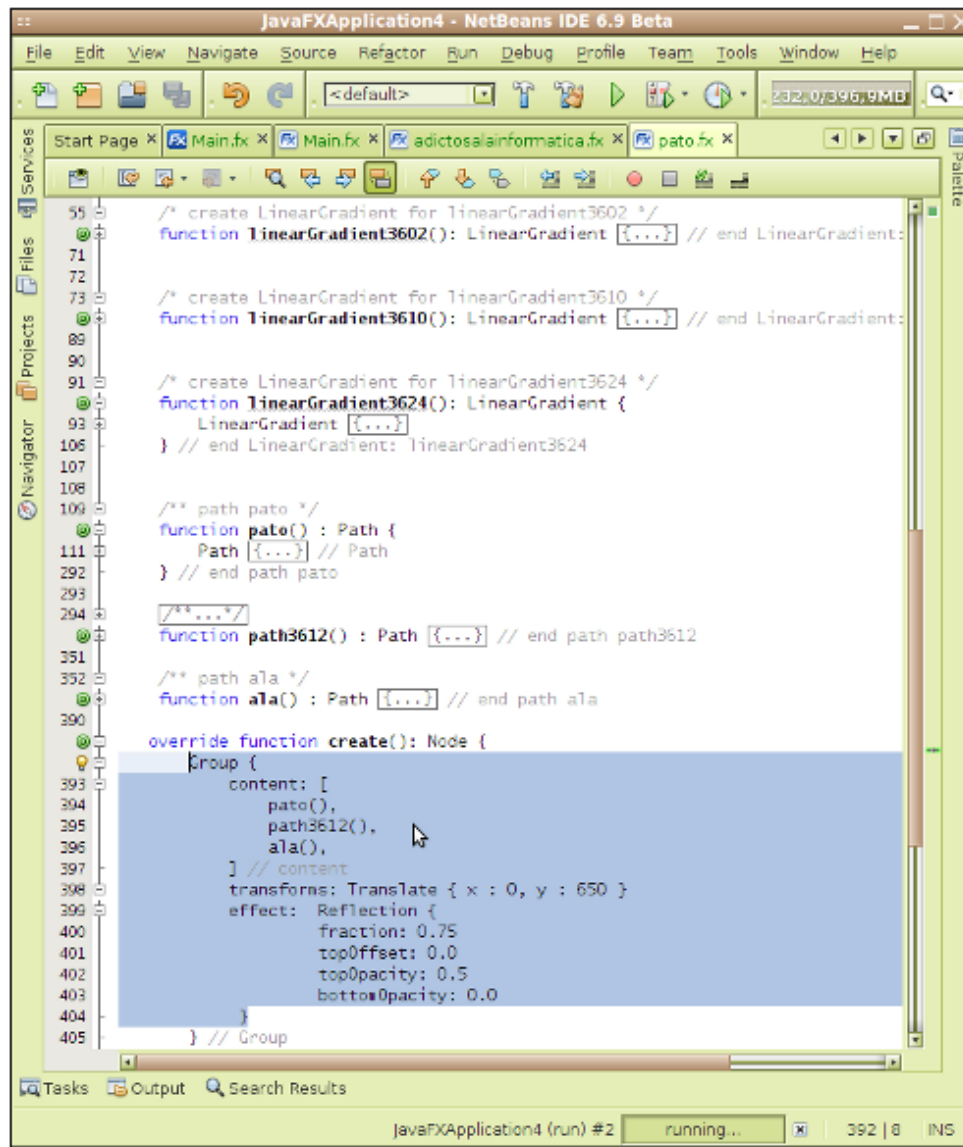
Volvemos a Netbeans , y veremos nuestro fichero fx , en mi caso pato.fx.

Editamos el fichero pato.fx , y observamos que hay una clase pato que extiende de CustomNode con un método para cada parte de nuestro dibujo , el pato , el ala , los degradados que hemos creado , un un grupo .



## Pantallazo-JavaFXApplication4 - NetBeans IDE 6.9 Beta-1

También hay un método que se llama create que es el encargado de devolver el contenido , en el group , le aplicaremos un efecto de reflejo , para que quede más bello.



Pantallazo-JavaFXApplication4 - NetBeans IDE 6.9 Beta-2

```
Group { //grupo
content: [
pato(), //dibuja el pato
path3612(), //esto varia segun el diseño
ala(), //dibuja el ala
] // content
transforms: Translate { x : 0, y : 650 } //posicionamos el
grupo en la escena , viene con valores
effect: Reflection { //efecto de reflejo
fraction: 0.75
topOffset: 0.0
```

```
topOpacity: 0.5
bottomOpacity: 0.0
}
} // Group
```

Ahora vamos al fichero main.fx , donde llamaremos a nuestro pato para que se pinte por pantalla.

Creamos un objeto de tipo Node , var pato:Node; , y le asignamos el valor creando un objeto pato , pato=new pato(); .

Añadimos el pato a la escena

Ejecutamos el proyecto , y graduamos el ancho de la escena en main.fx Scene , y la posición del gráfico donde os comenté arriba.

En mi caso tengo también 2 gráficos , me quedó así:

```
package javafxapplication4;

import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.text.Text;
import javafx.scene.text.Font;
import javafx.scene.Node;
import javafx.scene.effect.Reflection;

/**
 * @author scuraki
 */
var adicto:Node;
adicto=new adictosalainformatica();
var patito:Node;
patito=new pato();
Stage {
title: "Adictos a la informática"

scene: Scene {
width: 705
height:500
```



```

content: [
Text {
font : Font {
size : 10
}
effect: Reflection {
fraction: 0.75
topOffset: 0.0
topOpacity: 0.5
bottomOpacity: 0.0
}
x: 10
y: 10
content: "Scuraki"
},adicto,patito //añade los gráficos vectoriales
]
}
}
}

```



Pantallazo-Adictos a la informática

Bueno hasta aquí el tutorial , se puede apreciar algunos errores en el Path del ala y en la posición del degradado , pero puede ser útil ¿no os parece? , espero que os haya resultado interesante , hasta la próxima.