

# **A methodological proposal for the development of videogames for mobile phones.**

Alfonso López Baca<sup>1</sup>, Javier Martínez Torres<sup>1</sup> y Carmen Pagés Arévalo<sup>1</sup>

<sup>1</sup> Computer Science Department  
Escuela Técnica Superior de Ingeniería Informática  
Universidad de Alcalá de Henares. Spain  
{alfonso.lopez, jmartinez, carmina.pages}@uah.es

**Abstract.** Since SUN MicroSystems introduced the J2ME platform, the Software Market for mobile devices has grown rapidly, especially the videogame market for mobile telephones. This sort of application has some features that make them different from others. Therefore we cannot apply standard current methodologies to them. All this means that none methodology can be applied to the development of this sort of video-games. This article does not aim for suggesting a heavy methodology, not even a light methodology. It only advocates a methodological proposal composed of some recommendations, rather than proposing any methodology that would likely be rejected.

# Una propuesta metodológica para el desarrollo de videojuegos para teléfonos móviles

Alfonso López Baca<sup>1</sup>, Javier Martínez Torres<sup>1</sup> y Carmen Pagés Arévalo<sup>1</sup>

<sup>1</sup> Departamento de Ciencias de la Computación  
Escuela Técnica Superior de Ingeniería Informática  
Universidad de Alcalá de Henares.  
{alfonso.lopez, jmartinez, carmina.pages}@uah.es

**Abstract.** Desde que SUN propone su plataforma J2ME, el mercado del software para dispositivos móviles ha crecido desmesuradamente, y en particular el mercado del videojuego para teléfonos móviles. Este tipo de aplicaciones poseen unas características que las hacen especiales, provocando con ello que no se puedan aplicar las metodologías existentes. Todo esto implica que no se aplique ninguna metodología en el desarrollo de dichos videojuegos. El objetivo de este artículo no es proponer una metodología pesada, ni siquiera una metodología ágil. Simplemente se propone una propuesta metodológica formada por unas recomendaciones, para evitar el rechazo que una metodología sufriría en este campo.

## 1 Introducción

Desde hace unos años la sociedad está dando a los teléfonos móviles un papel mucho más relevante que el tradicional y donde estos dispositivos eran meros terminales con la posibilidad de comunicarse mediante llamadas; sin embargo, el crecimiento del negocio tanto de los juegos para PC como para teléfonos móviles, se ha incrementado muy rápidamente [1]. Nos encontramos en un momento donde los teléfonos móviles proporcionan muchos más servicios que los citados anteriormente. Con el paso de los años, lo que eran terminales primitivos han evolucionado hasta realizar tareas como el acceso a Internet, ejecutar programas de cierta complejidad, intercambiar información con otros dispositivos, y todo esto por medio de tecnologías como los infrarrojos, Bluetooth, GPRS, UMTS, etc.

Sin embargo, existe un claro punto de inflexión en la relación entre la informática y este tipo de dispositivos, data de 1999, cuando SUN anuncia la aparición de una plataforma para teléfonos móviles [2]. A partir de ese momento las compañías comienzan a dotar a los teléfonos móviles de una máquina virtual y un gran número de empresas comienzan a desarrollar software para este tipo de dispositivos. Los primeros programas eran sencillas aplicaciones que proporcionaban servicios como la conversión entre monedas, o calcular la diferencia horaria entre algunas capitales mundiales.

Pero con el tiempo el software de estos dispositivos se ha sofisticado hasta llegar a aplicaciones de intercambio de información con grandes sistemas, o videojuegos multijugador.

Es en este último tipo de aplicaciones, donde se está creando una gran industria, por los grandes beneficios que el desarrollo de videojuegos supone en España. Además, las estadísticas predicen que el mercado seguirá aumentando en los próximos años [3].

Cada vez son más las empresas que se involucran en la industria del videojuego para teléfonos móviles, también se debe considerar que las operadoras están muy interesadas en este negocio, ya que gracias a la descarga de videojuegos se ha disparado el ARPU (Average Revenue Per User) de datos en los últimos años [4], es por ello que las operadoras financian aplicaciones que intercambien información por medio de las redes GPRS, UMTS.

Todos estos factores proporcionan un caldo de cultivo ideal para que el sector crezca de manera desorbitada, y esta condición provoca que los proyectos de este tipo se realicen de manera precipitada y caótica.

Además se han de tener en cuenta las grandes diferencias entre este tipo de aplicaciones y las aplicaciones para PC, nos encontramos con que se necesitan unas fases y técnicas muy diferentes a las de las metodologías actuales. Pensemos en el caso de intentar aplicar metodologías como RUP, Métrica e incluso XP a este tipo de proyectos [5]. Todas ellas carecen de los roles y etapas propias de estos proyectos, esto es debido a las características de este tipo de aplicaciones que las hacen especiales, aunque existen metodologías específicas para juegos como es C3-SD Game Development Process [6]. A continuación se analizan cuales son las características especiales de estas aplicaciones que dan pie a proponer una guía metodológica, para su desarrollo.

## **2 Características de estas aplicaciones**

En este punto se exponen una serie de características propias de estas aplicaciones que nos ayudan a entender el rechazo que sufriría la implantación de una metodología tradicional en este tipo de proyectos.

### **2.1 Bajo número de participantes**

En la mayoría de proyectos informáticos el número de personas implicadas es alto todo lo contrario que ocurre en el desarrollo de videojuegos donde el número de participantes suele ser bajo. Ello provoca que no se apliquen los roles clásicos del desarrollo de aplicaciones. Por ejemplo, si intentásemos repartir los roles que define Métrica 3 [7], nos daremos cuenta, de que no tienen mucho sentido en este tipo de desarrollos, por las diferentes tareas que se han de implementar en el desarrollo de videojuegos. En apartados posteriores se definirán los roles necesarios para este tipo de proyectos y la intervención de cada uno de estos en las fases de desarrollo.

## **2.2 Carencia de estandarización de los dispositivos**

En las aplicaciones tradicionales para PC no se tienen en cuenta las dimensiones de pantalla, el tipo de teclado, etc. Esto se debe a la estandarización de estas máquinas.

Sin embargo en el desarrollo de videojuegos estos aspectos requieren de una atención especial. En los teléfonos móviles existen multitud de combinaciones en cuanto al tamaño de pantalla (101x80 píxeles, 160x128 píxeles, 176x208 píxeles, etc.), existen botoneras con teclas adicionales al teclado tradicional.

Todo esto provoca que se deban de generar un alto número de versiones de un mismo aplicativo, cada una dirigida a un número reducido de terminales. Esto provoca que la gestión de configuración sea compleja y se deban de definir responsabilidades especiales para este tipo de tareas en estos desarrollos. Más adelante veremos las consecuencias de esta característica en la propuesta, pero se ha de considerar este aspecto como uno de los fundamentales en este tipo de aplicaciones.

## **2.3 Diferencias en el software de la KVM**

Los teléfonos móviles que soportan J2ME suelen diferir en cómo realizar funciones especiales como los efectos sonoros y el manejo de la pantalla. Cuando desarrollamos un videojuego solemos utilizar sonidos y acciones sobre el refresco de la pantalla.

La máquina virtual de SUN con el perfil MIDP 1.0 no contempla este tipo de sentencias (sonidos y manejo de pantalla avanzado) y la solución que han adoptado la mayoría de compañías es añadir a sus teléfonos una serie de APIs que el programador puede utilizar para este tipo de funciones. Una consecuencia de esto es que cuando se realiza un juego para un terminal de cierta compañía, si queremos aprovecharlo para terminales de otra compañía (cosa habitual), debemos modificar nuestra aplicación, y utilizar los APIs propios de dicha compañía. Para citar un ejemplo breve que no necesite código para su comprensión podemos citar la clase Canvas de los APIs de SUN y la clase GameCanvas de los APIs de NOKIA.

La consecuencia de esto es que además de las versiones resultantes de considerar las diferencias hardware (dimensiones de pantalla, teclado, etc.), se deben generar versiones por las incompatibilidades entre el software base de cada uno de los dispositivos.

## **2.4 Limitaciones en cuanto a tamaño**

Las limitaciones en el hardware de estos dispositivos provocan que el tamaño de las aplicaciones esté limitado (normalmente a 64 KB). Esto no suele ocurrir cuando desarrollamos aplicaciones para PC. Este es posiblemente el caballo de batalla de los desarrolladores de videojuegos; tanto programadores, como diseñadores y grafistas deben tener muy presente en todo momento estas limitaciones de tamaño para conseguir videojuegos que funcionen en terminales reales.

Las técnicas utilizadas para conseguir aplicaciones de tamaño reducido suelen estar reñidas con los principios ingenieriles y de mantenimiento de aplicaciones. Se puede citar, como ejemplo, que al final del desarrollo del videojuego se suelen co-

mentar todos los bloques de código try/catch (herramientas para el tratamiento de excepciones) que el compilador permita, de este modo se reduce al máximo el tamaño de la aplicación.

## 2.5 Corta duración de estos proyectos

Mientras que cuando se desarrollan aplicaciones para PCs podemos hablar de planificaciones de hasta varios años, en el desarrollo de videojuegos para teléfonos móviles la duración del proyecto debe ser de corta duración debido a la propia evolución del mercado en juegos para móviles.

Esta característica sigue poniendo en evidencia el fracaso que conllevaría la implantación de una metodología en este tipo de proyectos. Sería complicado encontrar una empresa dispuesta a generar todas las etapas que implica seguir una metodología, para un proyecto donde la duración debe ser la menor posible.

## 2.6 Corta vida del producto

Mientras que cuando tratamos con otras aplicaciones para PC, encontramos un ciclo de vida largo, en este tipo de aplicativos debemos tener en cuenta que su periodo de demanda será corto, debido a la obsolescencia del mismo.

Esto no se debe a la tecnología como podría pensarse en un primer momento, sino a que la demanda de la industria del videojuego está muy influenciada por las modas, acontecimientos deportivos, etc. Este caso no se suele dar cuando tenemos que desarrollar una aplicación de gestión de la información, como podría ser una aplicación de nóminas.

## 3 Roles

Uno de los objetivos principales de este artículo es clasificar a los participantes en los proyectos de desarrollo de videojuegos. Con este apartado se consigue definir las responsabilidades, habilidades y perfiles de cada uno de ellos, evitando así que no se superpongan sus procesos ni queden áreas del proyecto sin cubrir.

A continuación se muestran los cuatro roles definidos para este tipo de proyectos. Excepto el jefe de proyecto que será único por cada proyecto, el resto de roles pueden aparecer varias veces en un mismo proyecto (por ejemplo, varios programadores):

**Programador:** su responsabilidad será entregar un juego que funcione correctamente en los emuladores estándar del Wireless Toolkit de SUN, para ello no puede utilizar API's propietarios. El código deberá ser lo más reducido y eficiente posible para que al añadir las sentencias necesarias para reproducir sonidos y efectos animados no superemos el límite del tamaño de la aplicación. En los lugares donde se debe insertar código para reproducir sonido, se debe comentar para facilitar la labor del responsa-

ble de pruebas (siguiente rol). Será un especialista en lenguajes de programación como J2ME o DoJa (Docomo Java).

**Responsable de pruebas:** se encargará de añadir al juego las sentencias necesarias para generar sonidos con los API's propietarios de cada compañía. También se ha de encargar de subir el videojuego a un servidor y realizar pruebas de descarga. Conocerá herramientas de edición de sonido. También se encargará de ofuscar el paquete JAR resultante para reducir su tamaño y evitar que sea descompilado. Se ha de especializar en cada uno de los API's de cada compañía. Si todas estas tareas fuesen realizadas por el programador romperían el ritmo de trabajo de este perfil, que necesita gran concentración para implementar los algoritmos utilizados en la programación de videojuegos, por esta y otras razones se aconseja tener un especialista en pruebas en cada uno de los proyectos.

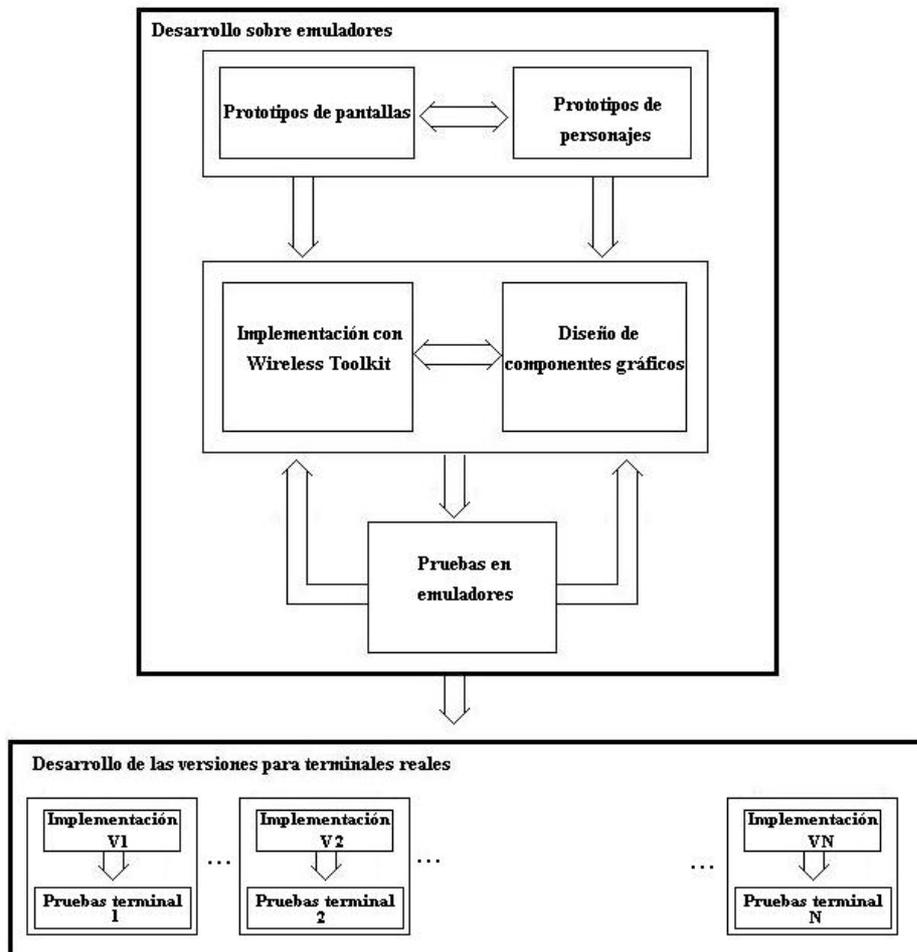
**Diseñador gráfico o grafista:** se encargará de generar todos los ficheros de imágenes del videojuego. Debe ser una persona creativa y experta en herramientas de edición de imágenes. Debe tener en cuenta en todo momento que las imágenes deben estar optimizadas al máximo y no poseer una calidad excesiva, ya que la prioridad debe ser disminuir el peso (tamaño) de las imágenes y no una nitidez excesiva; por muy nítidas que sean las imágenes no se apreciarán los detalles en la pantalla de un teléfono móvil, por tanto no merece la pena un grado alto de detalle, aunque con la evolución de estos dispositivos cada vez las exigencias son mayores.

**Jefe de proyecto:** entre sus responsabilidades se encontrarán las de validar los productos resultantes de las diferentes fases del proyecto, fijar unos objetivos y fechas de entrega realistas y proporcionar al resto del equipo el marco de trabajo adecuado para que el proyecto sea exitoso. Debe tener conocimientos sobre herramientas para la dirección de proyectos.

Con los cuatro perfiles anteriores una empresa se podría enfrentar a cualquier tipo de videojuego. Se ha de recalcar que dentro de un perfil podemos incluir varios recursos, por ejemplo es usual tener dos programadores o dos grafistas dentro del mismo proyecto. También puede ocurrir que se solapen varios perfiles sobre la misma persona, por ejemplo el programador y el grafista, recaen sobre el mismo recurso. Pero esta última opción aumenta considerablemente el tiempo de desarrollo del videojuego.

## 4 Fases en el desarrollo del proyecto

Una vez se han definido los perfiles que intervendrán en el desarrollo del proyecto, se definirá una serie de fases y sus relaciones por las que deberá pasar el proyecto para concluir con éxito. A continuación se muestra un gráfico que muestra las fases y sus transiciones:



**Fig. 1.** Figura que muestra las fases de un proyecto de desarrollo de un videojuego.

Ahora describiremos cada una de las fases del ciclo de desarrollo, pero en un primer vistazo se observa que el proyecto se divide en dos etapas: el desarrollo con el J2ME del Wireless Toolkit (sobre emuladores), y una segunda etapa usando los APIs propietarios de cada una de las compañías para modelos reales.

#### 4.1 Fase de desarrollo sobre emuladores

Dentro de la etapa de desarrollo sobre emuladores, nos encontramos con las siguientes fases:

**Prototipado:** el prototipado consiste en obtener bocetos de las pantallas, y personajes del videojuego. Para ello se pueden utilizar imágenes de videojuegos para PC modificadas, que sirvan como base al posterior trabajo del grafista. Durante esta fase intervendrán los perfiles de grafista y el jefe de proyecto, que validará los prototipos. Como resultado de esta fase se obtendrán una serie de prototipos que guiarán al programador en la colocación de los elementos en pantalla, y al grafista en los productos a obtener en fases posteriores.

**Implementación con el Wireless Toolkit de Sun:** esta etapa se divide en dos a su vez que son: implementación con J2ME de Sun y diseño de la interfaz. Pasemos a describir cada una de ellas.

En la fase de codificación, el perfil protagonista será el programador, que desarrollará el videojuego utilizando únicamente el Wireless Toolkit de Sun. Como objetivos fundamentales a la hora de programar debe de tener: la eficiencia del código (por los motivos expuestos al principio de este artículo), y el tamaño del paquete que deberá ocupar lo mínimo posible.

En paralelo a la fase anterior se diseñará el interfaz gráfico. Como es de esperar será el grafista el encargado de crear las imágenes del videojuego, que el programador incluirá en su código.

El producto final de esta fase será un fichero .jar que contenga las clases desarrolladas por el programador junto con las imágenes, será indispensable que el paquete se ejecute correctamente en un emulador.

Durante esta fase será importante la coordinación del equipo de trabajo, que será llevada a cabo por el jefe de proyecto, así como la validación del producto final.

**Pruebas en emuladores Sun:** durante la fase anterior se realizarán pruebas en emuladores continuamente. Pero esta fase hace referencia a la creación de varias versiones del juego, dependiendo de factores como el tamaño de pantalla de los diferentes emuladores. Los emuladores elegidos dependerán de los terminales a los que vaya dirigido el videojuego, se escogerán los emuladores con el tamaño de pantalla más parecido a los terminales reales, para ello se puede modificar la configuración de los emuladores software para que se ajusten al tamaño de pantalla de los terminales reales.

Al finalizar esta primera fase en la que sólo se ha utilizado el Wireless Toolkit, como base para nuestros programas, se debe de obtener un conjunto de versiones que funcionen correctamente sobre emuladores y con un tamaño lo suficientemente pequeño como para añadir efectos de sonido sin superar el límite del terminal. Tras esta primera fase, comenzará la migración a teléfonos reales, en esta fase se necesitará la ayuda del perfil del responsable de pruebas. A continuación se describe esta última fase del proyecto, que se ha denominado desarrollo para terminales reales.

## 4.2 Fase de desarrollo para terminales reales

En esta última fase de desarrollo se construirán las versiones definitivas para cada modelo de teléfono o pequeños conjuntos de teléfonos. La duración variará en función del número de teléfonos al que vaya dirigido el juego.

El objetivo de esta fase es añadir a las versiones para emuladores las clases necesarias para realizar funciones especiales como por ejemplo los efectos sonoros. Además de comprobar el correcto funcionamiento y descarga de la aplicación sobre el teléfono.

Como se ha dicho anteriormente el perfil protagonista será el responsable de pruebas que debe conocer en profundidad los APIs propietarios de cada compañía para poder de una manera sencilla añadir el código necesario a cada versión.

Al final de esta fase se irán obteniendo las diferentes versiones para cada uno de los terminales reales. El proyecto terminará con la finalización de todas las versiones solicitadas por el jefe de proyecto.

## Conclusiones

A la hora de desarrollar un videojuego para teléfonos móviles, seguir las etapas y asignar los roles detallados en esta propuesta, no garantizan el éxito del mismo. Hay varios aspectos importantes que quedan pendientes, como por ejemplo:

1. Definir técnicas de pruebas: las técnicas de pruebas de este tipo de aplicativos son un tanto especiales. Ello es debido a que el paso del emulador al terminal real no es trivial, ya que suele haber un gran número de errores que ocurren en el terminal pero no en el emulador y que son inherentes al propio terminal y marca del mismo.
2. Documentación: a pesar de la escasa importancia que se le da a la documentación en este tipo de proyectos, en cada una de las fases se han de generar una serie de documentos encaminados a ahorrar tiempo en desarrollos futuros, debiendo ser estos documentos lo más reducidos posible.
3. Control de versiones: como se citó anteriormente, este aspecto es complicado debido al elevado número de versiones que resultan de estos desarrollos. Para gestionar las versiones será necesario utilizar un repositorio y definir procedimientos para manipularlo.

No obstante, la aplicación de una metodología de trabajo conlleva el mejor funcionamiento de todo el equipo, así como, el óptimo cumplimiento de los objetivos marcados a la hora de realizar el videojuego, debido a que en desarrollos de este tipo, el parámetro objetivos/tiempo es muy importante.

Para finalizar debemos resaltar, que el éxito de este tipo de proyectos está supeditado a la experiencia y el conocimiento de los participantes. Pero lo que pretende este artículo es dar una serie de recomendaciones aplicables en proyectos reales, para que

en el futuro se puedan desarrollar las mejores herramientas y técnicas para cada una de las fases del ciclo de desarrollo. Debemos destacar la gran cantidad de aspectos que quedan por tratar en este tipo de proyectos.

## Referencias

1. T. Pelkonen. E-Content Report 3, (2004).
2. Sun Microsystems, Inc. J2ME Technology. <http://developers.sun.com/techttopics/mobility/j2me/>, (1999).
3. El mundo.es. El número de usuarios de videojuegos se duplicará en los próximos cinco años. <http://www.el-mundo.es/navegante/2004/05/12/juegos/1084351254.html> (2004).
4. Finanzas.com. Móviles prevé un alza del beneficio operativo anual de hasta el 17%. <http://www.finanzas.com/id.6094738/noticias/noticia.htm> (2003).
5. Martin Fowler. *The New Methodology*. <http://www.martinfowler.com/articles/newMethodology.html> (2003).
6. Small-device. Mobile Game Development [http://www.small-device.com/downloads/SD\\_Game\\_Service.pps](http://www.small-device.com/downloads/SD_Game_Service.pps) (2004!).
7. Consejo Superior de Informática. Métrica v3: Metodología de Planificación, Desarrollo y Mantenimiento de Sistemas de Información. Technical report, Ministerio de Administraciones Públicas. <http://www.csi.map.es/csi/metrica3/Participantes.pdf>. (2001).
8. M. Weiser: The Computer for the 21<sup>st</sup> Century. Scientific American, vol 265, no. 9, págs 66-75. (1991)
9. R. S. Pressman, Ingeniería del software. Un enfoque práctico, 4<sup>a</sup> ed McGraw Hill (1997).
10. D. A. Norman, *The invisible computer*, The MIT Press, Cambridge Massachusets (1998).
11. M. Weiser, Some computer Science issues in ubiquitous computing, *Communications of the ACM*, Vol. 36, No. 7(1993)
12. Hoyle David, ISO9000: Manual de sistemas de calidad, 4<sup>a</sup> ed. Paraninfo (1998).