

¿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)



[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 echarle 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>	 <p>autentia real business solutions</p>
---	---	--

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

Curso Web J2EE

Curso Avanzado en Desarrollo Web con J2EE

applet test

Software to test Java Applets inside website automatically

Master Java Élite Madrid

Master Java-J2ee-ejbs profesional. Prácticas en empresas. Becas. Exes.

Formación Empresas

Consultoría de Formación Tecnologías Web

Anuncios Goooooogle

Anunciarse en este sitio

Desarrollo de Applets

Uno de los desarrollos que más ha contribuido al arranque de Java, fue el desarrollo de las llamadas Applets.

Un applet es un pequeño componente Java de se enlaza a una página Html.

Al descargarse un navegador nuestra página Html, detecta la etiqueta que identifica al applet y arranca una máquina virtual. Posteriormente, solicita los ficheros necesarios para la ejecución e inicializa el programa.

Hay un punto clave que hay que tener en cuenta para que las applets funcionen... la configuración de seguridad del navegador ...

Los applets son elementos que se deben utilizar con cuidado porque mucha gente es reacia a autorizar que se ejecute su código... además, es muy posible que en Internet con líneas de comunicación lentas... no sea muy factible su utilización.

[Otro tutorial que tenemos sobre JFreeChart](#) ha presentado mucho interés... y mucha gente me ha preguntado como usarlo desde un applet.

En un primer paso... vamos a hacer una cosa más curiosa seguir usando nuestro servlet para generar las gráficas dinámicamente en el servidor (y no tener que distribuir las clases a través de internet... y desde un applet llamar a este servlet para que nos genere la grafica deseada... es decir, la imagen.

El código de nuestro applet es bastante sencillo...

La dinamica es sencilla

Extendemos una clase de applet

```
public class appletsimple extends Applet implements ImageObserver
```

Al arrancar el applet, cargamos la imagen

```
URL trayectoBase = this.getCodeBase();
depura("El trayecto base es: " + trayectoBase.toString());

URL trayectoImagen = new URL(trayectoBase,"imagen.jpg");

this.getAppletContext().showStatus(trayectoImagen.toString());
depura("El fichero es " + trayectoImagen.toString());

iMiGrafica = getImage(trayectoImagen);
```

Y pintamos en pantalla la imagen

```
g.drawImage(iMiGrafica,50,60,this);
```

```

/*
 * appletsimple.java
 *
 * Created on 21 de septiembre de 2003, 16:06
 */

package appletroberto;

/**
 *
 * @author Administrator
 */

import java.awt.*;
import java.applet.*;
import java.io.*;
import java.net.*;
import java.awt.image.*;

public class appletsimple extends Applet implements ImageObserver
{
    Image iMiGrafica = null;

    public void init()
    {
        depura("Inicializamos el applet");
    }

    public void paint(Graphics g)
    {
        g.drawString("Este es el mensaje", 50,50);

        depura("Pintamos cadena");
        g.drawImage(iMiGrafica,50,60,this);
    }

    public void start() {
        depura("Arranque de la App");

        try
        {
            URL trayectoBase = this.getCodeBase();
            depura("El trayecto base es: " + trayectoBase.toString());

            URL trayectoImagen = new URL(trayectoBase,"imagen.jpg");

            this.getAppletContext().showStatus(trayectoImagen.toString());
            depura("El fichero es " + trayectoImagen.toString());

            iMiGrafica = getImage(trayectoImagen);
        }
        catch (Exception e)
        {
            depura("Error al cargar Imagen");
        }
    }

    public boolean imageUpdate(Image img, int infoflags, int x, int y, int width, int height)
    {
        if (infoflags == ImageObserver.ALLBITS)
        {
            depura("Ya se ha cargado");
            repaint();
        }

        return true;
    }

    void depura(String pCadena)
    {
        System.out.println("Mensaje: " + pCadena );
    }
}

```

Ahora creamos una página html con la etiqueta adecuada

```

<html>
<body>

<p align="center">
<applet width="300" height="300" code="appletroberto.appletsimple.class"
codebase="."/>Applet que muestra imagen simple</applet>
</p>

</body>
</html>

```

Para probar nuestro programa ... ejecutamos la aplicación de prueba **appletviewer** que viene con el JSDK

```

C:\WINNT\system32\cmd.exe - appletviewer pruebaapplet.html

C:\java\appletbasico>appletviewer pruebaapplet.html
Mensaje: Inicializamos el applet
Mensaje: Arranque de la App
Mensaje: El trayecto base es: file:/C:/java/appletbasico/
Mensaje: El fichero es file:/C:/java/appletbasico/imagen.jpg
Mensaje: Pintamos cadena
Mensaje: Ya se ha cargado
Mensaje: Pintamos cadena

```

Y el resultado del programa es:



Ahora, si queremos que nuestro applet utilice el servlet que genera la gráfica, solamente debemos cambiar la URL para llamar al servlet.

Ahora modificamos nuestro servlet para pasarle los datos que queremos pintar

```

/*
 * servletGeneradorGraficaDinamica.java
 *
 * Created on 21 de septiembre de 2003, 17:46
 */
import java.io.*;
import java.net.*;

import javax.servlet.*;
import javax.servlet.http.*;

import org.jfree.data.*;
import org.jfree.chart.*;
import org.jfree.chart.plot.*;

/**
 *
 * @author Roberto Canales
 * @version
 */
public class servletGeneradorGraficaDinamica extends HttpServlet
{
    public JFreeChart crearChart(int array[])
    {
        XYSeries series = new XYSeries("Evolucion Sesiones");

        // inicializamos los valores pasados
        for (int i=0; i<array.length; i++)
        {
            series.add(i, array[i]);
        }

        XYDataset juegoDatos= new XYSeriesCollection(series);

        JFreeChart chart = ChartFactory.createLineXYChart("Sesiones en Adictos al Trabajo",
        "Meses", "Sesiones", juegoDatos,
        PlotOrientation.VERTICAL,
        true,true,true
        );

        return chart;
    }

    int getParamEntero(HttpServletRequest request,String pNombre, int pDefecto)
    {
        String param = request.getParameter(pNombre);

        if (param == null || param.compareTo("") == 0)
        {
            return pDefecto;
        }
    }
}

```

```

return Integer.parseInt(param);
}

/** Processes requests for both HTTP GET and POST methods.
 * @param request servlet request
 * @param response servlet response
 */
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    response.setContentType("image/jpeg");

    OutputStream salida = response.getOutputStream();

    int ancho = getParamEntero(request,"ancho",300);
    int alto = getParamEntero(request,"alto",300);

    // esto es una ñapa paro de momento solo queremos demostrar que funciona
    int iNumeroElemento = getParamEntero(request,"elementos",3);

    int array[] = new int[iNumeroElemento];

    // leemos variables al estilo y0=1 y1=2 y3=22
    for(int i = 0; i < array.length; i++)
    {
        array[i] = getParamEntero(request,"y" + i ,0);
    }

    JFreeChart grafica = crearChart(array);
    ChartUtilities.writeChartAsJPEG(salida,grafica,ancho,alto);

    salida.close();
}

/** Handles the HTTP GET method.
 * @param request servlet request
 * @param response servlet response
 */
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    processRequest(request, response);
}
}

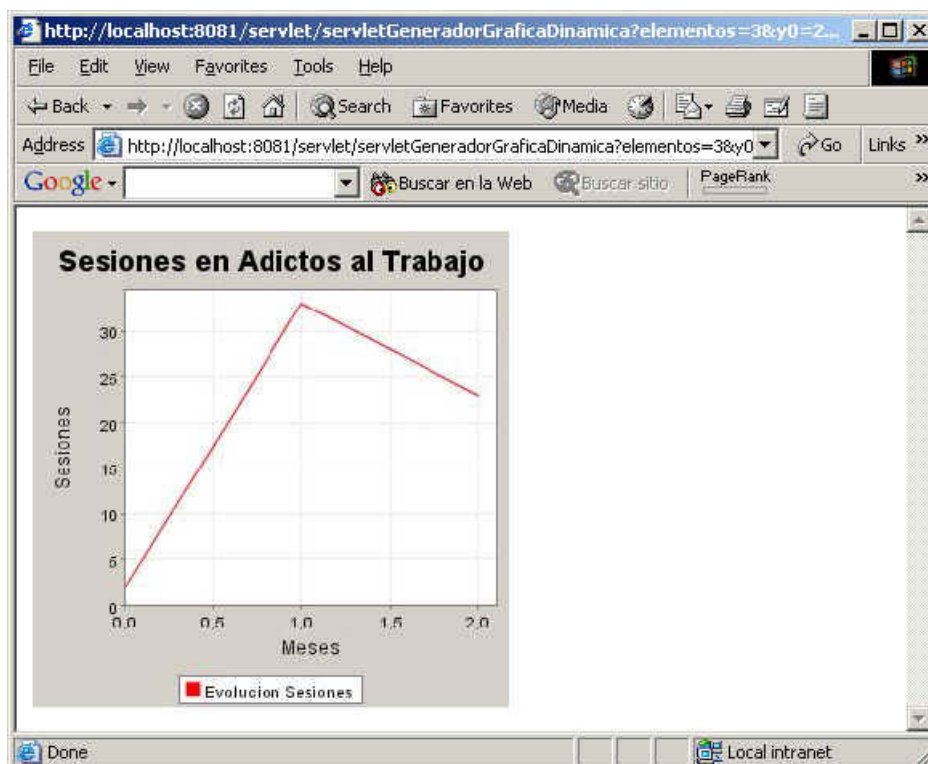
```

Vamos a probar el servlet por separado

Solicitamos nuestra URL

<http://localhost:8081/servlet/servletGeneradorGraficaDinamica?elementos=3&y0=2&y1=33&y2=23>

Y vemos el resultado



Y ahora readaptamos el applet para solicitar una imagen en la URL.

```
public void start() {
    depura("Arranque de la App");

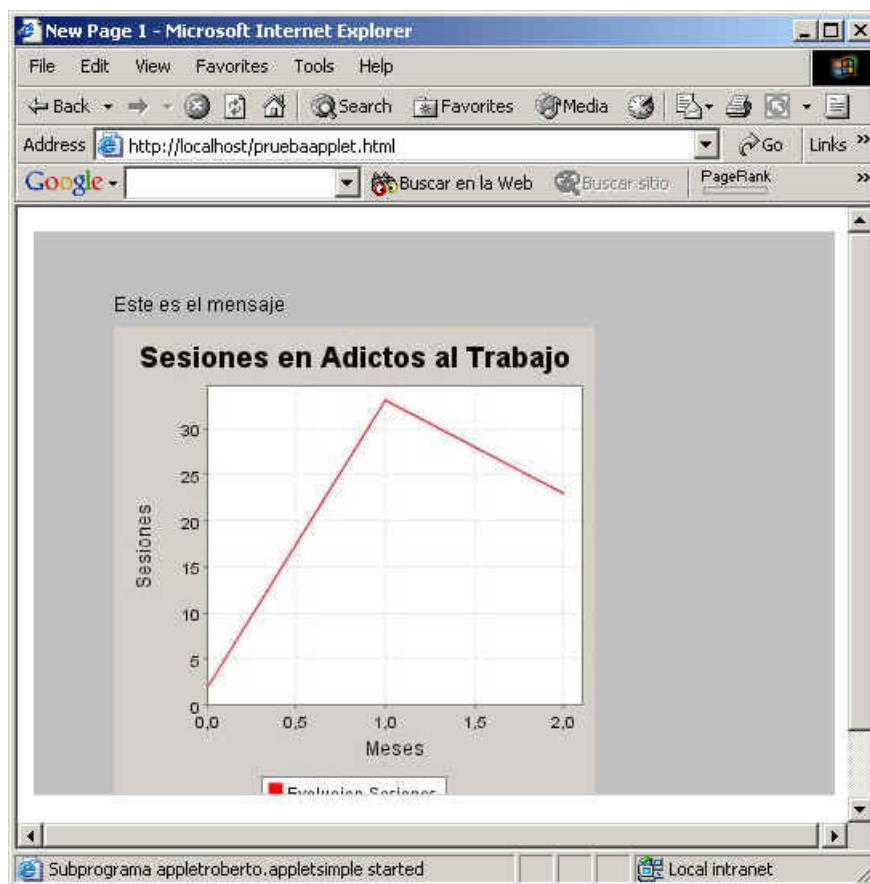
    try
    {
        URL trayectoBase = this.getCodeBase();
        depura("El trayecto base es: " + trayectoBase.toString());

        //URL trayectoImagen = new URL("http://localhost:8081/servlet/servletGeneradorGraficaDinamica?
        //elementos=3&y0=2&y1=33&y2=23&ancho=200&alto=300"); Tambien podriamos indicar tamaño

        URL trayectoImagen = new URL("http://localhost:8081/servlet/servletGeneradorGraficaDinamica?elementos=3&y0=2&y1=33&y2=23");

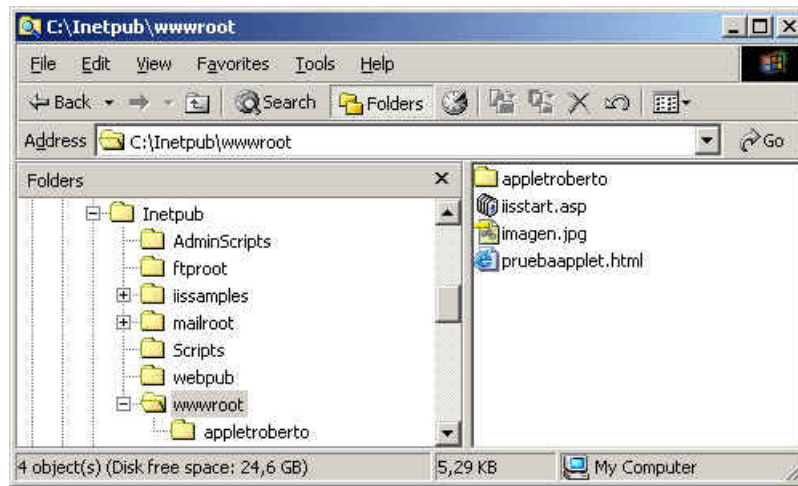
        this.getAppletContext().showStatus(trayectoImagen.toString());
        depura("El fichero es " + trayectoImagen.toString());

        iMiGrafica = getImage(trayectoImagen);
    }
    catch (Exception e)
    {
        depura("Error al cargar Imagen");
    }
}
```



Para que esto funcione en un navegador Web externo... debemos tener la precaución de copiar todos los ficheros del applet y la página HTML al directorio Web de nuestro servidor (pero no copiar los .java)

Es probable que en vuestra página Web no funcione bien mirar las opciones de **seguridad** y ya hablaremos de la **firma digital de applets** para evitar problemas peroeso será en otro tutorial ...



La verdad es que es bastante sencillo ¿verdad?

[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="button" value="Enviar"/>

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

Nombre Corto

[Introducción al Java Web Start](#)

[Firma digital de un Applet](#)

[Generar imagenes desde Servlets](#)

[Upload de ficheros en Java](#)

[Gráficas en Java con JFreeChart](#)

Descripción

Os mostramos como podéis distribuir aplicaciones de consola de un modo sencillo a través de Java Web Start. Es el mismo principio de los applets aplicado a aplicaciones.

Para que un applet Java pueda ejecutarse en un cliente Web con la configuración de seguridad por defecto y/o adquirir privilegios de seguridad, es necesario firmarla digitalmente. Alejandro Perez nos enseña como hacerlo de un modo rápido y sencillo.

Os mostramos como generar ficheros GIF desde un servlet java. Util para generear gráficas dinámicas, contadores, etc

Os mostramos como enviar ficheros a un servidor Web y manipularlos en un servlet en el servidor, gracias a APIs de apache

Os mostramos como generar gráficas profesionales, en aplicaciones y servlets, en Java con la librería gratuita JFreeChart

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](http://www.enredados.com) [Hosting en Castellano con soporte Java/J2EE](#)



www.AdictosAlTrabajo.com Optimizado 800X600