

## PRIMEROS PASOS PARA PROGRAMAR APLICACIONES MOVILES EN JAVA

### First steps for mobile applications in java programming

#### RESUMEN

Este documento enseña los pasos básicos para desarrollar una aplicación móvil, presentando con un ejemplo la realización de una historia clínica, adicionalmente se realiza una introducción de las aplicaciones necesarias para la creación de un proyecto de este tipo, para ejecutarlo y visualizarlo en un simulador.

**PALABRAS CLAVES:** CLDC, CSS, Device Configuration, Emulador, J2ME, J2ME-POLISH, MIDLET, MIDP, MobileApplication, NetBeans,

#### ABSTRACT

*This document shows the basic steps to develop a mobile application, presenting a sample implementation of a medical history, perform a further introduction of the applications necessary to create a project of this type, to run it and view it on an emulator.*

**KEYWORDS:** *Device Configuration, Emulador, J2ME, J2ME-POLISH, MIDLET, MIDP, MobileApplication, NetBeans,*

#### JUAN DAVID MONTOYA M.

Ingeniería de Sistemas y Computación  
Estudiante  
Universidad Tecnológica de Pereira  
[troyox@gmail.com](mailto:troyox@gmail.com)

#### SAULO DE JESUS TORRES

Ingeniero Eléctrico, M. Sc. Física “Ciencia de los Materiales”.  
Candidato a PHD en Informática: Ingeniería del Software con la Universidad Pontificia de Salamanca en Madrid España.  
Profesor de planta  
Universidad Tecnológica de Pereira Colombia.  
[Saulo.torres@utp.edu.co](mailto:Saulo.torres@utp.edu.co)  
[saulotor@etp.net.co](mailto:saulotor@etp.net.co)

#### PAULA ANDREA VILLA S.

Ingeniera de Sistemas y Computación  
Profesor auxiliar  
Universidad Tecnológica de Pereira  
[pavaji@utp.edu.co](mailto:pavaji@utp.edu.co)

## 1. INTRODUCCIÓN

Cada día son más los usuarios de dispositivos móviles, lo que ha convertido estos dispositivos en una parte esencial de la vida cotidiana de las personas, debido a las mejoras tanto en la posibilidad de acceso como en el desarrollo de software. Los sistemas operativos y las aplicaciones de los dispositivos móviles han estado en constante desarrollo como se menciona en [1], pero *Sun Microsystems* da un paso adelante dentro de su tecnología *Java* y presenta *J2ME (Java 2 Micro Edition)* que es una colección de funciones y procedimientos que ofrece una biblioteca para ser utilizada en los desarrollos de software. Esta librería está desarrollada en *java* y está orientada a productos de consumo como *PDA's*, teléfonos móviles o electrodomésticos como se menciona en [2].

## 2. PRIMEROS PASOS EN J2ME

En esta primera parte del artículo se indica cómo construir y ejecutar la primera aplicación móvil. Existen diferentes herramientas para construir aplicaciones en J2ME, como lo son el Sun One Studio de Sun Microsystems, Jbuilder de Borland y el J2ME Wireless Toolkit 2.0 que proporciona Sun. Este último es el más sencillo de utilizar y aunque no es uno de los más potentes, sirve para realizar aplicaciones sencillas que se deseen crear.

Para instalar *J2ME Wireless Toolkit*, primero se debe instalar las librerías necesarias de java. Se puede descargar la última versión de JDK desde la URL <http://java.sun.com/javase/downloads/index.jsp>. Cuando se descargue y se instale, se descarga el J2ME desde la URL <http://java.sun.com/j2me/download.html>.

Por último se instala el programa *NetBeans mobility* que es un entorno de desarrollo muy cómodo y muy potente como mencionan en [3]. En la página del autor

proporcionan varias versiones y se puede descargar la versión completa de desarrollo en muchos lenguajes o solo descargar la versión para móviles, la cual contiene *JAVA ME*, en el siguiente link <http://www.netbeans.org/downloads/index.html>

Para crear la primera aplicación móvil se inicia el software *Netbeans* y se va a *Archivo / Proyecto Nuevo* que brinda acceso a un asistente que permite seleccionar el tipo de desarrollo a ser implementado, seleccionándose en este caso *JAVA ME / Mobile Application*. Luego el asistente, permite seleccionar un nombre para el proyecto, una ubicación de almacenamiento y la posibilidad de crear una aplicación ejemplo llamada *Hello Midlet*. Seguidamente, aparece la configuración del dispositivo móvil de las funciones y librerías que soporta, si no se tiene un dispositivo móvil específico para el cual se va a hacer el desarrollo se selecciona *Device Configuration CLDC-1.1 y MIDP-2.0* que es el más general y abarca una gran cantidad de estos dispositivos. Cuando se hace clic en terminar, es el fin del asistente y automáticamente se crea una nueva carpeta dentro de *Mis Documentos* llamada *NetBeansProjects*, ahí van a ir guardados los proyectos realizados en *NetBeans*. Ver figura 1.

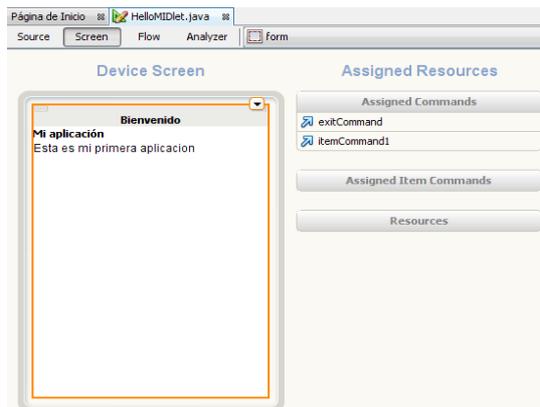


Figura 1. Sección donde se registran los proyectos en NetBeans  
Autor: Los autores.

El *NetBeans* proporciona la posibilidad de desarrollar en modo código, en modo gráfico o con un diagrama de flujos, es muy útil porque se puede visualizar lo que hay en pantalla o seleccionar la función de cada botón y arrastrarlo hacia otros formularios creando así el diagrama o si se quiere construir funciones y clases más específicas creando desarrollos por código.

En este momento del desarrollo, en *FLOW* se tiene un pequeño diagrama creado donde en el primer módulo, hay 2 botones creados llamados *Started* y *Resumed* y donde el botón *Started* quiere decir a donde se va cuando se inicia la aplicación, este lleva a un formulario que a su vez tiene 1 botón *ExitCommand* que lleva devuelta al primer módulo (Ver figura 2).

Este diagrama se puede modificar según el diseño y los requerimientos de la aplicación, crear botones y muchas otras funciones de la barra de herramientas *paleta*, como alertas, tablas, cajas de texto, funciones login, compositor de SMS, botones, etc. Ver figura 3.

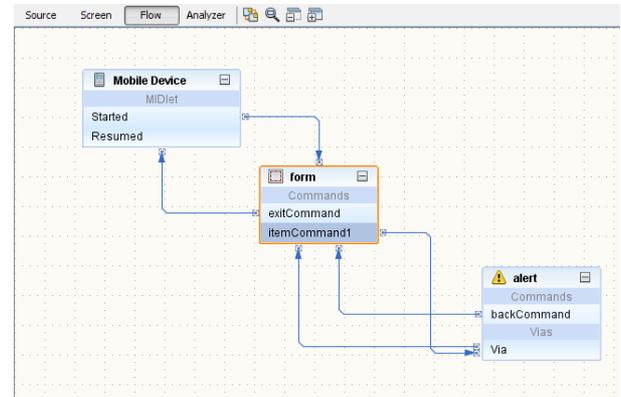


Figura 2. Ejemplo diagrama de flujo  
Autor: Los autores.

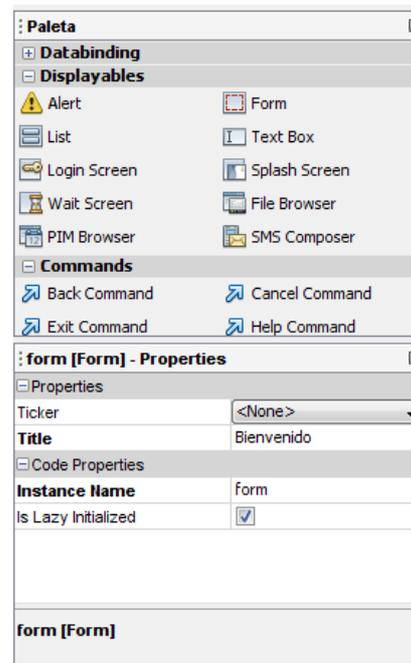


Figura 3. Barra de herramientas "PALETA"  
Autor: Los autores.

En *Screen* se puede definir rápidamente lo que aparecerá en la pantalla principal, es decir, el formulario especificado a donde se va después de correr la aplicación (Ver Figura 4.). Cuando se selecciona el formulario que está en la pantalla se puede observar en la parte derecha las propiedades de ese formulario donde se puede cambiar el nombre en la casilla *title*, además de otras propiedades. Se puede agregar un ítem *string* desde la paleta y cambiarle el *label* y el texto correspondiente, también se puede agregar un botón que al ser presionado se vaya a una alerta, esta alerta también se encuentra en

la paleta e igual que con las otras herramientas cambiar su comportamiento y sus opciones.

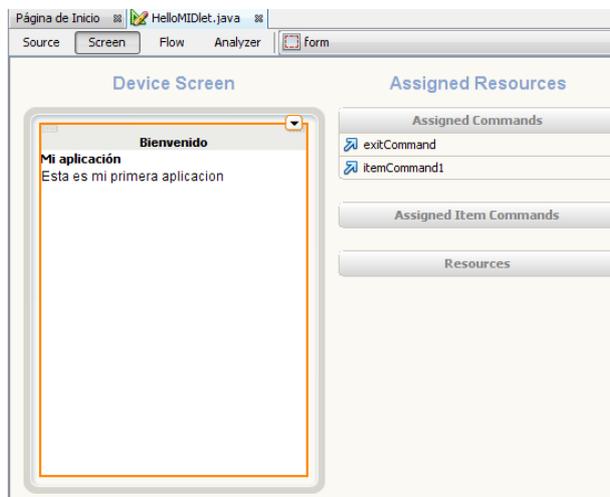


Figura 4. Sección "Screen"  
Autor: Los Autores.

Por último se guardan los cambios y en menú *ejecutar / Run Main Project* se abrirá el proyecto en un emulador para poder visualizar la aplicación antes de pasarla al dispositivo móvil. (Ver figura 5)

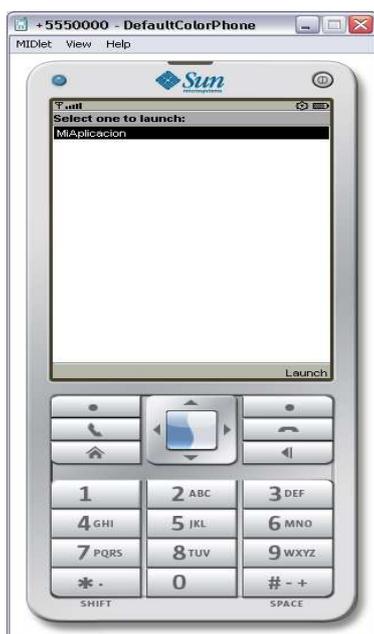


Figura 5. Pantalla simulación de la aplicación  
Autor: Los autores.

### 3. COMPLEMENTOS

A J2ME se pueden agregar otros complementos que trabajan en conjunto con el API de Java. Estos dos complementos que se explican a continuación o técnicas aplicadas a la programación de dispositivos móviles son

muy necesarios ya que se puede lograr una agradable navegación y a su vez permite ampliar la funcionalidad de un dispositivo móvil con el manejo de base de datos.

#### 3.2 J2ME-Polish

Este complemento se encarga no solo de mejorar la apariencia de los proyectos *J2ME* sino también de agregar funcionalidades de comportamiento como se menciona en [4], este complemento consiste en que Java entienda el código *Cascading Style Sheets (CSS)*, el cual es un código que sirve para agregar estilos, lo interprete y aplique las correspondientes funciones para agregar diseños y funciones que se desean. Antes de poder hacer todo esto se debe descargar e instalar el complemento desde <http://www.j2mepolish.org>, en el asistente de instalación se debe especificar la ubicación de las librerías de Java y si se quiere integrar el complemento a *NetBeans* también hay que especificar la ruta donde está instalado.

#### 3.2 Bases de datos

Otra adición que se puede usar en *J2ME* es la incorporación de bases de datos pero es necesario aclarar que *Java* no ha incorporado este sistema en sus librerías para móviles, pero hay varios métodos que se pueden implementar para lograr esta conexión, uno de estos métodos es conectarse a través de la red de datos de los dispositivos móviles por *http*, se hace la petición desde el dispositivo a un archivo *PHP* dentro del mismo servidor que contiene la base de datos, este archivo hace las correspondientes peticiones a la base de datos y se encarga también de enviar los resultados devuelta al dispositivo móvil.

### 4. DESARROLLO DE UNA HISTORIA CLÍNICA PARA UN DISPOSITIVO MÓVIL

Para el desarrollo de esta historia clínica se ha tenido en cuenta las buenas prácticas del desarrollo de software que se encuentran enunciadas en [5], para garantizar la articulación y compatibilidad con la aplicación web de un proyecto presentado en el artículo observatorio para el monitoreo de variables físicas y fisiológicas en niños y adolescentes en edad escolar [6]. Esto con el propósito de ofrecer la posibilidad a los usuarios del sistema de registrar los datos de la historia clínica en la aplicación sin disponer de un computador, ni de una red de internet, dado que este sistema funciona a través de la red celular.

Después de realizar el asistente para crear un nuevo proyecto, se observa que en la parte izquierda nos aparece una zona donde dice proyecto, aquí van a ir organizados por medio de un árbol los códigos, configuraciones y otra carpeta donde deben ir los recursos, es decir, las imágenes o sonidos que se incorporarán al proyecto. Dentro de la carpeta de los

códigos van a ir los *package* que son los paquetes contenedores de los códigos de la aplicación.

En este ejemplo se desarrollará una Historia Clínica donde se utilizará el *J2ME-POLISH* y la consulta a una base de datos instalada en un PC, los códigos llevan la siguiente estructura:

1. Especificar en qué *package* va a ir el nuevo código *Java* y las librerías que se deben importar para utilizar el *API* de *Java* para *J2ME*.

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

2. Luego hay que declarar la clase principal

```
public class HistorialClinico extends MIDlet
implements CommandListener { }
```

3. Dentro de esta clase hay que construir otras clases básicas y esenciales para el funcionamiento de la aplicación

```
public HistorialClinico () { }
public Display getDisplay () { } //Para elegir la
pantalla que se quiere mostrar
```

```
public void exitMIDlet() { } //Consiste en declarar
las variables nulas y llamar a destroyApp
```

```
public void startApp() { } //Es la que contiene el
orden o la forma de mostrar las clases
```

```
public void pauseApp() { } // Para pausar la
aplicación
```

```
public void destroyApp(boolean unconditional) { }
Para cerrar y liberar la aplicación
```

```
public void commandAction(Command cmd,
Displayable screen) { } // Para asignar la función de
los botones
```

4. Dentro de la función *HistorialClinico* irá el menú el cual se declara como una lista

```
##style mainScreen
menuScreen = new List("HISTORIAL CLINICO",
List.IMPLICIT);
```

```
##style mainCommand
this.menuScreen.append( "Registrar", null);
##style mainCommand
this.menuScreen.append("Consultar", null);
##style mainCommand
this.menuScreen.append("Actualizar", null);
##style mainCommand
```

```
this.menuScreen.append("Borrar", null);
##style mainCommand
this.menuScreen.append( "Cerrar Sesion", null);
```

En este caso la lista se llama *menuScreen* y se está definiendo cada uno de ítems de esa lista con *this.menuScreen.append* y tanto a la lista como a sus ítems se les está definiendo un estilo *J2ME-POLISH*, donde se observa que se declara con *##Style* y el nombre del estilo que tenemos en el archivo *CSS* dentro de la carpeta *resources* donde se le puede especificar fondos, columnas, fuentes, iconos, colores, márgenes y comportamiento al menú y pantalla.



Figura 6. Menú principal Historia Clínica  
Los autores

5. Ahora se le dará funcionalidad al nuevo menú, en el *CommandAction()*

```
if (cmd == List.SELECT_COMMAND)
{intselectedItem=
this.menuScreen.getSelectedIndex();
switch (selectedItem) {
case 0: // registrar en BD{
registrar(); break; }
case 1: // Consultar de la BD {
consulta(); break; }
case 2: // Actualizar {
actualizar(); break; }
case 3: // Borrar {
borrar(); break; }
case 4: // Cerrar Session {
logout(); break; }
```

```

    }
}

```

Para conectarse con la base de datos se puede usar el siguiente método:

Se declaran las variables

```

private String URL
URL="http://gpsis.utp.edu.co/Historial_Clinico/conexion.php";
URL+="?username="+username+"&password="+password+"&paciente="+paciente+"&tipo_consulta="+tipo_consulta;

```

Esta variable contiene la dirección donde están los archivos *PHP* y las variables que ayudarán a conectar con la base de datos, el archivo *PHP* además de configurar la conexión con la base de datos se reciben los datos, se hacen las sentencias de consulta y se envía la respuesta a la aplicación.

```

$paciente = trim($_GET["paciente"]);
$query = "SELECT cedula, nombre1, nombre2, apellido1, apellido2 FROM pacientes WHERE paciente='$paciente'";
$result = mysql_query($query, $db);
$cRegistros = 0;
$rawPost = "";
while(list($var1, $var2, $var3, $var4, $var5) = mysql_fetch_array($result)) {
    $rawPost.="|".$var1."|".$var2."|".$var3."|".$var4."|".$var5;
    $cRegistros++;
}
echo $cRegistros . $rawPost;
die();

```

Este método funciona muy bien, pero se debe tener en cuenta que hay que especificar en cada consulta cuantas variables se van a enviar y cuantas se van a recibir; si la consulta contiene menos variables que las especificadas no ocurre ningún problema, pero si la consulta contiene más variables se pierden datos.

#### 4. CONCLUSIONES

*J2ME* se ha convertido en una buena opción para crear juegos y aplicaciones en teléfonos móviles debido a que se puede emular en un computador durante la fase de desarrollo y luego implantarlos fácilmente al teléfono.

Las bases de datos en dispositivos móviles amplían la funcionalidad de las aplicaciones tanto empresariales como de entretenimiento, ya que se usa el dispositivo como intermediario para procesar información desde otro servidor y dejándolo de este modo libre de procesamiento y almacenamiento que son la desventaja de estos.

Las aplicaciones móviles están en constante desarrollo debido a que los dispositivos móviles van evolucionando y cada vez se parecen más a los propios computadores, así que se hace necesario una constante investigación sobre el tema.

#### 5. BIBLIOGRAFÍA

- [1] Alberto García Serrano [Online]. Available:
- [2] Sun Microsystem [Online]. Available: <http://java.sun.com/javame/index.jsp>
- [3] NeBeans [Online]. Available: [http://www.netbeans.org/index\\_es.html](http://www.netbeans.org/index_es.html)
- [4] J2ME-Polish [Online]. Available: <http://www.j2mepolish.org/tour/tour-es.html>
- [5] C. Ocampo, S. Valencia, P. Villa. Revista Scientia et Technica. Año XV, N°42. Agosto 2009 Modelos de calidad de software.
- [6] C. Nieto, S. Valencia, P. Villa. Revista Scientia et Technica. Año XV, N°43. Diciembre de 2009 Observatorio para el monitoreo de variables físicas y fisiológicas en niños y adolescentes en edad escolar.
- [7] J. García, J. Rodríguez, Aprenda Javacom si estuviera en primero, Escuela Superior de Ingenieros Industriales de San Sebastián, Universidad de Navarra.
- [8] R. Alarcón. Diseño orientado a objetos con UML. ISBN 84-88457-03-0
- [9] R.Pressman. Ingeniería del software, un enfoque práctico. Quinta edición, 2001.
- [10] Standard Glossary of Software Engineering Terminology, IEEE STD 610-1990, IEEE 1990.
- [11] The complete reference Java. Herbert Schildt. 2006. ISBN- 0072263857