



UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

Java Applets como ejemplo de “Frameworks”

Agustín J. González
ELO-329

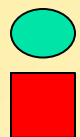


Frameworks

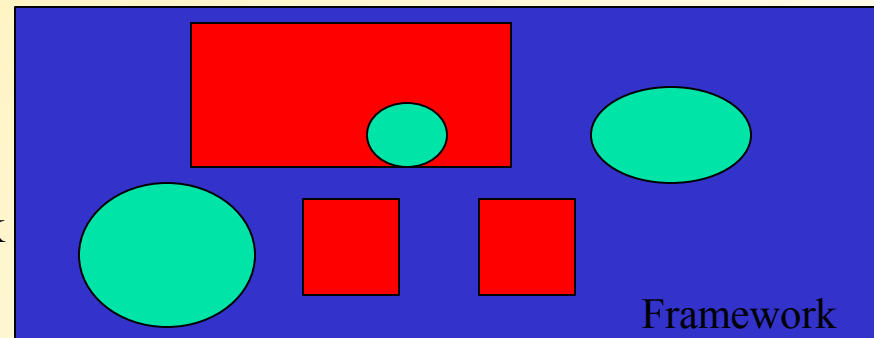
- Un framework (“marco de trabajo”) es un conjunto de clases que implementan todos los servicios comunes de un cierto tipo de aplicación.
- Para construir una aplicación, el programador deriva de alguna de las clases del framework y agrega las operaciones específicas de la aplicación.
- Por ejemplo: considere la clase Windows 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.



Programador
Ofrecido por el framework





Frameworks (cont)

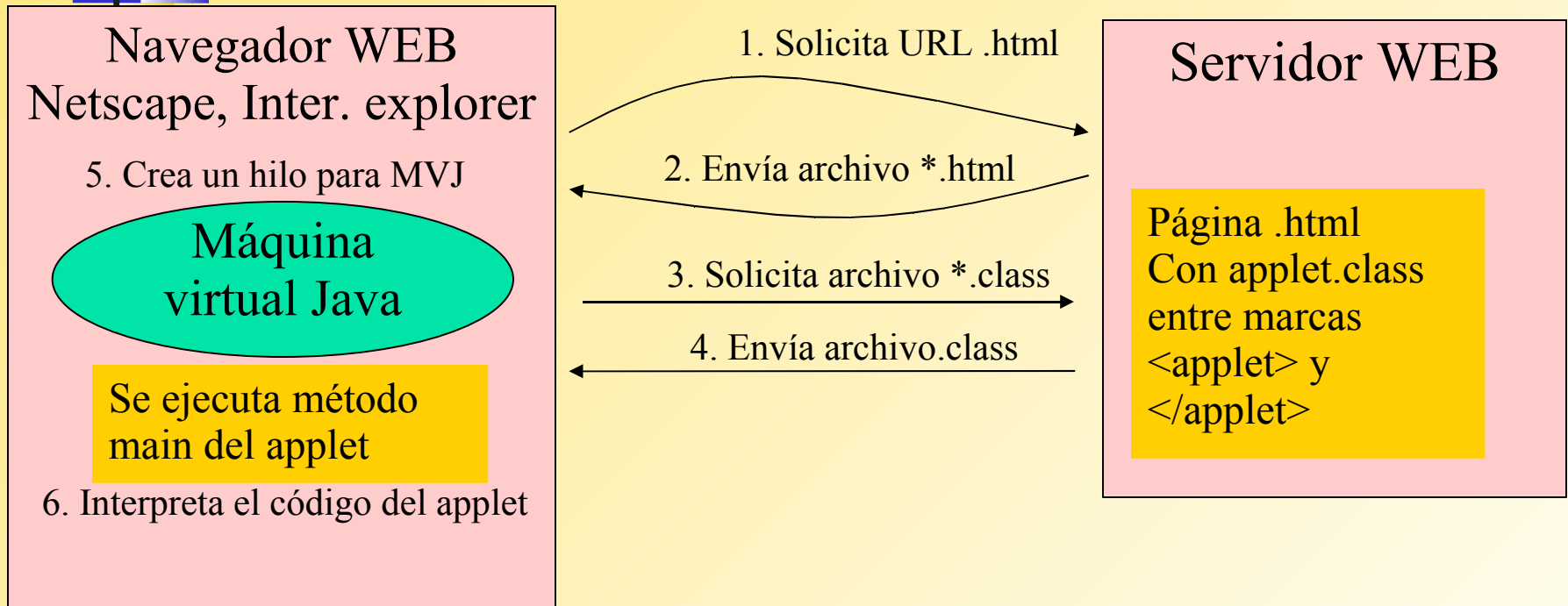
- La idea no es que el usuario deba conocer todos los detalles sobre el manejo la interfaz usuario, sino sólo las operaciones claves que le son de interés.
- Ejemplo: cuando un área que acepta scrolling es movida, la operación que maneja esta labor se encarga de solicitar el pintado de la porción de 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 impone el orden de ejecución de los métodos provistos por la clase derivada.
- La programación sobre un framework maneja eventos aislados como pintado, comandos de menu, clicks del mouse, etc.
- Cuando un método termina, el control es retornado al framework.
- Nuevamente reconocemos el comportamiento conocido como programación conducida por eventos (event-driven programming)



Applet como un framework simple

- Un applet es un programa Java especial que está preparado para ser ejecutado 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 de texto de Redes de Computadores)
- Un applet se ejecuta bajo el control de un navegador o el utilitario appletviewer.
\$ appletviewer <archivo.html con applet>
- La programación del applet se hace bajo un framework que implementa gran parte de la Graphical User Interface (GUI)
- Las applet poseen limitaciones por razones de seguridad; por ejemplo no se está permitido manipular archivos locales. Ver NotHelloWordApplet.java y su prueba.
- Ver TumbleItem.java y su prueba

Arquitectura para la ejecución de Applets



- 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.



Ciclo de vida de un Applet

Los pasos que tiene lugar cuando un applet es invocada son los siguientes:

- Cuando el navegador encuentra el rótulo `<applet>`, el navegador busca el código especificado en el parámetro y lo transfiere por la red hasta el browser (navegador).

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

- El navegador crea un hilo separado para que cada applet se ejecute concurrentemente.
- Una vez cargada el applet, un objeto de la clase "nombre del applet" es instanciado (creado) e inicializado (llamando a `init()`).
- Luego el método `start()` del applet es llamado y la apariencia del applet es desplegada (llamando a `paint()`).
- Luego 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 (cont)

- El browser maneja la applet invocando métodos heredados de la clase Applets, los cuales pueden ser sobremontados. La clase define métodos por defecto.
- `init()`: Llamada al iniciar una nueva applet, luego que el código es cargado o recargado. Operaciones muy largas (carga de archivos desde la red) deberían hacerse en un hilo separado.
- `start()`: Este método es llamado después de `init()`. Es llamado cada vez que regresamos a la página del applet. El método `paint()` es llamado automáticamente después de `start()`.
- `stop()`: Llamado para detener el applet. Normalmente significa detener cualquier thread (hilo) creado en `start()`. Es invocado por el navegador cuando éste cambia de página. El programador no lo llama directamente
- `destroy()`: Llamada para efectuar limpieza final. Es llamada cuando el navegador es cerrado.



Ciclo de vida de un Applet (cont)

- El applet puede transitar varias veces por los estado partida y detenida (start, stop).
- Hay un problema cuando cambiamos un applet porque el navegador no la vuelve a cargar al hacer un reload. Debemos cerrar el navegador o,
- Para forzar la recarga del applet ejecute el comando x en la consola Java del navegador, el cual limpia el cache de clases cargadas. Este comportamiento puede variar de una versión a otra. En mozilla por ejemplo, parece no ser necesario hacer la limpieza.



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`.



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`.
- Eliminar el método `main` de la aplicación. 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`



Seguridad en Applets

- Cada Applet posee un administrador de seguridad (Applet security Manager), el cual envía una SecurityException si se intenta violar alguna norma de seguridad.
- Applets no pueden correr programas locales.
- No se pueden comunicar con otra máquina, salvo aquella de donde proviene.
- No pueden leer o escribir al disco local.
- 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)



Creación de ventanas desde un applet

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



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 el `getAppletContext()` de `Applet`.
- Ejemplo: `Bookmark.html` `Bookmark.java`
Destacar: en `ActionListener` de `Bookmark`

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