

>> Programación de dispositivos móviles.

Nicolás Álvarez S.
nalvarezs@vtr.net

Juan Monsalve Z.
jmonsalve@elo.utfsm.cl



Tópicos a tratar

- Repaso de conceptos de programación.
- Conceptos básicos de Java
- Plataforma a utilizar en el desarrollo de aplicaciones para celulares.
- Utilizando J2ME.
- Ejemplos prácticos.

Repaso de conceptos básicos de programación

- Código Fuente / Archivo ejecutable
- Variables
- Condicionales (if-else)
- Iteraciones (do-while, while, for)
- Uso de IO (Input-Output)

Conceptos básicos de Java

- Lenguaje Multiplataforma
- Lenguaje totalmente orientado a objeto
- Máquina virtual de Java(JVM): aplicación que debe traducir el código a cada plataforma compatible con Java
- Cuando se compila un programa en Java, no se obtiene un ejecutable. Se obtiene un código precompilado que se puede ejecutar en cualquier JVM

Plataforma de desarrollo de aplicaciones.

- La mayoría de los teléfonos actuales son capaces de ejecutar aplicaciones desarrollados por terceros (ajenos al fabricante).
- Las aplicaciones generalmente se desarrollan en el lenguaje de programación Java, de Sun Microsystems.
- Dada las características de hardware limitadas de los celulares, existe una versión de Java optimizada para uso en móviles, denominada J2ME.



J2ME: Java 2 Micro Edition

- J2ME ofrece una plataforma de desarrollo que permite controlar una serie de funcionalidades del teléfono, como Bluetooth, cámara, conexión GSM, teclas, etc.
- J2ME es un estándar limitado de Java, ya que fue diseñado para desarrollar aplicaciones en celulares, tomando en cuenta su poco poder de procesamiento y memoria.
- Java es un lenguaje de programación orientado a objetos, no funcional como C.
- Para la programación de nuestras aplicaciones, el concepto de “orientación a objetos” pasará a segundo plano y trataremos de programar de manera más funcional (como en C).

J2ME: Java 2 Micro Edition

- Para poder ejecutar una aplicación en Java, es necesario que los fabricantes de celulares cumplan con el estándar desarrollado por SUN.



Sony Ericsson



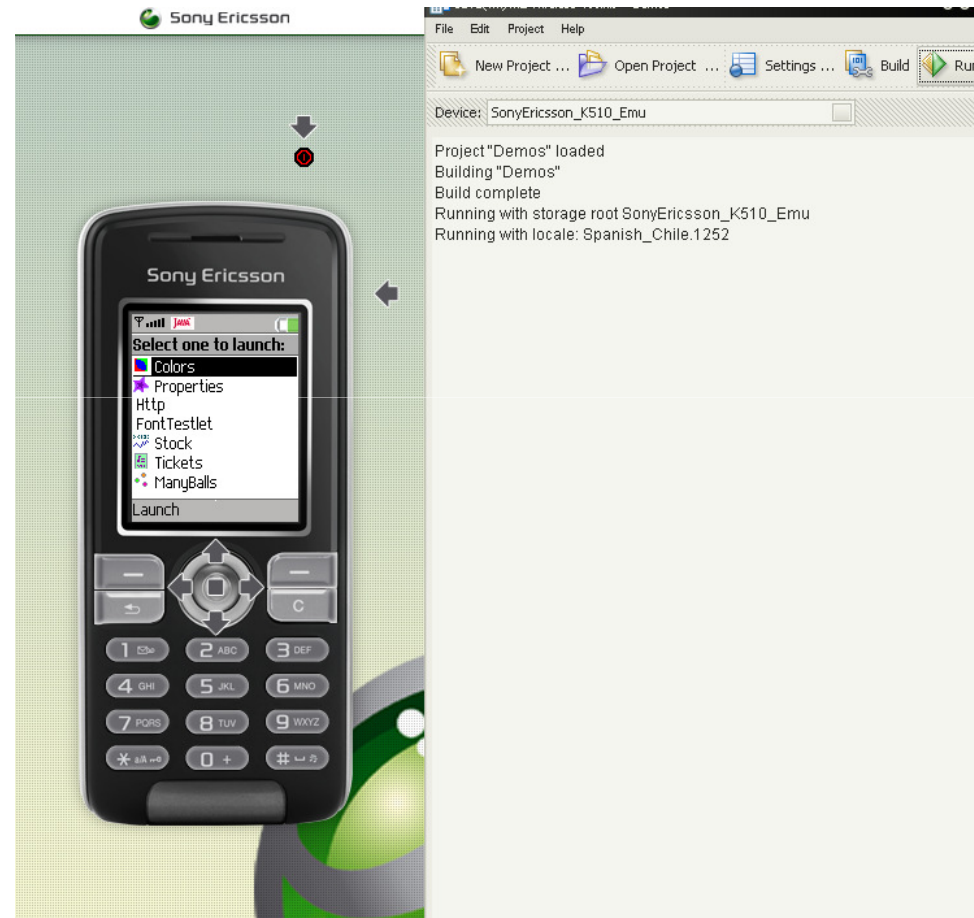
MOTOROLA



NOKIA
Connecting People

Desarrollando aplicaciones en J2ME

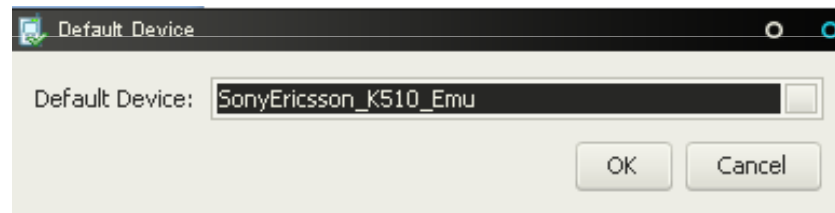
- Al igual que para C, se necesita un compilador para el lenguaje Java.
- Utilizaremos la herramienta *Java ME SDK for CLDC* proporcionada por Sony Ericsson.
- Esta SDK (Software Development Kit) trae ejemplos de ayuda para la programación, administrador de proyectos y emuladores de teléfonos.



Creando un proyecto en Java ME SDK for CLDC

- Configuraremos en primera instancia el emulador a utilizar. Usaremos como terminal predeterminada el SonyEricsson K510, que corresponden a los celulares que se tienen para trabajar.

Inicio>Programas>Sony Ericsson>Java ME SDK for CLDC>WTK2>Default Device Selection

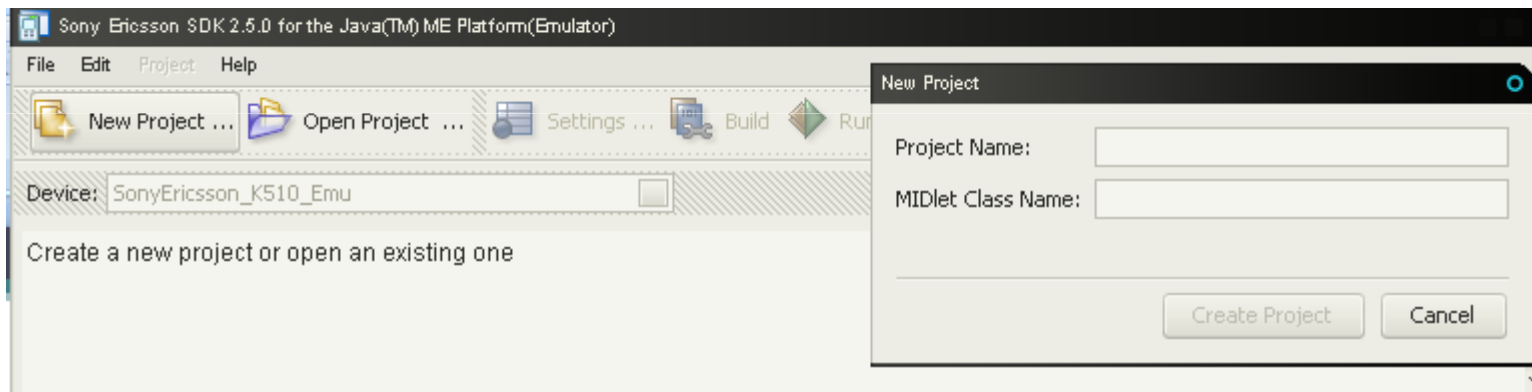


- Posteriormente, abriremos la KToolbar, que oficia de administrador de proyectos.

Inicio>Programas>Sony Ericsson>Java ME SDK for CLDC>WTK2>KToolbar

Creando un proyecto en Java ME SDK for CLDC

- Una vez abierta la KToolbar, click en *New Project*. Ingresaremos el nombre del proyecto (*Project Name*) y el nombre del archivo de código fuente a utilizar (*MIDlet Class Name*).



MIDlet: Programa en Java que se ejecuta en un terminal móvil.

Creando un proyecto en Java ME SDK for CLDC

- Una vez creado el proyecto, el directorio donde se almacenarán los archivos, corresponderá por omisión a:

C:\SonyEricsson\JavaME_SDK_CLDC\PC_Emulation\WTK2\apps\[NOM.PROY]

- Bajo el directorio “src” de ese directorio, almacenaremos los archivos fuentes de la aplicación.
- En C, los archivos fuente tienen extensión “.c”; en Java es “.java”.
- Crearemos en el directorio “src” un archivo de nombre “[*MIDlet Class Name*].java” (recordar el nombre del MIDlet al crear el proyecto).

Código Fuente en Java / Explicación

<http://www.elo.utfsm.cl/~iwg101/examples/ej1.java>

- En el archivo previamente creado, pondremos el siguiente código fuente:

```
import javax.microedition.lcdui.*;  
import javax.microedition.midlet.*;
```

```
public class ej1 extends MIDlet{
```

```
    Form mForm;
```

```
    public void destroyApp(boolean unconditional){  
        System.out.println("destroyApp");  
    }
```

```
    public void startApp(){  
        System.out.println("startApp");  
        mForm = new Form("Aplicacion de Prueba");  
        Display mDisplay;  
        mDisplay = Display.getDisplay(this);  
        mDisplay.setCurrent(mForm);  
    }
```

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

```
}
```

Librerías a utilizar por la aplicación, análogo en C al usar

Se define la "clase" de nombre

Se define la función "destroyApp", que controla lo que se hace cuando se cierra la aplicación. Muestra en el Ktoolbar el

La función startApp se ejecuta automáticamente al iniciar la aplicación (como la función "main" de C). A la variable mForm se le asigna el título de "Aplicación de Prueba". Se crea una variable de tipo "Display" con variable o.

Se define la función "pauseApp", que controla lo que se hace cuando se pausa la aplicación. Muestra en el Ktoolbar el mensaje "pauseApp" al pausar.

Otros Ejemplos

- <http://www.elo.utfsm.cl/~iwg101/examples/ej2.java>
(colocar texto en la ventana principal).
- <http://www.elo.utfsm.cl/~iwg101/examples/ej3.java>
(utilizar botones de comandos).
- <http://www.elo.utfsm.cl/~iwg101/examples/ej4.java>
(ingresar texto dinámicamente).
- <http://www.elo.utfsm.cl/~iwg101/examples/ej5.java>
(mostrar imágenes, el archivo de imagen debe colocarse en el directorio “res”).