

Cuadro comparativo de tecnologías para juegos en dispositivos portátiles. Versión 1.1

Investigación y Programación Ltda.

Documento liberado al público, bajo Licencia Creative Commons Reconocimiento - Compartir Igual
<http://creativecommons.org/licenses/by-sa/2.0/deed.es>

Se exponen aquí algunos elementos para tomar decisiones sobre los primeros pasos en programación de dispositivos móviles. Como resultado de una primerísima fase de investigación, se muestra una comparación de varias alternativas para implementar juegos y aplicaciones similares en un dispositivo móvil. Las referencias a cuestiones técnicas apuntan a construir criterios de costos y capacidad de soporte posventa que a mediano y largo plazo pueda tener la elección de uno u otro camino; y las decisiones y sugerencias que se encuentran en este documento usaron también las estadísticas citadas al final en la bibliografía.

1. Utilidad y contexto.....	1
2. Criterios de comparación acerca de la máquina.....	2
3. Aplicación Web o Aplicación de Escritorio: comparación de opciones dentro del modelo cliente-servidor.....	3
3.1. Análisis.....	3
3.2. En conclusión.....	5
4. Lenguajes de programación, algunas alternativas.....	6
5. Resumen de expectativas.....	6
6. Conclusión.....	7
7. Algunas lecturas adicionales sugeridas.....	7

1 Utilidad y contexto

- Al tomar decisiones para realizar un desarrollo, se recomienda salir de los aspectos técnicos de programación inicialmente, y evaluar la penetración que tiene un modelo de celulares en el mercado, y la penetración que tiene el sistema operativo para móviles de los celulares más populares. No es sólo un aspecto, sino ambos: bien podría ocurrir que el celular más comprado en un país o región, tenga el sistema operativo más incompatible, mientras que los celulares que le siguen en nivel de popularidad, comparten el mismo sistema operativo, compensando el hecho de no estar en primer lugar en popularidad. Nótese, por ejemplo, que al momento de escribir este documento, el sistema operativo Symbian tiene una penetración de alrededor de la mitad, dentro de la gama de competidores de sistemas operativos similares para dispositivos móviles. Véanse las siguientes gráficas:

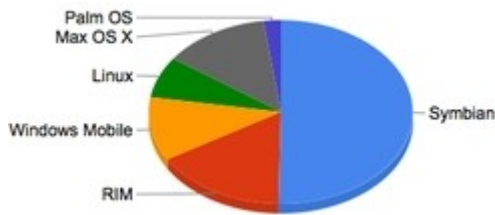


Fig. 1. Tomado del artículo citado en [7], el cual a su vez es construido con base en las estadísticas de Gartner [9].

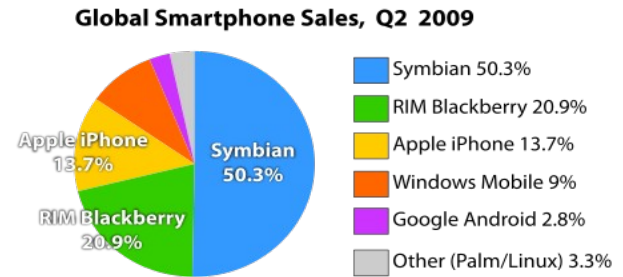


Fig. 2. Tomado de [4], artículo Smartphone, con base en las estadísticas de Canalys

2 Criterios de comparación acerca de la máquina.

Tabla comparativa. Nokia E71, iPhone, HTC P3600. Para incluir Palm u otro fabricante, es necesario conocer el modelo específico a comparar.

Característica	Nokia E71	iPhone	HTC P3600
Memoria de almacenamiento	110 MB, aunque se le pueden conectar tarjetas adicionales de máximo 8 Gb	De 4 a 64 Gb, depende del modelo	Se le puede conectar una tarjeta MiniSD con capacidad de 4Gb
Memoria RAM (MB)	128	128 a 256	64
Sistema operativo	Symbian OS, 9.2	iPhone OS	W. Mobile 5.0
Compatibilidad para	-C++, Java, OPL, AS3.	Objective -C	

<i>programación Posibles lenguajes de programación o entornos de desarrollo</i>	Los archivos se transfieren al celular a través de Nokia PC Suite, que está disponible para Windows. El intercambio de información entre PC y dispositivo se hace a través de un cable de datos, o de una conexión por Bluetooth, o a través de un puerto infrarrojo.	Debe ser programado en Apple-Macintosh.	.NET y compatibles con W. Mobile.
<i>Tamaño de la pantalla (pixeles X pixeles)</i>	320 x 240	320 X 480	240X320
<i>GPS</i>	Sí	Sí, en el modelo 3G	Sí
<i>Método de ingreso de datos por parte del usuario / operario.</i>	A través de teclado QWERT. Hay botones con funciones especiales.	Pantalla táctil. Para letras, hay un teclado virtual.	Touch, y unos pocos botones

En [4] (artículo Mobile Development) y en [5], hay comparativos adicionales para un primer panorama acerca de todas las alternativas disponibles. En [7], [8], [9], [10] hay estadísticas acerca del posicionamiento de diferentes alternativas para smartphones y dispositivos móviles en el mercado.

3 Aplicación Web o Aplicación residente: comparación de opciones dentro del modelo cliente-servidor.

3.1 Análisis

La siguiente comparación es válida exclusivamente dentro del contexto descrito en este documento.

	Aplicación WEB <i>Arquitectura: modelo de cliente ligero o cliente liviano</i>	Aplicación local <i>Arquitectura: modelo de cliente pesado o "aplicación residente"</i>
Conexión a Internet	Permanente durante la sesión de ingreso de datos, pues requiere que un servidor recoja los datos en tiempo real. Si se interrumpe la conexión, se interrumpe la	Puede interrumpirse la conexión, y almacenar los datos hasta que vuelva a tener conexión. La ausencia de conexión a Internet no arruina el desenvolvimiento de la aplicación.

	recogida de datos.	
Resolución de pantalla	Ambos casos deben ajustarse para la resolución del dispositivo móvil.	Ambos casos deben ajustarse para la resolución del dispositivo móvil.
Capacidad de almacenamiento	Determinada por el servidor. Localmente es nula, porque en esta opción la aplicación web no almacena datos localmente.	El Nokia E71 viene de fábrica con una memoria de 100Mb. Ahora bien, para un tamaño similar a los archivos de datos de texto actuales generados por algunos juegos, por cada 1Mb se pueden almacenar alrededor de 1000 sesiones distintas de recogida de datos, de solo texto, sin fotos. Si la aplicación requiere almacenar imágenes o fotos, en 1 Mb caben mínimo 20 fotos en una resolución que todavía permite distinguir rostros humanos.
Lenguajes de programación en lado de cliente.	<ol style="list-style-type: none"> 1. WAP, alta compatibilidad y estandarización. 2. HTML+CSS, pero en los dispositivos móviles esto cambia notablemente. 3. AS3. Depende de que el dispositivo móvil tenga Flash Player en la versión correcta. En esta alternativa, las pruebas deben incluir la exploración de tecnologías para aplicaciones RIA en dispositivos móviles. 	Para el caso del Nokia E71, además de AS3 (ver aplicación web), aquí es posible usar C++, Java, OPL, y un largo etcétera.
Requerimientos de procesador y memoria	El modelo de cliente liviano requiere menos recursos por parte de la máquina.	Requiere más memoria y capacidad de procesamiento por parte de la máquina.
Escalabilidad	<p>-Todas las validaciones deben hacerse en el servidor, lo cual implica mayor consumo de recursos. El servidor estará más cerca de saturarse con este modelo.</p> <p>-Necesita una primera conexión para cargar la pantalla de la interfaz de ingreso de datos. Luego de eso, una sesión rica en posibilidades para interactuar con el formulario, implica sostener conexiones abiertas al servidor durante los varios minutos que tarde cada sesión de ingreso de</p>	<p>-Se hacen validaciones en el lado del cliente (el dispositivo móvil), lo que reduce la carga para el servidor.</p> <p>-Las conexiones que se hacen con el servidor están destinadas a transferencia de datos exclusivamente, pero no a sostener ociosamente una conexión abierta con el servidor mientras el operario ingresa los datos.</p>

	<p>datos.</p> <p>-La funcionalidad de estas aplicaciones será más limitada en cuanto al uso de todas las posibles funciones del dispositivo.</p>	<p>-Permite aprovechar mejor las posibilidades de los dispositivos.</p>
<p>Costos, para un mismo dispositivo.</p>	<p>-Bajo costo de actualizaciones para la aplicación cliente: Se hace una sola vez en el servidor, puesto que en cada sesión el cliente carga de nuevo la aplicación.</p> <p>-Alto costo en transferencia de datos. En el modelo de que la aplicación pueda cargar con frecuencia, la interfaz de ingreso de datos necesitará cargarse con frecuencia. Si la señal a Internet es a través de teléfono, posiblemente se cobrará por número de kilobytes transmitidos, o por tiempo de conexión, y en cualquier caso resulta tener un costo alto.</p> <p>-No exige gran capacidad de procesamiento por parte de la máquina.</p> <p>-Exige conectividad permanente a Internet. Costos adicionales de logística por eventual ausencia de conexión a Internet (ejemplo: la necesidad de transcribir datos tomados manualmente).</p>	<p>Alto costo en actualizaciones para la aplicación cliente: Cada dispositivo debe actualizarse al realizar un cambio en la aplicación cliente.</p> <p>-Bajo costo en transferencia de datos: No exige conectividad permanente a Internet. Incluso, si la actualización de datos puede hacerse cada 4 a 8 horas hábiles, no es necesario que el dispositivo móvil tenga conexión a Internet: los datos pueden enviarse a un PC, que se encarga de enviarlos al servidor.</p> <p>-Exige mayores condiciones de procesamiento.</p> <p>-No exige conectividad permanente a Internet. Ello reduce costos adicionales.</p>
<p>Compatibilidad</p>	<p>Dependerá de la resolución de pantalla del dispositivo, y de la capacidad del navegador para interpretar correctamente las tecnologías de lado de cliente de una aplicación web, como html, css, javascript y AS3.</p>	<p>En general, deberá diseñarse para cada conjunto de dispositivos, siendo éste uno de los principales contras. No obstante, hay lenguajes y tecnologías comunes en varias plataformas diferentes para aplicaciones residentes.</p>

3.2 En conclusión

El modelo de un cliente pesado en una aplicación residente para el dispositivo móvil, implica mayores costos iniciales en tiempo (y por lo tanto en dinero) que el modelo de cliente ligero a través de una aplicación web. Para cada aplicación, parte del estudio es si este costo adicional compensa drásticamente las falencias de la alternativa del cliente liviano.

La principal desventaja de la aplicación web, es que está basado en una conexión a Internet permanente y estable. Y al pensar en ello, debe tenerse en cuenta que:

1. En algunos recintos internos se apantalla el ingreso de señales electromagnéticas, y por lo tanto no se recibe una buena señal de Internet.
2. Si el servidor se cae, toda la logística de recogida de datos puede colapsar, porque depende del servidor permanentemente para recoger los datos de cada sesión.
3. A veces las redes de Internet fallan a nivel internacional, y eso de nuevo implicaría el colapso del sistema.
4. La escalabilidad del número de dispositivos será menor, debido a que cada dispositivo implica un mayor consumo de recursos del servidor, tanto en conexiones simultáneas del servidor, como en procesamiento por parte del mismo.

De otro lado, la principal ventaja de la aplicación web es su rápida actualización.

En resumidas cuentas, la decisión implica otorgar un peso a cuáles factores son más importantes para la compañía. Listemos algunos de ellos, independientemente de que sean o no sean deseables:

1. Aspectos pos-venta y actualización: la aplicación ligera permite una mayor flexibilidad en general para cambios y actualizaciones, y reduce costos de actualizaciones. Sin embargo, encuentra una limitación en cuanto al acceso a funciones del dispositivo. Es decir, una aplicación de este tipo no podría acceder, por ejemplo, a leer un GPS.
2. Crecimiento de activos y documentación para el cliente . En el corto plazo, es una mejor opción la aplicación ligera, por la portabilidad del código. Pero el código fuente y documentación de los desarrollos en el modelo de cliente pesado representan un mayor activo intangible por parte del cliente.
3. Soporte técnico posterior del sistema. Es necesario en ambos casos, pues la aplicación depende del servidor, que es administrado por la parte que se acuerde. La aplicación de modelo pesado viene compilada, pero en el modelo de cliente ligero es posible acceder al código fuente en la aplicación cliente, ya que los lenguajes son interpretados en el lado del cliente.

4 Lenguajes de programación, algunas alternativas

Para una aplicación residente, existen varios lenguajes que se pueden usar en un comienzo:

- **C++**, de rápida ejecución y brinda excelentes posibilidades de comprensión. La objeción es el grado de universalidad de los desarrollos realizados aquí, y la dificultad actual de encontrar programadores en este lenguaje que puedan soportar los desarrollos que se realicen aquí para el cliente.

- **.NET (C#, VB.NET)**. Es una excelente alternativa, para la plataforma Windows Mobile. Sin embargo, tal parece que esta plataforma no es la más común en el mercado, y por lo tanto Windows Mobile parece no ser el mejor criterio de elección, de acuerdo a las estadísticas de [7], [8], [9], [10].
- **Java**, de gran -pero no perfecta- compatibilidad. Las posibles imperfecciones como total acceso a las funciones del dispositivo, parecen ser compensadas por el hecho de que el número de cambios a realizar en los programas al migrar de dispositivo pueden ser menores que en otros lenguajes.
- **OPL**. Lenguaje interpretado similar al basic, destinado sólo a Symbian, y que parece tener una reducida comunidad que lo soporte.
- **AS3+Flash**: Si bien hay muchas opciones para crear aplicaciones web con riqueza funcional, ésta parece ser la más universal actualmente. Presenta además la ventaja de que con el paso del tiempo puede participar la sección de Diseño Gráfico al usar esta tecnología.

5 Resumen de expectativas.

Pruebas de ejecución de programas creados por su empresa, en un dispositivo móvil específico.

Deben plantearse unas expectativas para el primer proyecto que usted programe para celulares. El desarrollo involucra realizar pruebas para creación de programas en uno o varios lenguajes de programación, con el fin de ser ejecutados en el equipo que usted seleccione. Estas pruebas buscan sondear el terreno tanto para la disponibilidad de funciones que se necesitan para lograr la aplicación buscada, como para reconocer posibles obstáculos y establecer un plan de trabajo con mayor precisión.

6 Conclusión

Una aplicación residente exige costos mayores pero tiene una mayor estabilidad operativa por todas las eventualidades que pueden ocurrir de desconexión a la red, y además permite una mayor escalabilidad al proyectar el sistema a varias decenas o centenas de dispositivos.

El Nokia E71 parece ser un arma de doble filo: por una parte tiene un número abrumador de opciones para desarrollar negocios a su alrededor, pero por otra parte, tiene un costo que no necesariamente se pueda cubrir por todos los clientes al hablar en términos masivos. Comercialmente, cada proyecto necesitará un conjunto muy específico de funciones, y difícilmente ello se necesitará usar todas las capacidades de un dispositivo como el Nokia E71. No obstante, la plataforma que usa es Symbian, que como puede verse en las estadísticas de [7]-[10], es la más común del mercado, y ello sugiere que como elemento de desarrollo y pruebas para lograr primeros negocios es una excelente opción. Una estrategia para cubrir costos de migración a mediano plazo, es usar una estrategia de programación que apunte a ser universal y altamente compatible.

Con los recursos actuales, sin inversiones ulteriores, es posible adelantar una buena investigación para tener una prueba de concepto del sistema actual. Dicha investigación arranca con un estudio preliminar de pruebas durante 10 a 15 días hábiles para determinar un cronograma puntual, con el

fin de generar pruebas con lenguajes diseñados para aplicaciones de ejecución en dispositivos móviles. No obstante se sugiere que al terminar estas pruebas con lenguajes de aplicaciones residentes se agregue la posibilidad de un nuevo lenguaje, a saber, AS3, que permite estudiar la posible participación de un trabajo de diseño gráfico, con un tiempo de estudio adicional en programación de 8 días hábiles.

Aspectos faltantes: Para acelerar esta primera fase de estudio, no se incluyeron aún decisiones en torno a la tecnología GPS. Para futuras comparaciones con más modelos de dispositivos, será necesario conocer el modelo específico con el que cuenta el cliente, y evaluar en ese caso, con los parámetros dados aquí, si es conveniente para el cliente realizar el desarrollo para ese modelo en específico.

7 Algunas lecturas adicionales sugeridas

Las referencias en Internet fueron accedidas en oct-nov de 2009.

- [1]. *Nokia E71, Full Phone Specifications*. Archivo en línea. Disponible en la url http://www.gsmarena.com/nokia_e71-2425.php
- [2]. Links a diferentes posibles vendedores del teléfono, para comparar costos y formas de pago: <http://www.gsmarena.com/compare.php3?idPhone1=2425>
- [3]. *Handling of temporally lack of network*. Allí se encuentra una breve comparación sobre la comunicación síncrona y asíncrona a la hora de diseñar aplicaciones, y algunas ideas adicionales que contribuyen a conceptos para este documento. Está disponible en <http://fragments.turtlemeat.com/handheld-mobile-programming/handling-lack-of-network.php>
- [4]. Artículos de Wikipedia: *Cliente Liviano*, en http://es.wikipedia.org/wiki/Cliente_liviano; *Mobile Development*, en http://en.wikipedia.org/wiki/Mobile_Development; *Smartphone*, artículo de http://en.wikipedia.org/wiki/Smart_phone
- [5]. *Start your engines: Mobile Application Development*. Se muestra aquí una comparación de tecnologías para crear RIAs, dentro del modelo de cliente liviano, y también se comparan tecnologías disponibles para aplicaciones residentes. Disponible en: <http://www.devx.com/SpecialReports/Article/37693/1954>
- [6]. *Mobile Web Best Practices 1.0. Basic Guidelines*. W3C Recommendation 29 July 2008. Esta lectura es útil para desarrollar aplicaciones web en Internet. Disponible: <http://www.w3.org/TR/mobile-bp/>
- [7]. TOFEL, Kevin C. *Smartphones: Symbian sinks under 50% share, Apple races past Windows Mobile*. Artículo publicado el 4 de diciembre de 2008. Disponible en <http://jkontherun.com/2008/12/04/smartphone-mark-2/>
Y ver también el artículo TOFEL, Kevin, *Smartphone Market: RIM jumps past Windows Mobile in 2Q08*, publicado el 12 de septiembre de 2008. <http://jkontherun.com/2008/09/12/smartphone-mark/>
En ambos artículos puede verse que hay un descenso en el porcentaje de Symbian, pero sigue llevando la delantera por varias veces. Se basa en las estadísticas de www.gartner.com
- [8]. DIGNAN, Larry, *Mobile OS wars: Symbian leads globally; MAC OS X surges*. Publicado en el 12 de marzo de 2009. Disponible en http://news.cnet.com/8301-1035_3-10195697-94.html
También se basa en las estadísticas de www.gartner.com
- [9]. *Gartner Says Worldwide Smartphone Sales Reached its Lowest Growth Rate With 11.5 Per cent Increase in Third Quarter of 2008*. <http://www.gartner.com/it/page.jsp?id=827912>
Ver también este artículo más actualizado:
Gartner Says Worldwide Mobile Phone Sales declined 6 Per Cent and Smartphones Grew 27 Per Cent in second quarter of 2009, publicado en agosto 12 de 2009. Disponible en :

<http://www.gartner.com/it/page.jsp?id=1126812>
[10]. Canalys, Symbian: Apple iPhone Already Leads Windows Mobile in the US Market Share, Q3, 2007. Disponible en <http://www.roughlydrafted.com/2007/12/14/canalys-symbian-apple-iphone-already-leads-windows-mobile-in-us-market-share-q3-2007/>

Observación de julio de 2010. Nota de actualización: En el presente documento no se incluyeron modelos como el BlackBerry, que parece haber ganado popularidad en Colombia durante el 2010. En una próxima publicación se informará sobre esta línea.