[**Animación con Timer en Java y Netbeans**](http://jc-mouse.blogspot.com/2010/01/animacion-con-timer-en-java-y-netbeans.html)

Se puede realizar animación en java utilizando el objeto Timer el cual realiza operaciones cada cierto tiempo indicado por un temporizador. En este caso la operación a realizar será la de presentar en pantalla una secuencia de imágenes JEPG las cuales simular la animación.

La animación consiste en una japo comiendo, ayudándose de un jSlider se podrá también acelerar o retardar la secuencia de imágenes

El proyecto consta de un JFRame con la siguiente distribucion de objetos

[](http://2.bp.blogspot.com/_aQGkfw6J-yo/S0POIuiy-8I/AAAAAAAAB-8/sTXFyr8OYw4/s1600-h/netb-form.jpg)

De una clase, la encargada de la animación animationclass.java, esta clase consiste en extender la propiedad **jLabel**y de añadirle un **Timer**, el cual es el encargado de la animación.  
  
El código de la clase animationclass.java es:

package animation;

public class Main {

public static void main(String[] args) {

// TODO code application logic here

form1 f = new form1();

f.show();

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

package animation;

import java.util.Timer;

import java.awt.\*;

import java.util.TimerTask;

public class animationclass extends javax.swing.JLabel {

private Timer tiempo ;

private TimerTask task;

private int speed = 60;

private int frame=0;

boolean run = false;

//se crea el objeto y se redimensiona al tamaño del contenedor

public animationclass(Dimension d){

this.setSize(d);

}

//coloca una imagen en el objeto jlabel

public void setimage(int n){

this.setIcon(new javax.swing.ImageIcon(getClass().getResource("/img/japo\_" + n + ".jpg")));

}

//coloca la velocidad en la variable

public void setvelocidad(int v){

this.speed= v;

}

//para la animacion

public void startAnimation() {

run=true;

tiempo = new Timer();

task = new TimerTask() {

public void run() {

frame++;

if (frame<=10){

setimage(frame);

}

else{frame=0;}

}

};

//se inicia la animacion

//System.out.println("Se inicia la animacion");

tiempo.schedule(task,0,speed);

}

//detiene la animacion

public void stopAnimation() {

tiempo.cancel();

task.cancel();

run=false;

//System.out.println("La animacion fue detenida");

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

package animation;

public class form1 extends javax.swing.JFrame {

/\*\* Creates new form form1 \*/

public form1() {

initComponents();

this.setLocationRelativeTo(null);

this.setTitle("Animacion Java - by mouse");

a = new animationclass(jPanel1.getSize());

jPanel1.add(a);

jPanel1.repaint();

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

if (jButton1.getText().equals("PLAY")){

jButton1.setText("STOP");

a.startAnimation();

}

else

{

a.stopAnimation();

jButton1.setText("PLAY");

}

}

private void jSlider1StateChanged(javax.swing.event.ChangeEvent evt) {

int v = jSlider1.getValue();

if (a.run==true){

a.stopAnimation();

jButton1.setText("PLAY");

}

a.setvelocidad(v);

jButton1.setText("STOP");

a.startAnimation();

}

El funcionamiento de la clase animationclass.java es la de crear un jLabel, otorgarle mediante el constructor el tamaño del contenedor (jPanel) donde se lo colocara, el método **setImage** coloca mediante la propiedad **setIcon**una imagen (de las 10 que se encuentran el package IMG) en el jlabel, el método encargado de la animación es **startAnimation()**, el cual crea el Timer correspondiente el cual ira incrementado la variable frame y hará un llamado al método **setimage()**, para actualizar la imagen del jLabel.

**La aplicación terminada**

[](http://2.bp.blogspot.com/_aQGkfw6J-yo/S0PP6n4yFdI/AAAAAAAAB_E/iH9979HF0jg/s1600-h/animacion-java-netbeans.jpg)