
Taller: Programando dispositivos móviles con software libre

David Fernández Vaamonde
davidfv@alfa21.com

Mobigame 2004

Universidad de Alcalá de Henares
Escuela Politécnica

Guión

- Motivación
- Instalando el software necesario.
- Un primer ejemplo.
- Ciclo de vida de las aplicaciones SuperWaba.
- Compilación para dispositivos móviles (Palm OS y Windows CE).

- Sistema de clases
- Eventos en SuperWaba.
- Interfaz de usuario.
- Otras cosas.
- Emuladores (Palm)

Motivación

- La mia:
 - ¡Quería programar mi palm!
- Las del resto del mundo:
 - Cada vez más dispositivos móviles
 - PDAs (palm, ipaq...)
 - Móviles
 - Etc...
 - Uso "industrial" de dispositivos móviles
 - Integración de sistemas

Motivación

- Posibilidades para programar un Palm desde Linux:
 - En C -> Muy difícil (prc-tools)
 - En python(pippy) -> De juguete
 - Con j2me ->
 - ¿Dónde está la máquina virtual?
 - ...
 - No es software libre...
- ¿Que uso?

Motivación

- SuperWaba
 - Subconjunto de Java
 - POO
 - Estructurado y conocido
 - Máquina virtual para dispositivos, libre (LGPL)
 - Palm OS
 - Windows CE
 - Similar a programar un Applet.
 - Muchas clases y sistema sencillo.
 - Clases en paquetes independientes (metemos lo que necesitamos)
 - "Waba New Generation" ;)
- ¡La solución perfecta!

Instalando el software necesario:

Ingredientes necesarios:

○ Java

○ Cualquier implementación:

- Compilador: GCJ, Javac

- Máquina Virtual: Kaffe, GIJ, Java(Sun o IBM)

○ En este taller:

- Compilador: GCJ

- Máquina Virtual: Java

○ (En cualquier distribución)

○ Clases SuperWaba

- <http://superwaba.com.br/en/downloads.asp> (Registro gratuito)

- (SuperWabaSDK)

```
export CLASSPATH=$CLASSPATH:/SuperWabaSDK/lib/SuperWaba.jar:/SuperWabaSDK/utils:.
```

Instalando el software necesario.

- Ejecutables java (incluidos en Superwaba)
 - Antes binarios -> ahora java (más portables)
 - Crearán los "formatos" para cada dispositivo.
 - Clases Exegen y Warp
- No necesitamos ninguna clase externa de Java.
- Máquinas virtuales para los dispositivos.
 - Específica para cada uno (Palm o Pocket PC)
 - Palm:
 - Superwaba.prc, SWNatives.prc (lib/palm/PalmOS...)
 - Superwaba.pdb (clases, lib/xplat)
 - WindowsCE:
 - SuperWaba.exe, SuperWaba.pdb, MSW.pdb

Instalando el software necesario.

Un entorno de desarrollo totalmente libre:

- **Compilador: GCJ (GNU)**
- **Máquina virtual: Kaffe o GIJ**
 - Ejecutan Exegen y Warp perfectamente.
- **Problema:**
 - Clase Applet (visualizador)
- **Posible solución:**
 - POSE (Palm OS Emulator)

Un primer ejemplo

- **Compilación:**

- gcj -C HolaMundo.java
- javac HolaMundo.java
- (Jikes NO lamentablemente...)

- **Ejecución:**

- java waba.applet.Applet HolaMundo
- gij waba.applet.Applet HolaMundo
- kaffe waba.applet.Applet HolaMundo

- **Visor de applets SuperWaba.**

Ejemplo: Holamundo en Superwaba

```
import waba.ui.*;
```

```
import waba.fx.*;
```

```
public class HolaMundo extends MainWindow{
```

```
    public void onPaint(Graphics g){
```

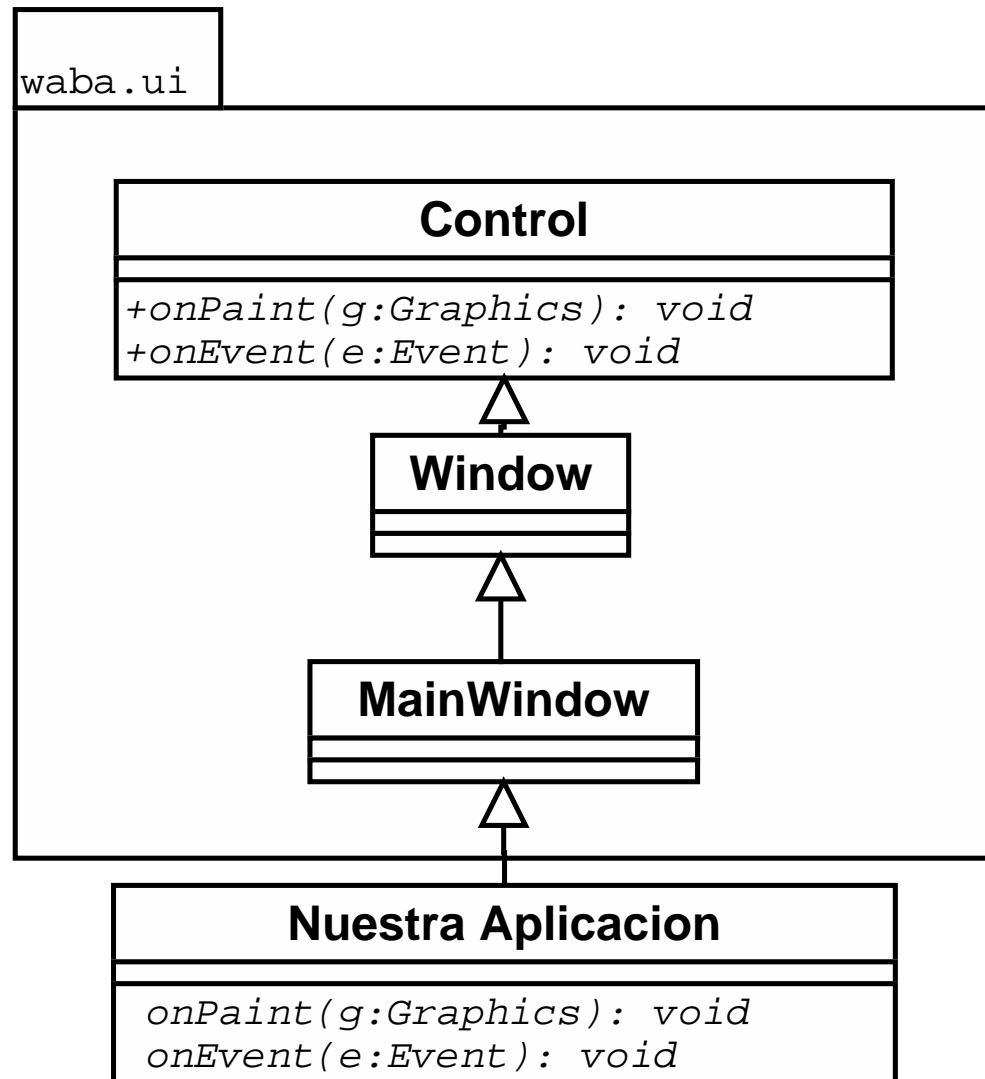
```
        g.setColor(255, 0, 0);
```

```
        g.drawText("Hola Mundo!", 10, 10);
```

```
    }
```

```
}
```

Ciclo de vida de las aplicaciones SuperWaba.



Ciclo de vida de las aplicaciones SuperWaba.

Metodos a redefinir:

- onStart()

- Se ejecuta al comenzar la aplicación.
- Ejecuta un repaint().

- onPaint()

- Se ejecuta cuando se realiza un repaint().

- onEvent()

- Es llamado cuando surge un evento.
- Se tratarán con el los eventos de la aplicación.

Ejemplo: HolaMundo2 en SuperWaba

```
import waba.ui.*;
import waba.fx.*;

public class HolaMundo2 extends MainWindow{
    int i;
    String cadena;

    public void onStart(){
        cadena=new String("Hola Mundo ---> onStart");
        i=20;
        this.repaint();
    }

    public void onPaint(Graphics g){
        g.setColor(0,0,0);
        g.drawText(cadena,0,i);
        i=i+15;
        cadena="Hola Mundo ---> onPaint";
    }
}
```

Compilación de aplicaciones para Palm OS y Windows CE

Clases Exegen y Warp

- Para palm:

- Exegen NombreFichero ClasePrincipal Nombreaplicación

- java Exegen HolaMundo HolaMundo Hola

- Fichero .prc

- Warp c NombreAplicacion RecursosNecesarios

- java Warp Hola HolaMundo.class

- Recursos asociados a la aplicación (Clases, iconos...)

- Fichero .pdb

Compilación de aplicaciones para Palm OS y Windows CE

○ Para Windows CE:

- warp c NombreAplicacion RecursosNecesarios

- java Warp Hola HolaMundo.class

- Recursos asociados a la aplicación

- Fichero .wrp

- exegeN NombreFichero ClasePrincipal NombreAplicación /p path

- java ExegeN Hola HolaMundo.class Hola /p "\\Program Files\\Scribble"

- Fichero .Ink

- Path completo a donde estar el .wrp

Compilación de aplicaciones para Palm OS y Windows CE

Makefile

- Crear targets para Exegen y Warp
- Permite automatizar muy bien las tareas

Ant

- Sistema de compilación en java
- Basado en XML
- Targets específicos

Sistema de Clases en SuperWaba

○ Clases básicas:

- waba.ui: Interfaz con el usuario
- waba.fx: Gráficos, imágenes, sonido,...
- waba.io: Entrada/salida, acceso a PDBs, puerto serie, sockets..
- waba.util: Vectores, Hash, ...

○ Extensiones:

- superwaba.ext.xplat.*
 - pimal: Clases de gestión de recursos
 - gps: Acceso al sistema de GPS
 - util.xml: XML "Tokenizer"
 - util.crypto: Criptografía (MD5, SHA1...)
 - html.ui: Contenedor HTML

Eventos en SuperWaba

- `onEvent(Event e)`

- Llamada con cada evento
- Obtenemos del evento tipo y quien lo genera.
- `e.type` y `e.target`
- Eventos posibles en:
 - `ControlEvent` : Eventos de control
 - `KeyEvent`: Eventos de teclas
 - `PenEvent`: Eventos del lapiz
- Como constantes.

Ejemplo: Eventos en SuperWaba

```
import waba.ui.*;
import waba.fx.*;

public class Eventos extends MainWindow{
    int i = 0;

    public void onPaint(Graphics g){
        g.setColor(0,0,0);
        g.drawText("Contador de pulsaciones:",0,0);
        g.drawText("Has pinchado " + i + " veces", 10, 20);
    }

    public void onEvent(Event e){
        if(e.type==PenEvent.PEN_DOWN){
            i=i+1;
            repaint();
        }
    }
}
```

Ejemplo: Eventos2 en SuperWaba

```
import waba.ui.*;
import waba.fx.*;

public class Eventos2 extends MainWindow{
    int x=0;
    int y=0;

    public void onPaint(Graphics g){
        g.setColor(0,0,0);
        g.drawLine(x-5,y-5,x+5,y-5);
        g.drawLine(x-5,y-5,x-5,y+5);
        g.drawLine(x-5,y+5,x+5,y-5);
        g.drawText(""+x,10,10);
    }
    public void onEvent(Event e){
        if(e.type==PenEvent.PEN_DRAG){
            x=((PenEvent)e).x;
            y=((PenEvent)e).y;
            repaint();
        }
    }
}
```

Interfaz con el usuario

- Clases waba.ui.*
- Componentes:
 - Edit
 - Button
 - TabPanel
 - Calendar
 - Radio
 - ...
- Se ha de dibujar el "rectangulo" que los rodea: `setRect()`;
- Se añaden a la ventana (a un contenedor): `add()`;
- Se repintan con un `repaint()`;

Ejemplo: GUI en SuperWaba

```
import waba.ui.*;
import waba.fx.*;

public class GUI extends MainWindow{
    int i=0;
    Button boton;
    Label etiqueta;

    public void onStart(){
        boton=new Button("Pulsame");
        etiqueta=new Label("Numero de pulsaciones: "+i);
        boton.setRect(10,10,80,30);
        etiqueta.setRect(10,50,160,10);
        add(boton);
        add(etiqueta);
    }
}
```

Ejemplo: GUI en SuperWaba (y II)

```
public void onPaint(Graphics g){  
}
```

```
public void onEvent(Event e){
```

```
    if(e.type==ControlEvent.PRESSED && e.target==boton){
```

```
        i=i+1;
```

```
        etiqueta.setText("Numero de pulsaciones: "+i);
```

```
        repaint();
```

```
    }
```

```
}
```

```
}
```

Ejemplo: GUI2 en SuperWaba

```
import waba.ui.*;
import waba.fx.*;
public class GUI2 extends MainWindow{
    Edit edit;
    Button boton;
    Label label;
    Label label2;
    Calendar cal;
    TabPanel tabBar;

    public void onStart(){

        String nombres[]={ "uno", "dos" };

        tabBar = new TabPanel(nombres);
        add(tabBar);
        tabBar.setGaps(2,2,2,2);
        tabBar.setRect(getClientRect());
        add(tabBar);
        tabBar.setGaps(2,2,2,2);
        tabBar.setRect(getClientRect());
        add(tabBar);
    }
}
```


Ejemplo: GUI2 en SuperWaba (y II)

```
// Obtenemos los dos paneles
Container Cont = tabBar.getPanel(0);
Container Cont2 = tabBar.getPanel(1);
// Creamos los objetos
edit=new Edit();
cal=new Calendar();
boton=new Button("Comando");
label=new Label("Comando: ");
label2=new Label("");
// Marcamos los rectángulos
edit.setRect(10,40,110,30);
boton.setRect (10,83,80,30);
label.setRect (10,113,200,30);
label2.setRect (10,150,200,30);
cal.setRect(5,20,150,160);
// Añadimos a cada panel
Cont.add(edit);
Cont.add(boton);
Cont.add(label);
Cont.add(label2);
Cont2.add(cal);

}
```

Ejemplo: GUI2 en SuperWaba (y III)

```
public void onEvent(Event e){
    if(e.type==ControlEvent.PRESSED && e.target==boton){
        label.setText("Comando: "+edit.getText());
    }
}
}
```

Otras cosas: Persistencia

Opciones:

- 1) Opción (gratis): File o Catalog (palm o Windows)
- 2) Opción (pago): SQL para acceso a PDBs (!!)

Características:

- Permite hacer persistentes objetos o datos
- Se tratan igual que ficheros
- Ideal para un patrón DAO (ligero)

Otras cosas: Persistencia (ejemplo)

```
Catalog c = new Catalog("MyCatalog", Catalog.READ_ONLY);
if (!c.isOpen())
    return;
DataStream ds = new DataStream(c);
int count = c.getRecordCount();
for (int i = 0; i < count; i++)
    if (c.setRecordPos(i))
    {
        String name = ds.readString();
        int age = ds.readShort();
        double salary = ds.readDouble();
        ...
    }
c.close();
```

Otras cosas: Imágenes

○ Clases:

- waba.fx.image
- superwaba.ext.xplat.fx.gif.GifImage
- superwaba.ext.xplat.fx.jpeg.JpegImage
- superwaba.ext.xplat.fx.png.PngImage

○ Permite mostrar imágenes.

```
import waba.ui.*;
import waba.fx.*;
public class Imagen extends MainWindow{
    Image Foto=new Image("gpul.bmp");
    Label Etiqueta=new Label("http://www.gpul.org");

    public void onPaint(Graphics g){
        Etiqueta.setRect(40,40,1300,10);
        add(Etiqueta);
        g.drawImage(Foto,40,60);
    }
}
```

Otras cosas: Juegos

- Superwaba.ext.xplat.game.*
- Framework completo para juegos:
 - Sprites
 - Scores
 - Animation
- Se extiende GameEngine en lugar de MainWindow
- Nuevos métodos a sobrecargar:
 - onGameInit()
 - onGameExit()
 - onTimer()
 - start()
 - stop()
 - onGameStart()
 - onGameStop()
- Ejemplo: Ping

Emuladores

- Las cosas pueden variar del aspecto del visor de Applets.
- PALM (para Linux):
 - POSE
 - XCopilot
- WindowsCE
 - MobiPocket :?
- Redondean nuestro entorno de desarrollo

Conclusiones:

Lo bueno:

- Entorno totalmente libre
- Muchas clases libres e información
- No está atado a una empresa
- Se pueden usar todas las técnicas y patrones de J2ME

Lo "menos bueno":

- No es un standard de facto como J2ME
- De momento no contempla Móviles (de momento...)
- Hay que limar la ejecución de la clase Applet ;)

Enlaces interesantes

¡Gracias por la atención!

Superwaba:

<http://superwaba.br.com>

Wiki de Superwaba:

<http://superwaba.sourceforge.net/cgi-bin/twiki/view>

Linux:

<http://www.gpul.org>

<http://lilo.uah.es>

Documentación en:

<http://www.davidfv.net/ponencias>