

Capítulo 5: Aplicaciones cliente

5.1 La aplicación cliente en la Pocket PC

La aplicación desarrollada para el cliente en un dispositivo móvil como corresponde a la Pocket PC necesita una capa muy delgada de aplicación, de forma tal que no exija tantos recursos al sistema en el que está siendo ejecutada como la cantidad de datos que se transmiten para llegar a la aplicación del servidor. El cliente para Pocket PC fue desarrollado utilizando el Compact Framework 1.1 de Microsoft.

5.1.1 Conociendo el compact Framework de Microsoft

El Microsoft Framework fue diseñado con una simple intención: la de desarrollar aplicaciones. Ese tipo de aplicaciones que despliegan, procesan y recolectan información, es decir, aquellas que ofrecen alguna razón para cargarlas. Simplemente necesitarán una interfaz pues la información se podrá encontrar de forma local o remota sin importarle al usuario de donde provengan los datos [MSDN,2004].

El Framework compacto de .NET pretende acelerar y simplificar el desarrollo de aplicaciones para dispositivo móviles o dispositivos “inteligentes” como se les ha comenzado a conocer. Lo que incluye Pocket PC, Pocket PC 2002, Pocket PC Phone Edition y dispositivos corriendo a partir de Windows CE 4.1 o superior [MSDN,2004].

Tanto el .NET Framework como el compact Framework de Microsoft se basan en dos componentes principales: el *Common Language Runtime* (CLR) y las librerías del .NET Framework.

El CLR es la base del .NET compact Framework, pues es el responsable de manejar los tiempos de ejecución, proveyendo a los procesos los servicios necesarios, administración de memoria y de administración de hilos [MSDN,2004].

Las librerías de .NET Framework permiten realizar aplicaciones rápidas por su colección de clases reutilizables, y como las mismas son ejecutadas sobre la máquina virtual de .NET existe portabilidad entre las aplicaciones desarrolladas y los dispositivos que trabajen esta máquina virtual [MSDN,2004].

5.1.2 Common Language Runtime

El CLR provee un ambiente de ejecución del código que administra todas las aplicaciones desarrolladas para el ambiente del Compact Framework. El CLR fue diseñado principalmente para mejorar el rendimiento de la computadora, con la finalidad de obtener resultados al momento (*Just In Time*). Además es posible incluir módulos (DLL's) trabajados en diferentes lenguajes que bajo la estructura del CLR puedes interactuar sin ningún problema de compatibilidad o perdida de funcionalidad. Así se obtienen aplicaciones que no es necesario recompilarlas para pasarlas a algún otro tipo de dispositivo móvil [MSDN,2004].

5.1.3 Librerías de clases del .NET Compact Framework

Las aplicaciones que se desarrollen harán uso de las clases en la librería para obtener funciones reutilizables por alguna otra aplicación. Como se podrá esperar de toda librería orientada a objetos .NET Compact Framework incluye clases para el manejo de XML, acceso a bases de datos, I/O, diseño de interfaces, por mencionar sólo algunas [MSDN,2004].

5.1.4 Implementación

La primera implementación que se decidió tomar para el desarrollo de la aplicación para la Pocket PC fue tratar de tomar el documento en cGML pero después de varias pruebas se decidió desechar esta opción porque resultaba insuficiente el medio de transporte para hacer llegar algo tan extenso. Por lo que fue necesario proyectar el documento cGML en una imagen y enviarla al dispositivo.

Por lo tanto, la aplicación cliente en Pocket PC aprovecha la versión compacta del .NET Framework que ofrece todos los servicios necesarios que Windows Compact puede ofrecer.

En particular el cliente cuenta con una colección de imágenes que puede descargar de Internet (en caso de tener una colección) o puede almacenarlos directamente al dispositivo.

En el caso de la primera ventana se presentan las opciones de ver un mapa almacenado o descargar alguno del Internet. Es conveniente mencionar que todos los mapas presentados a continuación incluyen una selección completa de la capa, como fue especificado en el capítulo de extracción de información geográfica de la base de datos.



Imagen 5.1 Presentación en la Pocket PC

En caso de no tener ningún mapa en el dispositivo se puede ir a la interfaz de descarga



Imagen 5.2 Solicitud al Web

Al finalizar la descarga es posible visualizar la misma regresando al menú principal.



Imagen 5.3 Visualización del mapa

Parte del desarrollo a futuro implica que estas descargas se puedan realizar utilizando WFS y con consultas espaciales desde el mismo dispositivo móvil. Como una gran ventaja es que los dispositivos móviles como las Pocket PC o Palms es que

aumentan su capacidad de procesamiento y espacio rápidamente, por lo que será cuestión de tiempo que puedan recibir y procesar documentos en cGML.

5.2 La aplicación cliente en el celular

Hasta ahora el resto de las aplicaciones antes mencionadas fueron desarrolladas utilizando el .NET Framework de Microsoft, pero no todos los dispositivos móviles, específicamente celulares, cuentan con tal plataforma por lo que es necesario desarrollar una aplicación que permita la visualización utilizando una plataforma más extendida dentro de los mismos. Con base en esta demanda fue necesario desarrollar dicha aplicación utilizando J2Me, que representa una máquina virtual o plataforma más extendida dentro de los teléfonos celulares actuales.

5.2.1 Conociendo la plataforma J2Me

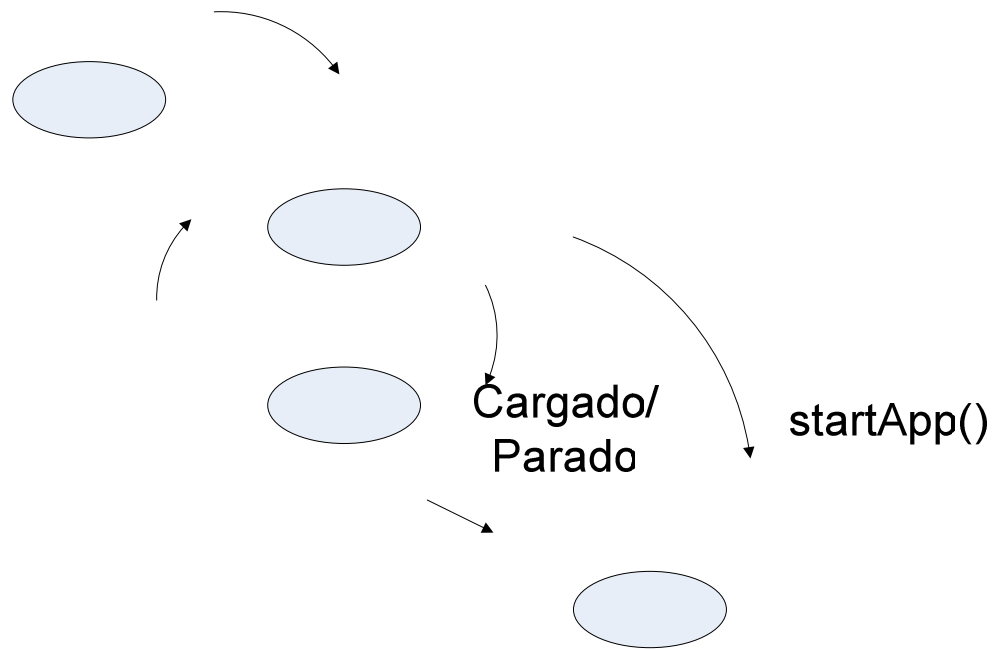
J2Me o *Java 2 Micro edition* es la microplataforma que Sun Microsystems ha desarrollado para dispositivos electrónicos y sistemas incrustados. La tecnología J2Me contiene dos tipos principales de componentes: de configuración y de perfiles [j2MeFaqs,2005].

El componente de configuración de J2Me se refiere a la tecnología que es utilizada en los dispositivos electrónicos que se conectan a una pared o se sostienen en la mano. La configuración se refiere a que la máquina virtual de J2Me utiliza librerías de

bajo nivel, separándose incluso en dos tipos de dispositivos: los que están entre 128 y 512 K de memoria y los que están arriba de 512 K [j2MeFaq,2005].

Los perfiles en cambio se basan en las librerías básicas que se utilizan en la configuración y se expanden los API's para ofrecer funcionalidades específicas para otro tipo de dispositivo y así satisfacer necesidades muy particulares de algún sector del mercado.

Ahora, tanto los clientes de Java 2 Micro Edition y Microsoft Compact Framework son máquinas virtuales demasiado delgadas, debido a la naturaleza de los componentes con los que se está trabajando, por eso mismo es que ha sido imposible utilizar diferentes utilerías que son fáciles de representar en una PC. Por ejemplo, una búsqueda de archivos o directorios y peticiones al Web. A continuación se presenta el ciclo de vida de un programa en J2Me (Véase figura 5.4):



Activar

Imagen 5.4 Ciclo de vida de un MIDlet

startApp()

Existen cuatro estados dentro de la ejecución de un MIDlet, el primero de ellos es el de carga o parado, el segundo es el de activación de donde se puede pausar o terminar su ejecución. El ciclo de vida del MIDlet es muy parecido al de cualquier proceso ejecutándose sobre la máquina virtual de JAVA como lo son los Applets.

pause

Pausar

A pesar de que J2Me se ha ido extendiendo como una máquina virtual incrustada a celulares, aún existe el problema de la versión que se tiene instalada, tanto en el perfil como en la configuración, esto lleva a una segunda clasificación dentro del API de J2Me, que corresponde a las instrucciones altas del J2Me y las de bajo nivel del mismo [j2MeFaqs,2005].

des

Las instrucciones altas de J2Me corresponden a pantallas (formas), cajas de texto, avisos, comandos, listas y diferentes comandos de etiquetas, espaciadores y grupos de selección. En las instrucciones de bajo nivel encontramos las de conexión con Internet, *bluetooth* y graficación de imágenes directo al *canvas* de la forma [j2MeFaqs,2005].

La diferencia entra utilizar solamente instrucciones altas de J2Me corresponde a que se logra sin ningún problema el intercambiar entre dispositivos y la aplicación sin que se afecte su funcionamiento o *look and feel*, en cambio las instrucciones de bajo nivel se ven muy relacionadas con el Hardware específico del dispositivo lo que hace que varias de estas aplicaciones se vean limitadas en portabilidad a otro tipo de dispositivos. En el caso de la implementación del cliente se tuvieron que hacer uso de ambos tipos de instrucciones con la finalidad de poder pintar en el *canvas* (instrucción de bajo nivel) del dispositivo y además utilizar el GUI (instrucción de alto nivel) para la interacción con el usuario.

Existen dos principales distribuciones del perfil MIDP (*Mobile Information Device Profile*), la versión 1.0 y la 2.0. La principal diferencia entre ambas versiones es la implementación de multimedia, mejor soporte de animaciones y conectividad. Por otro lado también existen dos principales distribuciones de la configuración CLDC (*Connected Limited Device Configuration*) que provee la capa básica de las interfaces para el desarrollo de aplicaciones para dispositivos móviles, como corresponde al manejo de operaciones con punto flotante. Lamentablemente existen bastantes dispositivos móviles que no soportan el manejo del punto flotante. Para observar detalladamente que

dispositivos utilizan que versiones del perfil y la configuración se puede revisar la especificación del dispositivo móvil [J2Me,2004].

5.2.2 Implementación del cliente en J2Me

La aplicación desarrollada en J2Me es un cliente bastante delgado, haciendo uso de instrucciones de alto nivel para el diseño e implementación de las interfaces y uso de instrucciones de bajo nivel para la representación de la imagen que se le envía al cliente.

El siguiente esquema representa a mayor detalle la aplicación:

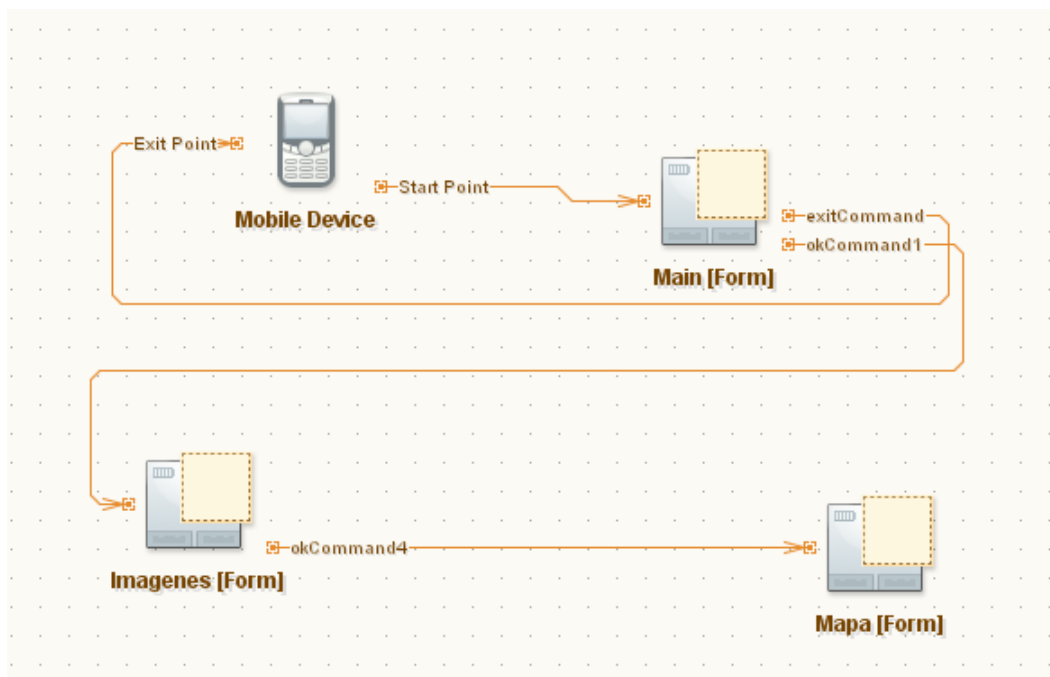


Imagen 5.5 Estructura de la aplicación

Donde la primera ventana consiste en la forma principal que permite seleccionar las acciones a ejecutar. En la aplicación actual sólo es posible salir o revisar alguna de las

imágenes precargadas en el dispositivo mediante el cambio de ventana a la forma Imágenes.



Imagen 5.6 Presentación del MIDlet

La siguiente ventana permite seleccionar la imagen del mapa que se desea presentar en el dispositivo móvil. Ahora, las imágenes están en formato JPG, GIF lo cual es más complicado pues el formato estándar de imágenes en celulares es PNG (que está basado en el algoritmo de GIF pero debido a la patente sobre el algoritmo de compresión se extendió su uso).



Imagen 5.7 Elección del mapa

Finalmente se presenta el mapa seleccionado en el dispositivo, la imagen es escalada al tamaño de pantalla que posea el dispositivo móvil, por lo que si la pantalla es demasiado pequeña esto representará una clara desventaja de usabilidad.



Imagen 5.8 Visualización del mapa

Es así como se puede llevar una representación cartográfica a cualquier dispositivo. La principal ventaja del programa expuesto es que utiliza las instrucciones más básicas tanto del perfil como la configuración del J2Me para poder ser transportada a cualquier dispositivo que soporte la máquina virtual. Las resoluciones de los celulares varían (sobre todo con modelos antiguos) entre 120 x 160 a 260 x 280 píxeles. Es parte del trabajo a futuro proveer un medio de comunicación entre el dispositivo móvil y alguna otra aplicación que permita evaluar y transmitir una respuesta (un mapa) ajustándose a la petición del dispositivo.

