

Java Applets como ejemplo de “Frameworks”

Agustín J. González
ELO-329

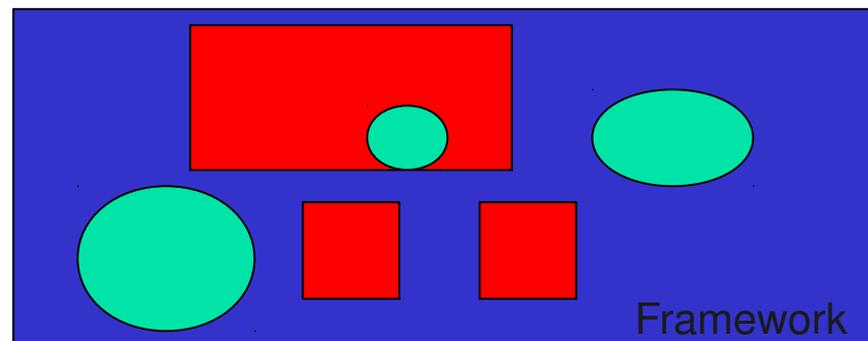
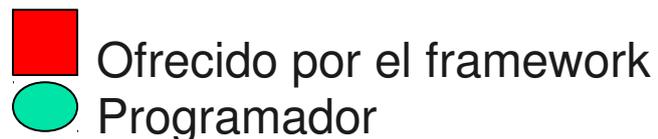
Frameworks

- Un framework (“marco de trabajo”) es un conjunto de clases que implementan todos los servicios comunes de un cierto tipo de aplicación.
- Para construir una aplicación, el programador deriva de alguna de las clases del framework y agrega las operaciones específicas de la aplicación.
- Swing es un framework para desarrollar interfaces gráficas.
- Por ejemplo: considere la clase JFrame para describir una ventana gráfica de una interfaz usuario. Se espera que otras clases sean derivadas de ésta para proveer implementaciones de operaciones como dibujar su contenido. Cada ventana debe saber como re-dibujarse cuando es restaurada desde el icono, o expuesta luego de ser cubierta por otra ventana.

Frameworks: Ideas relacionadas

Empresa constructora:

- Supongamos que usted tiene una empresa constructora. Por razones de economía y eficiencia, dispone de 5 tipos de casas para sus clientes.
- Como es razonable que el usuario participe definiendo sus requerimientos, la empresa le da la posibilidad que él elija algunos aspectos como: tipo de ventanas, piso, techo, y terminaciones en general.
- Además puede dejar cosas a su entera definición, es decir para que el cliente las haga por su parte.

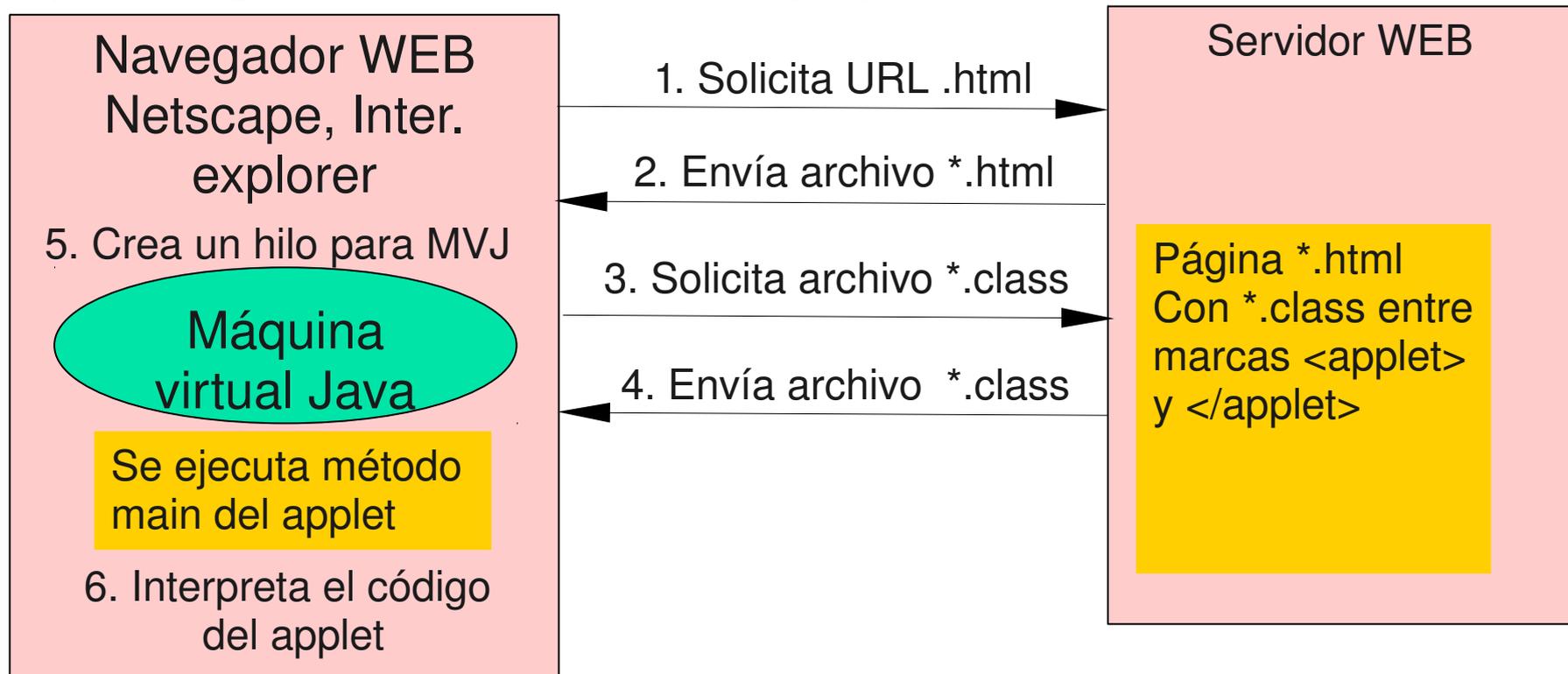


Frameworks (cont)

- Ejemplo: cuando un área que acepta scrolling es movida, la operación que maneja esta labor se encarga de solicitar el pintado de la porción de la imagen ahora expuesta. La clase base no tiene idea sobre qué se está pintando, pero sabe cuando el pintado es necesario.
- **La clase base llama métodos redefinidos por la clase derivada o métodos de alguna interfaz que otra clase debe implementar.**
- La programación sobre un framework maneja eventos aislados como pintado, comandos de menu, clicks del mouse, etc.
- Cuando un método termina, el control vuelve al framework.
- Nuevamente reconocemos el comportamiento conocido como programación basada en eventos (event-based programming)
- La idea no es que el usuario deba conocer todos los detalles sobre el manejo la interfaz usuario, sino sólo las operaciones claves que le son de interés.

Applet como un framework simple

- Un applet es un programa Java especial que está preparado para correr como parte de una página web.
- El applet es cargado y ejecutado por el navegador Web. Permite agregar interactividad y dinamismo a una página Web. (Ver página demo applets del jdk)



Evolución en modo de ejecución

- En sus orígenes las applets se ejecutaban sólo en el navegador HotJava de Sun. Luego la máquina virtual se incluyó en los Netscape e Internet Explorer, pero quedaba rápidamente obsoleta.
- Hoy Sun creó un Plug-in para permitir actualizaciones más rápidas. Ejecuta las applets usando el jre (java Runtime environment) externo al navegador.
- Los navegadores permiten ver qué plugins poseen; por ejemplo, en firefox ver poner en URL:
about:plugins

Inclusión de un Applet en una página

- En HTML (HyperText Markup Language) el rótulo `<applet></applet>` permite incluir un applet en la página.
- Cuando el navegador encuentra el rótulo `<applet>`, el navegador requiere del servidor WEB el archivo especificado por el parámetro "code".

- Ejemplo de applet en archivo *.html:

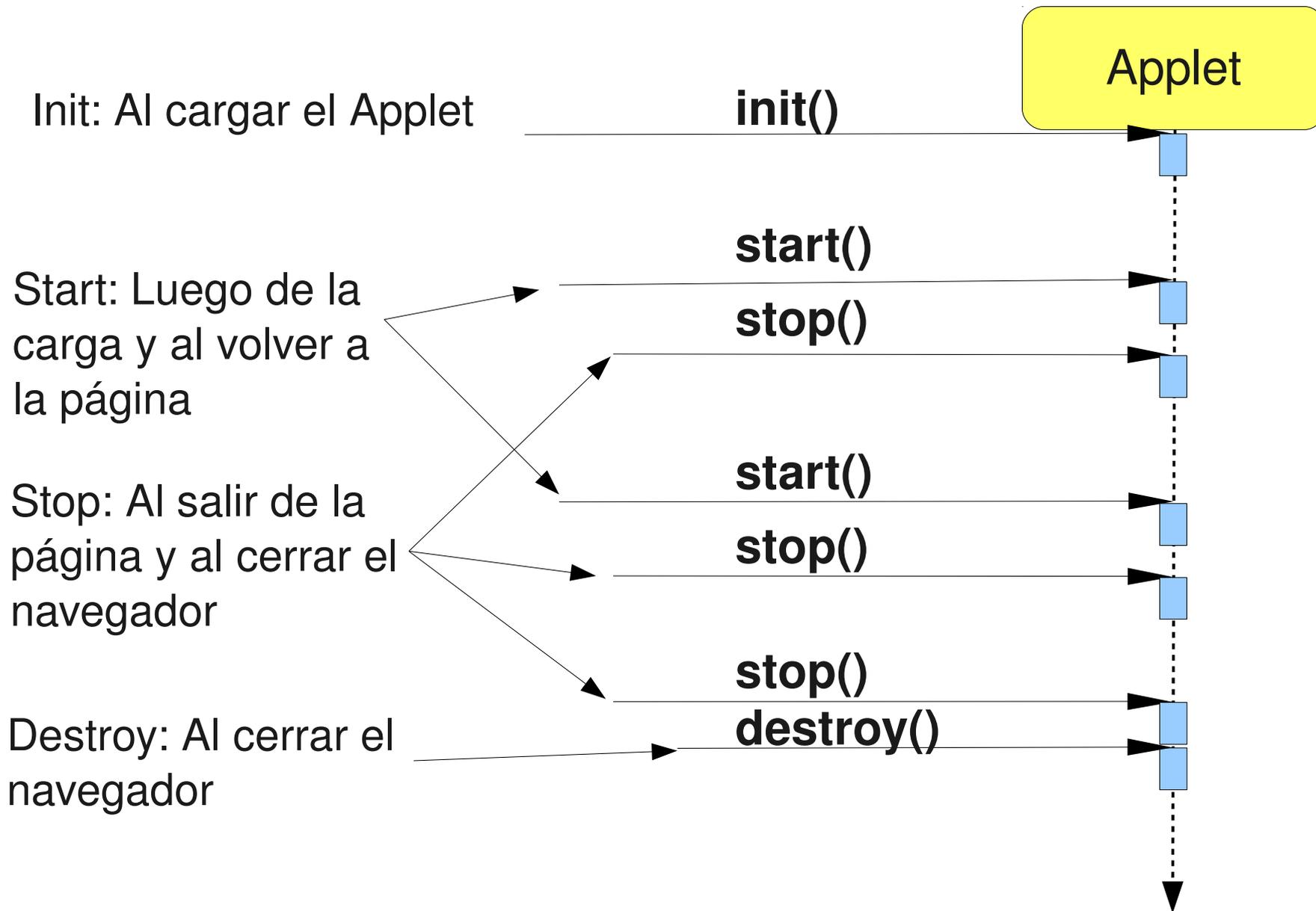
```
<applet code="NotHelloWorldApplet.class"  
width="300" height="100">  
</applet>
```

- El navegador crea un hilo separado para que cada applet se ejecute concurrentemente.

Ciclo de vida de un Applet

- Un applet debe ser subclase de Applet (o JApplet), la cual provee provee la interfaz entre el applet y el navegador.
- Una vez cargada el applet, una instancia del applet es creada e inicializada (llamando a `init()`).
- Luego se invocarán métodos sobre el applet para:
 - Iniciar su ejecución, `start()`
 - Detener su ejecución, `stop()`
 - Hacer limpieza final, cuando cerramos el navegador.
- Cuando el método `start()` es llamado y la apariencia del applet es desplegada (llamando a `repaint()`).
- El browser monitorea y controla los eventos asociados al applet (mouse, teclado, otros eventos , timers etc) y envía mensajes a los objetos manejadores de estos eventos que previamente deben ser registrados (event listeners) por el programador.

Ciclo de vida de un Applet



Eventos significativos y sus Métodos

- Para responder a los eventos significativos de un Applet debemos redefinir alguno de los siguientes métodos:
- **public void init()**: invocado al cargar el applet y para la inicialización del applet, no debería durar mucho para cargas rápidas. Éste puede ser pensado como el constructor.
- **public void start()**: es invocado luego de init, también es llamado cada vez que regresamos a la página.
- **public void stop()**: Si redefinimos start es buena práctica redefinir stop. Éste es invocada cada vez que salimos de la página. También al cerrar el navegador. Para ahorrar recursos deberíamos detener la ejecución iniciada con start.
- **public void destroy()**: Es llamado cuando el navegador es cerrado normalmente y después de llamar a stop. Se usa para liberar recursos no liberados por stop.

Ciclo de vida de un Applet (cont)

- Hay un problema cuando cambiamos un applet porque el navegador no la recarga al hacer un reload. Debemos cerrar el navegador o,
- Forzar la recarga del applet con el comando x en la consola Java del navegador, el cual limpia el cache de clases cargadas.
- Las nuevas versiones de firefox sólo son compatibles con consolas de JRE 6u2 o superior. :-)
- Un applet se ejecuta bajo el control de un navegador o con el utilitario appletviewer.
\$ appletviewer <archivo.html con applet>
- Así es más fácil hacer pruebas antes de dejarla en un servidor web.

Dibujando en un Applet

- La applet tiene un área de trabajo en la página del browser. El largo y alto son especificados en el rótulo `<APPLET>`
- Usualmente hacen uso de componentes GUI (Graphical User Interface) tales como botones, menús, etc. En este caso cada componente se encarga de su pintado.
- Si queremos hacer uso de componentes gráficas de Swing, debemos heredar nuestra applet de la clase `JApplet`.
- Si nosotros creamos contenido con el objeto `graphics`, debemos redefinir el método `paint`.

Conversión de una aplicación en Applet

- Crear un html con el rótulo <applet> y la invocación al código de la misma.
- Crear una subclase de Applet o JApplet. Hacer la clase public.
- La subclase de Applet o JApplet no debe tener el método main. No construir un frame para la aplicación. La clase Applet ya lo tiene.
- Mover el código del constructor del frame o JFrame al método init().
- Remover llamados a setSize, se define en el html.
- No llamar a setVisible ni setDefaultCloseOperation

Ejemplos. Calculadora

- Aplicación Java Calculator.java
- Applet creada: CalculatorApplet.java
la implementación sin frame es:
CalculatorPanel.java
- HTML asociado: CalculatorApplet.html
- Podemos ejecutar vía el navegador o a través de la aplicación appletviewer.
- > appletviewer CalculatorApplet.html

Creación de ventanas desde un applet

- Son simples, sólo crear instancias de JFrame, pero no invocar setDefaultCloseOperation.
- Ejemplo:
 - PopupCalculatorApplet.java
 - PopupCalculatorApplet.html

Seguridad en Applets

- Los visualizadores de Applets (navegadores) imponen restricciones de seguridad.
- Cada Applet posee un administrador de seguridad (Applet security Manager), el cual envía una SecurityException si se intenta violar alguna norma de seguridad.
- No pueden cargar bibliotecas ni definir métodos nativos.
- No pueden leer o escribir regularmente al disco local.
- Applets no pueden correr programas locales.
- No se pueden comunicar con otra máquina, salvo aquella de donde proviene.
- Las ventanas que el applet genera tienen un rótulo de advertencia.
- Para usar distintos niveles de seguridad, se puede usar un applet firmada (Signed applet)

Atributos de rótulo applet

- El rótulo html admite varios atributos.
- Posición:
width, height para el tamaño
align que puede ser: left, right, bottom, top, middle, entre otros.
- Ubicación del código:
code: indica el nombre del applet,
codebase: (opcional) indica el directorio relativo a la ubicación del html donde se encuentra el código,
archive: (opcional) lista de archivos .jar con código y otros recursos requeridos por el applet,
alt: texto alternativo a desplegar en caso que no se pueda desplegar el applet.

Paso de parámetros a un applet desde archivo html

- Se usa el rótulo html `<param>`. Éste posee dos atributos: `name` and `value`.

Name contiene el nombre del parámetro y `value` su valor. Ambos son de tipo texto.

- Luego desde Java es posible leer el valor del parámetro invocando el método `getParameter()` de la clase `Applet`.

- Ej: `<applet code="FontParameter.class" width="200" height="200">`
`<param name="font" value="Helvetica"/ >`

- Luego leemos los parámetros como sigue:

```
public class FontParameter extends JApplet {  
    public void init() {  
        String fontName=getParameter("font");  
  
        .....  
    }  
}
```

- Ejemplo: `Chart.java` `Chart.html`

Multimedia e interacción con Navegador

- Applets pueden manejar imágenes y audio.
- La clase URL es muy útil para esta manipulación.
- Notar métodos de la clase Applet:
 - getDocumentBase: retorna URL de la página donde proviene el html del applet
 - getCodeBase: retorna el URL del código del applet.
- getImage, getAudioClip, y play nos permiten la lectura de imágenes y sonido, el cual reproducimos con play. Todos son métodos de Applet.
- Ver: TicTacToe.html

Comunicación del applet con su entorno: Características más avanzadas

- Es posible comunicar varias applets de una misma página html.
- Es posible comunicar el applet con el navegador solicitando nuevas páginas html.
- El método usado es `getAppletContext()` de `Applet`.
- Ejemplo: `Bookmark.html` `Bookmark.java`

Destacar: en `ActionListener` de `Bookmark`

```
AppletContext context =getAppletContext();  
context.showDocument(url,"right");
```