

# Dispositivos Móviles y Multimedia.

César Tardáguila Moro.

Hace tiempo que los teléfonos dejaron de ser algo que servía para realizar llamadas, convirtiéndose en "dispositivos multimedia". Con el cambio de nombre, y de casos de uso, llegan múltiples oportunidades de tecnológicas y de negocio. En este documento se explora el estado actual de implantación de los dispositivos multimedia, algunos patrones de uso, y las oportunidades y retos que presenta el mercado asociado.

**Autor: César Tardáguila** tiene amplia experiencia en el campo de los dispositivos móviles como analista y jefe de proyecto, trabajando para varias consultoras de software. Actualmente, desarrolla aplicaciones para Mac e iPhone (<http://www.bambooapps.com>).

Primera edición del documento en octubre de 2006.

Revisado en enero de 2009.



Este texto se publica bajo licencia de "Reconocimiento – No Comercial – Sin obra derivada" 2.5 de Creative Commons. Más información en <http://mosaic.uoc.edu/commons.html>

## Índice

Introducción .....	4
¿A qué se puede llamar dispositivo móvil? .....	4
¿Qué tipos de dispositivos hay? .....	4
Historia de los dispositivos móviles .....	6
Plataformas y lenguajes soportados .....	12
Windows Mobile .....	12
Symbian .....	14
C++ .....	16
J2ME .....	15
Flash Lite .....	17
iPhone OS .....	19
Objective-C .....	19
Android .....	20
Acceso a internet .....	21
Herramientas de desarrollo .....	22
Consideraciones sobre el desarrollo. Limitaciones .....	23
Usos actuales de los dispositivos y perspectivas de futuro .....	25
Usos actuales .....	25
Perspectivas de futuro .....	27
Bibliografía .....	29
Libros .....	29
Sitios web .....	29

## Introducción

La vida actual no se podría entender sin computadoras. Lo que se conoció como "revolución de la microinformática" a principios de los años 80, con la popularización de los primeros ordenadores domésticos, ha terminado por resultar en la universalización de los ordenadores como herramientas y como electrodomésticos de consumo.

Es tal la presencia de dispositivos electrónicos en nuestras vidas, que a veces no nos damos cuenta de la relación tan estrecha que tenemos con ellos. Tan estrecha como para llevar uno de ellos siempre en el bolso.

Porque los dispositivos móviles no son más que otro miembro más de la familia de los ordenadores. El hermano pequeño, cierto, pero un hermano más a fin de cuentas.

A lo largo de este documento, iremos viendo en qué se parece el dispositivo móvil a su hermano mayor, el ordenador de sobremesa, para qué tareas está mejor preparado, qué se puede y qué no se puede hacer con él, y jugaremos a intentar adivinar el futuro de uno de los campos más excitantes en el desarrollo multimedia.

### ¿A qué se puede llamar dispositivo móvil?

Buena pregunta. La línea entre lo que es un dispositivo móvil y lo que no lo es puede ser un poco difusa, pero en general, se pueden definir como aquellos micro-ordenadores que son lo suficientemente ligeros como para ser transportados por una persona, y que disponen de la capacidad de batería suficiente como para poder funcionar de forma autónoma.

Normalmente, son versiones limitadas en prestaciones, y por tanto en funcionalidades, de los ordenadores portátiles o de sobremesa. Por cierto, los ordenadores portátiles no se consideran como dispositivos móviles, ya que consumen más batería y suelen ser un poco más pesados de lo que se espera de algo pensado para llevar siempre encima.

### ¿Qué tipos de dispositivos hay?

A grandes rasgos, y dependiendo del tamaño los dispositivos se pueden dividir en tres clases.

1.- Teléfonos. Son los más pequeños de la casa, y por tanto los más ligeros y más transportables. En general, también son los más baratos, aunque un teléfono de gama alta puede superar en precio a muchos de sus hermanos mayores, las PDAs. Su función primordial era clara: recibir y realizar llamadas; aunque parece que dentro de poco va a comenzar a ser complicado encontrar teléfono que sirvan para eso. Funcionalidades propias de ordenadores, o de dispositivos de otro tipo, como la grabación y edición de vídeo, realización de fotografías, lectura de documentos, localización en mapas, navegación por Internet, y muchas cosas más, son no sólo habituales, sino esperadas en cualquier teléfono moderno.



**Motorola DynaTAC 8000X, uno de los primeros teléfonos móviles fabricados en gran cantidad.**

2.- PDAs, organizadores electrónicos u ordenadores de mano. Su nombre (PDA) significa Personal Digital Assistant (asistente personal digital), un término acuñado en sus primeros años de historia, pero que resume bien su funcionalidad principal, que es servir como organizadores, con agenda, calendario, gestión de contactos, y que posteriormente han ido creciendo, de forma que actualmente sirven tanto como aparatos en los que leer un libro como en los que encontrarse en un mapa. La línea que los separa de los teléfonos es cada vez más difusa.



**Apple Newton MessagePad 2000**

3.- Consolas. En realidad esta categoría debería llamarse “dispositivos orientados a jugar”, porque son más que simples consolas. Los dos ejemplos actualmente en el mercado son la Sony PlayStation Portable (PSP) y la Nintendo DS, que no sólo sirven para jugar, sino que integran algunas de las funcionalidades típicas de una PDA, como reproducción de archivos multimedia, integración con agenda y calendario, o navegador de Internet.



**PSP**

## Historia de los dispositivos móviles

El primer dispositivo móvil, para muchos, es la Newton, desarrollada y comercializada por Apple, y que estuvo a la venta entre 1993 y 1998. La Newton era un dispositivo revolucionario para su tiempo, que implementaba un sistema de reconocimiento de escritura y que podía sincronizarse con un ordenador de sobremesa (de Apple, claro está). Fue tan revolucionaria, y se adelantó tanto a su tiempo, que fue un fracaso comercial, por lo que terminó retirándose del mercado.



**Panel de control de la Newton**

Pero aunque la Newton pueda considerarse como la primera PDA, desde luego no fue el primer dispositivo portátil programable.

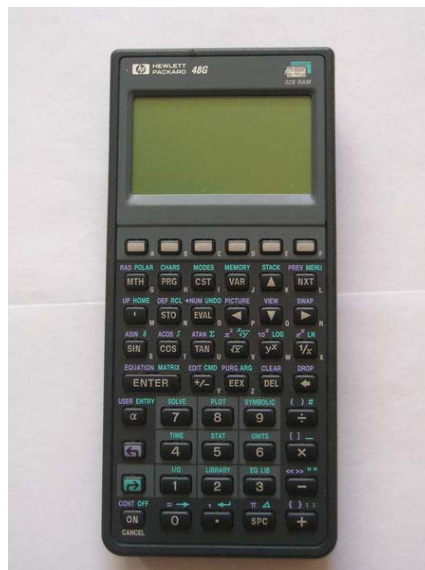
Durante los años 80, tanto Casio como Hewlett-Packard desarrollaron y comercializaron varias calculadoras programables, que si bien no tenían la capacidad de sincronizar sus datos con un ordenador de sobremesa, sí tenían capacidades gráficas, y accesorios que tal vez puedan sonar extravagantes ahora, como impresoras, o tarjetas de memoria extraíbles de 1Kb (sí, un "k").

Sirva como ejemplo la Casio FX-750P, de la que el autor de este texto conserva un ejemplar que funciona desde el año 1984, que tenía una pantalla de una sola línea, con 32 caracteres de 5x7 puntos, teclado QWERTY y teclado numérico, 4 KB de RAM, dos ranuras para tarjetas de memoria extraíbles de hasta 2 KB, y puerto serie por el que conectarla a una impresora. Y era programable en BASIC. ¿Especificaciones de risa, verdad? No tanto, si se tiene en cuenta que la configuración CLDC de J2ME sólo obliga a que el teléfono tenga 4KB de memoria, y que muchos de los móviles pre-Symbian tienen una memoria de 32 KB. Más adelante veremos con un poco más de detalle qué es eso de J2ME, y qué es la configuración CLDC, pero por ahora basta con saber que es la especificación que deben cumplir los móviles programables en java.



**Casio FX-750P**

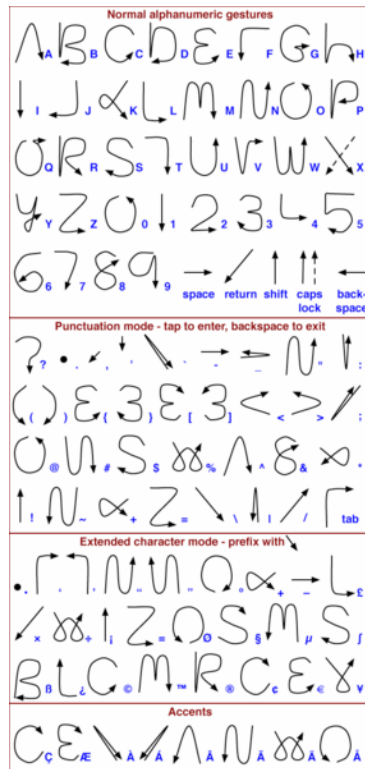
Las calculadoras programables de Casio rivalizaban en prestaciones con las de Hewlett-Packard. Si la FX-750P podría considerarse como el buque insignia de la marca japonesa, la serie HP48 lo era para los californianos. Las calculadores de esta serie, que se dividían en dos ramas, la S para los modelos estándar y la G para los de mayor funcionalidad, estuvo en producción entre los años 1990 y 2003. Las especificaciones comunes a todos los modelos de la gama eran una pantalla de 131x63 píxeles, un puerto de comunicaciones por infrarrojos y otro serie de 4 pines, y 512 KB de memoria. El modelo más alto de la gama, la HP 48GX, soportaba dos tarjetas de expansión de memoria, de forma que se podía llegar a acumular un total de 5MB.



**HP 48G**

Tras la muerte de la Newton, nacieron los dos dispositivos que durante unos años dominaron el mercado: la Pilot y el PocketPC.

Los dispositivos de Palm se adelantaron a los de Microsoft en un par de años. En parte por ello, su salida a producción fue un éxito de ventas, llegando incluso a acumular, en 2001, un 60 por ciento del mercado americano. Sin embargo, Palm se resintió del declive global del mercado de las PDA, comenzando un declive que coincidió prácticamente en el tiempo con la entrada en el mercado del gigante de la informática de consumo: Microsoft.



**Graffiti, el sistema de reconocimiento de caracteres de Palm OS, obligaba a escribir de forma determinada.**

En el año 2000 vio la luz el primer PocketPC, el hijo de la Newton, y que al contrario que ésta, sí ha sobrevivido hasta hoy, pese a no ser un éxito comercial de grandes dimensiones, gracias a la continuidad y el apoyo decidido de Microsoft.

Los primeros PocketPC tenían como sistema operativo el llamado Windows CE 3.0. Por su nombre podría parecer que era una versión aligerada del sistema operativo más utilizado en el mundo, pero en realidad no tenía mucho que ver con éste.



**Compaq iPaq 3630, el primer PocketPC de éxito**



La mayor razón para el éxito del PocketPC ha sido su relativa sencillez de manejo y su integración con ordenadores de escritorio basados en Windows. La integración es tal que para realizar una sincronización entre ambos basta con conectar el PocketPC al ordenador con un cable.

Actualmente, los PocketPC y las Palm tienen pantallas de resolución VGA, en prácticamente todos los casos incorporan protocolos de comunicaciones inalámbricos, como Bluetooth o Wifi, o unidades de GPS. Por prestaciones, son, sin duda, los hermanos mayores del mundo de la movilidad.

Pero si las PDAs son las reinas en lo referente a prestaciones, en cuanto a unidades en el mercado, el rey indiscutible es el teléfono móvil.

¿Quién no tiene un teléfono móvil? En realidad, lo raro es que haya quien no tenga un teléfono totalmente funcional guardado en un cajón, porque se ha comprado otro que, en el plazo de unos meses, multiplicaba por diez o por cien las prestaciones de su antiguo terminal.

Si la Newton, la Palm y el PocketPC han sido los que han abierto el camino, los que realmente han entrado con toda la fuerza posible a ocupar ese mercado han sido los teléfonos móviles. Actualmente, se calcula que en España hay más terminales móviles (unos 44 millones) que habitantes. En el 37% de los hogares hay un móvil, en el 34% hay dos terminales, y en el 17% hay tres, según datos de Cetelem de finales de 2005.

Pero ¿cuál es el motivo o los motivos para que haya tantos teléfonos? Es difícil de decir con total seguridad, pero se puede resumir principalmente en tres.



**Motorola MicroTAC**

En primer lugar, el precio. Se pueden conseguir terminales por cero euros en ocasiones excepcionales, como al darse de alta con algún operador, pero en todo caso los precios de los teléfonos de gama baja suele ser bastante asequible.

El segundo factor es más complejo, pero puede resumirse en “mira lo moderno que soy, qué móvil tengo”. Hay quien lo llama *techno-lust*, hay quien lo llama “ganas de aparentar o de chulear a las amistades”, pero básicamente se resume en que todo el mundo quiere tener el móvil más moderno y con más funcionalidades, sobre todo si es más moderno y hace más cosas que el del vecino. Esa obligación por tener el dispositivo más moderno viene motivada también en parte por la continua mejora de la tecnología.

Cada vez los móviles pueden hacer más cosas, y eso crea una doble necesidad. Por un lado, es necesario tener un dispositivo que permita hacer esas cosas, sean las que sean (grabar vídeo, escuchar mp3, ver los goles de tu equipo favorito en tiempo real). Pero eso también provoca que cada vez se intenten hacer más cosas con el teléfono, lo que vuelve a hacer necesario un terminal más potente, que a su vez creará la necesidad de contenidos de más calidad, lo que a su vez... Esto supone nuestro tercer factor: un círculo vicioso para el usuario pero que beneficia por igual a los fabricantes y a las operadoras.

Sin embargo, los móviles que verdaderamente pueden considerarse equivalentes a las PDAs son los de gama alta, o *smartphones*.

El término *smartphone* es engañoso. Su traducción literal sería “teléfonos inteligentes”, y se utiliza indiscriminadamente para hacer referencia a cualquier teléfono de gama alta, englobando tanto a dispositivos de la Serie 60 o superior de Symbian, como a los que funcionan bajo Windows Mobile o bajo Palm OS. De todas formas, la marca comercial *SmartPhone* es propiedad de Microsoft.



**Sony Ericsson P910i**

Porque, efectivamente, hay una línea de teléfonos que funcionan bajo Windows Mobile. Por un lado, los PocketPC Phone Edition, que son híbridos de teléfono y PocketPC y por otro los llamados comercialmente SmartPhones. La diferencia entre ambos está, fundamentalmente, en la pantalla táctil. Los PocketPC Phone Edition tienen, mientras que los SmartPhones no, y que se manejan de forma similar a los móviles Symbian de gama alta. Tanto PocketPC Phone Edition como PocketPC se engloban dentro de la gama *Windows Mobile* de Microsoft.<sup>2</sup>

También Palm mantiene su irreductible nicho de mercado dentro de los teléfonos de gama alta: la serie Treo. Los Treo son híbridos de teléfono y PDA, con teclado QWERTY, y funcionan, salvo uno de los modelos (en concreto el Treo 700w, que utiliza PocketPC Phone Edition) bajo PalmOS.



Treo 700w

En todo caso, el mercado de los teléfonos de gama alta estaba copado, hasta hace muy poco, por los modelos con sistema Symbian.

Sin embargo, en los últimos dos años hemos asistido a la irrupción del representante por excelencia del “techno-lust” el iPhone de Apple. Si Apple fue la que abrió el camino con la Newton, es el que ha vuelto a revolucionar el mercado con el iPhone, en parte porque ha sido el único fabricante que realmente ha conseguido desarrollar un dispositivo que integrara a todos los anteriores existentes por separado (teléfono, reproductor mp3 y PDA), reinventando la forma en la que se interactúa con él.

#### Para los más curiosos

PDA en wikipedia:

<http://es.wikipedia.org/wiki/PDA>

Newton en wikipedia y en inglés:

[http://en.wikipedia.org/wiki/Apple\\_Newton](http://en.wikipedia.org/wiki/Apple_Newton)

Breve historia de la PDA:

<http://www.telediariodigital.com.ar/weblogs/tic/leer.asp?idx=9>

Los 10 fracasos más sonados en la historia de las PDAs (algo desactualizado, pero interesante):

<http://www.canalpda.com/2005/01/18/279-10+fracasos+sonados+historia+pda>

Reviews de todos los dispositivos con palm OS

<http://www.palminfocenter.com/news/3628/all-reviews-of-pdas-handhelds-and-smartphones/>

Lista de los dispositivos con palm OS:

[http://en.wikipedia.org/wiki/List\\_of\\_Palm\\_OS\\_Devices](http://en.wikipedia.org/wiki/List_of_Palm_OS_Devices)

Otra lista de dispositivos con palm OS:

<http://www.geocities.com/netbrakr/>

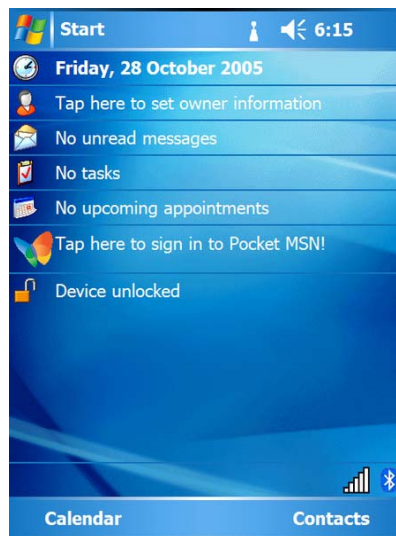
## Plataformas y lenguajes soportados

Cada una de las plataformas tiene sus particularidades, no sólo en cuanto al manejo del dispositivo por el usuario, sino también a la hora de desarrollar aplicaciones para las mismas.

No será lo mismo programar una aplicación para Windows Mobile que para Symbian, entre otras cosas porque no todas las plataformas soportan los mismos lenguajes de programación.

## Windows Mobile

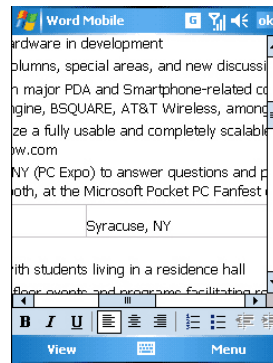
Como ya hemos visto con anterioridad, el sistema operativo en el que están basados los PocketPC actualmente se llama Windows Mobile.



**Pantalla de inicio típica de Windows Mobile 6**

Este sistema se pretende vender como una versión muy aligerada de Windows, pero en realidad no tiene mucho que ver con él. El paradigma de funcionamiento es similar, basado en ventanas, aunque éstas se comportan de forma muy diferente a como lo hacen en su hermano mayor de escritorio. Por ejemplo, al cerrar la ventana de un programa éste no se cierra realmente, sino que se sigue ejecutando como si se hubiera minimizado.

El punto fuerte de estos dispositivos es que ofrecen funcionalidades similares a las de sus hermanos mayores. Por ejemplo, se pueden editar documentos de word, hojas de cálculo de excel, leer libros en formato pdf o chm, recibir y enviar correo electrónico, manejar una agenda, la libreta de contactos, sincronizar datos con el PC, navegar por internet, utilizar un GPS... en definitiva, casi lo mismo que en un PC, pero sin teclado y con la pantalla bastante pequeña.



**Versión para Windows Mobile de Microsoft Word**

La entrada de datos se realiza a través de la pantalla, que es táctil, y gracias al sistema de reconocimiento de escritura que implementan, que permite trabajar de una forma bastante natural y rápida.

La mayoría de las aplicaciones para estos dispositivos se desarrollan en .Net, la plataforma de desarrollo de Microsoft, o directamente en C++, aunque aún quedan, como reminiscencia de sus inicios como Windows CE, algunas aplicaciones escritas en Embedded C++ o Embedded Basic, que eran dos entornos de desarrollo basados respectivamente en C++ y Basic, pero con muchas limitaciones.

Pero en la actualidad casi todos los desarrollos para Windows Mobile se realizan en C++ o en .Net, basándose en el Compact Framework.

El principal problema a la hora de desarrollar aplicaciones para estos dispositivos es el precio de las herramientas de desarrollo. Estas herramientas (Visual Studio, un entorno de desarrollo que soporta varios lenguajes como C#, C++, J#, JScript o ASP .Net) son de altísima calidad, pero de un precio que no está al alcance de todos los que se quieran aventurar a escribir una aplicación para PocketPC.

Por otra parte, el hecho de que Microsoft esté detrás de la plataforma, dándole todo su apoyo, se nota en la cantidad y la calidad de la documentación disponible para los desarrolladores. Newsletters, una sección sólo para dispositivos en el MSDN, blogs de los ingenieros de Microsoft que trabajan en Windows Mobile, actualizaciones de la documentación en DVDs periódicos... en resumen, mucho y muy bueno.

Tal vez por eso mismo, y pese al precio del entorno de desarrollo, la competencia es muy alta. Hay muchas compañías que producen aplicaciones para Windows Mobile, y que invierten mucho dinero en su desarrollo. No es, por tanto, un nicho de mercado en el que sea sencillo introducirse.

En cualquier caso, siempre que se vaya a desarrollar aplicaciones para esta plataforma, deben tenerse en cuenta las limitaciones específicas de la misma, sobre todo a la hora de intentar desarrollar interfaces que sean lo más point-and-click posibles, y que necesiten de la menor introducción de textos posible.

#### Para los más curiosos

Página principal de Visual Studio en MSDN:

<http://msdn.microsoft.com/es-es/vstudio/default.aspx>

Visual Studio Express, versiones gratis pero con limitaciones de las herramientas de desarrollo de Microsoft:

<http://www.microsoft.com/Express/>

Windows Mobile Developer Centre:

<http://msdn.microsoft.com/en-us/windowsmobile/default.aspx>

Los grandes errores a evitar en el desarrollo para Windows Mobile:

<http://www.developer.com/tech/article.php/3588021>

Blog con contenido de calidad relacionado con el desarrollo para Windows Mobile:

<http://www.christec.co.nz/blog/>

## Symbian

Si los PocketPC están basados en un sistema operativo de Microsoft llamado Windows Mobile, la gran mayoría de los teléfonos móviles funcionan gracias a un sistema operativo llamado Symbian. Hace unos años, a estos teléfonos se les llamaba con el nombre genérico de SmartPhones.

Symbian es un consorcio en el que participan los mayores fabricantes de teléfonos móviles, con Nokia a la cabeza. Sony Ericsson, BenQ (que hace poco absorbió a Siemens), Fujitsu, Lenovo, Motorola, Panasonic, Samsung o Sharp están representados en Symbian, y son por tanto, participes en el desarrollo y la expansión del sistema.

De todos estos fabricantes, el que mayor cuota de mercado tiene es Nokia (que con algunas fluctuaciones, viene siendo desde hace años de alrededor del 40%), que además es el fabricante que primero apostó por este sistema operativo, y el que más uso hace de él. Por tanto, a partir de ahora, al referirme a teléfonos Symbian, lo haré sobre todo a teléfonos Nokia.

Symbian es un sistema operativo escrito en C++, por lo que presenta muy bajo consumo de recursos del dispositivo, a la vez que se ejecuta con gran rapidez. El sistema operativo Symbian se presenta en varios "sabores". En concreto, Nokia divide sus dispositivos Symbian en tres familias, que se llaman respectivamente Serie 40, Serie 60 y Serie 80.

La Serie 40 es la que agrupa a los teléfonos Symbian con pantallas más pequeñas (en general, hasta de 240x320 píxeles), y tiene ciertas limitaciones sobre las otras series, sobre todo en lo referente a la cantidad de recursos del teléfono que puede poner a disposición de las aplicaciones que se ejecuten sobre él.

La Serie 60 es la más extendida, y es donde suelen estar los teléfonos de gama media y alta de Nokia, a los que muchas veces se hace referencia como teléfonos multimedia. Con alguna excepción, suelen tener pantallas más grandes que los de la serie 40, y suelen ser teléfonos específicamente optimizados para ejecutar aplicaciones J2ME.

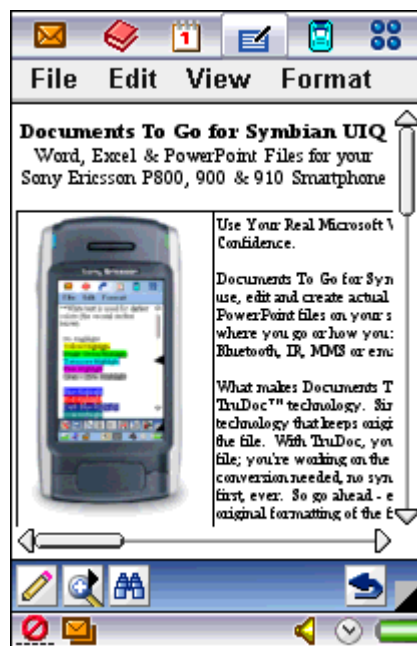


### Captura de una interfaz Series 60

La Serie 80, finalmente, es la de los llamados Communicators, esos teléfonos que se abren longitudinalmente y que esconden teclados QWERTY. En realidad son un paso intermedio entre el teléfono y la PDA.

Aparte de la implementación que hace Nokia de Symbian, Sony Ericsson implementa otro sabor más de ese sistema, llamado UIQ, que se maneja por teclado o a través de una pantalla táctil. Estos dispositivos suelen implementar sistemas de reconocimiento de escritura, y los más conocidos son, como he dicho, la serie P de Sony Ericsson.

Actualmente, en el mercado de los teléfonos móviles Symbian es el sistema más extendido, aunque no el único. La mayoría de los fabricantes siguen implementando sus propios sistemas, por lo que sigue sin haber una uniformidad o coherencia entre dispositivos.



Symbian UIQ

Las tres grandes posibilidades tecnológicas a la hora de desarrollar aplicaciones para teléfonos móviles son C++, J2ME y Flash Lite.

### C++

Es el lenguaje de programación por excelencia para aplicaciones que necesitan extraer el máximo del terminal, tanto en capacidad de procesamiento y por lo tanto en velocidad de ejecución, como en utilizar las posibilidades de hardware que ofrezca el dispositivo. El ejemplo más claro de aplicación candidata a ser realizada en C++ sería un juego de conducción, con capacidades multiusuario a través de bluetooth.

Es ideal, por tanto, para aplicaciones críticas (como por ejemplo, sistemas operativos para teléfonos como Symbian).

C++ es un lenguaje de programación, de los que se llaman orientados a objetos, que en realidad es una evolución del lenguaje más utilizado en la historia de la informática: el lenguaje C. Desarrollar en C++ es bastante complicado para casi cualquier programador. Hacen falta, por tanto, programadores de un perfil muy especializado o muy alto, y el lenguaje en sí no hace



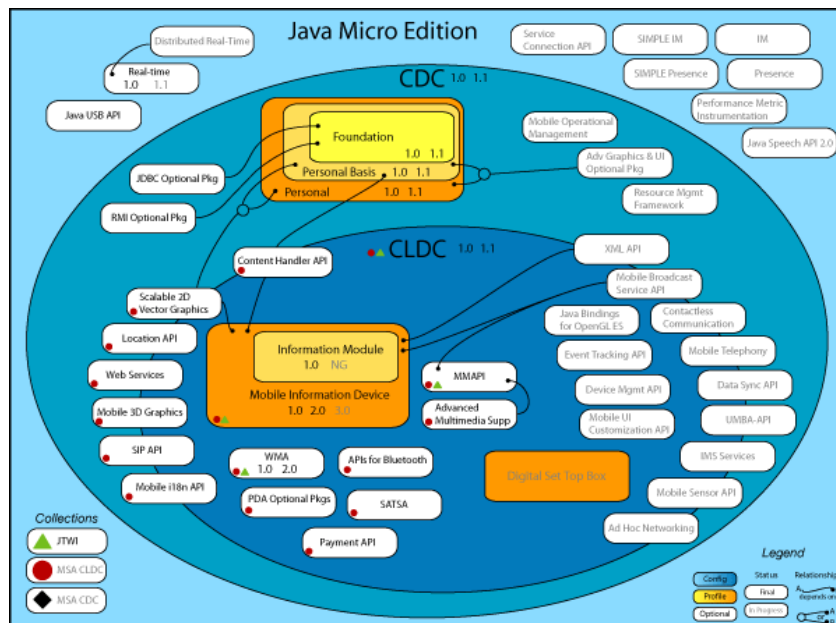
posible el desarrollo de aplicaciones con gran rapidez, con lo que los proyectos desarrollados en C++ suelen ser más caros que aquellos producidos con otros lenguajes.

**J2ME**

J2ME es un subconjunto del lenguaje Java. ¿Pero qué es el lenguaje Java? Es un lenguaje de programación desarrollado a principios de los años noventa por James Gosling y algunos de sus colegas de Sun Microsystems. Las principales características de este lenguaje son, por un lado, que es de los llamados orientados a objetos, lo que supuestamente permite desarrollar aplicaciones más complejas con mayor facilidad, y por otro, que fue diseñado para ser independiente de la plataforma, lo que supuso una gran novedad en su momento.

Sin embargo, el caso de J2ME es especial, ya que es de todo, menos independiente de la plataforma.

La especificación J2ME está dividida en dos grandes grupos, dependiendo de la cantidad de funcionalidades para las que se quiera dar soporte. Esos grupos son las llamadas Configuraciones (en toda la documentación sobre J2ME se suele hacer referencia a ellas por su nombre en inglés: Configurations). Hay dos configuraciones, que dividen la plataforma en dos grandes grupos de dispositivos.



**El mapa con las configuraciones y perfiles J2ME. ¿Lioso?**

Por un lado, los dispositivos más potentes, son los que soportan la Configuración CDC. Durante mucho tiempo en este grupo sólo cabían las PDAs (dispositivos como la Palm, por ejemplo), dadas las exigencias de memoria y tamaño de pantalla para cumplir con la especificación.

La otra configuración, la llamada CLDC, es la que agrupaba a los dispositivos con menor capacidad de procesamiento, y es donde siempre han estado los móviles.

Actualmente, Nokia, el mayor fabricante de teléfonos móviles, está en proceso de portar toda su gama de teléfonos Symbian a la Configuración CDC, un indicador más de cómo el hardware va mejorando progresivamente en prestaciones.

Dentro de cada una de las configuraciones, no todos los dispositivos soportan las mismas funcionalidades. Éstas está divididas en los llamados Perfiles (o en su nombre original, Profiles). Los dos perfiles de la CLDC son el MIDP 1.0 y el MIDP 2.0. Como podría suponerse, el perfil MIDP 2.0 incorpora librerías que no aparecen en los teléfonos MIDP 1.0, con las que se



pueden implementar aplicaciones que accedan a la cámara, que puedan capturar imágenes, vídeos, que accedan a servicios web, etc.

Los dispositivos que se adhieran a la especificación de cada uno de esos perfiles tienen que dar soporte a una serie de funcionalidades obligatoriamente. Esas funcionalidades vienen empaquetadas en lo que se llama especificaciones. Algunas de esas especificaciones son obligatorias para poder decir que el dispositivo soporta uno u otro de los perfiles, mientras que otras son opcionales y pueden o no ser implementadas por el fabricante. Por ejemplo, un teléfono que implemente gráficos 3D deberá hacerlo soportando la especificación jsr-184.

¿Pero por qué es todo tan complicado? ¿Porqué es necesaria tanta especificación, perfiles, etc?. Para asegurar un mínimo de compatibilidad entre dispositivos desarrollados por fabricantes dispares, y para asegurar al programador un mínimo común denominador en el que su aplicación funcione.

En todo caso, la adhesión a una especificación no asegura que las cosas funciones igual y bien en todo tipo de dispositivos, por eso es importante testear pronto y testear a menudo, y en el mayor número de dispositivos posibles.

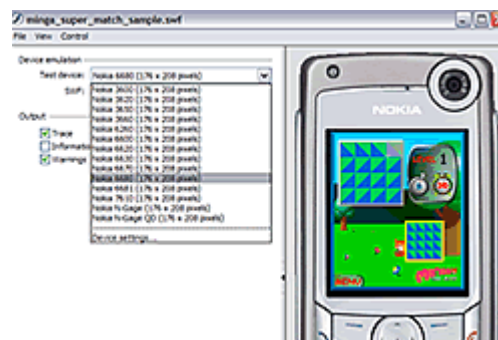
Además, debe tenerse en cuenta que ninguna especificación obliga a un tamaño de pantalla determinado. Por tanto, hay que ser extremadamente cuidadoso en delegar la presentación en pantalla en el propio dispositivo. Lo cual tampoco es necesariamente un problema, ya que la propia plataforma funciona así, dejando al dispositivo que sea el que pinte las cosas. Eso tiene como ventaja que si se hace bien, siempre saldrá bien, y como inconveniente, que no se puede suponer que las aplicaciones se vayan a ver igual en dos teléfonos distintos. Pero donde es más importante esa limitación es a la hora de desarrollar juegos, a que suelen ser a pantalla completa, y con elementos de interfaz propios, y en los que se suele buscar a toda costa la consistencia entre dispositivos.

En todo caso J2ME es la plataforma de desarrollo para móviles más extendida, porque no está limitada sólo a los teléfonos con sistema Symbian, porque durante mucho tiempo ha sido la única disponible para los desarrolladores, y porque algunos años en los que no ha sido la única, programar en J2ME era la única forma de implementar muchas funcionalidades. De hecho, aún hoy, sigue siendo la plataforma más extendida, con mayor número de desarrolladores, y con mayor número de aplicaciones disponibles.

## Flash Lite

Flash Lite es un adaptación de la plataforma Flash para dispositivos móviles. Actualmente coexisten varias versiones, Flash Lite 1.1, Flash Lite 2.0 y Flash Lite 2.0

Flash Lite 1.1, es la versión más extendida, sobre todo en Asia, y basada en la versión 4 de player de flash. Y es la más extendida por dos razones primordiales; en primer lugar porque es la que más tiempo lleva disponible, y en segundo lugar porque es la que necesita de hardware menos potente para su ejecución.



Emulador incorporado en Adobe Flash 8

Al estar basado en la versión 4 del player de flash, sólo soporta la sintaxis de ActionScript de Flash 4, una sintaxis bastante extraña y alejada de las convenciones de programación modernas. Esa forma de programar necesita de vinculaciones muy fuertes entre el código y los gráficos, por lo que es muy difícil que los desarrolladores que no estén acostumbrados a ella se puedan adaptar con facilidad.

Además, para desarrollar cualquier aplicación con cierta interactividad, termina siendo necesario repartir el código entre multitud de elementos gráficos, por lo que la complejidad del desarrollo crece exponencialmente con la complejidad de la aplicación. Por no hablar del coste de mantenimiento.

Tampoco se pueden obviar las limitaciones propias de flash Lite 1.1. Por ejemplo, no se pueden guardar datos en la memoria del teléfono, no es nada fácil cargar datos en tiempo de ejecución, y directamente no es posible cargar imágenes, por ejemplo.

Pese a todo, al igual que para crear aplicaciones complejas, Flash Lite 1.1 no es, desde luego, la mejor herramienta, sí es ideal para realizar otro tipo de aplicaciones, como fondos de pantalla animados, salvapantallas, o incluso microjuegos muy sencillos. ¿Por qué? Precisamente por estar basado en Flash 4, que permite desarrollar animaciones con interactividad sencilla en tiempos mucho menores que cualquier otra plataforma.

Todo cambió, sin embargo, con la llegada de Flash Lite 2.0.

En primer lugar, Flash Lite 2.0 está basado en el Flash Player 7. En realidad, la funcionalidad que permite es muy similar a la de cualquier aplicación flash basada en esa versión del player.

Por tanto, ahora es posible guardar datos en la memoria del teléfono o cargar ficheros XML con estructuras de datos complejas a través de la conexión GPS. Para el programador la mejora ha sido también sustancial, ya que puede aplicar las técnicas modernas de programación: programación orientada a objetos, patrones de diseño, aplicaciones dirigidas por eventos... Ahora es posible desarrollar una aplicación Flash Lite de la misma forma en la que se haría una J2ME, por ejemplo.

Aunque las mejoras son sustanciales, también hay un aspecto en el que Flash Lite 2 cojea, y es los altos requerimientos de hardware que necesita para su ejecución. Por ahora sólo está disponible para teléfonos de la gama alta de la Serie 60 de Nokia.

Flash Lite 3 es la evolución lógica de la plataforma, soporta vídeo incluyendo H.264, y un entorno de desarrollo rápido. Sin embargo, al menos en Asia, donde al menos en cuestión de dispositivos se está a la cabeza del desarrollo mundial, y por varios cuerpos de distancia, la implantación de la plataforma es mínima.

#### Para los más curiosos

Symbian OS en wikipedia:

[http://en.wikipedia.org/wiki/Symbian\\_os](http://en.wikipedia.org/wiki/Symbian_os)

Symbian Developer network:

<http://developer.symbian.com/>

Una breve introducción a C++

<http://www.cprogramming.com/tutorial/lesson1.html>

¿Qué es J2ME? Según Sun, claro...

[http://java.com/es/download/faq/whatis\\_j2me.xml](http://java.com/es/download/faq/whatis_j2me.xml)

Java a tope: J2ME. Libro gratis sobre J2ME

<http://www.lcc.uma.es/~galvez/J2ME.html>

Sobre las configuraciones J2ME

<http://www.java-samples.com/showtutorial.php?tutorialid=440>

Vídeo de introducción a Flash Lite 3

[http://www.adobe.com/devnet/devices/articles/flash\\_lite3\\_training\\_video.html](http://www.adobe.com/devnet/devices/articles/flash_lite3_training_video.html)

BlocketPC

<http://www.blocketpc.com/>

Flash Lite 4 Nokia:

<http://www.flashlite4nokia.com/>

## **iPhone OS**

---

El iPhone, el último en llegar, rompe también con las líneas maestras en cuanto a desarrollo se refiere, marcadas por sus antecesores.

El sistema operativo utilizado es una versión aligerada de Mac OS X, el sistema detrás de los ordenadores de Apple, y por tanto, las herramientas de desarrollo que se deben utilizar son las mismas que para trabajar en escritorio.

### **Objective-C**

El lenguaje que debe utilizarse para desarrollar aplicaciones nativas para iPhone OS es Objective-C, un superset de C (una especie de C enriquecido), de forma que el desarrollador se apoya en un extenso Framework orientado a objetos, escalable y altamente modular, llamado Cocoa.

En realidad, Cocoa es una colección de frameworks, que proporcionan todas las piezas necesarias para construir una aplicación: desde elementos de interfaz hasta gestión de tráfico de red.



Una aplicación para iPhone, casualmente desarrollada por el autor.

Pese a lo modular y extenso de los frameworks, el mayor problema con el que se encuentran los recién llegados a la plataforma es que la curva de aprendizaje de Objective-C es bastante acusada. Especialmente si, además, se viene de un entorno de desarrollo Windows, ya que no sólo hay que cambiar de lenguaje, sino de forma de desarrollo por completo.

Además, para poder desarrollar aplicaciones para iPhone es necesario estar registrado (previo pago) como desarrollador con Apple, para así poder obtener los certificados digitales necesarios para que las aplicaciones funcionen en los dispositivos. Por si fuera poco, la única forma de poner esas aplicaciones en esos dispositivos, es a través de una tienda de aplicaciones controlada por Apple.

#### Para los más curiosos

Apple Developer Connection:

<http://developer.apple.com/iphone/>

Introduction to Objective-C

[http://developer.apple.com/documentation/Cocoa/Conceptual/ObjectiveC/Introduction/chapter\\_1\\_section\\_1.html](http://developer.apple.com/documentation/Cocoa/Conceptual/ObjectiveC/Introduction/chapter_1_section_1.html)

## Android

Android es un sistema operativo para teléfonos, basado en el núcleo de Linux, y que, aunque disponible para cualquier fabricante como open-source, actualmente es el motor de los dispositivos comercializados por Google (en el momento de escribir este texto, sólo existe un dispositivo, el conocido en Asia como HTC Dream, y en el resto del mundo como G1).

Android también proporciona al desarrollador un completo Framework Java, un intento de normalización de la selva de especificaciones en que se ha convertido J2ME, que al igual que en el caso del iPhone, está orientado a facilitar y hacer más rápido el desarrollo, por un lado, y a proporcionar un "look and feel" específico y reconocible de la plataforma.

También se proporciona una tienda de aplicaciones, donde se pueden vender desarrollos comerciales.

En este momento, aún está por ver si la plataforma termina por alcanzar el momento suficiente como para terminar despegando.

#### Para los más curiosos

Android en wikipedia:

<http://es.wikipedia.org/wiki/Android>

Sitio web oficial, con acceso a la tienda de aplicaciones y a la documentación para desarrolladores:

<http://www.android.com/>

### Acceso a internet

Hemos hablado de aplicaciones, pero no de Internet. ¿Es que no se puede navegar por Internet desde un dispositivo móvil?

Sí, por supuesto, se puede navegar por Internet desde un dispositivo móvil. Pero, como tantas otras cosas en estos ordenadores, con ciertas limitaciones. Y esas limitaciones vienen dadas por los navegadores a utilizar.

En el caso de los PocketPC, Windows Mobile incluye una versión de Internet Explorer muy rebajada de funcionalidad, y que no soporta muchas de las cosas que soporta su hermano mayor.

También, desde hace relativamente poco tiempo, hay una versión de Firefox, llamada Minimo, pero que sólo funciona bajo Windows Mobile 5.0, y que es un poco propensa a leaks de memoria, como por otra parte cabe esperar de una aplicación cuyo número de versión es 0,013

En los dispositivos Symbian, sin embargo, el navegador que se está convirtiendo en la mejor opción para muchos usuarios es Opera. Opera Mobile es un navegador sólido, que soporta los estándares web, y que permite navegar sitios web "normales", que no tienen porqué ser WAP necesariamente. Y del que también hay versión para Windows Mobile.



Web del diario El País vista con Opera Mobile

En todos los casos, debe tenerse en cuenta, a la hora de desarrollar webs para dispositivos móviles, una serie de aspectos que harán que la web desarrollada sea accesible sin problemas para el usuario.

Por un lado es muy importante separar contenido de presentación. La utilización de marcado XHTML estricto en las páginas, junto con hojas de estilo en cascada para formatear los datos, permite, por un lado, que los navegadores sean capaces de renderizar la página sin problemas, y por otro, asignar una css específica para dispositivos móviles.

De esa forma, se pueden servir a los usuarios que utilicen, por ejemplo, un teléfono para visitar una web páginas sin imágenes, que se cargarán en menos tiempo, y serán más ligeras, por lo que le costarán menos dinero.

Además, hay que tener en cuenta que es bastante normal que los usuarios deshabiliten la carga de imágenes a la hora de navegar por Internet desde un dispositivo móvil, para de esa forma reducir no sólo los precios de espera sino los costes de la conexión, que en los casos de GPRS o 3G se tarifica por Kb. Por tanto, a la hora de plantear la estructura de la web, debemos evitar depender de imágenes para que se pueda navegar de forma correcta. Es buena idea, por tanto, hacer que todos los elementos de los menús sean sólo texto. Tampoco tiene sentido plantear efectos dinámicos, como rollovers, y hay que ser extremadamente cuidadosos a la hora de asignar los estilos de los enlaces visitados y no visitados, para que sean visibles.

También hay que ser precavidos con los tipos de archivos que se muestran. Por ejemplo, no se deben incluir archivos swf, quicktime, o cualquier otro que no sea estrictamente html. También hay que tener cuidado con los posibles archivos que se cuelguen para descarga (zip, rar, incluso archivos de Word), que pueden no ser legibles para el dispositivo.

En el caso del iPhone, la versión de Safari incluida permite navegar webs sin hojas de estilo especiales. Sin embargo, al no soportar ningún plugin externo, el único contenido multimedia visualizable es el que esté en formato quicktime.

## Herramientas de desarrollo

Las herramientas de desarrollo a utilizar dependerán de la plataforma para la que hayas planteado tu aplicación.

Si tu aplicación debe funcionar en dispositivos Windows Mobile, deberás utilizar el Visual Studio de Microsoft, un entorno de desarrollo bastante caro, como ya hemos dicho, pero que es casi sin discusión el mejor de todos los disponibles, y para el que vas a contar con más documentación de la que puedas llegar a absorber.

Deberás basar tu aplicación, o bien en código C++ nativo, con lo que necesitarás unos conocimientos técnicos bastante elevados antes de comenzar el desarrollo, o bien apoyándote en el .Net Compact Framework, que es un subconjunto del .Net Framework, pudiendo programar en C#.

Para dispositivos Symbian, la opción más extendida es, como hemos visto, J2ME. Las herramientas de desarrollo para J2ME no están tan refinadas como las de Windows Mobile, pero aún así, hay un entorno específico, el NetBeans Mobility Pack, y plugins y add-ons para los IDEs java más populares: Eclipse e IDEA.

Los tres entornos, una vez instalados los plugin correspondientes, permiten desarrollar, empaquetar y testear aplicaciones J2ME en los emuladores incorporados. Pero nunca debe olvidarse que los emuladores no son el dispositivo real, sino, como su nombre indica, emulaciones del dispositivo que están corriendo en una máquina que es varios órdenes de magnitud más rápida. Por tanto, lo que en un emulador funciona bien, puede que en el dispositivo no funcione tan bien.

Por eso, siempre que se desarrolle para un dispositivo móvil, pero sobre todo si se está haciendo en J2ME o Flash Lite, hay que probar los programas en el dispositivo según se están desarrollando. Y cuanto antes se empiece a probar, mejor, de esa forma los posibles problemas aflorarán antes, y serán más fáciles de resolver.

En el caso del iPhone será necesario un Mac con procesador Intel (no existe SDK para Windows ni para Linux), y aprender un lenguaje y una metodología de desarrollo completamente nueva y, a día de hoy, no excesivamente documentada.

Pero a la hora de decidir sobre la tecnología a utilizar para desarrollar una aplicación destinada a ser ejecutada en un dispositivo móvil, conviene preguntarse, entre otras, al menos un par de las siguientes preguntas:

¿A quién va dirigida la aplicación?. ¿Es una aplicación para todo tipo de usuarios, o es una aplicación empresarial?. ¿Los usuarios van a tener un entrenamiento específico para manejarla?

¿Se está desarrollando un juego o una aplicación?. La diferencia es muy importante, sobre todo si es un juego, más aún si se toma en consideración también la pregunta anterior. Dependiendo de el tipo de usuario al que está dirigido se puede suponer el tipo de dispositivo para el que se deberá hacer el desarrollo.

¿Hay algún requerimiento previo?. Tal vez sea una pregunta obvia, pero si el cliente sólo quiere dar soporte a dispositivos Symbian...

¿Qué tal tus conocimientos técnicos? A todos nos gusta aprender nuevas tecnologías, pero a la hora de sacar adelante un proyecto es mejor atenerse a lo ya conocido y a aquello en lo que se es eficaz y productivo.

¿Qué presupuesto tienes? Al final, ése es el factor determinante, que terminará de resolver cualquier duda que tengas.

### Consideraciones sobre el desarrollo. Limitaciones.

Hay dos limitaciones fundamentales a que tener en cuenta a la hora de desarrollar un aplicación para ser ejecutada en un dispositivo móvil, y son las particularidades de hardware y las de la conexión.

Aunque los dispositivos cada vez tienen más memoria y más capacidad de proceso, siguen siendo hermanos pequeños de los ordenadores de escritorio. Por tanto, debe tenerse especial cuidado con no realizar demasiadas animaciones, ni gastar ciclos de proceso en procedimientos que no sean estrictamente necesarios. La pantalla es pequeña, mal iluminada, y se suele mirar en movimiento, como ya hemos visto con anterioridad, por lo que podemos aprovechar eso en nuestro beneficio.

Por ello la entrada de datos debe ser lo más sencilla posible. En el caso de las PDAs, se realizará escribiendo en la pantalla con un lápiz, para que sea el dispositivo el que realice el reconocimiento de la escritura. Aunque ya no hace falta escribir de forma especial, como ocurría con el sistema de reconocimiento de escritura de las primeras Pilot, llamado Graffiti, para que el dispositivo entienda la letra, tampoco se puede ser demasiado descuidado con la caligrafía.

En los móviles, sin embargo, la entrada de datos se va a hacer, lo más probablemente, con una sola mano, y en condiciones no muy apropiadas. El teclado, en la mayoría de los casos, tiene sólo 10 teclas útiles, que se van a utilizar en movimiento.

En cualquiera de los dos casos, hay que intentar, por tanto, formatear las pantallas de entrada de datos para que se pueda realizar el mayor número de operaciones o bien punteando encima de botones, en el caso de las PDAs, o bien navegando con el joystick y confirmando con alguno de los botones principales en el caso de los teléfonos; evitando en lo posible que el usuario tenga que dirigirse al teclado. Siempre es preferible que haya, por ejemplo, un desplegable en el que elegir entre "Calle", "Vía", "Avenida" y "Paseo", a que haya un campo de texto en el que se tenga que escribir, letra a letra, "Calle".

Además, no hay que olvidar que los interfaces de los distintos dispositivos son desesperantemente heterogéneos. Además, ni el tamaño de las pantallas, ni el tamaño ni la ubicación de las teclas ayuda a la hora de introducir datos por el usuario. Por eso, siempre hay que intentar desarrollar los interfaces de las aplicaciones de forma que no sean necesarias muchas pulsaciones de teclas, e intentar que, cuando sea necesario pulsar alguna tecla sea alguna de las softkeys, o teclas especiales. En el caso de los dispositivos móviles, más que en cualquier otro, menos es más.



No sólo menos es más, sino que la colocación de los elementos de ese menos son los que van a marcar la diferencia. En realidad, no es sólo una cuestión de colocación en pantalla, sino de organización de la información a presentar, de lo que se llama arquitectura de la información.

La premisa es bastante sencilla: dadas las dificultades para la introducción de datos, hay que intentar simplificarla al máximo. Para ello, es muy importante organizar el interfaz de la aplicación de forma que los procesos de entrada de datos sean lo más naturales posibles, y que el usuario nunca pase por pantallas cuya presentación no sea estrictamente necesaria. Pero en realidad esas precauciones no hay que tomarlas sólo a la hora de introducir datos, sino también a la hora de presentar información, y a la hora de estructurar la navegación por la aplicación.

Normalmente es mucho más efectivo presentar la información agrupada de forma jerárquica, para de ese modo discriminar cuanto antes qué es lo que va a necesitar ver el usuario, y no mostrarle nada que no le resulte estrictamente necesario.

En cuanto a las conexiones, lo primero que hay que tener siempre en cuenta es que, al contrario que en las aplicaciones web estándar, no están siempre disponibles. Por tanto, nunca se puede suponer que el dato que es necesario para poder seguir adelante en la ejecución de la aplicación, va a poderse obtener a través de una conexión a Internet.

Además, debe tenerse en cuenta que antes de realizar una conexión a cualquier servicio tarificable, el dispositivo va a solicitar confirmación al usuario, y que éste puede denegarla, lo cual es perfectamente posible, dado el precio de las conexiones, que son facturables por peso, por cantidad de bytes enviados y descargados, no por tiempo, por lo que siempre hay que optimizar todo lo posible lo que se vaya a enviar en uno u otro sentido.

Por si no fueran dificultades suficientes, la latencia de la red, es decir, el tiempo de espera entre una petición de datos y el comienzo de la llegada de la respuesta, es mucho mayor que la de cualquier aplicación web normal. Por tanto, hay que ser muy cuidadosos a la hora de informar al usuario sobre lo que está ocurriendo en todo momento, si se está en espera de datos, si ya se están recibiendo, dando siempre, si es posible, estimaciones sobre el tiempo que resta para la terminación de todas las tareas.

Tampoco se puede suponer que la ejecución de la aplicación sea lineal y completa. La función básica de muchos dispositivos es emitir y recibir llamadas telefónicas. Por tanto, nuestra aplicación debe ser capaz de comportarse adecuadamente al recibir una llamada, entre otras cosas porque la función de teléfono tiene asignada la prioridad más alta de todas las del dispositivo, es decir, se va a parar la ejecución de cualquier aplicación cuando entre una llamada nueva. Se debe tener prevista una forma de congelar el estado de la aplicación justo antes de recibir la llamada, para que cuando ésta termine, y tras la confirmación del usuario, se pueda volver a ese estado. Por tanto, si es un juego, todos los elementos del mismo deben permanecer estáticos hasta que la llamada termine y el usuario confirme que desea continuar con la aplicación.

En general, hay que intentar probar las aplicaciones en el dispositivo para el que se vayan a desarrollar cuanto antes. De hecho, hay que probar mucho, y comenzar a probar lo antes posible. Nunca hay que fiarse de emuladores, entre otras cosas, porque se ejecutan en hardware cuya capacidad de procesamiento es superior a la del dispositivo en varios órdenes de magnitud.

Por tanto, al desarrollar para dispositivos móviles, es muy importante comenzar a probar pronto, y probar a menudo, a fin de evitar sorpresas desagradables.

#### Para los más curiosos

Artículo algo antiguo pero aún vigente:

[http://www.alzado.org/articulo.php?id\\_art=445](http://www.alzado.org/articulo.php?id_art=445)



iPhone evaluation report

<http://www.uigarden.net/english/iphone-evaluation-report>

7 usability guidelines for websites on mobile devices

<http://www.webcredible.co.uk/user-friendly-resources/web-usability/mobile-guidelines.shtml>

## Usos actuales de los dispositivos y perspectivas de futuro

### Usos actuales

Durante todo este documento hemos estado hablando de las plataformas, los lenguajes de programación, los sistemas operativos, pero en realidad, ¿para qué sirven? ¿Qué se puede hacer con un móvil? ¿Qué se está haciendo actualmente?.

Actualmente, el desarrollo para dispositivos móviles se puede dividir en dos categorías. Por un lado, el desarrollo web puro y duro (webs hechas para ser navegadas desde móviles), y por otro, la programación de aplicaciones para ser ejecutadas en el propio móvil.

No hay que olvidar que, actualmente, casi cualquier dispositivo implementa algún protocolo de comunicación, sea éste WI-FI o Bluetooth, por lo que es sencillo que, si no es capaz de conectar a Internet por sí mismo, como haría un teléfono, pueda hacerlo emparejándose con algún otro dispositivo que sí pueda. Dicho de otra forma, con un teléfono se puede uno conectar a Internet. Con un PocketPC o una Palm es probable que no, pero se puede hacer sin muchos problemas emparejándolo con un teléfono.

Actualmente hay varios portales de servicios basados en web que proporcionan interfaces específicos para ser navegados desde dispositivos móviles. Compra de entradas para espectáculos, consulta de callejeros, de mapas de carreteras, obtención de información sobre restaurantes o farmacias de la zona en la que se encuentra el usuario, son algunos de los servicios más extendidos.

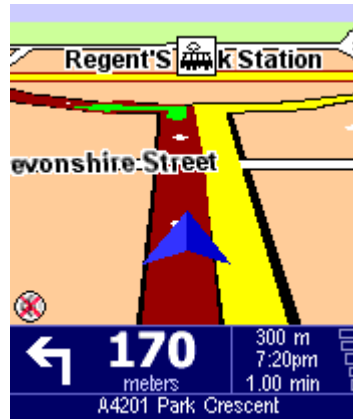


**Metro, un programa multiplataforma con planos de Metro de varias ciudades de todo el mundo**

El apartado de las aplicaciones ejecutables en el teléfono está sin duda dominado por los microjuegos. Las limitaciones gráficas y sonoras, sobre todo de los teléfonos, no sólo no son un problema sino que ayudan a proporcionar a los juegos cierto aire "retro", de máquina recreativa de los 80, que les hace más atractivos. Su bajo peso, además, les hace ideales para ser descargados directamente al teléfono mediante envío de sms a números especiales, por

ejemplo, por lo que su sistema de distribución tiene todas las ventajas de las distribuciones virales.

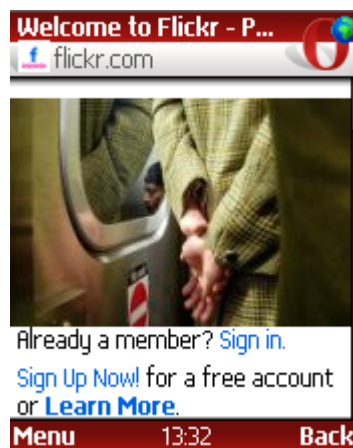
Además de los minijuegos, otro campo en el que están surgiendo muchas oportunidades para los desarrolladores es en la construcción de aplicaciones basadas en servicios de localización, utilizando datos de posicionamiento de unidades GPS, y combinando esos datos con los obtenidos sobre el entorno del usuario desde un servicio web.



Tom Tom Mobile en un Symbian Serie 60

También se está trabajando bastante en aplicaciones que puedan servir como apoyo a personas con minusvalía, tanto visual como auditiva. Sistemas de localización, de reconocimiento óptico, sistemas de traducción de y a Braille portátiles...

Por otro lado, los teléfonos son pequeños terminales con completas capacidades multimedia, e integran cámaras fotográficas que son capaces de grabar vídeo y reproductores multimedia que no tienen nada que envidiar a los desarrollados para los ordenadores de escritorio. De esas cámaras integradas han salido algunas de las fotografías o vídeos más famosos en los últimos meses, como los de los atentados de Londres, o los de algunos niños pegando a sus compañeros de clase.



Acceso a flickr.com desde Opera Mobile

Sin embargo, pese al amplio abanico de funcionalidades de los dispositivos, pese a su enorme índice de penetración en el mercado, pese a las cantidades de dinero que se alrededor de las descargas de contenidos multimedia o del envío de sms, aún no se ha producido el despegue definitivo del desarrollo de aplicaciones. Hablamos mucho de lo cercano que está el despegue definitivo de las aplicaciones móviles, de lo que se ve a poder hacer con los terminales de

última generación y las conexiones de banda ancha, pero lo cierto es que ese discurso se lleva escuchando, casi sin alteraciones, desde hace ya varios años.

Lo cierto es que el mercado mueve millones anualmente en descargas de contenidos multimedia, como tonos o fondos de pantalla. Es muy sencillo descargarse un tono polifónico enviando un sms, y al realizarse el pago no en dinero contante y sonante, sino descontando el importe del saldo o de la próxima factura, tampoco se tiene sensación de estar realizando un gasto real. Sin embargo, el mismo tono puede editarse en un ordenador de sobremesa, convertirlo en un formato compatible con el teléfono, y enviarlo a éste por bluetooth, por ejemplo, con lo que el coste final sería cero.

El porqué se gasta tanto dinero en servicios que puede realizar uno mismo con un poco de trabajo a los mandos de su ordenador, mientras otros, que han costado millones de euros en desarrollo, como por ejemplo la videoconferencia por 3G, prácticamente no se utilizan, es uno de los misterios más grandes de este nuevo siglo, al menos para los que han desarrollado esos servicios.

Sin embargo, estudiando con detenimiento las pautas de comportamiento del usuario medio, entendiendo como tal el que no tiene porqué saber lo que quiere decir Symbian, parece evidente que las últimas novedades tecnológicas no son más que soluciones en busca de un problema. Lo que parece demandar el mercado son servicios y aplicaciones que le simplifiquen la vida, que no requieran de mucha intervención por su parte para funcionar correctamente.

Merece consideración aparte el caso del iPhone, por múltiples razones. La existencia de un mercado cerrado, controlado por el fabricante y la necesidad de pasar por un proceso de certificación (de pago), no han impedido que ha día de hoy, haya más de 15.000 aplicaciones disponibles.

Sin embargo, hay varios riesgos asociados a un crecimiento tan rápido de una plataforma. El riesgo de implosión es muy alto, y dada la gran rapidez con la que aparecen aplicaciones nuevas, es muy difícil que un desarrollador se plantee trabajar en una aplicación por un tiempo considerable, lo que hace que el porcentaje de aplicaciones que realmente pueden considerarse como "killer" es mínimo.

## Perspectivas de futuro

---

Es muy difícil predecir el futuro. Más aún cuando estamos intentando predecir el futuro de un campo en el que la innovación es la máxima, lo que todos los implicados buscan, ya sean fabricantes de terminales, desarrolladores de aplicaciones, u operadoras y proveedores de servicios.

Pero sí que tenemos unos cuantos datos con los que atrevernos a aventurar una previsión para el futuro cercano.

En primer lugar, el hardware va a seguir evolucionando. Según la Ley de Moore, cada 18 meses la potencia de los ordenadores se duplica. Esta ley es igual de aplicable para un PC de escritorio, para el superordenador Mare Nostrum, para un PocketPC o para un móvil. Como decía Don Hilarión, la ciencia adelanta que es una barbaridad.

Por tanto, podemos esperar un avance considerable en la capacidad de proceso de los dispositivos en un plazo relativamente corto, un avance que además va a ser constante, se va a seguir produciendo a lo largo del tiempo.

Y a más capacidad de proceso, mayor calidad de los contenidos. Así pues, lo primero que podemos esperar es que los juegos sean más complejos, con mejores gráficos y mejor sonido, por ejemplo, o aplicaciones complejas con integración de gráficos 3D (aplicaciones sobre medicina o astronomía, por ejemplo).

Pero mejor hardware no sólo trae consigo aplicaciones de más calidad, sino mejores posibilidades de comunicación e interacción entre dispositivos y por lo tanto entre los dueños de los mismos. Bluetooth de alta velocidad, Wi-Fi, voz ip, comunicaciones por push... son varias las tecnologías que van a servir para mejorar la comunicación y el intercambio de datos entre dispositivos.

Nadie quiere quedarse sin al menos una parte del pastel del futuro. Mientras algunos están desarrollando pantallas interactivas que se puedan enrollar para transportar, otros están trabajando en interfaces de usuario que parten de paradigmas totalmente distintos a los que estamos acostumbrados a utilizar y otros están trabajando en el dispositivo que los una a todos.

Esa última rama de investigación, la ya famosa convergencia, puede ser una de las que más frutos dé en un futuro cercano. La idea detrás de esa línea de desarrollo es el intentar buscar un dispositivo que pueda realizar las funciones de todos los que habitualmente solemos utilizar: reproductor mp3, teléfono, agenda electrónica, incluso navegador GPS. Todo en un único aparato, de dimensiones reducidas, pero con una pantalla lo suficientemente grande como para ser funcional. El penúltimo intento fue el Origami de Microsoft, pero han sido varios los que lo han intentado previamente, sin mucho éxito: BenQ, Sony-Ericsson, Apple... Sin embargo, sí parece que los netbooks, especialmente en Asia, donde es raro ver otra cosa, están cuajando en el mercado.

Si bien parece claro que los dispositivos, en un futuro cercano, van a tender a converger a un único super-dispositivo, ¿qué se puede esperar de cómo serán por tanto las aplicaciones del futuro? Siempre es complicado adivinar lo que va a ocurrir, pero puede esperarse un aumento de las aplicaciones con cierto componente social, al igual que está ocurriendo en la web, que permitan interactuar a los usuarios, jugar en grupo, compartir archivos multimedia con facilidad.

También es de suponer que cada vez haya más aplicaciones especializadas en la gestión y reproducción de archivos multimedia, sobre todo música y vídeo, aplicaciones integradas con servicios de descarga a través del PC de escritorio.

En todo caso, lo que único que parece seguro es que cada vez va a haber más dispositivos móviles circulando por las calles en manos de sus dueños, dispositivos con mayores capacidades de proceso y multimedia. Una oportunidad que sería una pena dejar pasar.

## Bibliografía

De entre los múltiples libros y sitios web dedicados al mundo de la tecnología móvil, se recomienda la lectura o visita de los siguientes.

### Libros:

---

- WEISS, S.W. Handheld Usability. John Wiley and Sons Ltd, 2002
- JONES, MATT y MARSDEN, GARY. Mobile Interaction Design. John Wiley and Sons Ltd, 2005

### Sitios web

---

- [www.evolucy.com](http://www.evolucy.com)
- [www.cadius.com](http://www.cadius.com)
- [www.alzado.org](http://www.alzado.org)
- [www.macromedia.com/es/devnet/mobile](http://www.macromedia.com/es/devnet/mobile)
- [www.microsoft.com/windowsmobile](http://www.microsoft.com/windowsmobile)
- [www.palm.com/es](http://www.palm.com/es)
- [www.forum.nokia.com](http://www.forum.nokia.com)
- [www.oldschool.net/newton/](http://www.oldschool.net/newton/)
- [java.sun.com/j2me](http://java.sun.com/j2me)
- [developer.apple.com/iphone](http://developer.apple.com/iphone)
- [code.google.com/android](http://code.google.com/android)