Teknoda

Notas técnicas de JAVA Nro. 6 - White Paper

(Lo nuevo, lo escondido, o simplemente lo de siempre pero bien explicado)

JAVA "Basics": Entendiendo las applets

Tema: Applets, servlets, JVM,

Descripción: Este tip define el modelo de programación de applets, confrontándolo con otros tipos de programa JAVA, y explica los fundamentos, métodos y permisos en torno a las applets.

Nivel: Intermedio

Fecha pub: Febrero 2005

---

*"Notas Técnicas de JAVA" se envía con frecuencia variable y absolutamente **sin cargo** como un servicio a nuestros clientes. Contiene notas/recursos/artículos técnicos desarrollados en forma totalmente objetiva e independiente. Teknoda es una organización de servicios de tecnología informática y **NO comercializa hardware, software ni otros productos**. Si desea suscribir otra dirección de e-mail para que comience a recibir los tips envíe un mensaje desde esa dirección a develop@teknoda.com, indicando su nombre, empresa a la que pertenece, cargo y país.*

Lista de Tips publicados hasta la fecha:

1. JAVA Basics: Cómo conformar un entorno de programación JAVA (serie de varios tips). Parte I: Selección e instalación de un IDE gratuito.
2. Una introducción a JDBC (Java Database Connectivity) (Acceso a bases de datos desde JAVA)
3. Manejo del error "Bad Magic Number"
4. Java Basics: Entendiendo la Java Virtual Machine
5. Organización de memoria en JAVA Vs. Modelo Tradicional

Próximos Tips:

Nivel Técnico avanzado

- JAVA Vs. C++

Nivel Básico

- JAVA Basics: Entendiendo los servlets
- JAVA Basics: Mitos y Verdades sobre JAVA

Tabla de contenido

El objetivo del tip es explicar los fundamentos de las applets.. Los siguientes puntos serán cubiertos en el mismo :

I. Introducción

Las *applets* (término de origen francés que significa mini-aplicación) son mini-programas escritos en Java que ejecutan dentro de un Web Browser, motorizados por la Java Virtual Machine embebida dentro del browser.

Precisamente por las características propias del lenguaje JAVA, el código JAVA puede resolverse en cualquier entorno dotado de una JAVA Virtual Machine que lo interprete. (Ver Tip anterior "Java Basics: Entendiendo la Java Virtual Machine"). Esto hace que existan múltiples contextos y situaciones donde es posible ejecutar código JAVA, cada una con características particulares, y distintos modelos de programación, según se trate de Applets, Servlets, o Aplicaciones.

Las Applets son un tipo de programa muy típico del ambiente JAVA, y fueron determinantes en la difusión de este lenguaje, como un método de distribución de programas en la Web. Sirven, entre otras cosas, para "dar vida" a las páginas (interacción en tiempo real, inclusión de animaciones y sonidos, siempre dentro del formato de una pequeña ventana).

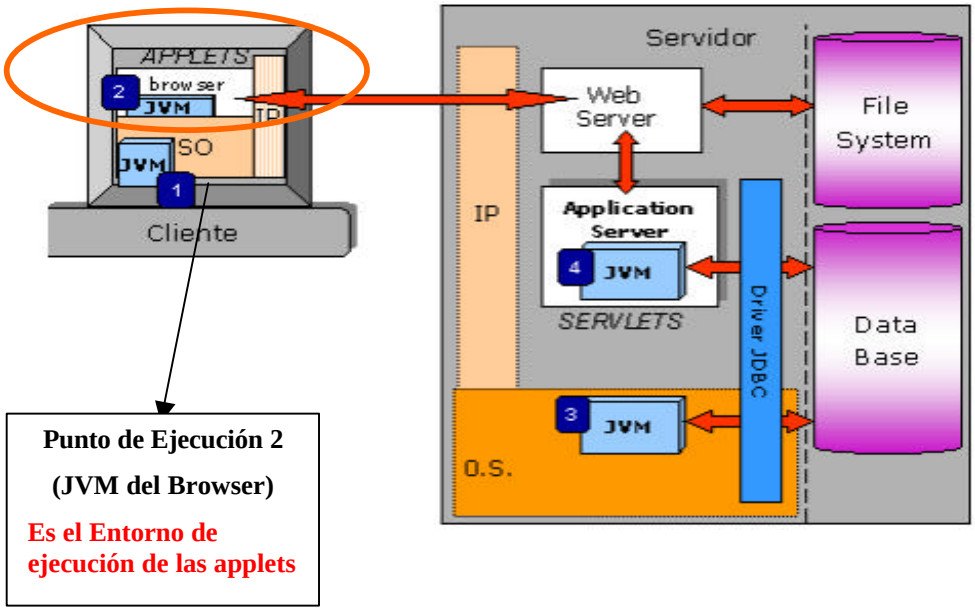
Las applets se cargan desde una determinada URL, y se ejecutan luego en el navegador. Para que esto ocurra tan sólo *hace falta que el navegador sea capaz de interpretar Java (Java Enabled)*. Al ser ejecutados en la **máquina cliente**, por lo tanto no existe disminución de la velocidad de transmisión por la saturación del módem o del ancho de banda.

A las páginas que contienen *applets* se las denomina páginas *Java-Powered*. Las *applets* pueden ser visualizadas con la herramienta *appletviewer*, incluido en el JDK de Java.

Existen no pocas controversias sobre la conveniencia de utilizar applets, frente a otras formas de programación JAVA, como las aplicaciones o las servlets. Por ello, antes de profundizar las características de las applets, es interesante diferenciarlas y contrastarla éstas últimas.

II. Applets, aplicaciones y servlets: Puntos de ejecución de JAVA

En el gráfico "*Puntos de Ejecución en Java*", se visualizan los distintos puntos de ejecución de todas las aplicaciones JAVA y se destaca específicamente el entorno de ejecución correspondiente a las aplicaciones applets.



Puntos de ejecución en Java *Copyright Teknoda S.A*

1. Aplicaciones Java en el cliente (JVM del OS)

Comúnmente, son aplicaciones de escritorio que corren en el cliente, utilizan la **JVM "local"**. Las mismas pueden ser ejecutadas mediante un archivo del tipo .bat o un archivo .exe. La JVM se instala ad-hoc, o está provista como parte del sistema operativo.

2. Applets Java (JVM del Browser)

Las Applets son aplicaciones web que se ejecutan en el **JVM local del Web Browser** . Son pequeñas ventanas que se despliegan del lado de cliente. Por ejemplo, Internet Explorer, o Netscape Navigator tienen incorporada una JVM como parte de su funcionalidad. Las applets son comunes en aplicaciones acotadas, distribuidas generalmente a través de Internet.

3.Aplicaciones Java en el Server (JVM del OS)

Estas " Aplicaciones Web " corren gracias a la JVM del lado del Servidor, es decir el cliente va poder utilizar esta " Aplicación Web " independientemente que posea o no una máquina virtual de Java. Es habitual que este tipo de aplicaciones estén conectadas a una base de datos a través de una conexión JDBC

Para más información (ver tip *Una introducción a JDBC (Java Database Connectivity) Acceso a bases de datos desde JAVA*)

4.Servlets Java (JVM del Application Server)

Las Servlets se ejecutan también en la JVM del Application Server del lado del Servidor, pero son gestionadas por requerimientos emanados del cliente, a través de una URL. **Requieren de un software específico, denominado JAVA WEB APPLICATION SERVER**. Nuevamente se repite la situación de independencia de la JVM del lado del cliente. El cliente puede trabajar con la aplicación desarrollada con servlets sin necesidad de tener instalada la maquina virtual de Java .

Como se aprecia en el gráfico, las *applets* no son exactamente aplicaciones Java. Presentan las siguientes diferencias respecto a las aplicaciones tradicionales Java:

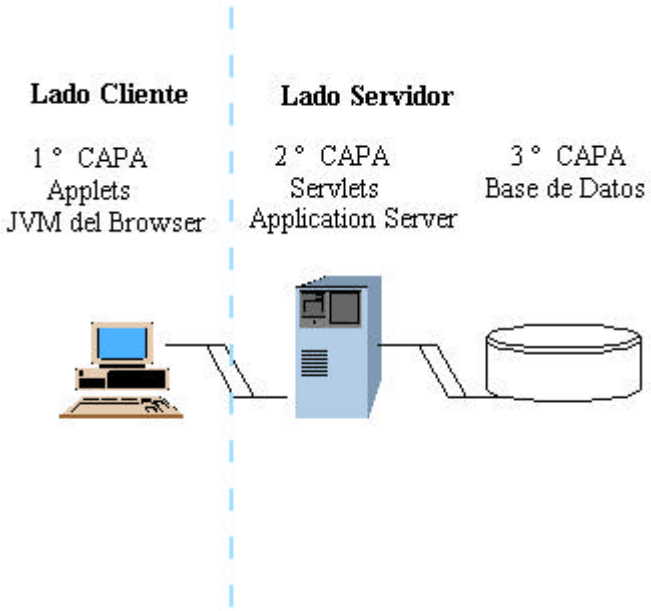
- Se cargan mediante un navegador, no siendo lanzados por el intérprete Java.
- Son cargados a través de la red por medio de páginas HTML y **no residen** en el disco duro de la máquina que los ejecuta.
- Poseen un ciclo de vida diferente: mientras que una aplicación se lanza una sola vez , una *applet* se arranca (inicia) cada vez que el usuario recarga la página en la que se encuentra la *applet*.
- Tienen menos derechos que una aplicación clásica, por razones de seguridad. De modo predeterminado **no pueden ni leer ni escribir archivos, ni lanzar programas en el puesto que las ejecuta, ni cargar DLLs**. Sólo pueden comunicarse con el servidor Web en el cual se encuentra la página Web que las contiene.

Observamos también en el gráfico, que existe otro tipo de aplicación Web llamada *Servlet*. Es importante asimismo conocer las características distintivas entre applets y servlets. Con este objetivo, se detallan las características de las Servlets:

- Se ejecutan en el Application Server del **Servidor**.
- Son ideales para aplicaciones Web que necesiten trabajar con la Base de Datos (por su rápido acceso).
- Manejan fácilmente la información respecto al estado y la sesión del usuario.
- Utiliza formularios de tipo HTML (fáciles de codificar para el programador).

No por esto debe interpretarse que *Applets* y *Servlets* son antagónicas o excluyentes en cuanto a su utilización. Todo lo contrario, ambas se complementan para dar como resultado una aplicación con las mejores características.

Esta cooperación entre Applets y Servlets , se ve reflejada en el siguiente esquema de 3 capas.



En la primera capa, las Applets sirven para dar mayor interacción con el usuario y aportan flexibilidad para realizar cambios, además de proveer una interfaz para formularios que precisen más complejidad. En la segunda capa , las Servlets al ejecutarse, permiten conectarse y trabajar de manera eficaz con la Base de Datos del Servidor.

Si bien pueden existir requerimientos de programación donde pudiera optarse por uno u otro modelo de programa JAVA, en términos generales, Applets y Servlets brindan *funcionalidades diferentes y* cada una de ellas responde mejor en determinadas situaciones.

III. Métodos esenciales en la ejecución de una applet

Las aplicaciones Applets presentan algunas características en sus métodos que no son habituales en las demás clases que componen las aplicaciones Java tradicionales. A continuación mencionaremos las siguientes *características exclusivas de las applets* :

• Las applets no tienen un método main ( ) con el que comience la ejecución. El papel central de su ejecución lo asumen otros métodos (que se verán posteriormente en la siguiente sección: *II métodos esenciales de las applets*)

•Todas las applets derivan de la clase **java.applet.Applet**. Las applets deben redefinir ciertos métodos heredados de Applet que controlan su ejecución: init( ) ,start( ), stop( ) , destroy( ) .

• Se heredan otros muchos métodos de las super-clases de Applet que tienen que ver con la generación de interfaces gráficas de usuario (AWT).

•Las applets también suelen redefinir ciertos métodos gráficos: los más importantes son paint() y update(), heredados de Component y de Container; y repaint() heredado de Component.

•Las applets disponen de métodos relacionados con la obtención de información, como por ejemplo: getAppletInfo(), getAppletContext(), getParameterInfo(), getParameter(), getCodeBase(), getDocumentBase(), y isActive().

Habitualmente, el programador tiene que sobrescribir algunos métodos (que **son los encargados de controlar la ejecución de las applets)** para que luego sean invocados por el Browser.

Los siguientes métodos son los que conforman el ciclo de vida de una applet:

- init ( )

Cuando el browser carga el applet, inmediatamente, el método **init** es el primero en ser llamado.
init se ocupa de todas las tareas de inicialización, realizando las funciones del constructor del applet.

- start ( )

El método **start** se llama automáticamente en cuanto el applet se hace visible, después de haber sido inicializada y también si la misma ha estado oculta y quiere hacerse visible nuevamente. En este método es habitual crear *threads* para aquellas tareas que requieren más tiempo tales como animaciones y sonidos.

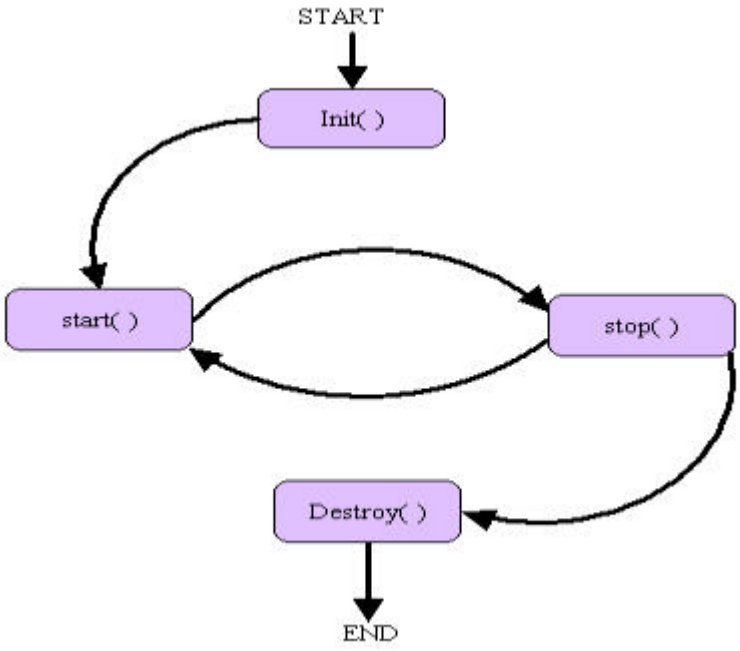
- stop ()

El método **stop** es llamado de manera automática al ocultarse el applet. Comúnmente, es el encargado de detener los *threads* que estén corriendo en el applet.

- destroy ( )

En general, es un método que no se sobrescribe y se utiliza cuando el applet va a ser descargada. Así, la misma puede liberar los recursos que haya reservado.

El siguiente esquema ilustra el ciclo de vida de una applet:



IV. Las consideraciones de seguridad en torno a las applets

Java tiene muchas medidas de seguridad que minimizan el riesgo de la ejecución de las *applets*, pero estas medidas también limitan a los programadores de *applets* en su capacidad de programación.

El modelo de seguridad para *applets* en Java trata al *applet* como: "código no fiable ejecutándose dentro de un entorno fiable". Por ejemplo, cuando un usuario instala una copia de un navegador Web en una máquina "está confiando" que su código será funcional en el entorno. Una *applet*, por el contrario, se carga desde la red sin ninguna comprobación de su fiabilidad.

Estas normativas de seguridad son implementadas para verificar que los códigos de byte de las clases de las applets, no rompen las reglas básicas del lenguaje ni las restricciones de acceso en tiempo de ejecución. Sólo cuando estas restricciones son satisfechas se le permite a la *applet* ejecutar su código. Cuando se ejecuta, se la marca para señalar que se encuentra dentro del intérprete. Esta marca permite a las clases en tiempo de ejecución determinar cuándo a una fracción del código se le permite invocar a cierto método. Por ejemplo, una *applet* está restringida en los *hosts* en los que se puede abrir una conexión de red o en un conjunto de URLs a las que puede acceder.

En su conjunto estas restricciones constituyen una política de seguridad. La actual política de seguridad afecta a los recursos que una *applet* puede usar, cuyos principales puntos son:

- Los accesos que pueden realizar a los archivos son restringidos. En particular si se trata de escribir en archivos y/o leerlos, las mismas no tendrán capacidades estándar que se puedan realizar en los navegadores que soportan *applets* de Java.

- Las conexiones de red serán restringidas a conectar **sólo** con el *host* de la que proviene.

- Una *applet* no es capaz de usar ningún método que pueda resultar en una ejecución arbitraria, código no revisado o ambos. Esto incluye métodos que ejecuten programas arbitrarios (métodos nativos) así como la carga de bibliotecas dinámicas.

El modelo original de seguridad proporcionado por la plataforma Java, conocido como el modelo "sandbox", existió para proporcionar un entorno muy restrictivo en el que ejecutar código no firmado obtenido desde una red abierta. En este entorno, el código local tiene total acceso a los recursos vitales del sistema, por ejemplo el sistema de archivos. Pero el código descargado remotamente proveniente de una applet, sólo puede tener acceso a recursos limitados proporcionados dentro del "sandbox". Un controlador de seguridad es el responsable en cada plataforma de determinar qué accesos a recursos están permitidos.

Concepto de Applet certificada (firmada)

A partir de la versión JDK 1.1 se introdujo el concepto de "applet certificada". Una applet con certificación digital es tratada como código local, con total acceso a los recursos, si se usa la clave pública para verificar la firma. Las que no son certificadas, aún se ejecutan dentro del sandbox. Las certificadas, se envían con sus respectivas firmas, en archivos JAR (Java ARchives) "firmados".

Clases encargadas de la seguridad en las applets

Cualquier intento que una applet haga para acceder a un archivo local lanzará una excepción. Cada browser tiene un objeto **SecurityManager** que implementa las políticas de seguridad. Cuando este objeto detecta una violación, lanza la excepción **SecurityException**. La aplicación applet puede tratar esta excepción y reaccionar adecuadamente. Si la excepción no es capturada por un *catch*, la applet dejará de funcionar, pero ningún archivo será accedido por la misma.

Existe una situación poco habitual para applets firmadas digitalmente. Estas applets tendrán privilegios adicionales, que hacen que el Browser muestre un dialog box, preguntando al usuario si desea aceptar la identidad del autor de la aplicación applet.

En caso de presentarse la situación anterior, se debería rechazar la solicitud propuesta a fin de evitar daños en el sistema en donde esa aplicación applet intenta descargarse.

Consideraciones finales:

El applet descargado a través de un Browser NO PUEDE:

- CARGAR bibliotecas o DEFINIR métodos nativos.
- LEER o ESCRIBIR ARCHIVOS en el host que lo está ejecutando (cliente) de forma normal.
- Establecer conexiones de red salvo con el host de origen.
- EJECUTAR ningún programa en el cliente.
- Leer algunas propiedades del sistema.
- Las ventanas que levanta un applet son visualmente distintas a la de las aplicaciones.

La siguiente URL provee el acceso a una applet que se podrá utilizar para un simple test sobre los conceptos seguridad de applets expresados en el tip.

http://www.informagen.com/Java/AppletSigning/Object.html

V. Dónde Obtener Información Adicional

Sitio de Sun: http://java.sun.com

Foro de desarrolladores : http://forum.java.sun.com

Seguridad de applets : http://www.wutka.com/hackingjava/ch3.htm

Tutoriales y códigos fuentes de applets ordenadas por categorías : http://javaboutique.internet.com/applets/

Ejemplos de Applets que se utilizan para simular desde campanas de Gauss a movimiento brownianos http://www.stat.duke.edu/sites/java.html

Ejemplos maliciosos para demostrar lo seguras que son las applets: http://java.sun.com/sfaq/

Dudas o consultas ***develop@teknoda.com***