

# **Java a fondo**

## **Apéndice B**

**Ing. Pablo Augusto Sznajdleder**



## Contenido

---

B.1	Introducción .....	4
B.2	Comenzando con Applets .....	4
B.2.1	El ciclo de vida de un applet.....	6
B.2.2	El contexto del applet .....	7
B.2.3	Pasarle parámetros a un applet.....	7

## B.1 Introducción

---

Un *applet* es un programa Java preparado para correr dentro de una página Web, dentro de un navegador. Si una página Web incluye uno o varios *applets* entonces cuando el usuario acceda a esta página los *applets* serán descargados a su computadora y se ejecutarán localmente insumiendo sus propios recursos: memoria, procesador, etcétera.

En los inicios, a mediados de los años noventa, los navegadores traían incorporada la máquina virtual Java. Esto permitía que los usuarios de Internet accedieran a páginas que, sin que ellos lo supieran, incluían, descargaban y ejecutaban *applets*.

Hace varios años que los navegadores dejaron de incluir la máquina virtual, por lo tanto para poder acceder a una página que incluye un *applet* primero será necesario descargar un *plugin* desde el sitio de Java (<http://www.oracle.com/technetwork/java/index.html>).

El objetivo de los *applets* es brindar funcionalidad dinámica a las páginas Web. En los años noventa estaba bien pero hoy existe tecnología mucho más eficiente que los ha dejado prácticamente sepultados.

Probablemente el lector sea usuario de *GMail* o de las utilidades online de *Google* (*Google Docs*, *Google Spreadsheet*, etc.). Toda la funcionalidad dinámica que proveen estas “páginas Web” está basada en *Java Script*, particularmente en *AJAX*. Para acceder a estas aplicaciones, no es necesario descargar ningún *plugin* y como se trata de texto HTML, son extremadamente rápidas y no necesitan ser “descargadas” como sí sucede con los *applets*.

No obstante, existen muchas aplicaciones basadas en *applets* que aún están en uso y necesitan ser mantenidas. Por este motivo, decidí incluir este apéndice que, si fuera el caso del lector, le permitirá tener una noción introductoria al tema.

Este apéndice asume que el lector ha leído (o tiene la base de conocimientos expuestos en) los Capítulos 2 (“Programación Orientada a Objetos”) y 5 (“Interfaz Gráfica”). De no ser así, le recomiendo que primero lea estos contenidos.

## B.2 Comenzando con Applets

---

Un *applet* es una instancia de una clase que extiende a las clases base `java.awt.Applet` o `javax.swing.JApplet`. Esto lo habilita a ser incluido en una página Web para correr dentro del contexto definido por el navegador.

Por una cuestión de simplicidad en este apéndice trabajaremos con la clase `Applet` (de AWT). La clase `Applet` es una subclase de `Panel`, por lo tanto los *applets* son *containers* de componentes gráficos.

A continuación, veremos el código de un *applet* que dice “Hola Mundo Applet” dentro de su área gráfica.

```
package libro.apendiceB;
import java.applet.Applet;
import java.awt.FlowLayout;
import java.awt.Label;

public class HolaMundoApplet extends Applet
{
    public HolaMundoApplet()
    {
```

```
setLayout(new FlowLayout());  
Label lbl = new Label("Hola Mundo Applet");  
add( lbl );  
}  
}
```

Como podemos ver no hay ninguna diferencia entre un *applet* y un *panel* ya que, de hecho, son la misma cosa.

Claro que para ejecutar el *applet* tendremos que escribir una página Web que lo incluya. Para esto utilizaremos el *tag* `APPLET` de la siguiente manera:

```
<applet code="libro.apendiceB.HolaMundoApplet"  
        height="300"  
        width="300">  
</applet>
```

Un archivo de texto con estas líneas de código y con extensión *.html* se convierte en una página Web que incluye al *applet* `HolaMundoApplet`.

El lector estará apurado por hacer “doble *click*” sobre este archivo y probar el *applet* en su navegador pero debe recordar que a menos que tenga instalado el *plugin* correspondiente no obtendrá ningún resultado.

Si por el momento no queremos instalar ningún *plugin* entonces podemos utilizar una herramienta que se provee para este fin junto con el JDK: el *appletviewer*.

El *appletviewer* es un navegador mínimo que solo reconoce el *tag* `APPLET`. Utilizaremos esta herramienta para ejecutar y probar nuestro `HolaMundoApplet`.

Suponiendo que llamamos “index.html” al archivo que contiene el código HTML (con el *tag* `APPLET`) entonces:

- El archivo debe estar ubicado en el *package root*.
- Desde la línea de comandos ejecutamos el comando: *appletviewer index.html*

El resultado será el siguiente.

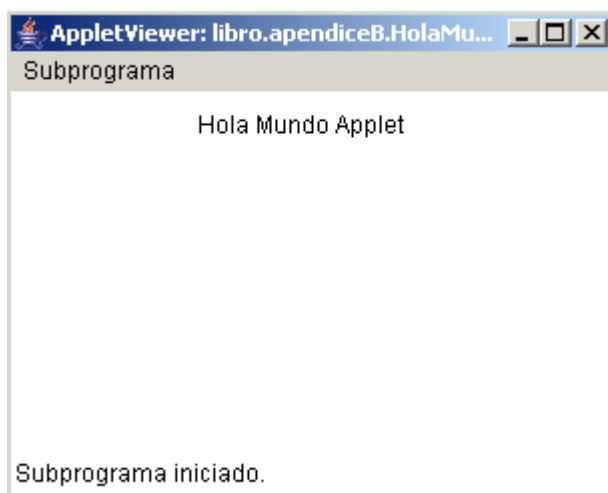


Fig. B.1 El *appletviewer* ejecuta un *applet*.

### B.2.1 El ciclo de vida de un applet

El *applet* vive dentro de una página Web, dentro del marco impuesto por el navegador. Para adaptarse a este contexto, hereda de la clase `Applet` cuatro métodos a través de los cuales podemos notificarnos (como programadores) sobre la evolución de su ciclo de vida. Estos métodos son los siguientes:

- `init` - Se llama cuando el contexto está listo y a punto de ejecutar al *applet*.
- `start` - Se llama luego del `init` y cada vez que se restaura la ventana del navegador.
- `stop` - Se llama cuando se minimiza la ventana del navegador y antes del `destroy`.
- `destroy` - Se llama cuando se cierra el navegador.

Es decir, podemos sobrescribir estos métodos y probar su funcionamiento minimizando, restaurando y cerrando el *appletviewer*.

---

```
//...
public class HolaMundoApplet extends Applet
{
    //...

    public void init()
    {
        System.out.println("init...");
    }

    public void start()
    {
        System.out.println("start...");
    }

    public void stop()
    {
        System.out.println("stop...");
    }

    public void destroy()
    {
        System.out.println("destroy...");
    }
}
```

---

Luego de correr el *appletviewer*, minimizarlo, restaurarlo y cerrarlo, la salida por consola será la siguiente:

```
init...
start...
stop...
start...
stop...
destroy...
```

## B.2.2 El contexto del applet

Como comentamos anteriormente, el *applet* vive dentro del contexto del navegador. Esto significa que será este quien definirá, por decirlo de alguna manera, las pautas de convivencia.

El `HolaMundoApplet` así como lo programamos tiene un error importante: se inicializa (en nuestro caso arma la interfaz gráfica) en el constructor.

Cuando una clase corre dentro de un contexto ajeno, generalmente, el constructor no es el lugar indicado para inicializarse, porque que se ejecute el constructor no necesariamente implica que el contexto dentro del que va a correr esté preparado.

Lo correcto será inicializar el *applet* en el método `init`. Esto nos permitirá, por ejemplo, acceder a parámetros que se pueden pasar dentro de la página Web.

## B.2.3 Pasarle parámetros a un applet

Dentro del `tag` `APPLET` podemos especificar parámetros de la siguiente manera:

```
<applet code="libro.apendiceB.HolaMundoApplet"
        height="300"
        width="300">
    <param name="nombre" value="Pablo" />
</applet>
```

Los parámetros se pasan por nombre. Dentro del *applet* podemos acceder a su valor especificando el nombre del parámetro en el método `getParameter` que heredamos de `Applet`.

---

```
package libro.apendiceB;

import java.applet.Applet;
import java.awt.FlowLayout;
import java.awt.Label;

public class HolaMundoApplet extends Applet
{
    public void init()
    {
        setLayout(new FlowLayout());
        String p = getParameter("nombre");
        Label lbl = new Label("Hola Mundo Applet " + p);
        add( lbl );
    }
}
```

---

Ahora el *applet* mostrará el valor del parámetro `nombre` luego de la frase “Hola Mundo Applet”.