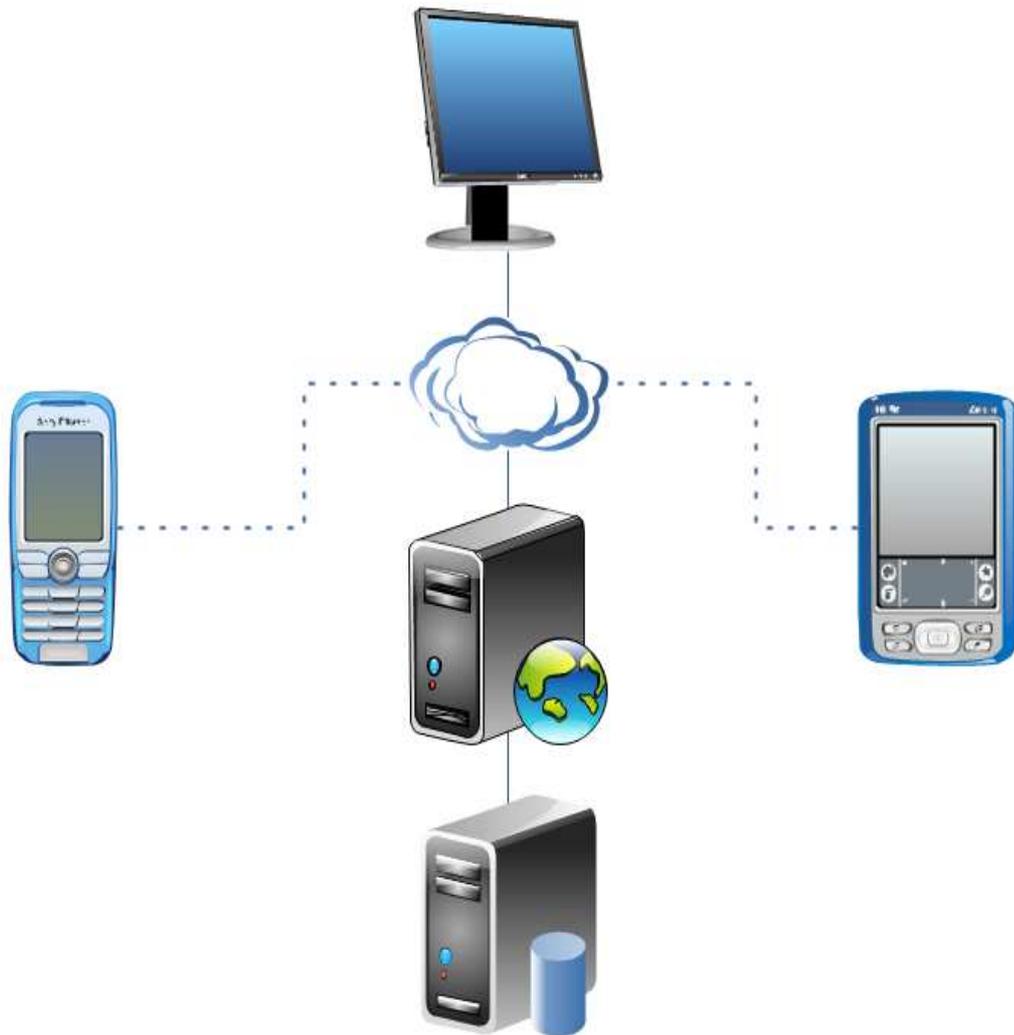


COMPUTACIÓN MÓVIL, Principios y técnicas

# COMPUTACIÓN MÓVIL

Principios y técnicas



Víctor Viera Balanta

Noviembre de 2010

Ing. Víctor Viera Balanta

COMPUTACIÓN MÓVIL, Principios y técnicas

# COMPUTACIÓN MÓVIL

Principios y técnicas

Pre edición para el público

[vieravictor@hotmail.com](mailto:vieravictor@hotmail.com)

Noviembre de 2010

Ing. Víctor Viera Balanta

COMPUTACIÓN MÓVIL, Principios y técnicas

# COMPUTACIÓN MÓVIL

Principios y técnicas

Víctor Viera Balanta

[vieravictor@hotmail.com](mailto:vieravictor@hotmail.com)

Ing. Víctor Viera Balanta

## **COMPUTACIÓN MÓVIL, Principios y técnicas**

### **Computación Móvil Un Avance en la Tecnología**

El continuo avance de la tecnología y las necesidades básicas de comunicación, han permitido el crecimiento de la computación móvil, como un elemento de apoyo en la cotidianidad de las personas, en la agilización y optimización de los procesos de las organizaciones e instituciones educativas.

Es por eso que el término computación móvil, se ha convertido en una situación familiar en nuestra sociedad, si pensamos por ejemplo, en los teléfonos celulares, los cuales han tenido un crecimiento vertiginoso, llegando cada día a formar parte de la vida de las personas, estos poseen una gran acogida debido a las diversas aplicaciones que traen preinstalados, con la posibilidad de incluir muchos programas o aplicaciones cuando el usuario desee.

La funcionalidad de la computación móvil, se ha convertido en un fuerte apoyo a las características que brinda la computación tradicional, y no como se piensa en algunos sectores que viene a reemplazarla. Se habla de apoyo porque permite realizar procesos que la computación tradicional no realiza de forma óptima y más aun, si reconocemos que en algunos ambientes el factor de la movilidad juega un papel súper importante, por ejemplo es el caso del sector salud, en la toma de datos para pacientes en sitio, tal es el caso de el personal que atiende en las ambulancias, por otro lado en los sectores que trabajan atención de desastres, también se puede vislumbrar en sectores académicos, donde se trabaja con esquemas de Aprendizaje Móvil (MLearning), entre otros.

La conciencia de las personas está evolucionando y preparándose para enfrentarse al paradigma de la computación móvil.

Enfréntate a la vida y vence tus temores, porque el nuevo reto que se tiene es: La Computación Móvil.

#### **Alexander Mondragón Valencia Ing. Sistemas y Telemática**

Alexander Mondragón Valencia es Ingeniero de Sistemas y Telemática de la Universidad Santiago de Cali, Colombia. Especialista en Docencia para la Educación Superior de la misma Universidad. Actualmente es estudiante de la Maestría en Ingeniería énfasis en Sistemas y Computación de la Universidad del Valle. Es docente del Programa de Tecnología en Sistemas de la Universidad Santiago de Cali.

**A** Principios del nuevo milenio las tecnologías móviles, obviamente, no estaban tan evolucionadas como en la actualidad. Entre otros, se tenían los celulares por un lado, los busca personas por otro, y las PDA (Asistentes Personales Digitales), a las que tenía acceso el autor eran las de familia PALM, el color que predominaba en la pantalla de estos dispositivos era el verde. Para la comunicación se disponía de un accesorio denominado modem; este accesorio se utilizaba para enviar y recibir datos por medio de la línea telefónica.



Figura No 1. Emulador Palm OS, principios de los 2000

Es de destacar que las PALM venían con un cable especial para la comunicación directa con el Computador; se Utilizaba el software llamado HotSync para intercambiar datos con el PC y la PDA.

## COMPUTACIÓN MÓVIL, Principios y técnicas

En aquellos años la compañía para la que laboraba tenía la siguiente necesidad:

En la compañía “Arrocera la Esmeralda S.A.” se necesitaba una solución donde se pudiera, por medio de un dispositivo móvil, digitar los pedidos en el mismo lugar donde ocurría la venta; luego almacenar estos pedidos en el dispositivo y posteriormente enviarlos a la empresa desde la casa del vendedor.



Figura No 2. Toma del Pedido, almacenado en el Dispositivo móvil y Envió posterior por medio de la línea telefónica.

Con estas condiciones se diseñó la aplicación móvil para la compañía “La Arrocera la Esmeralda S.A.”.

Procedimiento:

Primero: Los dispositivos estaban definidos, familia PALM de la época con cable para conexión al PC y modem para comunicaciones por medio de la línea telefónica.

Segundo: En el Dispositivo móvil, ¿que software y base de datos se utilizará para crear la aplicación?.

Tercero: En el servidor que recibiría los pedidos, ¿ que software y base de datos se utilizara para crear la aplicación?.

## **COMPUTACIÓN MÓVIL, Principios y técnicas**

Cuarto: Instrucciones para elaborar el programa para enviar los datos a la empresa.

Quinto: Capacitación de los vendedores y personal en la empresa.

Sexto: Creación de la Aplicación de pedidos; es preciso mencionar que existía el módulo de facturación.

Séptimo: Generación de la Factura a partir de los Pedidos recibidos, depuración y confirmación de pedidos.

Octavo: del éxito del proyecto dependía mi permanencia en la empresa

Noveno: Como sería el diseño de la Base de datos del dispositivo y de la aplicación en el servidor.

Finalmente el proyecto se finalizó exitosamente y se llegaron a realizar más de 2000 pedidos por medio del dispositivo móvil.

Para este caso, en primer lugar, se definieron los archivos que contendría el dispositivo. Luego se establecieron los campos e información que llevarían estos archivos.

Los archivos que se almacenaron en el dispositivo móvil eran los siguientes:

- Clientes
- Lista de Precios
- Archivo de Pedidos(almacenar pedidos tomados ).
- Ítems o Productos.
- Vendedor

Por diseño y rendimiento solo se tenían en el dispositivo móvil la información específica que necesita un vendedor; como ejemplo se

## **COMPUTACIÓN MÓVIL, Principios y técnicas**

puede mencionar que un vendedor únicamente tenía en el dispositivo la información de sus propios clientes.

En el dispositivo no se colocaron los archivos con todos los campos o atributos, se seleccionaron de cada uno de ellos los campos relevantes, así que para la información del cliente solo se guardó: el código, nombre, la dirección y el teléfono;

Por otro lado, el servidor contaba con toda la información necesaria para el cumplimiento del proceso de pedidos.

Para responder el segundo cuestionamiento, del software y base de datos utilizados en el dispositivo, en aquellos días existía un programa para Dispositivos móviles que poseía ciertas características específicas; este software se denominaba ThinkDB, este era un programa muy amigable que facilitaba la creación de la base de datos y la aplicación.

ThinkDB tenía un complemento para instalar en el servidor, por medio del cual se podía recibir y enviar información desde el dispositivo móvil.

Para realizar las pruebas a la aplicación de “Pedidos Móviles” se acompañó a los vendedores en el proceso de venta; justo en el lugar donde se creaban los pedidos, en los almacenes y tiendas de la ciudad.

Después de la toma de pedidos, el vendedor conectaba el dispositivo a cualquier línea telefónica, marcaba un número telefónico y se daba inicio al proceso de envío de pedidos hacia la empresa.

El anterior proceso tenía algunas dificultades pero en la mayoría de los casos el resultado era satisfactorio.

## ACERCA DEL AUTOR

---

**S**oy Víctor Viera Balanta, ingeniero de sistemas de la Universidad Santiago de Cali. y Tecnólogo en Sistemas de información de la Universidad del Valle. En el momento de escribir este libro adelanto mis estudios de postgrado en Gerencia de Instituciones Educativas.

Durante 10 años he consolidado expectativas y experiencias en el diseño y creación de software; trabajé en empresas nacionales e internacionales como Arrocera la Esmeralda, Universidad del Valle, Emcali-Acuacali, Sysgold, Innosoft, Enigma Developers y Multisys. De igual forma, he realizado proyectos para el Hospital Universitario del Valle, Coches del Pacifico y SMI-Popayan.

De las áreas afines a mi profesión, han salido mis conferencias y reflexiones sobre Computación móvil, Generadores de código, Programación avanzada, Células madre computacionales, publicadas por Internet. Además, coproductor con el empresario, Carlos Londoño, del Software SoftcarManager. En la actualidad, pertenezco al grupo de investigaciones COMBA I+D de la Universidad Santiago de Cali, grupo reconocido por colciencias. Soy profesor tiempo completo, de la Institución Universitaria Antonio José Camacho y he sido docente hora cátedra de la facultad de Ingeniería de la Universidad Cooperativa y del Centro de Estudios Profesionales.

**COMPUTACIÓN MÓVIL, Principios y técnicas**

**A mi sobrino Fernando...Q.E.P.D.**

**A mi esposa , familiares e hijos.**

**En especial a mis padres.**

**Un aporte**

## AGRADECIMIENTOS

---

La producción de este libro es la mejor ocasión para agradecer a la **Institución Universitaria Antonio José Camacho** por darme la oportunidad de hacer parte esta prestigiosa Institución y participar en procesos interesantes como la docencia, la creación de Semilleros de Investigación y el registro calificado del Programa de Ingeniería de Sistemas, entre otros.

También agradezco, especialmente, al **Licenciado Franklyn Benitez** por su asesoría en la revisión textual y paciencia. Además, lo más importante, creer en esta obra.

Igualmente al **Ingeniero William Díaz S.** por su credibilidad en el proyecto de computación móvil en la Institución, a todos los compañeros y compañeros de la facultad de Ingeniería, y otras facultades, y dependencias de la Institución.

A las universidades que hicieron parte de mi proceso de formación profesional como la Universidad del Valle, Universidad Santiago de Cali y de las Instituciones educativas que he trabajado como docente.

Agradecimientos al Ingeniero **Alexander Mondragón Valencia**, compañero en la universidad y ahora en la academia, muchas gracias por su aporte a este libro.

Un agradecimiento especial al Empresario **Carlos Londoño** por su afinidad de equipo y realización del proyecto. Finalmente, agradezco a todas aquellas que me distinguen y considero amigos.

<a href="#">PRÓLOGO</a> .....	5.
<a href="#">INTRODUCCIÓN</a> .....	14
<a href="#">Sincronización</a> .....	18
<a href="#">Dispositivo móvil</a> .....	19
<a href="#">CAMBIO DE PARADIGMA</a> .....	23
<a href="#">SISTEMAS DE CONEXIÓN</a> .....	28
<a href="#">Sistema de Conexión Desconectado</a> .....	28
<a href="#">Sistema de Conexión Semiconectado</a> .....	30
<a href="#">Sistema de Conexión conectado</a> .....	31
<a href="#">Ventajas y Desventajas</a> .....	31
<a href="#">SINCRONIZACION</a> .....	33
<a href="#">Protocolos de Sincronización</a> .....	34
<a href="#">SlowSync</a> .....	34
<a href="#">FastSync</a> .....	36
<a href="#">SyncML</a> .....	37
<a href="#">Eventos de Sincronización</a> .....	38
<a href="#">PROTOCOLO DE COMUNICACIÓN Y ENVIO DE DATOS</a> .....	39
<a href="#">Envío de datos desde la aplicación hacia el celular</a> .....	48
<a href="#">Protocolo de Comunicación BlueTooth</a> .....	53
<a href="#">DISEÑO DE BASES DE DATOS</a> .....	55
<a href="#">Envío de datos de la aplicación central hacia el dispositivo</a> .....	57
<a href="#">VIDEO JUEGOS Y MULTIMEDIA MÓVIL</a> .....	61
<a href="#">Juegos en Red</a> .....	67
<a href="#">Multimedia y Dispositivos móviles</a> .....	69
<a href="#">ALMACENAMIENTO</a> .....	71
<a href="#">Record Managment System</a> .....	71
<a href="#">ACTUALIDAD Y TENDENCIAS</a> .....	76
<a href="#">Computación Wearable</a> .....	78

## **COMPUTACIÓN MÓVIL, Principios y técnicas**

<a href="#">Realidad Aumentada</a> .....	78
<a href="#">Tendencias</a> .....	79
<a href="#">CÓDIGO FUENTE HOLA MUNDO J2ME</a> .....	81
<a href="#">CÓDIGO FUENTE VIDEOJUEGO CON ANIMACIÓN</a> .....	84
<a href="#">CÓDIGO FUENTE VIDEOJUEGO TRES EN LINEA</a> .....	95
<a href="#">CÓDIGO FUENTE TRANSMISIÓN DE DATOS VIA HTTP</a> .....	128
<a href="#">BIBLIOGRAFIA</a> .....	135

---

## **INTRODUCCIÓN**

**N**o es un secreto que las aplicaciones móviles tienen gran acogida en la actualidad. Las grandes compañías creadoras de hardware y software invierten millones de dólares en el diseño y creación de una gran variedad de dispositivos móviles.

Muchos países incluyen en sus planes de desarrollo el incentivo para el sector de software; incluido la creación de productos para dispositivos móviles.

Este libro está diseñado para aquellos o aquellas que desean aprender o simplemente conocer el mundo del desarrollo de aplicaciones móviles.

El enfoque del libro toma distancia de la enseñanza de un lenguaje de programación específico, para ello ya existen otros libros e información en Internet que se referencian en la bibliografía. Este libro se basa en el diseño y creación de aplicaciones móviles. Los ejemplos se plantean con J2ME Y C#, que son lenguajes de programación utilizados en la creación de software para dispositivos móviles, pero siempre con el énfasis “análisis y diseño primero”.

## **COMPUTACIÓN MÓVIL, Principios y técnicas**

Los planteamientos desarrollados en este libro van dirigidos a nivel de comprensión para profesionales de la ingeniería, estudiantes, empresarios, directores de proyectos y docentes, respectivamente.

Se trata de brindar un panorama novedoso, donde se exponen estrategias, trucos, dificultades y experiencias en el diseño y creación de aplicaciones móviles.

El presente libro es el resultado de mas de 10 años de experiencia en el desarrollo de aplicaciones tanto móviles y no móviles, también es fruto de la experiencia en el campo académico e investigativo en reconocidas Instituciones Universitarias.

Es grato comentar que varios estudiantes de computación móvil guiados por el autor. están cosechando éxitos en este campo.

El libro esta dividido en diez capítulos. El primero presenta una perspectiva de la influencia de la tecnología móvil y la importancia de está en la actualidad.

El segundo capítulo hace referencia a la arquitecturas de aplicaciones móviles, donde se describen los elementos principales de este tipo de aplicaciones; la aplicación central, el proceso de sincronización y el dispositivo móvil.

En su recorrido temático el libro, en el tercer capítulo, presenta los cambios que genera la industria de la tecnología móvil en el desarrollo de aplicaciones y la eficiencia que obtiene un proceso aplicando esta modalidad.

En el cuarto avance circulan los conceptos básicos para comprender los diversos tipos de diseño existentes referente a sistemas de conexión para aplicaciones móviles. Aquí se describe, detalladamente, el sistema conectado, semi conectado y desconectado.

El lector en el capítulo quinto encuentra técnicas, algoritmos y protocolos de sincronización entre un dispositivo y una aplicación central; en el sexto se explican los protocolos de comunicación que se

## **COMPUTACIÓN MÓVIL, Principios y técnicas**

pueden usar para enviar datos entre un dispositivo y una aplicación central.

El capítulo séptimo aborda las técnicas de diseño de bases de datos para aplicaciones desarrolladas en tecnología móvil. El capítulo octavo presenta un detallado panorama de la creación de juegos para dispositivos móviles en lo que concierne a la tecnología J2me. En este apartado hay algunos ejemplos de videojuegos y fragmentos de código fuente, claves para la compresión y creación de los videojuegos; en el capítulo noveno usted encontrará ejemplos para su comprensión. Finalmente en el capítulo décimo hay información actualizada y tendencias de la computación móvil; aquí se abordan temas como la computación wearable, realidad aumentada y la banca móvil.

La última parte del libro cuenta con apéndices que se describen a continuación:

Apéndice A, código fuente del programa “HolaMundo”.

Apéndice B, código fuente de videojuego con animación.

Apéndice C, código fuente videojuego “tres en línea”.

Apéndice D, código fuente de transmisión de datos vía Http.

Se solicita a los lectores enviar sugerencias, comentarios, vivencias, enlaces, a la dirección de correo [vieravictor@hotmail.com](mailto:vieravictor@hotmail.com); o para compartir una conferencia, charla, etc.

## ARQUITECTURA APLICACIONES MÓVILES

**E**xisten varios escenarios en los cuales se puede establecer la arquitectura de aplicaciones móviles; aquí se abordaran dos de ellos.

En el primero participan tres elementos:

- La aplicación central
- El proceso de sincronización
- La aplicación en el dispositivo móvil



Figura No 2.1. Elementos aplicaciones móviles, escenario 1.

A la aplicación central le corresponde el almacenamiento de los datos enviados por el dispositivo; también le corresponden los procesos de mayor consumo de memoria y tiempo e interfaces para los clientes con dispositivos no móviles.

### **Sincronización**

**La sincronización** es un proceso por medio del cual se mantiene la integridad y consistencia de los datos almacenados, tanto en el dispositivo móvil como en la aplicación central.

En este modelo, cuando se envía un dato desde el dispositivo móvil, el dato pasa primero por el proceso de sincronización, el proceso de sincronización valida y guía al dato a su destino en la aplicación central, luego la sincronización devuelve una respuesta al dispositivo.

El proceso de sincronización puede ser un programa totalmente independiente de la aplicación central; o por el contrario puede ser un módulo, subprograma o componente de la aplicación central; hasta hoy, se considera conveniente que el proceso de sincronización no esté en el dispositivo móvil.

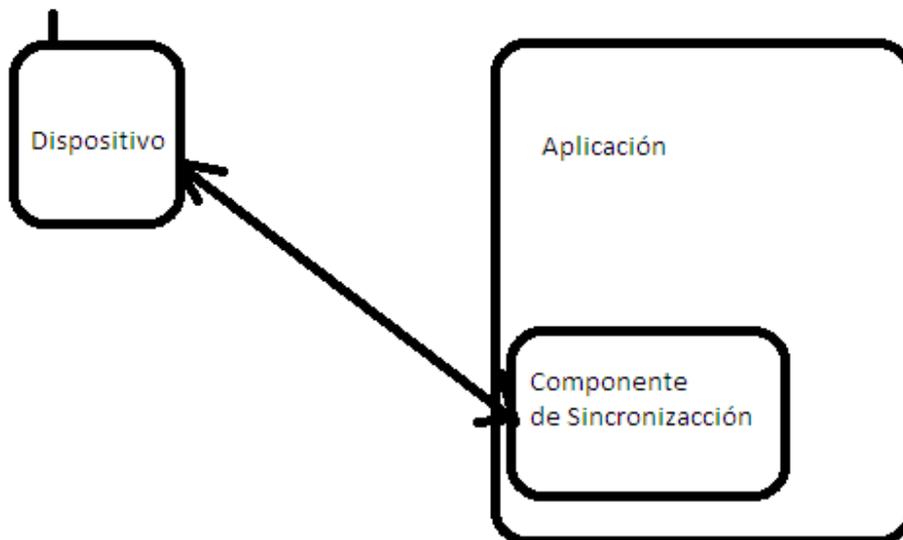


Figura No 2.2. La sincronización hace parte de la Aplicación Principal.

## **COMPUTACIÓN MÓVIL, Principios y técnicas**

En el capítulo No 5, se abordaran los detalles de la sincronización. Por ahora, se establece que se refiere al proceso por el cual se intercambian y validan datos entre el dispositivo y la aplicación; este intercambio de datos puede tener como objetivo puntual, la creación, el borrado o la actualización de los datos de un cliente que este registrados en el dispositivo móvil. La sincronización es útil para mantener una armonía entre el dispositivo móvil y la aplicación central.

Para dar un ejemplo de sincronización, se puede pensar en el proceso de conectar el celular al computador para obtener una copia de seguridad del dispositivo; así se tendrán los datos del dispositivo actualizados en el Computador. La utilización de la sincronización depende de las características y objetivos del sistema a diseñar, construir e implementar; puede necesitarse que los dispositivos móviles estén habilitados para realizar algunos procesos, desconectados temporalmente del sistema central.

### ***Dispositivo móvil***

**El dispositivo móvil** realiza el proceso básico de recolección y envío de datos hacia la aplicación central. El dispositivo móvil contiene algunos procesos importantes; pero los procesos de mayor complejidad, en lo posible, se deben realizar en la aplicación central.

**En el segundo escenario participan dos elementos:**

- El dispositivo móvil
- La aplicación central

## COMPUTACIÓN MÓVIL, Principios y técnicas

También se pueden encontrar sistemas computacionales móviles donde los datos no pasan por un proceso de sincronización propiamente dicho; estas aplicaciones poseen las validaciones necesarias en formularios o ventanas que se encuentran en el dispositivo; estas aplicaciones tienen la característica que están conectadas permanentemente a la aplicación central; se puede afirmar que la aplicación en el dispositivo móvil no funciona sin estar conectado o en línea con la aplicación central. Ver Figura 1.3.

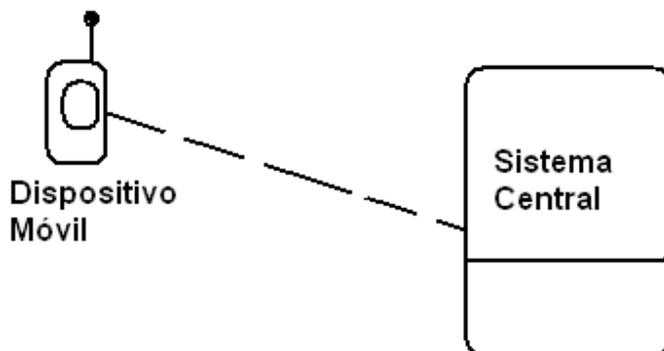


Figura No 2.3. Aplicación sin módulo de sincronización de datos.

Otra particularidad de estos sistemas hace referencia a los formularios que se visualizarán en el dispositivo móvil, estarán ubicados en la aplicación central, este caso se presenta cuando el dispositivo móvil posee o tiene la capacidad de interpretar un formulario Web por medio de un navegador o "Browser". Ver figura 2.4.

## COMPUTACIÓN MÓVIL, Principios y técnicas

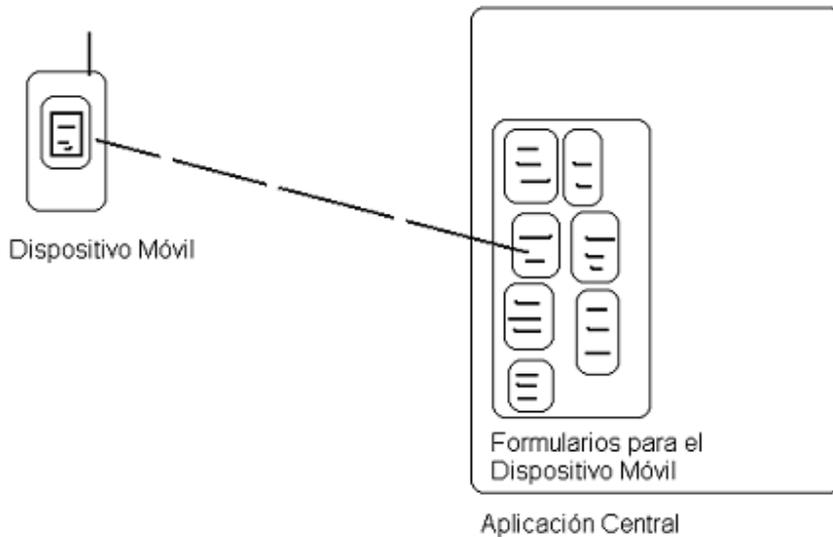


Figura No 2.4. Formularios ubicados en la aplicación central, el dispositivo móvil interpreta estos por medio de un navegador Web o programa similar.

Adicionalmente, en este tipo de escenario, conectado, puede darse el caso de que se cree una aplicación para el dispositivo móvil, y esta aplicación contenga las ventanas que hacen posible la captura y posterior envío de datos a la aplicación central. En este escenario, las ventanas están almacenadas en el dispositivo, y estas se comunican con procesos de la aplicación central.

## COMPUTACIÓN MÓVIL, Principios y técnicas

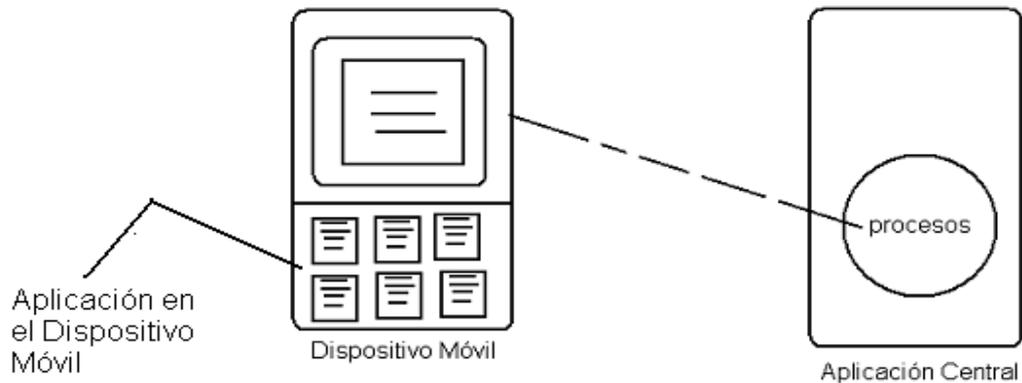


Figura No 2.5. Dispositivo móvil con una aplicación que permite la captura de datos y su posterior envío al sistema central.

En el dispositivo únicamente están almacenadas las ventanas y procesos básicos de validación de datos; el almacenamiento de datos, se presenta solo en la aplicación central.

---

## **CAMBIO DE PARADIGMA**

**A**nteriormente muchas aplicaciones o programas se diseñaban y construían con interfaces exclusivas para entrada de datos provenientes de un teclado de computador. Además, se elaboraba un formato o formulario para llenarse a mano en el lugar donde sucedía la acción o se establecía la fuente de datos; los datos se ingresaban en la aplicación a partir de la información contenida en los formatos.

Para citar un ejemplo, en el escenario donde un cliente lleva su vehículo averiado para que este sea reparado en un taller automotriz. al momento de llegada del vehículo al taller, se realiza un diagnóstico o estudio del daño presentado en el vehículo; este proceso se lleva a cabo con la ayuda de un formato especial para este tipo de diagnóstico. El formato para realizar el diagnóstico contiene los campos respectivos para describir las actividades de la reparación y el valor o costo aproximado de esta.

Luego, el formato diligenciado, a mano, se entregaba a una persona; esta persona interpretaba los datos escritos, después digitaba lo interpretado, de la información contenida en el formato, en los campos de la aplicación; en el mejor de los casos, la persona que digitaba los datos, entendía e interpretaba adecuadamente lo escrito en el formato o formulario.

## COMPUTACIÓN MÓVIL, Principios y técnicas

En ocasiones la misma persona que realizaba el diagnóstico al vehículo, se encargaba de llenar el formato e interpretar y digitar los datos en la aplicación; aun así, esto podía ocasionar errores y tardanzas. Finalmente, se entregaba la cotización al cliente, bien presentada, con los costos de la reparación. En la figura No 3.1 se muestra la cadena del proceso.

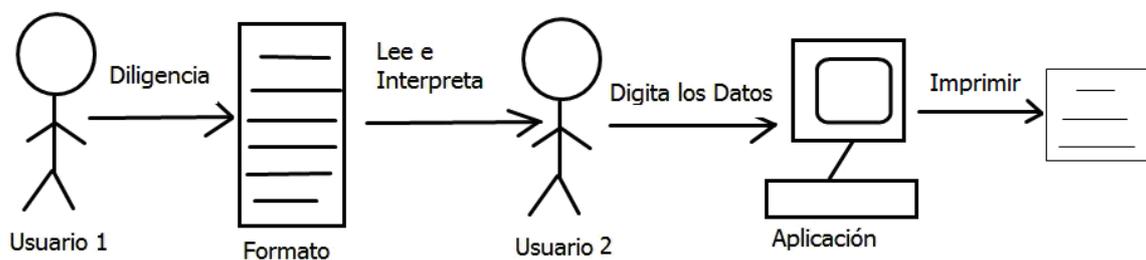


Figura No 3.1 Proceso no soportado por la computación móvil. Se llena un formato; este se interpreta por otro usuario, y este usuario digita en la aplicación lo interpretado.

En la Figura 3.2, lado izquierdo, se observa la cantidad de formatos atrasados y pendientes por interpretar y digitar en la aplicación; el usuario debe sacrificar gran parte de tiempo para cumplir con las tareas no terminadas. En el lado derecho se observa como se evita los procesos o reproceso de interpretación y digitación de los datos escritos en los formatos o formularios; aquí los datos ya han sido introducidos en la aplicación en el lugar donde ocurre el proceso; y mejor aun, los datos fueron digitados y enviados por medio de un dispositivo móvil.

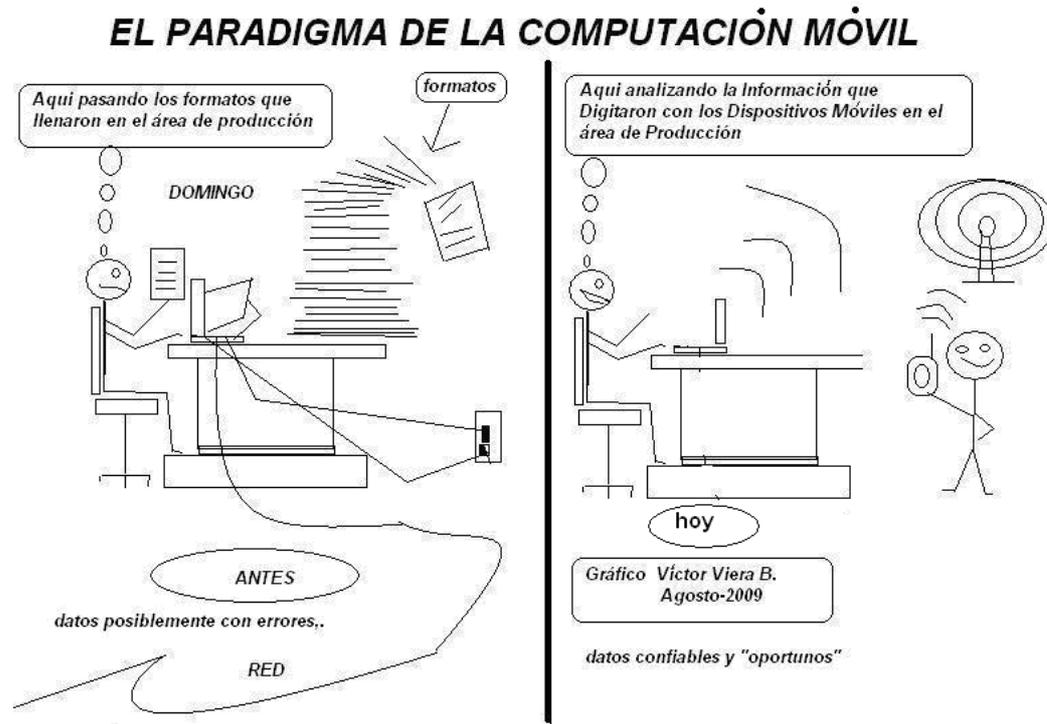


Figura No 3.2. Paradigma computación Móvil, lado Izquierdo cables, formatos y muchos errores de interpretación; lado derecho agilidad en procesos y ahorro de recursos.

**Se puede afirmar lo siguiente: los procesos soportados por la computación móvil son mucho mas ágiles, presentan menos errores, un ahorro significativo de recursos y el usuario puede estar, en el área de cobertura, en constante interacción con la aplicación central. Donde cambiamos el papel por el dispositivo.**

Entonces, es factible la representación de un proceso soportado por la computación móvil. Ver Figura No 3.3.

## COMPUTACIÓN MÓVIL, Principios y técnicas

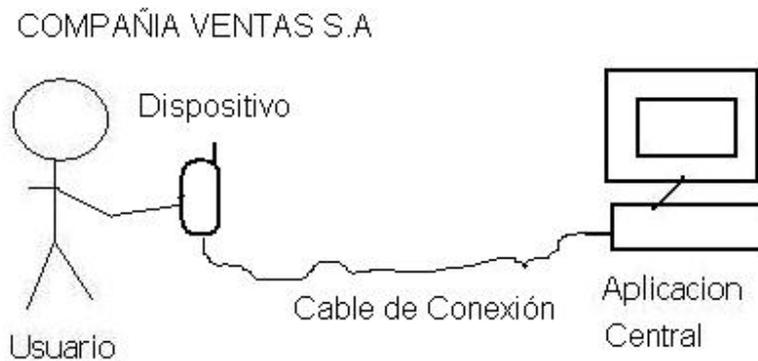


Figura No 3.3. Proceso soportado por la computación móvil.

El entorno muestra que la concepción de los sistemas y procesos esta cambiando; se puede observar como las compañías de apuestas han dotado a sus vendedores y vendedoras de dispositivos móviles; donde es posible realizar una apuesta en muchos sitios y el vendedor se dirige al cliente con el dispositivo; cabe anotar que estas apuestas se registran en línea en la aplicación central; este registro ocurre en el mismo instante que las apuestas se digitan en el dispositivo móvil.

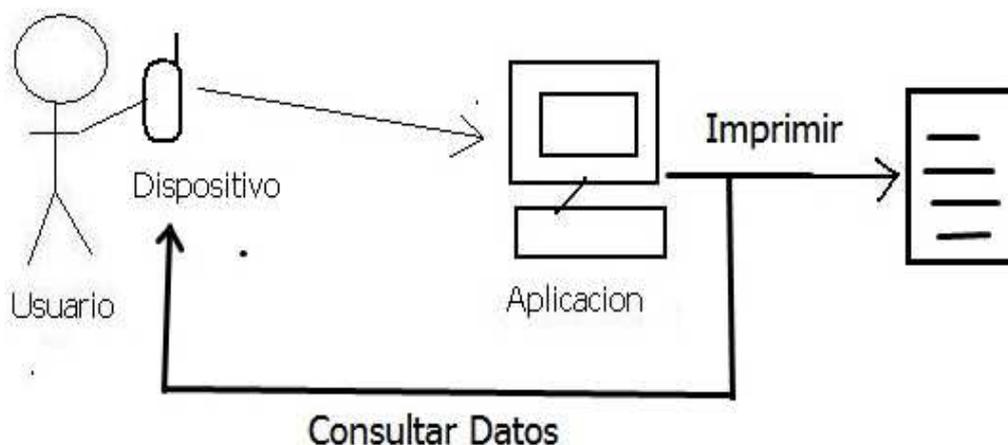


Figura No 3.4. Proceso soportado por la computación móvil. Con impresión y consulta de datos por medio del dispositivo.

## **COMPUTACIÓN MÓVIL, Principios y técnicas**

En la figura 3.4, se puede apreciar otra ventaja de los dispositivos móviles, estos permiten la consulta de información, la visualización de reportes y, entre otras, la actualización de información.

Otro ejemplo se puede apreciar con los vendedores que llevan dispositivos móviles para la toma de pedidos.

Existen otros sectores, como el de la salud, donde la computación móvil comienza a ser fuertemente utilizada. Los médicos pueden tener acceso a las historias clínicas en todo momento, también podrían realizar cambios y adiciones a los registros en cualquier lugar de la clínica. Lo anterior, claro esta, con los debidos controles de auditoria y seguridad.

Esta época exige que se aborden las soluciones informáticas de una forma diferente.

## **SISTEMAS DE CONEXIÓN**

---

Las aplicaciones móviles se pueden diseñar y construir con tres tipos diferentes de sistemas de conexión:

- a) Desconectado
- b) Semiconectado
- c) Conectado

De estos tipos de conexión dependerá la forma en que la aplicación envíe los datos a la aplicación.

### *Sistema de Conexión Desconectado*

En este sistema de conexión, los procesos se realizan con el dispositivo móvil totalmente desconectado de la aplicación central. Después de terminar los procesos y almacenar la información en el dispositivo se procesa a conectar el dispositivo y ha enviar los datos almacenados a la aplicación central.

Actualmente existen aplicaciones móviles como la de captura de pedidos, donde el vendedor visita a los clientes les toma los pedidos y almacena los estos en el dispositivo móvil, el dispositivo contiene la aplicación para realizar la captura de pedidos. Al

## COMPUTACIÓN MÓVIL, Principios y técnicas

finalizar la jornada, el vendedor lleva el dispositivo a la empresa o lugar de conexión y sincroniza el dispositivo con la aplicación central. En este momento ocurre de intercambio de datos entre el dispositivo y la aplicación central. Este proceso se denomina: sincronización.

El proceso de conexión con la aplicación se puede realizar por medio de un cable y dispositivo especial para esta tarea este se denomina “cuna” o en inglés “Cradle”. Esto se ilustra en la figura 4.1.



Figura No 4.1. Dispositivo Móvil conectado y realizando el proceso de sincronización con la aplicación central. Conexión por medio de cable y cuna.

El proceso de conexión y sincronización también se puede realizar por medio de una red inalámbrica, cuando el dispositivo está en el área de cobertura. Figura 4.2.

## COMPUTACIÓN MÓVIL, Principios y técnicas



Figura No 4.2. Dispositivo Móvil conectado y realizando el proceso de sincronización con la aplicación central. Conexión por medio de una red inalámbrica.

Cabe aclarar que el término desconectado significa que la sincronización se hace al terminar el proceso o procesos, posteriormente, se envía la información en bloque, del dispositivo a la aplicación central. Luego, se presenta almacenamiento de información, relacionada con el proceso, en el celular.

### *Sistema de Conexión Semiconectado*

En este tipo de sistema de conexión, el dispositivo en ocasiones permanecerá desconectado de la aplicación central; se establece que el dispositivo pueda seguir trabajando cuando se presenten caídas de la red o en el momento que el dispositivo se encuentre fuera del área de cobertura de la red. Este tipo de conexiones prepara a la aplicación en el dispositivo para trabajar enviando datos a la aplicación central en línea; pero también se construye para el trabajo con el

## COMPUTACIÓN MÓVIL, Principios y técnicas

dispositivo sin depender, hasta cierto punto, de la conexión con la aplicación central.

Existen aplicaciones desarrolladas en los talleres de autos, donde los diagnósticos se pueden realizar en línea, en las instalaciones del taller; o en forma desconectada, para realizar un diagnóstico a un vehículo que se encuentra fuera del área de cobertura de la red del taller. Al hacer el diagnóstico, fuera del área de cobertura se debe almacenar este en el dispositivo móvil para, luego, ser sincronizado con la aplicación central.

### *Sistema de Conexión conectado*

En este tipo de sistema de conexión, se espera que el dispositivo móvil esté siempre conectado con la aplicación central. En caso de no haber comunicación, no se podría trabajar con el dispositivo, porque la aplicación en el dispositivo no posee base de datos o archivos de almacenamiento; todo se realiza, por medio del dispositivo, en la aplicación central.

Este tipo de aplicaciones conectadas se observan en las empresas de apuestas, donde el juego, realizado por medio del dispositivo, debe quedar registrado inmediatamente en la aplicación central. No se permiten apuestas almacenadas en el dispositivo.

### *Ventajas y Desventajas*

Los sistemas de Conexión presentan características que se deben tener en cuenta al momento de establecer un modelo para soportar una aplicación móvil. Estas características son:

## COMPUTACIÓN MÓVIL, Principios y técnicas

En el sistema desconectado, puede ser de gran ayuda poseer cierta independencia de la aplicación central. En caso de fallos, los dispositivos móviles pueden seguir trabajando, recolectando información y evitando que los procesos se detengan. Una desventaja de los sistemas desconectados es el tiempo que transcurre desde el momento que se procesa la información en el dispositivo; y que esta se sincroniza con el servidor central. En un caso extremo el dispositivo se puede averiar o perder, antes de sincronizar la información.

En el sistema de conexión SemiConectado se deben diseñar y especificar muy bien los momentos de sincronización para evitar pérdidas de información por datos atascados en el dispositivo. En muchas ocasiones estas tareas dependen del usuario que porta el dispositivo móvil. Se puede presentar el no envió de datos a tiempo, por descuidos o problemas en el sincronismo que deben tener este tipo de procedimientos.

En el sistema desconectado no existe, en la mayoría de los casos, almacenamiento en el celular; el dispositivo depende totalmente del sistema central.

**La escogencia de un sistema de conexión, depende de las características y necesidades del negocio o empresa para la cual se construirá la aplicación móvil.**

## SINCRONIZACION

Los algoritmos y técnicas de sincronización representan el proceso central de la computación móvil. Se presentan tanto en aplicaciones móviles como en los procesos de actualización y copias de seguridad de datos de Celulares, Agendas electrónicas, Pdas, otros.

La Sincronización es el proceso por el cual se intercambian datos, entre dos sistemas, los datos intercambiados afectan la información de ambos sistemas. La sincronización implica el intercambio de datos; este proceso se realiza por medio de protocolos de sincronización, donde ambos sistemas conocen la forma de intercambiar datos.

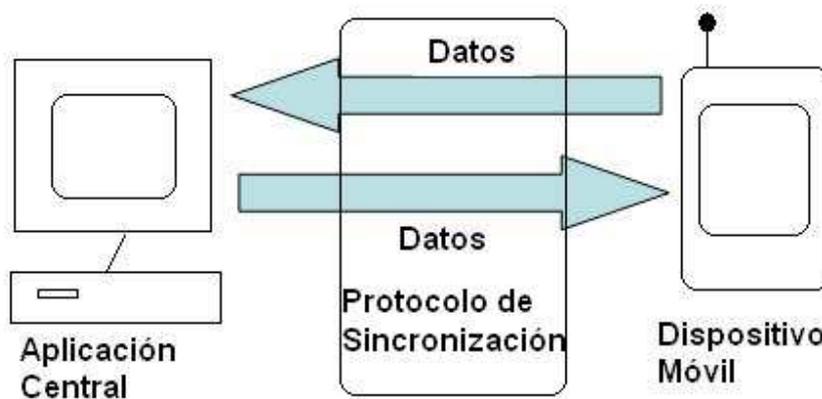


Figura No 5.1. Sincronización, intercambio de datos entre sistemas, por medio de protocolos de sincronización.

## COMPUTACIÓN MÓVIL, Principios y técnicas

La sincronización mantiene la integridad de la información tanto en el dispositivo como en la aplicación central. La sincronización también es la responsable de la concordancia de los procesos del dispositivo; se mantiene el dispositivo actualizado respecto a la aplicación central.

### **Protocolos de Sincronización**

#### **SlowSync**

En el protocolo **SlowSync (sincronización Lenta)** se transmiten todos los datos, del dispositivo a la aplicación central, y se comparan uno a uno, para establecer cual registro presenta cambios, eliminaciones o adiciones.

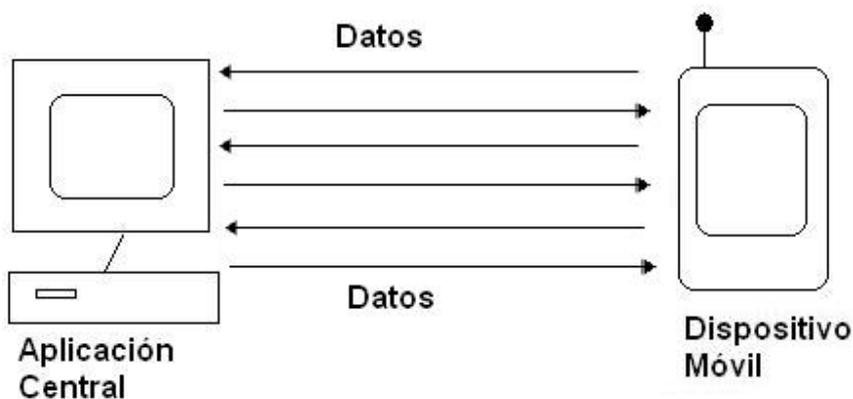


Figura No 5.2. Sincronización por medio de **SlowSync**, todos los datos del dispositivo, nuevos y viejos, se envían y se confrontan en la aplicación central.

## COMPUTACIÓN MÓVIL, Principios y técnicas

Este protocolo es lento porque sí el dispositivo y la aplicación central presentan datos comunes, la sincronización SlowSync, vuelve y sincroniza estos datos.

A manera de ejemplo, tenemos lo siguiente: Si se tiene una aplicación de ventas, donde el dispositivo y la aplicación central presentan los mismos registros de clientes; en el caso de que el vendedor cree un nuevo cliente en el dispositivo, y posteriormente exista una sincronización, el dispositivo enviara y comparara todos los datos, los que ya existen y los nuevos. Lo anterior se ilustra en la figura 5.3.

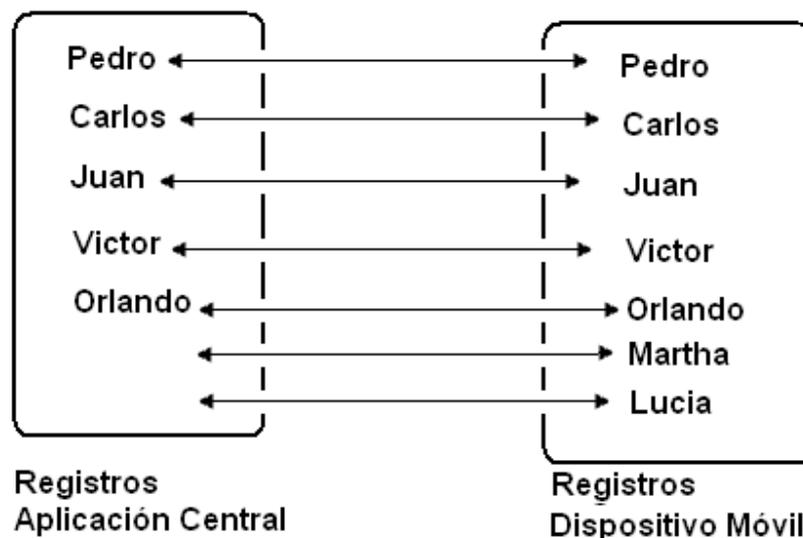


Figura No 5.3. Sincronización por medio de **SlowSync**, tanto los registro nuevos como los viejos se sincronizan con la aplicación central.

## COMPUTACIÓN MÓVIL, Principios y técnicas

El protocolo siguiente es más rápido que el protocolo anterior.

Veamos:

### ***FastSync***

En el protocolo **FastSync (sincronización Rápida)**, únicamente se sincronizan los datos que presentan cambios y los datos nuevos. Los datos comunes entre el dispositivo y la aplicación central no se modifican. Figura 5.4.

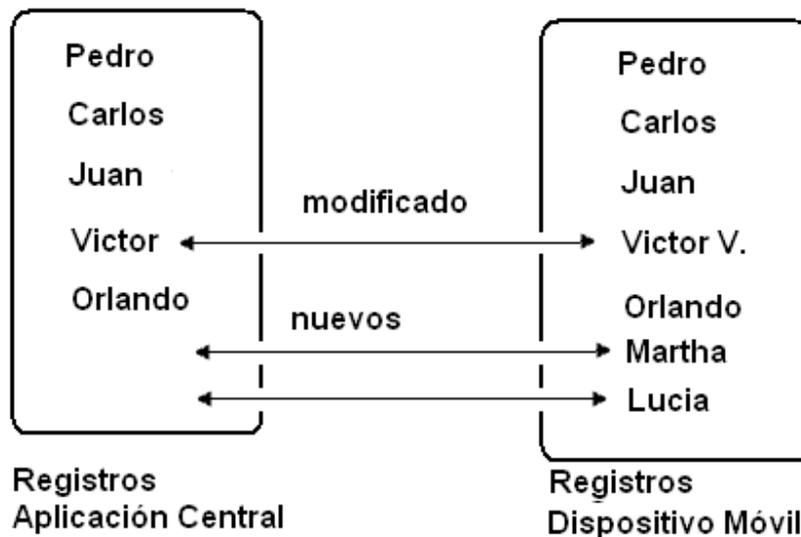


Figura No 5.4. Sincronización por medio de **FastSync**, Únicamente se sincronizan los datos nuevos o Modificados.

Este protocolo de sincronización utiliza un variable de estado para marcar los registros, los registros se marcan como: registros nuevos, registros actualizados y registros eliminados.

## COMPUTACIÓN MÓVIL, Principios y técnicas

Otro protocolo de comunicación es el **SyncML**, este protocolo es basado en XML<sup>1</sup> ( Extensible Markup Language) , **SyncML**, es utilizado para la comunicación remota entre dispositivos móviles y la aplicación central. Esto se muestra en la figura 5.5.

### **SyncML**

**SyncML**, fue creado por la **Oma<sup>2</sup>** (**Open Mobile Alliance**) , Figura No 5.5.

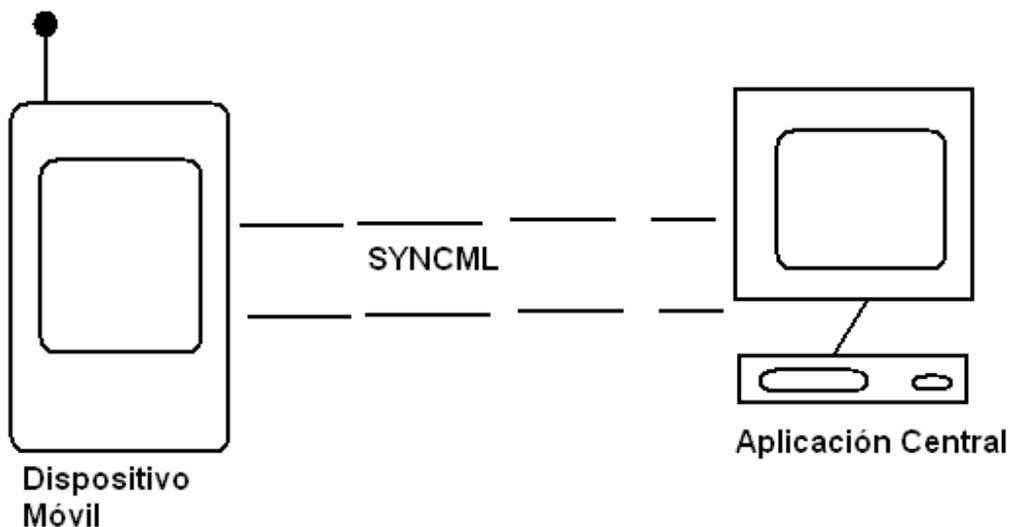


Figura No 5.5. Sincronización por medio de SyncML,

---

<sup>1</sup>Guía breve tecnologías ML,W3C,2008

<http://www.w3c.es/divulgacion/guiasbreves/tecnologiasxml>

<sup>2</sup> Open Mobile Alliance, Desarrolla estándares abiertos para la Industria de telefonía móvil

<http://www.openmobilealliance.org/index.html>

## ***Eventos de Sincronización***

Al momento de sincronizar la aplicación central y el dispositivo móvil se pueden presentar los siguientes eventos:

**Inserción:** El registro Marcado como nuevo, en el dispositivo, es creado en la aplicación central.

**Modificación:** El registro en el Dispositivo móvil, reemplaza al registro en la aplicación central.

**Borrado:** se borra el registro en la aplicación central. La sincronización también puede borrar un registro existente en el dispositivo móvil.

## **PROTOCOLO DE COMUNICACIÓN Y ENVÍO DE DATOS**

Los sistemas de conexión abordados en el capítulo 4 se complementan con el protocolo de comunicación utilizado para la transmisión de datos entre el dispositivo y el servidor. En este capítulo se utilizará el protocolo de comunicación Http (Hypertext Transfer Protocol). En español, Protocolo de transferencia de hipertexto. Este sistema es la base de las transacciones de la WWW (World Wide Web); Http fue creado por el Consorcio World Wide Web (W3C)<sup>3</sup>.

El protocolo Http, cuenta con dos tipos de comandos: Get y Post.

Con ambos comandos, Get y Post, se puede interactuar con el servidor, enviando peticiones (Request) y recibiendo respuestas (Response).

---

<sup>3</sup> <http://www.w3c.es/>

## COMPUTACIÓN MÓVIL, Principios y técnicas

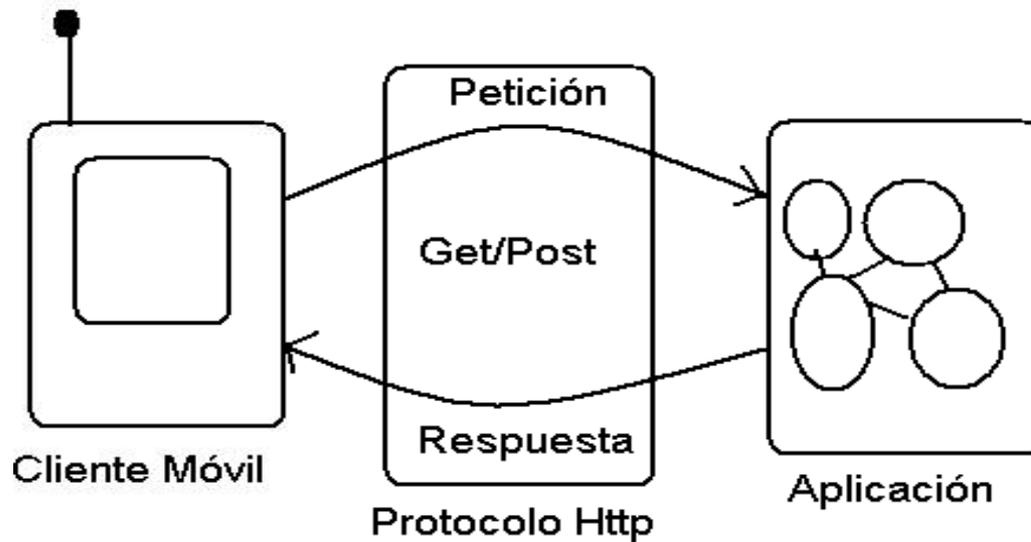


Figura No 6.1. Esquema comunicación por medio del protocolo Http.

Los dispositivos móviles y los lenguajes de programación actuales soportan Http como protocolo de comunicación.

Este protocolo funciona tanto en redes inalámbricas como en redes que necesitan cables para el envío de datos.

```
public void envioMae()
{
    try{
        String url = "http://127.0.0.1/recibir.php?codigo="+codigo.getString();
        //se abre la conexión y se usa la Interface HttpURLConnection,lectura-escritura
        HttpURLConnection hc = (HttpURLConnection)Connector.open(url, Connector.READ_WRITE);
        hc.close();
    }catch(Exception e){
        //ocurrio un error
        e.printStackTrace();
    }
}
```

Las anotaciones en el código indican: 'servidor' apunta a '127.0.0.1', 'proceso' a 'recibir.php', 'variable' a 'codigo' y 'dato enviado' a 'codigo.getString()'.

## COMPUTACIÓN MÓVIL, Principios y técnicas

Figura No 6.2. Fragmento de código J2me para envío de datos http por medio del método Get, desde el Dispositivo

```
private void enviar_cadenahttp()
{
    String url = "http://127.0.0.1/proceso.php?codigo="+codigo.getString();
    try
    {
        HttpWebResponse response;
        //procesa la dirección Web
        WebRequest request = WebRequest.Create(url);
        request.Credentials = CredentialCache.DefaultCredentials;
        //recibe la respuesta del servidor
        response = (HttpWebResponse)request.GetResponse();
    }
    catch (WebException mm)
    {
        MessageBox.Show("falló la conexión" + mm.Message+mm.Message);
    }
    catch (UriFormatException)
    {
        MessageBox.Show("Dirección Incorrecta");
    }
}
```

Figura No 6.3. Fragmento de código en el lenguaje C# para el envío de datos http por medio del método Get, desde el dispositivo.

Las figuras 6.2 y 6.3 ilustran las instrucciones utilizadas para enviar datos por medio del protocolo Http, en este caso, el dato se envía a un servidor de red con dirección o número IP 127.0.0.1, en este servidor se encuentra el programa "proceso" esperando las solicitudes.

Hasta este punto sólo se tiene el código de envío de datos desde un dispositivo móvil; Ahora, se explica lo concerniente al servidor.

## **COMPUTACIÓN MÓVIL, Principios y técnicas**

Para recibir y responder solicitudes Http se necesita un servidor Web; estos servidores son programas especializados en este tipo de procesos; estos soportan protocolos Http. Los programas de servidores Web o Http, trabajan como intermediarios entre el dispositivo móvil y la aplicación central.

Aquí se mencionan servidores Http de libre distribución:

Software Apache Server<sup>4</sup>, programa ampliamente utilizado para soportar las aplicaciones web.

Software Apache Tomcat<sup>5</sup>, programa utilizado regularmente con la familia de programas Java.

Software Cherokee<sup>6</sup>, hace referencia a un programa que cumple con las principales funciones de los servidores Web.

Además de contar con el servidor web, se debe tener un lenguaje de programación donde se elaboren los programas de la aplicación y parte del sistema de sincronización.

Entre los lenguajes de programación existentes de libre distribución tenemos:

Php<sup>7</sup>, lenguaje de programación utilizado para la creación de aplicaciones

---

<sup>4</sup> <http://www.apache.org/>

<sup>5</sup> <http://tomcat.apache.org/>

<sup>6</sup> <http://www.cherokee-project.com/>

## **COMPUTACIÓN MÓVIL, Principios y técnicas**

Java<sup>8</sup>, lenguaje de programación de gran acogida para el desarrollo de aplicaciones

Python<sup>9</sup>, lenguaje robusto de utilizado en la creación de aplicaciones

C#<sup>10</sup>, Es un lenguaje potente y moderno, de libre distribución, se puede utilizar un entorno de desarrollo comercial.

Ruby on Rails<sup>11</sup>, lo comprenden un conjunto de utilidades especiales para la creación de aplicaciones Web, creadas en el lenguaje de programación Ruby.

Para completar los componentes, se necesita una base de datos, aquí se mencionan algunas bases de datos:

MySQL<sup>12</sup>, de gran acogida para acompañar y soportar el sistema de bases de datos necesario en las aplicaciones web.

PostgreSQL<sup>13</sup>, potente sistema de bases de datos, muy utilizado en el entorno de desarrollo web.

Firebird<sup>14</sup>, sistema de bases de datos robusto y confiable,

---

<sup>7</sup> <http://php.net/>

<sup>8</sup> <http://java.sun.com/>

<sup>9</sup> <http://www.python.org/>

<sup>10</sup> [http://en.csharp-online.net/CSharp\\_Language\\_Specification](http://en.csharp-online.net/CSharp_Language_Specification)

<sup>11</sup> <http://rubyonrails.org/>

<sup>12</sup> <http://dev.mysql.com/>

<sup>13</sup> <http://www.postgresql.org/>

## COMPUTACIÓN MÓVIL, Principios y técnicas

Es pertinente resaltar que también existen sistemas de bases de datos, servidores Web y lenguajes de programación, comerciales potentes y confiables.

La figura 6.4 ilustra la arquitectura que combina: el servidor, el lenguaje de programación y el programa en el dispositivo móvil.

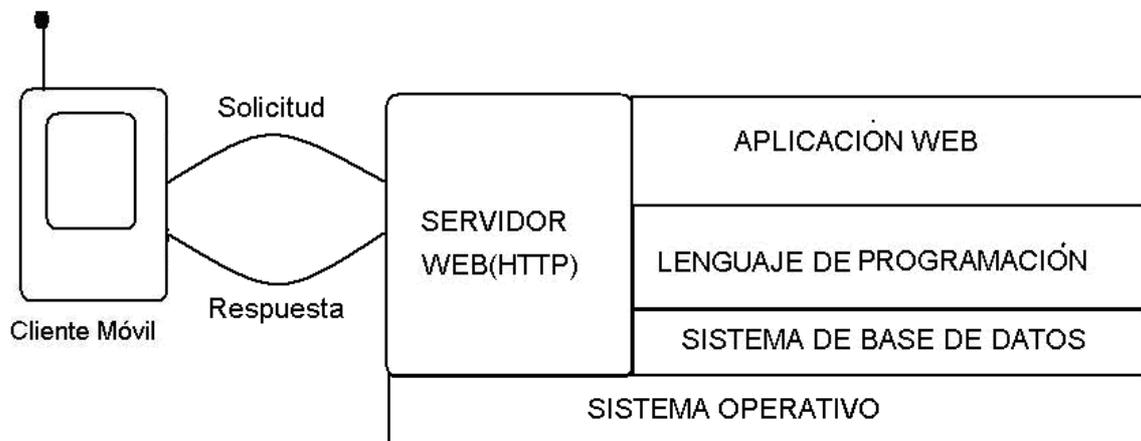


Figura No 6.4. Arquitectura de sistema de Computación Móvil.

El dispositivo móvil debe tener una composición de software como:

- a) Sistema Operativo
- b) Programa escrito en un lenguaje de programación
- c) Otras utilidades

---

<sup>14</sup> <http://www.firebirdsql.org/>

## **COMPUTACIÓN MÓVIL, Principios y técnicas**

Entre los sistemas operativos para dispositivos móviles se tienen: Symbian Os, Android, Linux, Windows Mobile, Blackberry Os, iPhone Os, Palm Os.

Entre los lenguajes de programación utilizados para escribir programas para dispositivos móviles se tienen:

C#, utilizado en Pdas y Teléfonos Inteligentes familia Microsoft.

J2Me , de gran aceptación y difusión, pionero en el entorno móvil.

C, utilizado en varios dispositivos móviles.

El lenguaje de programación utilizado, aquí, para recibir los datos provenientes del clientes móvil, será Php<sup>15</sup> un lenguaje de programación de gran utilización para el desarrollo Web. Ver figura No. 6.5.

---

<sup>15</sup> <http://php.net/>

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
<?PHP
    $codigo=$_GET['codigo'];
    echo "LOS DATOS RECIBIDOS SON \n";
    //conexion base de datos
    //$myconn = @mysql_connection("servidor","user","pass");

    $conexion = mysql_connect("localhost", "root", "");

    echo "procesado";
    if (!$conexion){
        echo "Error al intentar conectarse con el servidor MySQL";
        exit();
    }else{
        echo "conectado";
        //selecciono la base de datos
        mysql_select_db("nueva", $conexion);
        $cadena = "insert into ciudad(codigo) values('$codigo)";
        //se ejecuta el query
        $resultado = mysql_query($cadena, $conexion) or die(mysql_error());
    }
?>
```

Recibe el Dato Código

Figura No 6.5. Fragmento de código en Php para atender el envío de datos desde un cliente móvil.

La figura 6.6 ilustra la apariencia de la ventana de envío de datos del celular hacia la aplicación central.

## COMPUTACIÓN MÓVIL, Principios y técnicas

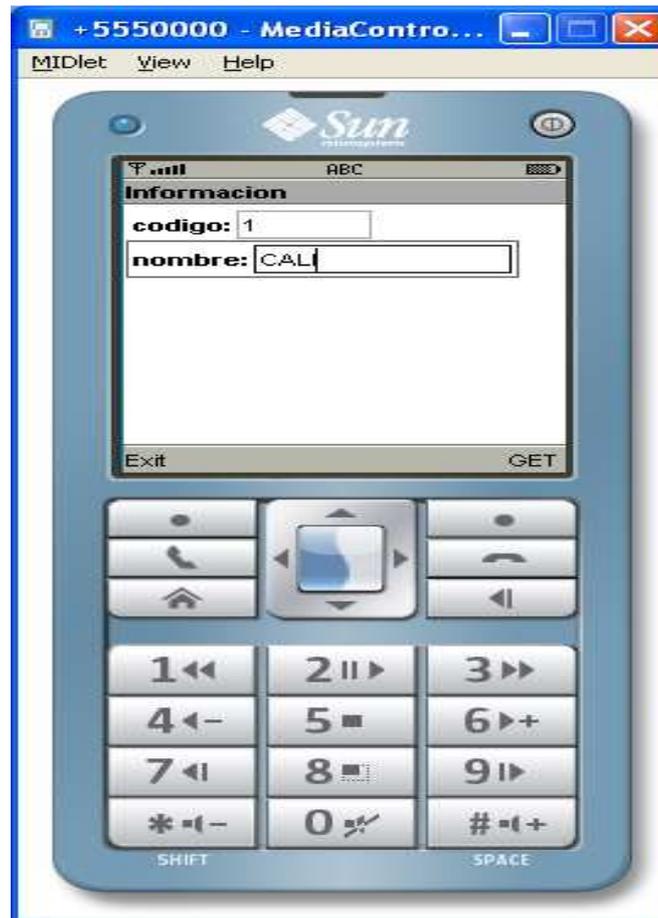
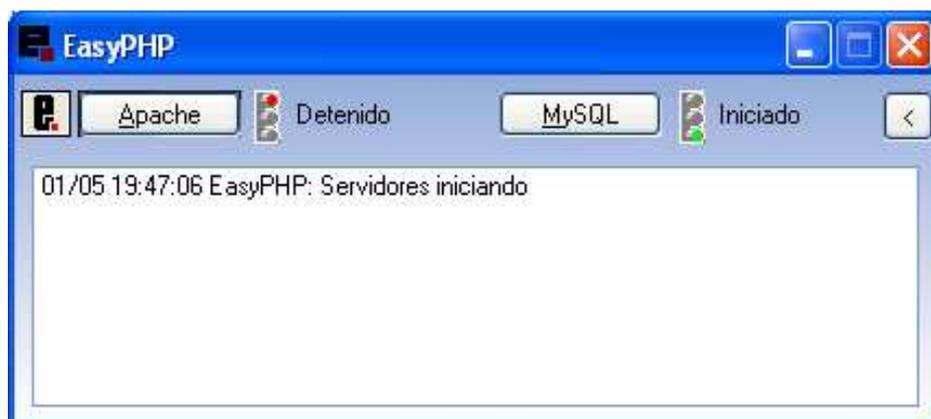


Figura No 6.6. Imagen con los datos a ser enviados al servidor por medio del protocolo Http.



## COMPUTACIÓN MÓVIL, Principios y técnicas

Figura No 6.7. Imagen del software EasyPhp, el cual lo conforman: Php, Servidor Apache y base de datos Mysql.

La imagen 6.8 visualiza el registro guardado en la base de datos de la aplicación central.

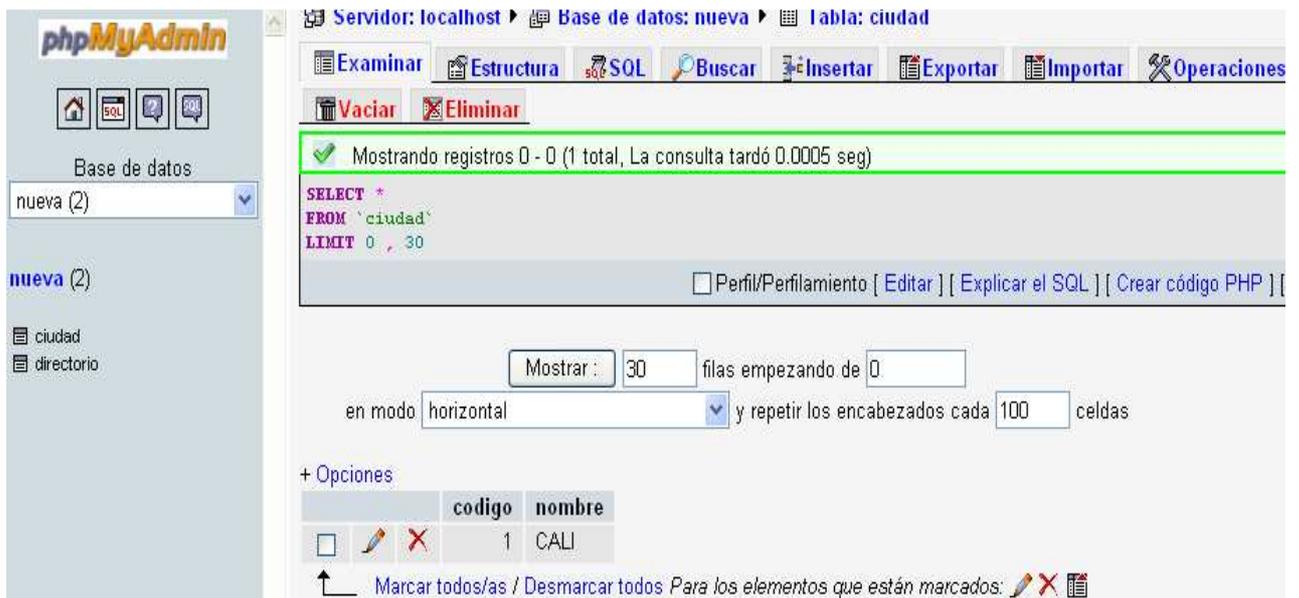


Figura No 6.8. Imagen donde se visualiza el registro en la base de datos enviado desde el dispositivo.

Se envía nuevamente otro registro, la base de datos presenta dos registros, los cuales fueron enviados desde el dispositivo. El protocolo Http permite realizar este proceso de forma fácil y rápida.

### ***Envío de datos desde la aplicación hacia el celular***

En el lenguaje Php, existe el comando denominado “echo” que permite visualizar información en la pagina Web de la aplicación por

## COMPUTACIÓN MÓVIL, Principios y técnicas

ejemplo, si se desea visualizar el mensaje “hola mundo” se escribe la línea de código.

**echo "hola mundo" ;**

La figura No 6.9 muestra el código fuente necesario para recibir los datos provenientes de la aplicación central.

```
public void envioGet()
{
    boolean ret = false;
    try{
        String url = "http://localhost/recibir.php?codigo=" +
            dato1.getString()+"&nombre=" + dato2.getString();
        HttpURLConnection hc = (HttpURLConnection)Connector.open(url);
        hc.setRequestMethod(HttpURLConnection.GET);
        InputStream in = hc.openDataInputStream();
        String cadena= "";
        int ch=0;
        for(int i=0;i<hc.getLength();i++)
        {
            ch= in.read();
            if(ch==-1){
                break;
            }
            cadena+= ((char)ch);
        }
        resultado.setString(cadena);
        hc.close();
    }catch(Exception e){
        e.printStackTrace();    //Muestra Toda la traza del error
    }
}
```

Se procesa la cadena que proviene del servidor

Cadena procesada

Figura No 6.9. Código escrito en el lenguaje J2me para recibir y procesar datos provenientes de la aplicación Web.

## COMPUTACIÓN MÓVIL, Principios y técnicas

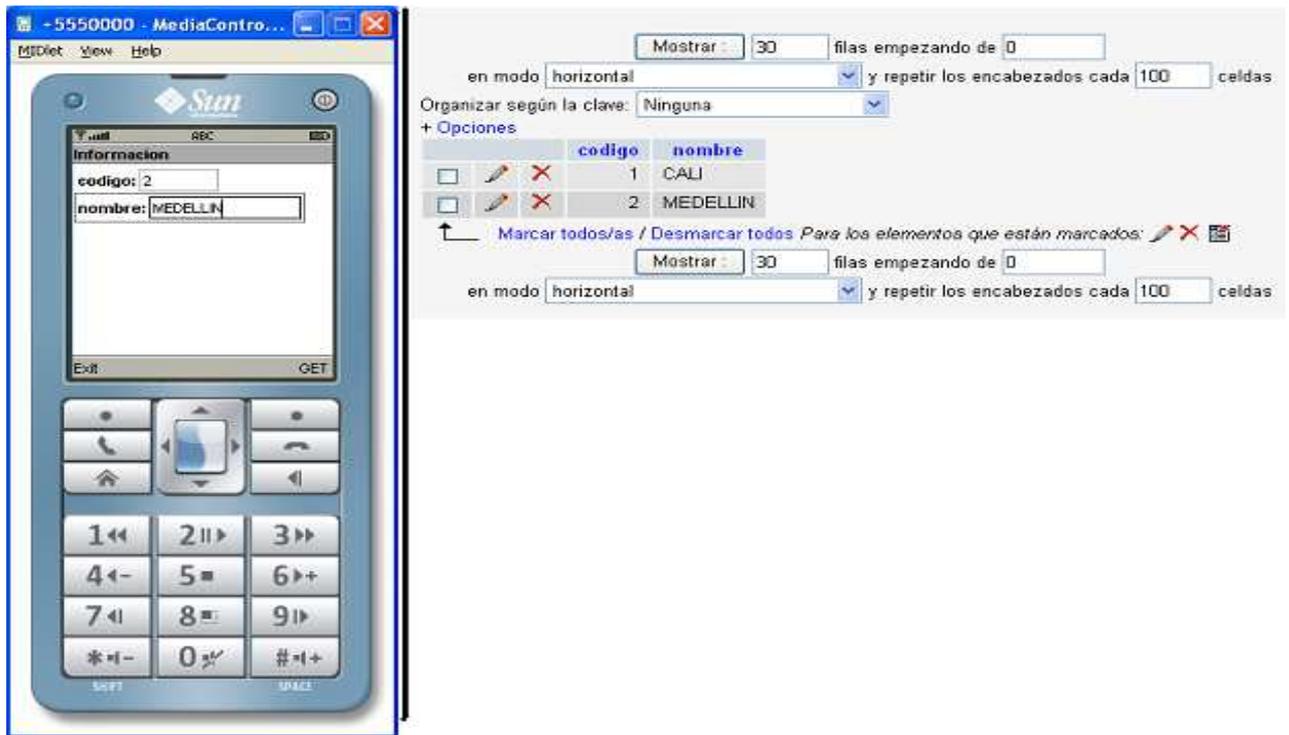


Figura No 6.10. Nuevo registro enviado al servidor y recibido por la aplicación.

El mismo proceso de envío de datos se puede realizar desde un formulario Web, es de afirmar que, en este caso, no se cambió una sola línea de código en el proceso que recibe los datos. La figura 5.10 ilustra los datos ha ser enviados desde un formulario Web, y los datos enviados y almacenados en la base de datos.

## COMPUTACIÓN MÓVIL, Principios y técnicas

Dirección

Código

Nombre

**Formulario**

**phpMyAdmin**

Organizar según la clave:

+ [Opciones](#)

			codigo	nombre
<input type="checkbox"/>			3	CARTAGENA
<input type="checkbox"/>			1	CALI
<input type="checkbox"/>			2	MEDELLIN

[↑](#) [Marcar todos/as](#) / [Desmarcar todos](#) Para los elementos que están marcados:   

### **Base de Datos**

Figura No 6.11. Proceso de envío de datos desde un formulario Web, se utiliza el mismo proceso en el servidor.

La figura 6.12 ilustra esta característica, un proceso en la aplicación central que es utilizado para trabajar de una manera diversa, con un cliente que envía datos desde un dispositivo móvil y otro que lo hace desde un formulario Web en una estación de trabajo.

## COMPUTACIÓN MÓVIL, Principios y técnicas

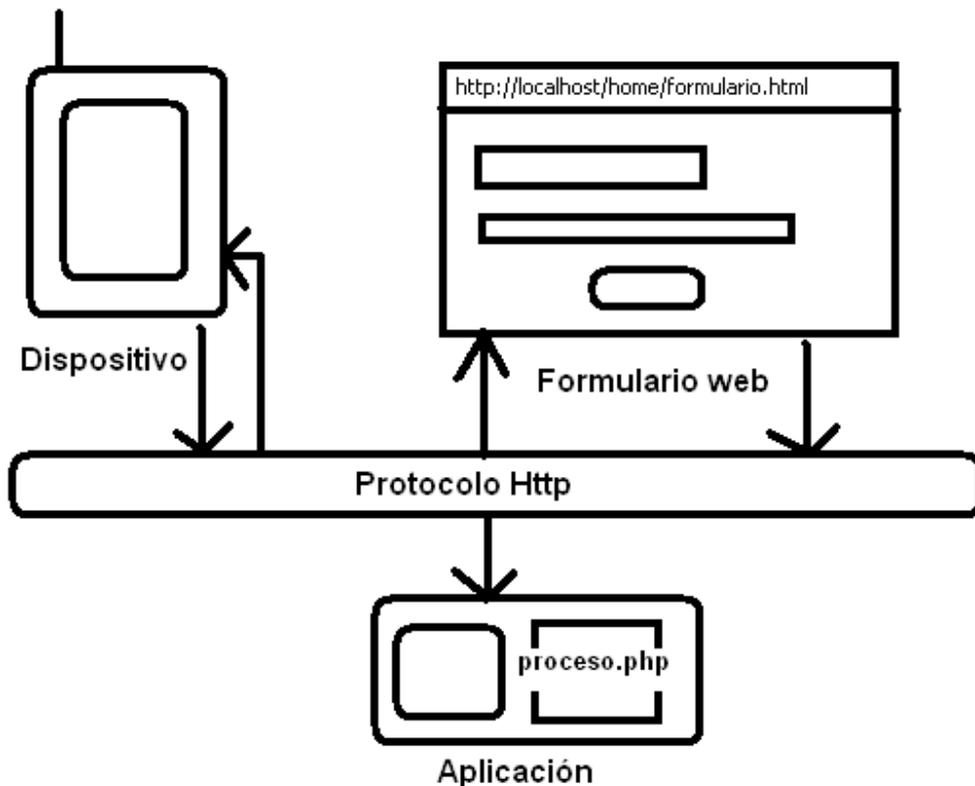


Figura No 6.12. El Proceso de envío se puede realizar desde un dispositivo o formulario Web sin cambiar una sola línea de código en el servidor.

Aquí se abordó la comunicación y envío de datos desde un dispositivo a una aplicación en el servidor, por medio del protocolo Http. Otra manera de hacerlo es por medio de Web Services<sup>16</sup>, en español, Servicios Web, donde los dispositivos y aplicaciones consumen o utilizan estos, los Servicios Web integran una serie de estándares y tecnologías como : Xml, Soap, Wsdl, seguridad, entre otros, que no

<sup>16</sup> <http://www.w3.org/TR/ws-arch/>

## COMPUTACIÓN MÓVIL, Principios y técnicas

se abordaran aquí, pero que se mencionan para que el lector pueda investigar sobre estos.

La utilización de Servicios Web, permite construir aplicaciones robustas, seguras e integrables a un variado número de aplicaciones; En otras palabras, la mayoría de aplicaciones profesionales están construidas con Web Services.

### ***Protocolo de Comunicación Bluetooth***

**Bluetooth** es un protocolo que permite la transmisión de voz entre dispositivos móviles. Este protocolo es inalámbrico, hasta el momento se considera de alcance reducido o baja cobertura. Este se basa en enlace por radio frecuencia, la especificación **Bluetooth 3.0** permitió avances significativos en las comunicaciones móviles, ahora se habla de la especificación 4.0 que consumirá poca energía y se podrá incluir en dispositivos móviles de bajo costo.

Un área donde es muy utilizada la comunicación por medio de **Bluetooth** es la Domótica que estudia la automatización de una vivienda por medio de diferentes sistemas y tecnología, entre ellas la tecnología móvil. En la Institución Universitaria Antonio José Camacho existe el grupo de investigación INTELLIGO el cual trabaja, entre otros, con proyectos sobre Domótica uno de estos es el trabajo de

## **COMPUTACIÓN MÓVIL, Principios y técnicas**

investigación denominado Bluelight. Sistema para el Control de dispositivos eléctricos por medio de Bluetooth. La revista Sapiencia de la Institución Universitaria Antonio José Camacho en su volumen 2 numero 7 publicó este artículo.

## **DISEÑO DE BASES DE DATOS**

**E**n la base de datos de la aplicación central se deben realizar ciertos cambios, preferiblemente, para la correcta integración de la aplicación central y el dispositivo móvil. Estos cambios en la base de datos se realizan por medio de la creación de campos adicionales, estos campos se refieren a:

- Dispositivo móvil que envió el registro
- Hora y fecha de envío del registro
- Usuario

Lo anterior se puede trabajar con un archivo de registro de transacciones del sistema, en otras palabras, el llamado “log” del sistema.

Se explicara el proceso de envío de datos con el ejemplo de una aplicación clásica de pedidos, en la cual los vendedores envían los datos desde sus dispositivos móviles a la aplicación central.

**COMPUTACIÓN MÓVIL, Principios y técnicas**

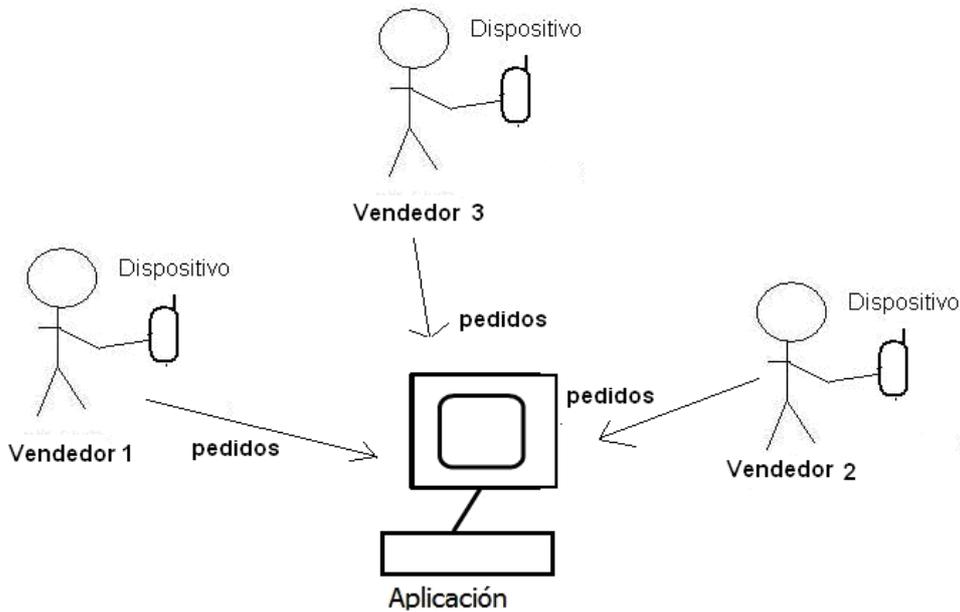


Figura No 7.1. Sincronización, intercambio de datos entre sistemas, por medio de protocolos de sincronización.

Un posible diseño para una base de datos podría ser la siguiente:

Tradicionalmente un diseño de pedido debe tener una tabla Maestra de pedido y una tabla Detalle. Aquí se abordara sólo la tabla Maestra de pedidos y algunos de sus campos.

Pedido						
Nopedido	codCliente	FechaHora	codCiudad	vendedor	NoDispositivo	FechaHoraS

Campos adicionales  
Dispositivo Móvil

Figura No 7.2. Tabla de pedidos propuesta, donde se indican los campos adicionales para el registro de las transacciones desde el dispositivo móvil.

## COMPUTACIÓN MÓVIL, Principios y técnicas

La figura 7.2 ilustra los campos adicionales que puede tener una tabla de pedidos para soportar el trabajo con dispositivos móviles, el primer campo “NoDispositivo”, se utiliza para establecer el dispositivo que envió el registro a la aplicación central. El segundo campo “FechaHoraS”, es utilizado para indicar la fecha y hora de haber sido almacenado el registro proveniente del dispositivo móvil.

El campo “FechaHora”, es utilizado para registrar la fecha y hora de la toma del pedido.

En los sistemas de conexión Desconectado y Semiconectado se pueden enviar los registros mucho después de haberse almacenado los datos en el dispositivo móvil.

En un sistema de conexión Conectado, los datos son guardados directamente a la aplicación central desde el dispositivo móvil;

No es necesario el almacenamiento de datos en el dispositivo. Por consiguiente los campos “FechaHoraS” y “FechaHora”, contienen los mismos datos y se puede prescindir de uno de ellos.

### ***Envío de datos de la aplicación central hacia el dispositivo.***

Supongamos que en el mismo sistema de pedidos se necesita enviar un registro nuevo, una referencia, a uno o varios vendedores. Se establece que se está trabajando sobre un sistema Conectado o Semiconectado.

## **COMPUTACIÓN MÓVIL, Principios y técnicas**

Para este efecto se puede contar con una tabla de sincronización en la aplicación central que facilite este proceso. En la figura 7.3 se ilustra esta tabla.

**Tabla Básica de Sincronización**

<b>campo1</b>	<b>campo2</b>	<b>campo3</b>	<b>tabla</b>	<b>paraDispositivo</b>	<b>TomoDispositivo</b>	<b>fechaHora</b>

Figura No 7.3. Tabla Básica de Sincronización.

Antes de explicar el funcionamiento de esta tabla, se debe mencionar que este proceso se puede realizar de diversas formas y cada una de estas formas tiene sus ventajas y desventajas. Adicionalmente, hasta el momento de crear este libro, este no ha sido un proceso muy divulgado. Entonces aquí se plantearán estos procesos.

En la tabla de sincronización se encuentran tres campos: campo1, campo2 y campo3; estos campos se utilizan, en este ejemplo, se pueden utilizar para almacenar la información que se necesita enviar al dispositivo como, código, nombre y precio de la nueva referencia.

El campo tabla se utilizará para indicar en que tabla o archivo del dispositivo se almacenará la información.

El campo paraDispositivo indica hacia que dispositivo se enviará el dato almacenado en campo1, campo2, campo3.

## COMPUTACIÓN MÓVIL, Principios y técnicas

El campo tomoDispositivo se utiliza para determinar que dispositivo deprecionó el dato, este dato debe coincidir con la información en el campo paraDispositivo.

Y el campo fechaHora es utilizado para establecer la fecha y la hora que el dato fue recibido por el dispositivo.

Tabla Básica de Sincronización

campo1	campo2	campo3	tabla	paraDispositivo	TomoDispositivo	fechaHora
2311	Producto1	2,000	Product	PDA1	PDA1	01/06/2010 3:15
2311	Producto1	2,000	Product	PDA2	PDA2	04/06/2010 1:25

Figura No 7.4. Tabla Básica de Sincronización con datos

La tabla de la figura 7.4 muestra la tabla con datos enviados hacia el dispositivo móvil.

Otra posible forma de enviar datos hacia un dispositivo móvil es la de tener la instrucción para ser ejecutada en el dispositivo. Para mostrar un ejemplo, se necesita insertar el código y el nombre en la tabla ciudad del dispositivo, para esto se puede utilizar la siguiente instrucción Sql:

```
“insert into ciudad (codigo,nombre) values ('10','Cali');”
```

La instrucción anterior es enviada al dispositivo y ejecutada en este.

La figura 7.5 visualiza la estructura de la tabla que contiene la instrucción

## COMPUTACIÓN MÓVIL, Principios y técnicas

Sincronizar		
instruccion	dispositivo	sincronizo
insert into ciudad (codigo,nombre) values ('10','Cali');	1	0
insert into ciudad (codigo,nombre) values ('10','Cali');	2	1

Figura No 7.5. Tabla Básica de Sincronización con instrucciones que serán ejecutadas en el dispositivo.

El campo “dispositivo” hace referencia hacia que dispositivo va el registro; el campo “sincronizo”, indica si el dispositivo ya se sincronizó con la aplicación central, en caso de que el dispositivo este sincronizado se coloca 1 en este campo, en caso contrario se encuentra un cero(0).

La anterior es una técnica que se puede utilizar cuando se esta trabajando con un modelo de conexión Semiconectado o Desconectado.

En realidad existen varias maneras de crear un sistema de sincronización para la aplicación central y los dispositivos móviles, aquí solamente se han mencionado algunas opciones que cuentan con ventajas y desventajas.

## **VIDEO JUEGOS Y MULTIMEDIA MÓVIL**

Los video juegos para dispositivos móviles tienen gran demanda. En la actualidad, se puede observar como los programas de televisión, en especial los programas de dibujos animados tienen como alternativa sitios en Internet donde se pueden bajar video juegos especiales para dispositivos móviles. La industria de los videojuegos y los dispositivos móviles está en crecimiento y sumado a esto, es un sector que genera millones de dólares; los dispositivos móviles y los video juegos para estos presentan una buena posibilidad para el emprendimiento.

En la creación de videojuegos para dispositivos móviles se debe considerar aspectos como el lenguaje de programación utilizado, el sistema operativo y las características del dispositivo.

Aquí vamos a plantear la creación de videojuegos con J2me y se presentaran algunas arquitecturas de videojuegos móviles en red.

Para la construcción de un videojuego se pueden utilizar las librerías gráficas de los lenguajes de programación, estas librerías contienen instrucciones para dibujar círculos, cuadrados, óvalos ,etc. También se pueden utilizar gráficos diseñados en programas especializados e incluirlos en los videojuegos. A continuación se muestra el juego del tres en línea, desarrollo por el estudiante Miguel García de la

## COMPUTACIÓN MÓVIL, Principios y técnicas

Institución Universitaria Antonio José Camacho, en el trabajo realizado como miembro del semillero de investigación Games.Net.

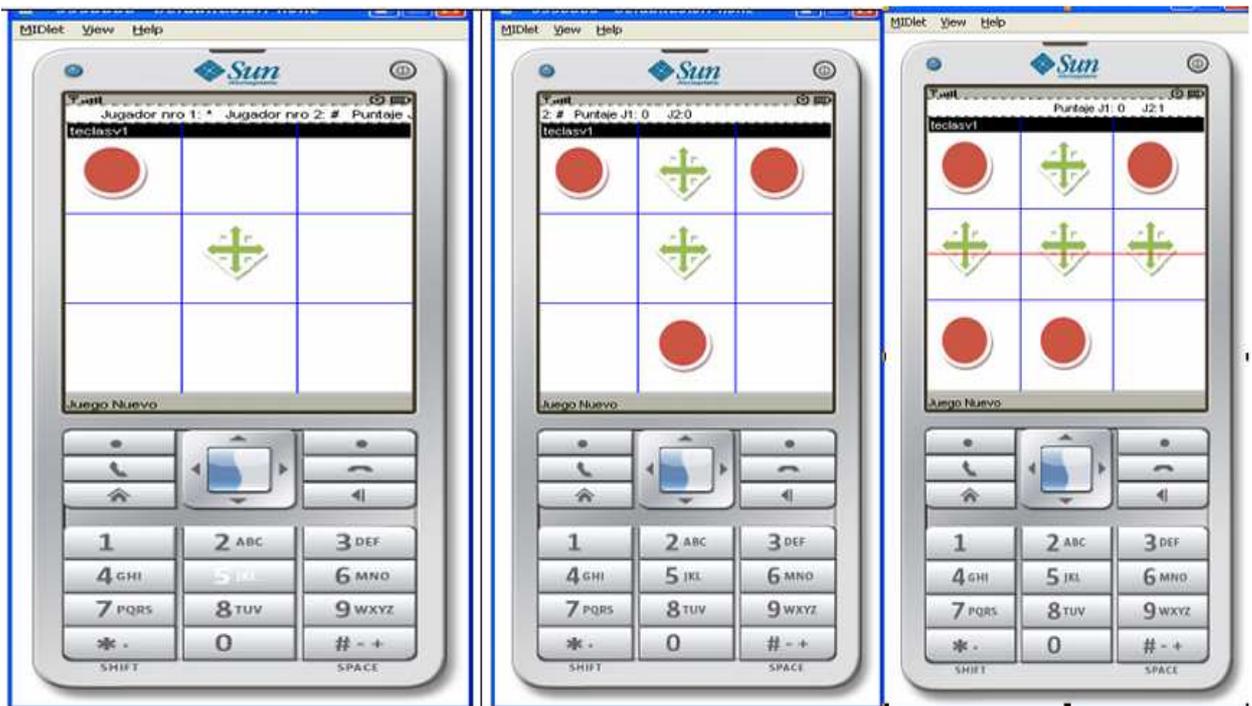


Figura No 8.1. Juego tres en línea, donde se combinan imágenes e instrucciones para dibujo de figuras geométricas.

La Figura No 8.2 muestra la estructura del juego para el dispositivo

## COMPUTACIÓN MÓVIL, Principios y técnicas

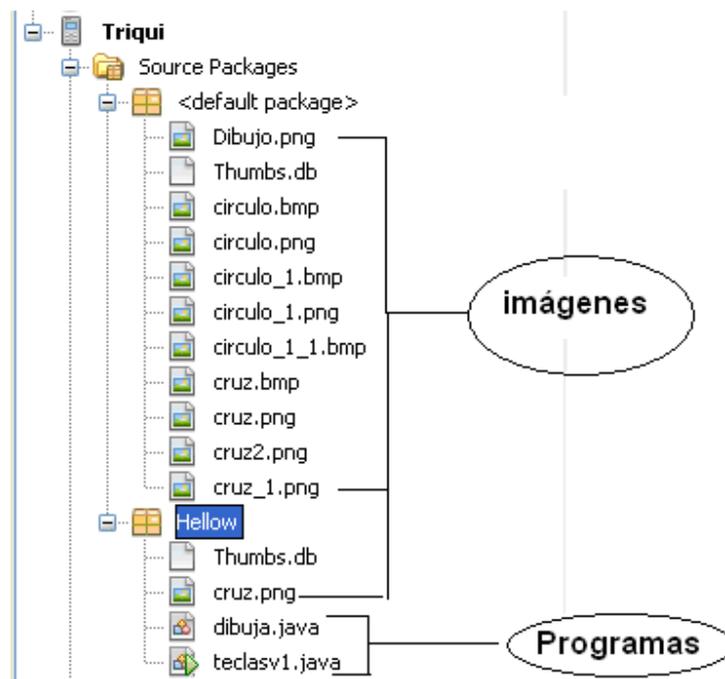


Figura No 8.2. Estructura del juego tres en línea

Este juego combina las imágenes creadas con instrucciones para el dibujo de figuras. La figura 8.3 muestra el dibujado de una línea.

```
g.setColor(255, 0, 0);  
g.drawLine(0, nuAlto/6, nuAncho, nuAlto/6);
```

Figura No 8.3. Instrucción para el Dibujado de una Línea

También se presentan las instrucciones para la definición de la variable que contendrá la imagen, la carga de la imagen y su posterior visualización. La figura No 8.4 visualiza la porción de código que realiza este proceso.

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
public static Image f=null;
protected void paint(Graphics g)
{
    try {
        f2 = Image.createImage("/cruz2.png");
    } catch (IOException ex) {
        ex.printStackTrace();
    }
    try {
        f = Image.createImage("/circulo.png");
    } catch (IOException ex) {
        ex.printStackTrace();
    }
    g.drawImage(f, 60, 50, 10);
}
```

Figura No 8.4. Fragmento de código para la definición, carga y visualización de una imagen.

El siguiente videojuego consiste en lo siguiente: personaje debe pasar una avenida, obviamente este debe cumplir el objetivo sin dejarse atropellar por un vehiculo.

Este videojuego tiene un diseño más elaborado que el anterior, pero trabaja con un conjunto de gráficas e imágenes, adicionalmente, cuenta con animación; los vehículos y las nubes se mueven autónomamente, el personaje principal el movido por el jugador. Ver figura No 8.5

## COMPUTACIÓN MÓVIL, Principios y técnicas



Figura No 8.5. Imágenes del videojuego donde el objetivo es pasar una avenida. Este videojuego presenta animación. Construido por el estudiante Miguel García.

## COMPUTACIÓN MÓVIL, Principios y técnicas

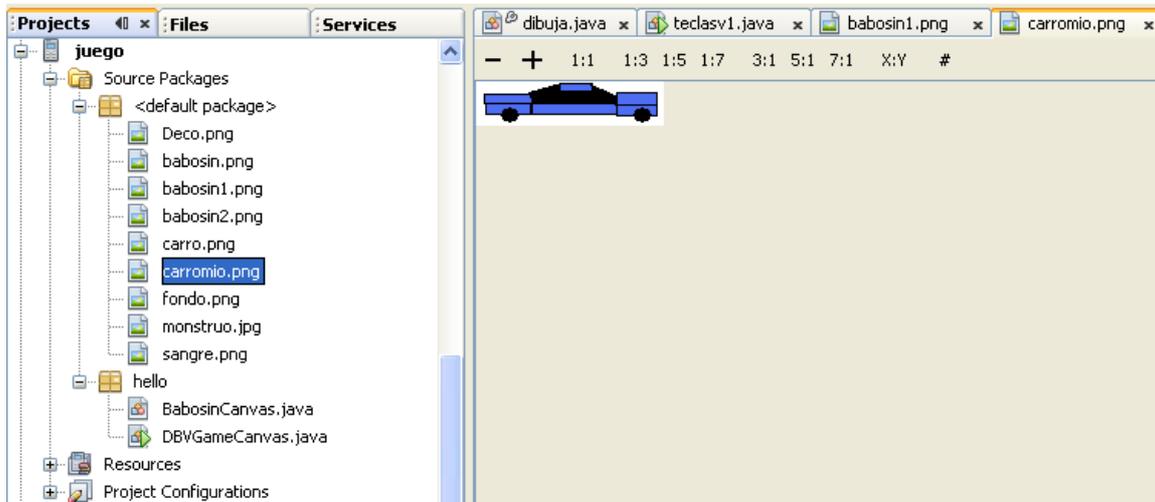


Figura No 8.6. Estructura del videojuego, lado izquierdo e imagen de un vehiculo, lado derecho.

El código fuente de este programa se presentara en el apéndice A. en la figura No 8.7 se visualiza un fragmento del código que dibuja el escenario del videojuego.

```
private void dibujaPantalla(Graphics g)
{
    // lineas montañas
    g.setColor(150,255,255);
    g.fillRect(0, 0, getWidth(), getHeight());
    g.setColor(105, 105, 105);
    g.drawLine(0,195,getWidth(),195);
    g.fillRect(0,230,getWidth(),20);
    g.drawLine(0,280,getWidth(),280);
    g.setColor(255,255,0);

    sangre.setPosition(sangrex, sangrey);
    sangre.paint(g);

    g.fillRoundRect(5, 10, 30, 30, 30, 30);
    g.setColor(0,205,0);
    g.fillTriangle(0, 150, 100, 0, getWidth(), 150);
    g.setColor(100, 50, 0);
    g.fillTriangle(7, 120, 45, 65, 70, 120);
    g.setColor(200, 100, 0);
    g.fillRect(10, 120, 60, 60);
    g.setColor(150, 75, 0);
    g.fillRect(31, 160, 15, 20);
    g.fillRect(15, 130, 15, 15);
    g.fillRect(50, 130, 15, 15);
    x1+=1;x2+=1;x3+=1;
    x4+=1;x5+=1;x6+=1;
    x7+=1;x8+=1;x9+=1;
}
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

Figura No 8.7. Fragmento de código fuente del videojuego para pasar una avenida.

Para actualizar la pantalla se utiliza la instrucción de J2me “**flushGraphics();**”, aquí se indica por ser de gran importancia para el efecto de animación en el videojuego.

### *Juegos en Red*

Los videojuegos anteriores se instalan en el dispositivo, no necesitan de una red para funcionar, a continuación se presenta el diseño general de un videojuego en Red.

Por medio de las funciones de red de los lenguajes de programación para dispositivos móviles se puede crear videojuegos donde interactúen varios jugadores por medio de una red inalámbrica.

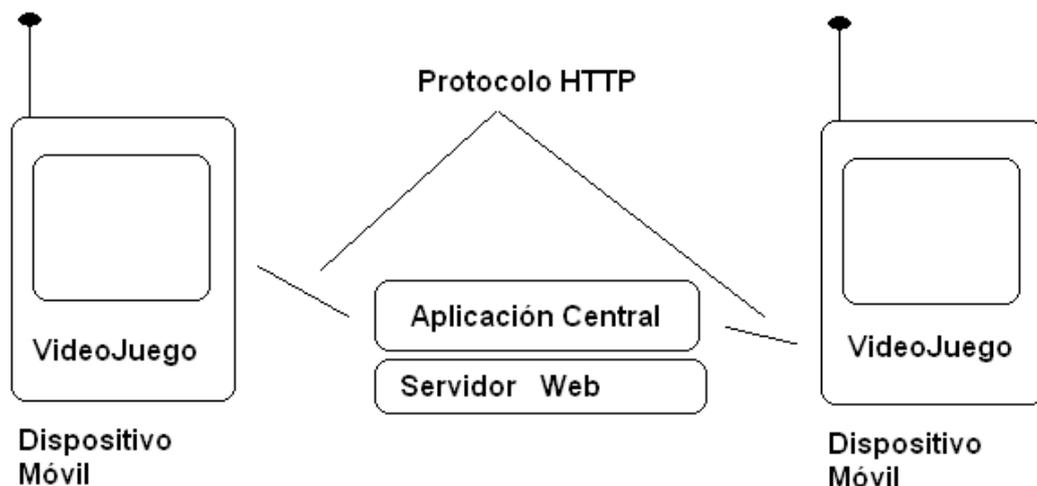


Figura No 8.8. Esquema para un videojuego en red utilizando el protocolo de comunicación Http.

## COMPUTACIÓN MÓVIL, Principios y técnicas

Se puede construir un videojuego en red como el tres en línea, donde cada jugador conoce por medio de la red el estado del juego; en este caso se enviaría por la red la posición de la jugada de cada jugador y en el dispositivo se plasmaría para cada jugador el nuevo estado del juego.

Un videojuego con este principio fue construido en una tesis de grado de la Institución Universitaria Antonio José Camacho. Esta tesis de grado fue dirigida por el autor en el semestre I-2010, se trataba del videojuego de batalla naval, construido por los estudiantes Marco Antonio Mayor y Viviana Varela.

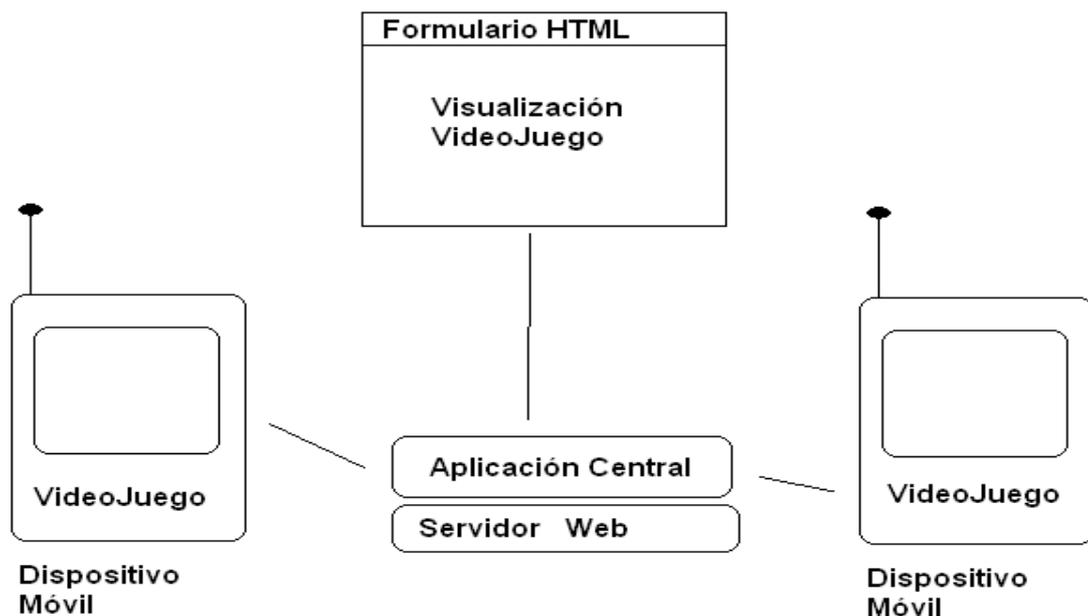


Figura No 8.9. Esquema para un videojuego en red utilizando el protocolo de comunicación Http y visualización por medio de una pagina Web.

## **COMPUTACIÓN MÓVIL, Principios y técnicas**

La figura 8.9 muestra un esquema de un videojuego donde se cuenta con una página Web que muestra el desarrollo del juego, cada jugador puede ver el desarrollo del juego y adicionalmente otros espectadores pueden ver este por medio de la página Web.

Se puede diseñar un variado tipo de esquemas de videojuegos para dispositivos móviles uno de ellos podría ser un videojuego en línea sin la intervención de una aplicación central este videojuego utilizaría la tecnología BlueTooth<sup>17</sup>

### ***Multimedia y Dispositivos móviles***

La multimedia hace parte de los videojuegos pero actualmente se pueden construir aplicaciones para dispositivos móviles, diferentes a videojuegos, ricas en multimedia. Este es el caso de software educativo, aplicaciones empresariales, software de simulación, entre otros.

En la actualidad se presentan cursos que aprovechan todo el poder multimedial de los dispositivos móviles. En Internet se encuentran páginas donde se puede encontrar todo tipo de aplicaciones y cursos para dispositivos móviles entre ellas tenemos:

- Todo para tu móvil gratis

---

<sup>17</sup> Tecnología Bluetooth <http://www.bluetooth.com/Spanish/Technology/Pages/default.aspx>

[http://www.bluezona.com/index.php?option=com\\_content&task=view&id=25&Itemid=50/](http://www.bluezona.com/index.php?option=com_content&task=view&id=25&Itemid=50/)

## **COMPUTACIÓN MÓVIL, Principios y técnicas**

<http://todoparatumovilgratis.blogspot.com/>

- Simplemente celulares

<http://simplementecelulares.blogspot.com/>

- Curso de Ingles para dispositivos móviles

<http://www.cosasdeeducacion.es/cursos-de-ingles-gratis/>

- Todo PocketPc

<http://www.todopocketpc.com>

El tema sobre Bluetooth se amplia en el capítulo 6, protocolos de comunicación y envío de datos.

## **ALMACENAMIENTO**

---

**E**l almacenamiento en los dispositivos móviles depende en gran medida de la familia de dispositivo; esta familia está regida por un sistema operativo, lenguaje o lenguajes de programación y un sistema de almacenamiento de archivos.

En dispositivos móviles de la familia Android <sup>18</sup>, se tiene la base de datos SQLite<sup>19</sup>, este es un sistema de gestión de datos relacional, que contiene las características necesarias transaccionales. Esta es compuesta, entre otros, por tablas, índices, definiciones, datos. SQLite está bajo licencia **Copyright**, que significa que ésta contribución y el código fuente son de dominio público.

### ***Record Management System***

Aquí se abordará el almacenamiento con J2ME, este almacenamiento es más primitivo y se basa en registros. El almacenamiento con j2me se denomina RMS (Record Management System).

---

<sup>18</sup> Sitio oficial sistema operativo para dispositivos móviles familia **Android** <http://www.android.com/>

º2345456j

<sup>19</sup> Sitio oficial SQLite <http://www.sqlite.org/>

## COMPUTACIÓN MÓVIL, Principios y técnicas

Se parte de una zona de almacenamiento; una zona de almacenamiento es un espacio especial en la memoria del dispositivo; en esta zona cada registro almacenado en el soporte de registros tiene un identificador único o “record id”, ver figura 9.1

recordid	datos
1	Victor Viera
2	Marta Balanta
3	Beatriz Duarte

Figura No 9.1 Números únicos (recordid) de registros y datos

J2me contiene ésto funcionalidad en el paquete RMS, específicamente en **javax.microedition.rms**.

El siguiente código muestra el programa completo de manejo de registros por medio de Rms.

Paquete para manejo de registros

```
import javax.microedition.rms.*;
```

Declaración de registros

```
private RecordStore reg=null
```

Declaración del nombre de la zona de almacenamiento

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
static final String REC_STORE="Zonal"
```

Los siguientes fragmentos representan una idea básica de las diferentes operaciones que se realizan en el sistema de almacenamiento, no se trata de su explicación a fondo para esto existen otros libros que cumplen este objetivo.

### Ejemplo de método para abrir la zona de almacenamiento

```
public void abrirRegistros()  
{  
    try {  
        reg = RecordStore.openRecordStore(REC_STORE,  
true);  
    } catch (Exception e) {  
        System.out.println(e);  
    }  
}
```

### Cerrar la zona de almacenamiento

```
public void cerrarRegistros() {  
    try {  
        reg.closeRecordStore();  
    } catch (Exception e) {  
        System.out.println(e);  
    }  
}
```

### Guardar un registro en la zona de almacenamiento

```
public void guardarRegistro(String cadena)  
{  
  
    byte[] registro = cadena.getBytes();
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
        try
        {
            record.addRecord(registro, 0, registro.length);
        }
        catch(Exception e)
        {
        }
    }
```

### Visualizar registros

```
public void Visual(){
try{
    if(reg.getNumRecords(>0){

        Buscador visual=new Buscador(numero.getString());
        RecordEnumeration re=reg.enumerateRecords(visual,
null, false);

        if (re.numRecords(>0){
            Resultado.setText("Registros " + new String
(re.nextRecord()));
        }

        re.destroy();
    }
}catch (Exception e){}
}
```

Borrado de un registro, se pasa el índice del registro representado, en este ejemplo en dato

```
public void BorrarDatos(int dato){
try{
    reg.deleteRecord(dato);

}
catch (RecordStoreException e){}
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

}

Existen videojuegos para dispositivos construidos con J2ME que utilizan el almacenamiento rms ya sea temporal o permanentemente.

En la actualidad el avance en las tecnologías, los lenguajes de programación para dispositivos móviles permiten contar con bases de datos que facilitan el proceso de almacenamiento de información en un dispositivo.

## **ACTUALIDAD Y TENDENCIAS**

**E**n el momento la gente del común, y los usuarios comienzan a percibir los beneficios que trae la computación móvil; en términos más amplios, la ubicuidad. Es así como algunos bancos envían mensajes a los celulares donde se especifica los movimientos realizados en sus cuentas, por ejemplo este es el formato que contiene un mensaje enviado al dispositivo; el mensaje indica el pago de nómina consignado en la cuenta del usuario.

**“(Banco)14/07/2010 13:21 PAGO NOMINA CON ABONO EN TU CTA POR \$ 5,000 TU SALDO ES \$6,000 Remite 221114 Centro de mensajes +22222 enviado 14-jul-2010 01:26:52 pm”**

Esta información es enviada al dispositivo móvil del usuario y es preciso mencionar que es de gran utilidad contar con esta información oportunamente.

Otro mensaje importante es el donde se indica el retiro de la cuenta del usuario.

**“(Banco)17/07/2010 15:30 RETIRO DE TU CTA 86811 POR \$ 2,000 EN CAJERO ATT TU SALDO ES \$ 4,000 Remite 2288 centro de mensajes ++22222 enviado: 5-jul-2010 12:0627 pm”.**

## **COMPUTACIÓN MÓVIL, Principios y técnicas**

Tener a disposición este tipo de información en cualquier lugar es verdaderamente importante, tanto por seguridad como por control de los movimientos realizados en la cuenta.

Otro servicio es el llamado Banca Móvil en el cual se pueden realizar una serie de transacciones por medio del dispositivo móvil; estas transacciones desde el dispositivo facilitan realizar giros, recargar minutos, pagar cuentas de servicios públicos<sup>20</sup>.

En algunas empresas ya está incluida la computación móvil en diferentes procesos, para nombrar solo un ejemplo, las empresas de apuestas tienen equipados a sus vendedores con dispositivos móviles.

En lo que concierne al ámbito académico, las Universidades, en Colombia, incluyen ahora asignaturas relacionadas con la computación móvil; ya sea como electivas o base del programa. Las Universidades anticipatorias tienen grupos de investigación en el área; incluyendo grupos en Colciencias<sup>21</sup>.

El auge de las redes inalámbricas y los nuevos dispositivos, a precios bajos, con capacidades de conexión a estas, proponen un buen terreno para el crecimiento y acogida de esta “nueva” tecnología.

Ya empiezan a aparecer los primeros cursos multimedia para dispositivos móviles, donde el usuario puede ver, escuchar, leer en cualquier lugar.

---

<sup>20</sup> Referencia: diario el País, en su edición de 16 de agosto de 2010

<sup>21</sup> <http://www.colciencias.gov.co/web/guest/home>

## **COMPUTACIÓN MÓVIL, Principios y técnicas**

Otra área que tiene participación en la computación móvil es la del entretenimiento, especialmente los videojuegos; inclusive, muchos dibujos animados ya cuentan con su respectivo vídeo juego para dispositivo móvil.

La misma red de redes Internet contiene muchas páginas especialmente diseñadas para ser visualizadas correctamente en dispositivos móviles.

Ahora se mencionaran las diferentes manifestaciones de la computación ubicua.

### ***Computación Wearable***

La **Computación Wearable** o para tratar de hacer una traducción computación “Usable”, se refiere a los artefactos computacionales que se llevan literalmente puestos; a diferencia de los dispositivos móviles como los teléfonos inteligentes con los cuales interactuamos principalmente con las manos, los dispositivos Wearable se integran mucho mejor con el hombre y siempre están listos para usarse. Como ejemplo se puede mencionar prendas inteligentes que responden adecuadamente ante el frío, calor o signos vitales del usuario. Otros artefactos son las cámaras montadas en un casco que posibilitan la transición de imágenes y sonidos hacia una computadora.

### ***Realidad Aumentada***

Esta es una tecnología que se utiliza para apoyar el proceso de comprensión del entorno, el proceso de interpretación, en otras

## **COMPUTACIÓN MÓVIL, Principios y técnicas**

palabras agrega a la percepción del entorno elementos virtuales, creando con esto una realidad conjunta, a diferencia de la realidad virtual, ésta no sustituye el entorno, por el contrario lo complementa.

La realidad Aumentada incluye el uso de pantallas, computación gráfica que permite colocar textos, imágenes, sonidos sobre las imágenes captadas por la cámara o dispositivos. La Realidad Aumentada tiene aplicabilidad en diferentes campos, Medicina, entretenimiento, publicidad, y educación.

Existen otros nombres y tecnologías pero en esencia se trata de dispositivos inalámbricos que se utilizan o utilizarán para asistir procesos de nuestro que hacer diario.

### *Tendencias*

La computación móvil será una parte importante de nuestro acontecer diario. Se tendrá mayor número de redes inalámbricas con velocidades asombrosas, dispositivos más pequeños y livianos a menor precio.

Proliferarán los cursos de todo tipo diseñados especialmente para dispositivos móviles.

Los programas de televisión, la publicidad y el Internet se volcará hacia el área móvil.

Y siendo menos conservadores se puede asegurar que las libretas de apuntes y los mismos cuadernos desaparecerán; las personas realizarán sus anotaciones en los dispositivos. En los colegios,

## **COMPUTACIÓN MÓVIL, Principios y técnicas**

universidades se tendrán bases de datos y aplicaciones que soportaran muchos procesos por medio de dispositivos móviles.

Las personas serán asistidas por los dispositivos, para esto se crearan aplicaciones inteligentes, la Inteligencia Artificial, los sistemas distribuidos y los agentes móviles tendrán su oportunidad.

Tampoco se debe contar con una bola de cristal para mencionar que los dispositivos no tendrán la apariencia que tiene hoy, o que nuestro dispositivo nos reconocerá por medio de nuestro ADN (Ácido Desoxirribonucleico). Los dispositivos móviles podrán estar dentro o fuera de nuestro cuerpo.

Nuestros dispositivos contarán con muchas funcionalidades para interactuar con los aparatos de nuestro hogar, como la nevera, la calefacción y nuestro robot asistente.

Para lograr todo esto, las grandes compañías le seguirán apostando a los dispositivos móviles, produciendo hardware y software más eficientes y económicos. Estas grandes compañías se dividirán cada vez más y seguirán compitiendo por los primeros puestos del mercado. Las compañías estarán divididas pero los estándares predominaran.

## CÓDIGO FUENTE HOLA MUNDO J2ME

**E**l código fuente que se muestra a continuación es el ejemplo básico del programa “holamundo.java”

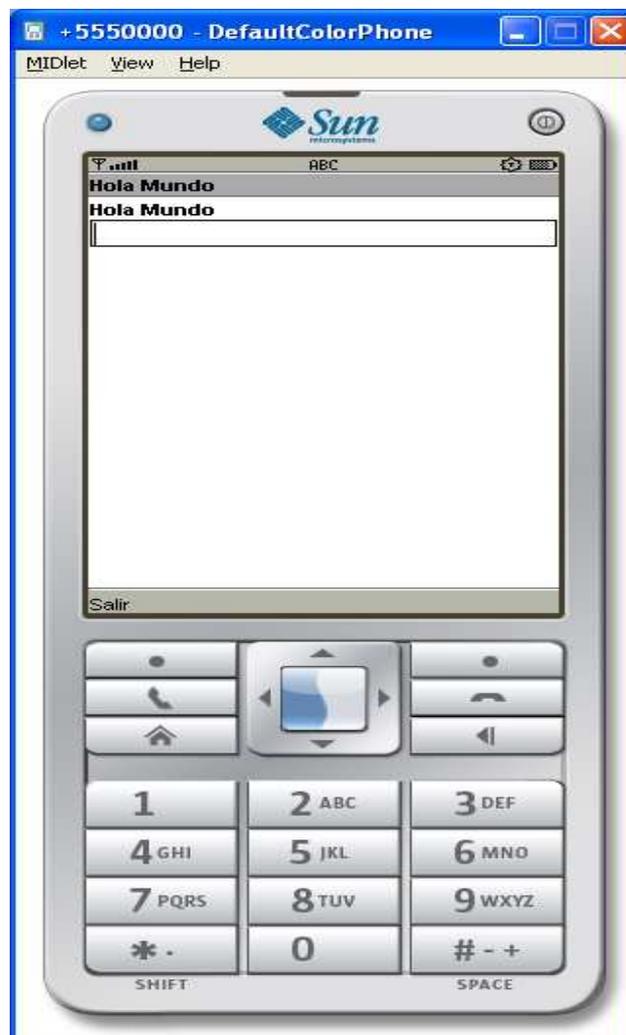


Figura No A.1. Programa HolaMundo.java

## COMPUTACIÓN MÓVIL, Principios y técnicas

**Código Fuente los comentarios se inician con “//”**

### **holamundo.java**

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
/** @author Victor Viera** */
public class holamundo extends MIDlet implements CommandListener {
    //variable de visualizacion
    public Display mostrar;
    //Ventana del Formulario
    public Form ventana;
    //Boton de Salir
    public Command salir;
    //Campo de TExto
    public TextField Campo;
    //Construtor
    public void holamundo() {
    }
    //inicio
    public void startApp() {
        //creación de la ventana
        ventana = new Form("Hola Mundo");
        //creación del campo de texto
        Campo = new TextField("Hola Mundo", "", 20, TextField.ANY);
    }
}
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
//creación del boton salir
salir = new Command("Salir", Command.EXIT,1);

//variable de visualizacon
mostrar = Display.getDisplay(this);

//se adicionan los objetos creados a la ventana
ventana.append(Campo);
ventana.addCommand(salir);

//escucha o atiende los mensajes
//y eventos de la ventana
ventana.setCommandListener(this);

//variable de visualizacion y formulario
mostrar.setCurrent(ventana);
}

public void pauseApp() {
}

public void destroyApp(boolean unconditional) {
}

//si el usuario presiona el boton de salir
public void commandAction(Command c, Displayable s) {
    if(c==salir) {
        destroyApp(false);
        notifyDestroyed();
    }
}
}
```

## CÓDIGO FUENTE VIDEOJUEGO CON ANIMACIÓN

**E**l código fuente que se muestra a continuación tiene instrucciones de dibujo de figuras geométricas, animación, visualización de imágenes, control y movimiento del personaje principal por parte del usuario. Este videojuego fue construido por el estudiante Miguel García en el semillero de Investigación Games.Net de la Institución Universitaria Antonio José Camacho.

La figura B.1 visualiza la estructura del juego, el código fuente del videojuego lo contienen los archivos “**BabosinCanvas.java**” y “**DBVGameCanvas.Java**”.

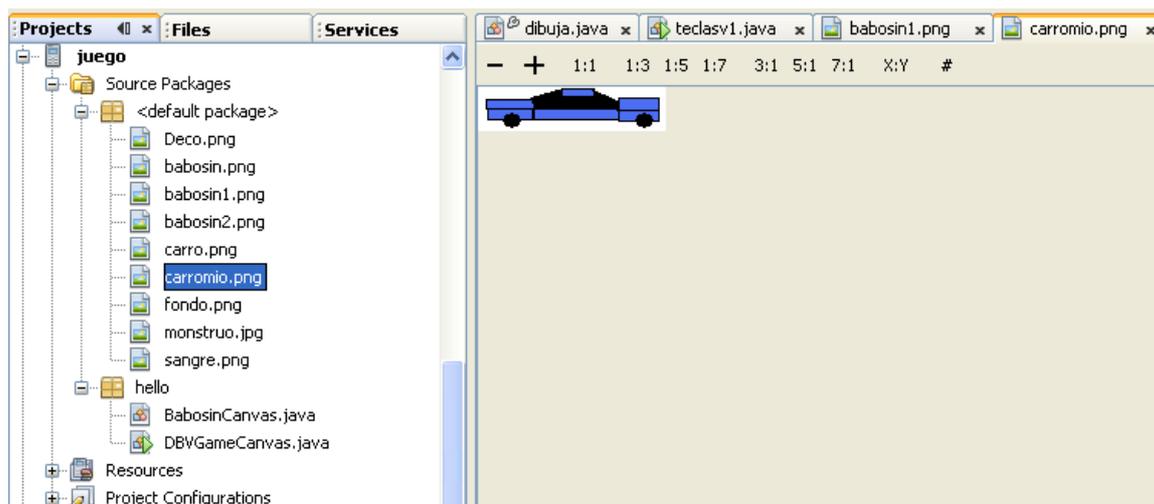


Figura No B.1. Estructura del videojuego, lado izquierdo e imagen de

## COMPUTACIÓN MÓVIL, Principios y técnicas



Figura No B.2. Imágenes del videojuego donde el objetivo es pasar una avenida.

**Código Fuente** los comentarios se inician con “//”

## COMPUTACIÓN MÓVIL, Principios y técnicas

### Archivo BabosinCanvas.java

```
package hello;

//Autor Miguel Garcia
//Estudiante Institucion Universitaria Antonio Jose Camacho
import javax.microedition.lcdui.*;
import javax.microedition.lcdui.game.*;

public class BabosinCanvas extends GameCanvas implements Runnable{
private boolean jugando; // Indica que la partida está en curso
private long retraso; // Retraso entre cada ciclo

private int babosinX,
babosinY,carromiox,carromioy,carrox,carroy,sangrex,sangrey;
// Coordinadas del personaje

private int ancho; // Ancho de la pantalla
private int alto; // Alto de la pantalla
private int posbabosin; // babosin actual del personaje (0-4)
private Sprite babosin,carro,carromio,sangre; // Sprites del personaje
public static Image image,image2,image3,image4;
int x1 =10,x2=25,x3=40,x4=150,x5=165,x6=180,x7=180,x8=195,x9=210;
public BabosinCanvas() throws Exception {
super(true);
ancho = getWidth();
alto = getHeight();
babosinX = ancho/2;
babosinY = alto / 2;
carrox=120;
carroy=250;
carromiox=120;
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
carromioy=198;
sangrex=10;sangrey=15;
retraso = 20;
// Carga el personaje principal
image = Image.createImage("/babosin2.png");
babosin = new Sprite (image);
image2 = Image.createImage("/carro.png");
carro = new Sprite (image2);
image3 = Image.createImage("/sangre.png");
sangre = new Sprite (image3);
image4 = Image.createImage("/carromio.png");
carromio = new Sprite (image4);

}
// Crea y arranca la hebra del juego
public void start() {
jugando = true;
Thread t = new Thread(this);
t.start();
}
public void stop() { jugando = false; }
// Bucle principal del juego
public void run() {
Graphics g = getGraphics();
while (jugando == true) {
leeTeclado(); // Lee acciones del usuario
dibujaPantalla(g); // Dibuja la escena actual
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
try {
Thread.sleep(retraso);
}catch (InterruptedException ie) {}
}
}
// Método para gestionar el teclado
private void leeTeclado() {
try{
int keyStates = getKeyStates();
if ((keyStates & LEFT_PRESSED) != 0) { // Izquierda
if(babosinX>=31 && babosinX<=34 && babosinY==162)
babosin.setVisible(false);
if( babosinY>=102 &&babosinY<=161 && babosinX==70)
System.out.println("nooooooooooooooooooooooooooooo");
else{
System.out.println("x: "+babosinX+" "+"Y: " +babosinY );
babosinX = Math.max(0, babosinX - 1);
posbabosin=(posbabosin+1)%5;
babosin.setFrame(posbabosin);
babosin.setTransform(Sprite.TRANS_NONE);
}
}
if ((keyStates & RIGHT_PRESSED) !=0 ) { // Derecha
if(babosinX>=31 && babosinX<=34 && babosinY==162)
babosin.setVisible(false);
if( babosinY>=102 &&babosinY<=161 && babosinX==0)
System.out.println("nooooooooooooooooooooooooooooo");
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
else{
System.out.println("x: "+babosinX+" "+"Y: " +babosinY );
babosinX = Math.min(ancho-16, babosinX + 1);
posbabosin=(posbabosin+1)%5;
babosin setFrame(posbabosin);
babosin.setTransform(Sprite.TRANS_MIRROR);
}

}if ((keyStates & UP_PRESSED) != 0) { // Arriba
if(babosinX>=31 && babosinX<=34 && babosinY==162)
babosin.setVisible(false);
if( babosinX>=8 && babosinX<=60 && babosinY==160)
System.out.println("nooooooooooooooooooooooooooooo");
else{
System.out.println("x: "+babosinX+" "+"Y: " +babosinY );
babosinY = Math.max(0, babosinY - 1);
babosin setFrame(0);
}
}

if ((keyStates & DOWN_PRESSED) !=0) { // Abajo
if(babosinX>=31 && babosinX<=34 && babosinY==162)
babosin.setVisible(false);
else{
babosin.setVisible(true);}
System.out.println("x: "+babosinX+" "+"Y: " +babosinY );
babosinY = Math.min(alto-24,babosinY + 1);
babosin setFrame(4);
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
}  
}catch(Exception e){  
  
    System.out.println("Error al leer del teclado");  
}  
  
}  
// Dibuja la pantalla  
private void dibujaPantalla(Graphics g) {  
// lineas montañas  
g.setColor(150,255,255);  
g.fillRect(0, 0, getWidth(), getHeight());  
g.setColor(105, 105 ,105);  
g.drawLine(0,195,getWidth(),195);  
g.fillRect(0,230,getWidth(),20);  
g.drawLine(0,280,getWidth(),280);  
g.setColor(255,255,0);  
sangre.setPosition(sangrex,sangrey);  
sangre.paint(g);  
g.fillRoundRect(5, 10, 30, 30, 30, 30);  
g.setColor(0,205,0);  
g.fillTriangle(0, 150, 100, 0, getWidth(), 150);  
g.setColor(100, 50, 0);  
g.fillTriangle(7, 120, 45, 65, 70, 120);  
g.setColor(200, 100, 0);  
g.fillRect(10, 120, 60, 60);  
g.setColor(150, 75, 0);
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
g.fillRect(31, 160, 15, 20);
g.fillRect(15, 130, 15, 15);
g.fillRect(50, 130, 15, 15);
x1+=1;x2+=1;x3+=1;
x4+=1;x5+=1;x6+=1;
x7+=1;x8+=1;x9+=1;
if(x1>getWidth()){
x1=x2-15;x2=x3-15;x3=0;
}
if(x4>getWidth()){
x4=x5-15;x5=x6-15;x6=0;
}
if(x7>getWidth()){
x7=x8-15;x8=x9-15;x9=0;
}

g.setColor(0,255,255);
g.fillRoundRect(x1, 45, 20, 20, 30, 30);
g.fillRoundRect(x2, 45, 20, 20, 30, 30);
g.fillRoundRect(x3, 45, 20, 20, 30, 30);
g.fillRoundRect(x4, 10, 20, 20, 30, 30);
g.fillRoundRect(x5, 10, 20, 20, 30, 30);
g.fillRoundRect(x6, 10, 20, 20, 30, 30);
g.fillRoundRect(x7, 50, 20, 20, 30, 30);
g.fillRoundRect(x8, 50, 20, 20, 30, 30);
g.fillRoundRect(x9, 50, 20, 20, 30, 30);
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
carromiox-=2;
if(carromiox<-115){
    carromiox=getWidth();
}
carrox+=3;
if(carrox>=getWidth()){
    carrox=0;
}
carromio.setPosition(carromiox,carromioy);
carromio.paint(g);
carro.setPosition(carrox,carroy);
carro.paint(g);
babosin.setPosition(babosinX,babosinY);
babosin.paint(g);

if(babosin.collidesWith(carro, true)){
    sangrex=babosinX+10;
    sangrey=babosinY+10;
    babosin.setVisible(false);
}
if(babosin.collidesWith(carromio, true)){
    sangrex=carromiox+10;
    sangrey=carromioy+10;
    babosin.setVisible(false);
}
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
flushGraphics(); // actualiza la pantalla
}
}
```

### **Archivo DBVGameCanvas.java**

```
package hello;
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
public class DBVGameCanvas extends MIDlet implements CommandListener {
private Display mostrar;
private BabosinCanvas babosinCanvas;
private Command salir;
public DBVGameCanvas() {
mostrar = Display.getDisplay(this);
try{
babosinCanvas= new BabosinCanvas();
}catch(Exception e){
System.out.println("nada");
}
babosinCanvas.start(); // Inicia la hebra del canvas
salir = new Command("Salir", Command.EXIT, 0); // Boton para salir
babosinCanvas.addCommand(salir);
babosinCanvas.setCommandListener(this);
}
public void startApp() {mostrar.setCurrent(babosinCanvas);
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
}  
public void pauseApp() { }  
public void destroyApp(boolean unconditional) {  
    if (babosinCanvas != null) {  
        babosinCanvas.stop(); // Para la hebra del canvas  
    }  
}  
public void commandAction(Command c, Displayable s) {  
    if (c.getCommandType() == Command.EXIT) {  
  
        //System.gc(); // Llama al recolector de basura  
        //destroyApp(true);  
        //notifyDestroyed();  
    }  
}  
}
```

## CÓDIGO FUENTE VIDEOJUEGO TRES EN LINEA

**E**l código fuente de este videojuego presenta la combinación de instrucciones para dibujar figuras geométricas y la visualización de imágenes; Fue diseñado por el estudiante Miguel García del semillero de investigación Games.Net de la Institución Universitaria Antonio José Camacho.

La figura C.1 visualiza la estructura del videojuego. El código fuente del videojuego lo contienen los archivos “**dibuja.java**” y “**teclasv1.Java**”.

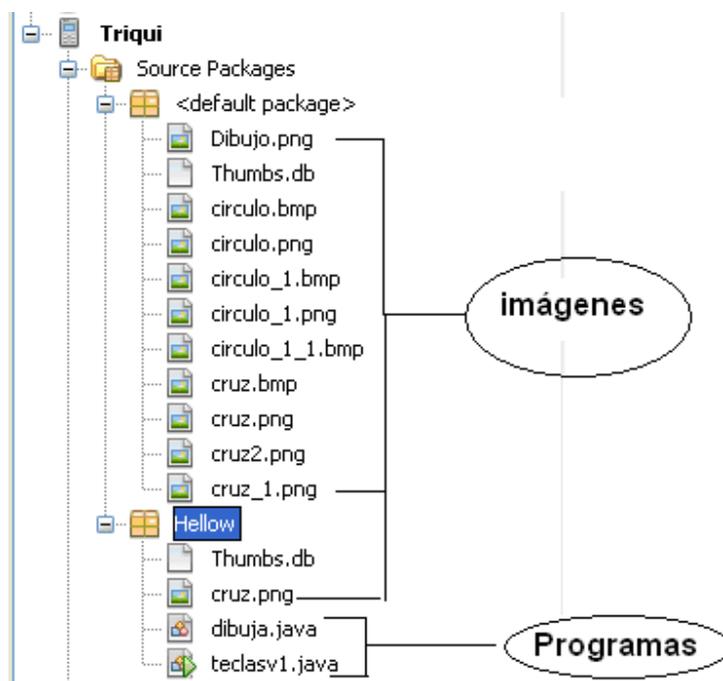


Figura No C.1. Estructura del juego tres en línea

## COMPUTACIÓN MÓVIL, Principios y técnicas

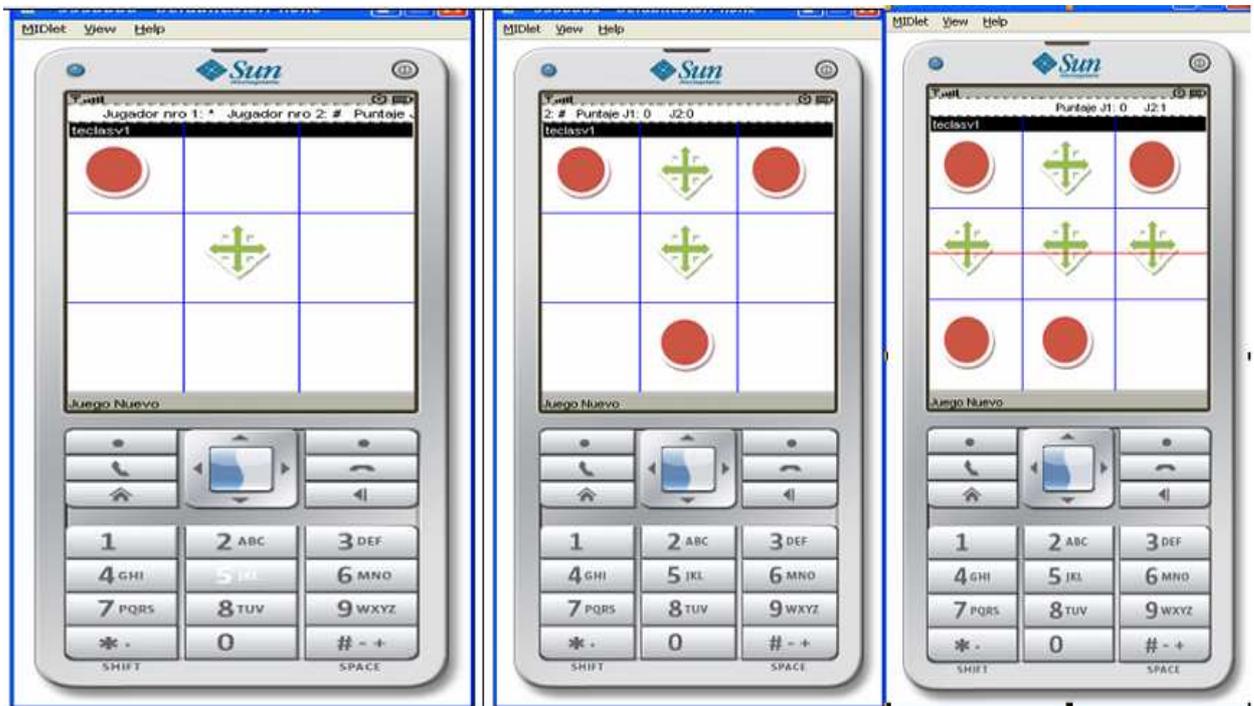


Figura No C.2. Imágenes del videojuego tres en línea

**Código Fuente** los comentarios se inician con “//”

**Archivo** `dibuja.java`

```
package Hellow;  
  
import java.io.IOException;  
import javax.microedition.lcdui.*;  
  
public class dibuja extends Canvas implements CommandListener{  
    private teclasv1 mid;  
  
    private Command juegoNuevo;  
  
    private Ticker mensaje;  
  
    int nuAlto,nuAncho,nuDAlto,nuDAncho;
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
boolean c1,c2,c3,c4,c5,c6,c7,c8,c9;//Cuadrantes
public static int turnoJugador=0;
public static int matrizGanar[][]=new int[3][3];
public static Image f=null;
public static Image f2=null;
public static int jg_j1=0;
public static int jg_j2=0;
public static Graphics g;
public dibuja(teclasv1 mid){
this.mid=mid;
mensaje=new Ticker("Jugador nro 1: * Jugador nro 2: # Puntaje J1: " + jg_j1+"
J2:"+jg_j2+""");
juegoNuevo=new Command("Juego Nuevo",Command.EXIT,0);
this.setTicker(mensaje);
this.addCommand(juegoNuevo);
this.setCommandListener(this);
nuAlto=this.getHeight();
nuAncho=this.getWidth();
nuDAlto=nuAlto/3;
nuDAncho=nuAncho/3;
c1=c2=c3=c4=c5=c6=c7=c8=c9=false;
}
protected void paint(Graphics g) {
try {
f2 = Image.createImage("/cruz2.png");
} catch (IOException ex) {
ex.printStackTrace();
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
    }
    try {
        f = Image.createImage("/circulo.png");
    } catch (IOException ex) {
        ex.printStackTrace();
    }
if (c1){
if(turnoJugador==1){
g.setColor(0,0,255);
g.drawImage(f, 60, 50, 10);
matrizGanar[0][0]=1;
int j=verificarGanador(1);
if (j == 1){
g.setColor(255, 0, 0);
g.drawLine(0,nuAlto/6,nuAncho,nuAlto/6);
matrizGanar[0][0]=0;
jg_j1++;
mensaje=new Ticker("Puntaje J1: " + jg_j1+ "   J2:"+jg_j2+");
this.setTicker(mensaje);
}
int j2=verificarGanador2(1);
if(j2==1){
g.setColor(255, 0, 0);
g.drawLine(nuAncho/6,0,nuAncho/6,nuAlto);
matrizGanar[0][0]=0;
jg_j1++;
mensaje=new Ticker("Puntaje J1: " + jg_j1+ "   J2:"+jg_j2+");
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
this.setTicker(mensaje);
}
String t=verificarGanador3(1);
if (t.equals("! GANASTE¡"))
{
g.setColor(255, 0, 0);
g.drawLine(1, 1, nuAncho, nuAlto);
matrizGanar[0][0]=0;
jg_j1++;
mensaje=new Ticker("Puntaje J1: " + jg_j1+"   J2:"+jg_j2+""");
this.setTicker(mensaje);
}
c1=false;
}
if(turnoJugador==2){
g.setColor(0,0,255);
g.drawImage(f2, 60, 50, 10);
matrizGanar[0][0]=2;
int j=verificarGanador(2);
if (j == 1){
g.setColor(255, 0, 0);
g.drawLine(0,nuAlto/6,nuAncho,nuAlto/6);
matrizGanar[0][0]=0;
jg_j2++;
mensaje=new Ticker("Puntaje J1: " + jg_j1+"   J2:"+jg_j2+""");
this.setTicker(mensaje);
}
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
int j2=verificarGanador2(2);
if(j2==1){
    g.setColor(255, 0, 0);
    g.drawLine(nuAncho/6,0,nuAncho/6,nuAlto);
    matrizGanar[0][0]=0;
    jg_j2++;
    mensaje=new Ticker("Puntaje J1: " + jg_j1+"    J2:"+jg_j2+""");
    this.setTicker(mensaje);
}
    String t=verificarGanador3(2);
    if (t.equals("! GANASTE¡"))
    {
        g.setColor(255, 0, 0);
        g.drawLine(1, 1, nuAncho, nuAlto);
        matrizGanar[0][0]=0;
        jg_j2++;
        mensaje=new Ticker("Puntaje J1: " + jg_j1+"    J2:"+jg_j2+""");
        this.setTicker(mensaje);
    }
c1=false;
}}
if (c2){
if(turnoJugador==1){
g.setColor(0,0,255);
    g.drawImage(f, 145, 50, 10);
matrizGanar[0][1]=1;
int j=verificarGanador(1);
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
if (j == 1){
    g.setColor(255, 0, 0);
    g.drawLine(0,nuAlto/6,nuAncho,nuAlto/6);
    matrizGanar[0][1]=0;
    jg_j1++;
    mensaje=new Ticker("Puntaje J1: " + jg_j1+"    J2:"+jg_j2+""");
    this.setTicker(mensaje);
}
int j2=verificarGanador2(1);
if(j2==1){
    g.setColor(255, 0, 0);
    g.drawLine(nuAncho/2,0,nuAncho/2,nuAlto);
    matrizGanar[0][1]=0;
    jg_j1++;
    mensaje=new Ticker("Puntaje J1: " + jg_j1+"    J2:"+jg_j2+""");
    this.setTicker(mensaje);
}
c2=false;
}
if(turnoJugador==2){
    g.setColor(0,0,255);
    g.drawImage(f2, 145, 50, 10);
    matrizGanar[0][1]=2;
    int j=verificarGanador(2);
    if (j == 1){
        g.setColor(255, 0, 0);
        g.drawLine(0,nuAlto/6,nuAncho,nuAlto/6);
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
matrizGanar[0][1]=0;
jg_j2++;
mensaje=new Ticker("Puntaje J1: " + jg_j1+"   J2:"+jg_j2+");
this.setTicker(mensaje);
}
int j2=verificarGanador2(2);
if(j2==1){
    g.setColor(255, 0, 0);
    g.drawLine(nuAncho/2,0,nuAncho/2,nuAlto);
    matrizGanar[0][1]=0;
    jg_j2++;
    mensaje=new Ticker("Puntaje J1: " + jg_j1+"   J2:"+jg_j2+");
    this.setTicker(mensaje);
}
c2=false;
}
}
if(c3){
    if(turnoJugador==1){
        g.setColor(0,0,255);
        g.drawImage(f, 220, 50, 10);
        matrizGanar[0][2]=1;
        int j=verificarGanador(1);
        if (j == 1){
            g.setColor(255, 0, 0);
            g.drawLine(0,nuAlto/6,nuAncho,nuAlto/6);
            matrizGanar[0][2]=0;
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
jg_j1++;
mensaje=new Ticker("Puntaje J1: " + jg_j1+"   J2:"+jg_j2+""");
this.setTicker(mensaje);
}
int j2=verificarGanador2(1);
if(j2==1){
    g.setColor(255, 0, 0);
    g.drawLine(nuAncho-45,0,nuAncho-45,nuAlto);
    matrizGanar[0][2]=0;
    jg_j1++;
    mensaje=new Ticker("Puntaje J1: " + jg_j1+"   J2:"+jg_j2+""");
    this.setTicker(mensaje);
}
String t=verificarGanador3(1);
if (t.equals("! GANASTE2i"))
{
    g.setColor(255, 0, 0);
    g.drawLine(nuAncho,1, 1, nuAlto);
    matrizGanar[0][2]=0;
    jg_j1++;
    mensaje=new Ticker("Puntaje J1: " + jg_j1+"   J2:"+jg_j2+""");
    this.setTicker(mensaje);
}
c3=false;
}
if(turnoJugador==2){
    g.setColor(0,0,255);
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
g.drawImage(f2, 220, 50, 10);
matrizGanar[0][2]=2;
int j=verificarGanador(2);
if (j == 1){
    g.setColor(255, 0, 0);
    g.drawLine(0,nuAlto/6,nuAncho,nuAlto/6);
    matrizGanar[0][2]=0;
    jg_j2++;
    mensaje=new Ticker("Puntaje J1: " + jg_j1+ "   J2:"+jg_j2+");
    this.setTicker(mensaje);
}
int j2=verificarGanador2(2);
if(j2==1){
    g.setColor(255, 0, 0);
    g.drawLine(nuAncho-45,0,nuAncho-45,nuAlto);
    matrizGanar[0][2]=0;
    jg_j2++;
    mensaje=new Ticker("Puntaje J1: " + jg_j1+ "   J2:"+jg_j2+");
    this.setTicker(mensaje);
}
String t=verificarGanador3(2);
if (t.equals("! GANASTE2i"))
{
    g.setColor(255, 0, 0);
    g.drawLine(nuAncho,1, 1, nuAlto);
    matrizGanar[0][2]=0;
    jg_j2++;
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
mensaje=new Ticker("Puntaje J1: " + jg_j1+" J2:"+jg_j2+""");
this.setTicker(mensaje);
}
c3=false;
}
}
if (c4){
if(turnoJugador==1){
g.setColor(0,0,255);
g.drawImage(f, 60, 130, 10);
matrizGanar[1][0]=1;
int j=verificarGanador(1);
if (j == 1){
g.setColor(255, 0, 0);
g.drawLine(0,nuAlto/2,nuAncho,nuAlto/2);
matrizGanar[1][0]=0;
jg_j1++;
mensaje=new Ticker("Puntaje J1: " + jg_j1+" J2:"+jg_j2+""");
this.setTicker(mensaje);
}
int j2=verificarGanador2(1);
if(j2==1){
g.setColor(255, 0, 0);
g.drawLine(nuAncho/6,0,nuAncho/6,nuAlto);
matrizGanar[1][0]=0;
jg_j1++;
mensaje=new Ticker("Puntaje J1: " + jg_j1+" J2:"+jg_j2+""");
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
this.setTicker(mensaje);
}
c4=false;
}
if(turnoJugador==2){
g.setColor(0,0,255);
g.drawImage(f2, 60, 130, 10);
matrizGanar[1][0]=2;
int j=verificarGanador(2);
if (j == 1){
g.setColor(255, 0, 0);
g.drawLine(0,nuAlto/2,nuAncho,nuAlto/2);
matrizGanar[1][0]=0;
jg_j2++;
mensaje=new Ticker("Puntaje J1: " + jg_j1+" J2:"+jg_j2+");
this.setTicker(mensaje);
}
int j2=verificarGanador2(2);
if(j2==1){
g.setColor(255, 0, 0);
g.drawLine(nuAncho/6,0,nuAncho/6,nuAlto);
matrizGanar[1][0]=0;
jg_j2++;
mensaje=new Ticker("Puntaje J1: " + jg_j1+" J2:"+jg_j2+");
this.setTicker(mensaje);
}
c4=false;
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
}  
}  
if (c5){  
if(turnoJugador==1){  
g.setColor(0,0,255);  
g.drawImage(f, 145, 130, 10);  
matrizGanar[1][1]=1;  
int j=verificarGanador(1);  
if (j == 1){  
g.setColor(255, 0, 0);  
g.drawLine(0,nuAlto/2,nuAncho,nuAlto/2);  
matrizGanar[1][1]=0;  
jg_j1++;  
mensaje=new Ticker("Puntaje J1: " + jg_j1+" J2:"+jg_j2+"");  
this.setTicker(mensaje);  
}  
int j2=verificarGanador2(1);  
if(j2==1){  
g.setColor(255, 0, 0);  
g.drawLine(nuAncho/2,0,nuAncho/2,nuAlto);  
matrizGanar[1][1]=0;  
jg_j1++;  
mensaje=new Ticker("Puntaje J1: " + jg_j1+" J2:"+jg_j2+"");  
this.setTicker(mensaje);  
}  
String t2=verificarGanador3(1);  
if (t2.equals("! GANASTE2;"))
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
{
g.setColor(255, 0, 0);
g.drawLine(nuAncho,1, 1, nuAlto);
matrizGanar[1][1]=0;
jg_j1++;
mensaje=new Ticker("Puntaje J1: " + jg_j1+"   J2:"+jg_j2+""");
this.setTicker(mensaje);
}
String t=verificarGanador3(1);
if (t.equals("! GANASTE¡"))
{
g.setColor(255, 0, 0);
g.drawLine(1, 1, nuAncho, nuAlto);
matrizGanar[1][1]=0;
jg_j1++;
mensaje=new Ticker("Puntaje J1: " + jg_j1+"   J2:"+jg_j2+""");
this.setTicker(mensaje);
}

c5=false;
}
if(turnoJugador==2){
g.setColor(0,0,255);

g.drawImage(f2, 145, 130, 10);
matrizGanar[1][1]=2;
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
int j=verificarGanador(2);
if (j == 1){
    g.setColor(255, 0, 0);
    g.drawLine(0,nuAlto/2,nuAncho,nuAlto/2);
    matrizGanar[1][1]=0;
    jg_j2++;
    mensaje=new Ticker("Puntaje J1: " + jg_j1+"    J2:"+jg_j2+""");
    this.setTicker(mensaje);
}
int j2=verificarGanador2(2);
if(j2==1){
    g.setColor(255, 0, 0);
    g.drawLine(nuAncho/2,0,nuAncho/2,nuAlto);
    matrizGanar[1][1]=0;
    jg_j2++;
    mensaje=new Ticker("Puntaje J1: " + jg_j1+"    J2:"+jg_j2+""");
    this.setTicker(mensaje);
}
String t2=verificarGanador3(2);
    if (t2.equals("! GANASTE2;"))
    {
        g.setColor(255, 0, 0);
        g.drawLine(nuAncho,1, 1, nuAlto);
        matrizGanar[1][1]=0;
        jg_j2++;
        mensaje=new Ticker("Puntaje J1: " + jg_j1+"    J2:"+jg_j2+""");
        this.setTicker(mensaje);
    }
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
    }  
    String t=verificarGanador3(2);  
    if (t.equals("! GANASTE¡"))  
    {  
    g.setColor(255, 0, 0);  
    g.drawLine(1, 1, nuAncho, nuAlto);  
    matrizGanar[1][1]=0;  
    jg_j2++;  
    mensaje=new Ticker("Puntaje J1: " + jg_j1+ "   J2:"+jg_j2+");  
    this.setTicker(mensaje);  
    }  
    c5=false;  
    }  
    }  
  
    if (c6){  
    if(turnoJugador==1){  
    g.setColor(0,0,255);  
    g.drawImage(f, 220, 130, 10);  
    matrizGanar[1][2]=1;  
    int j=verificarGanador(1);  
    if (j == 1){  
    g.setColor(255, 0, 0);  
    g.drawLine(0,nuAlto/2,nuAncho,nuAlto/2);  
    matrizGanar[1][2]=0;  
    jg_j1++;  
    mensaje=new Ticker("Puntaje J1: " + jg_j1+ "   J2:"+jg_j2+");
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
this.setTicker(mensaje);
}
int j2=verificarGanador2(1);
if(j2==1){
    g.setColor(255, 0, 0);
    g.drawLine(nuAncho-45,0,nuAncho-45,nuAlto);
    matrizGanar[1][2]=0;
    jg_j1++;
    mensaje=new Ticker("Puntaje J1: " + jg_j1+"    J2:"+jg_j2+""");
    this.setTicker(mensaje);
}

c6=false;
}
if(turnoJugador==2){
    g.setColor(0,0,255);
    g.drawImage(f2, 220, 130, 10);
    matrizGanar[1][2]=2;
    int j=verificarGanador(2);
    if (j == 1){
        g.setColor(255, 0, 0);
        g.drawLine(0,nuAlto/2,nuAncho,nuAlto/2);
        matrizGanar[1][2]=0;
        jg_j2++;
        mensaje=new Ticker("Puntaje J1: " + jg_j1+"    J2:"+jg_j2+""");
        this.setTicker(mensaje);
    }
}
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
int j2=verificarGanador2(2);
if(j2==1){
    g.setColor(255, 0, 0);
    g.drawLine(nuAncho-45,0,nuAncho-45,nuAlto);
    matrizGanar[1][2]=0;
    jg_j2++;
    mensaje=new Ticker("Puntaje J1: " + jg_j1+ "   J2:"+jg_j2+");
    this.setTicker(mensaje);
}
c6=false;
}
}

if (c7){
if(turnoJugador==1){
    g.setColor(0,0,255);
    g.drawImage(f, 60, 225, 10);
    matrizGanar[2][0]=1;
    int j=verificarGanador(1);
    if (j == 1){
        g.setColor(255, 0, 0);
        g.drawLine(0,nuAlto-45,nuAncho,nuAlto-45);
        matrizGanar[2][0]=0;
        jg_j1++;
        mensaje=new Ticker("Puntaje J1: " + jg_j1+ "   J2:"+jg_j2+");
        this.setTicker(mensaje);
    }
}
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
int j2=verificarGanador2(1);
if(j2==1){
    g.setColor(255, 0, 0);
    g.drawLine(nuAncho/6,0,nuAncho/6,nuAlto);
    matrizGanar[2][0]=0;
    jg_j1++;
    mensaje=new Ticker("Puntaje J1: " + jg_j1+"    J2:"+jg_j2+""");
    this.setTicker(mensaje);
}

String t2=verificarGanador3(1);
if (t2.equals("! GANASTE2i"))
{
    g.setColor(255, 0, 0);
    g.drawLine(nuAncho,1, 1, nuAlto);
    matrizGanar[2][0]=0;
    jg_j1++;
    mensaje=new Ticker("Puntaje J1: " + jg_j1+"    J2:"+jg_j2+""");
    this.setTicker(mensaje);
}

c7=false;
}

if(turnoJugador==2){
    g.setColor(0,0,255);
    g.drawImage(f2, 60, 225, 10);
    matrizGanar[2][0]=2;
    int j=verificarGanador(2);
    if (j == 1){
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
g.setColor(255, 0, 0);
g.drawLine(0,nuAlto-45,nuAncho,nuAlto-45);
matrizGanar[2][0]=0;
jg_j2++;
mensaje=new Ticker("Puntaje J1: " + jg_j1+"   J2:"+jg_j2+");
this.setTicker(mensaje);
}
int j2=verificarGanador2(2);
if(j2==1){
    g.setColor(255, 0, 0);
    g.drawLine(nuAncho/6,0,nuAncho/6,nuAlto);
    matrizGanar[2][0]=0;
    jg_j2++;
    mensaje=new Ticker("Puntaje J1: " + jg_j1+"   J2:"+jg_j2+");
    this.setTicker(mensaje);
}
String t2=verificarGanador3(2);
    if (t2.equals("! GANASTE2!"))
    {
g.setColor(255, 0, 0);
g.drawLine(nuAncho,1, 1, nuAlto);
matrizGanar[2][0]=0;
jg_j2++;
mensaje=new Ticker("Puntaje J1: " + jg_j1+"   J2:"+jg_j2+");
this.setTicker(mensaje);
    }
c7=false;
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
}  
}  
  
if (c8){  
if(turnoJugador==1){  
g.setColor(0,0,255);  
g.drawImage(f, 145, 225, 10);  
matrizGanar[2][1]=1;  
int j=verificarGanador(1);  
if (j == 1){  
g.setColor(255, 0, 0);  
g.drawLine(0,nuAlto-45,nuAncho,nuAlto-45);  
matrizGanar[2][1]=0;  
jg_j1++;  
mensaje=new Ticker("Puntaje J1: " + jg_j1+ " J2:"+jg_j2+");  
this.setTicker(mensaje);  
}  
int j2=verificarGanador2(1);  
if(j2==1){  
g.setColor(255, 0, 0);  
g.drawLine(nuAncho/2,0,nuAncho/2,nuAlto);  
matrizGanar[2][1]=0;  
jg_j1++;  
mensaje=new Ticker("Puntaje J1: " + jg_j1+ " J2:"+jg_j2+");  
this.setTicker(mensaje);  
}  
c8=false;
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
}
if(turnoJugador==2){
g.setColor(0,0,255);
g.drawImage(f2, 145, 225, 10);
matrizGanar[2][1]=2;
int j=verificarGanador(2);
if (j == 1){
g.setColor(255, 0, 0);
g.drawLine(0,nuAlto-45,nuAncho,nuAlto-45);
matrizGanar[2][1]=0;
jg_j2++;
mensaje=new Ticker("Puntaje J1: " + jg_j1+ " J2:"+jg_j2+");
this.setTicker(mensaje);
}
int j2=verificarGanador2(2);
if(j2==1){
g.setColor(255, 0, 0);
g.drawLine(nuAncho/2,0,nuAncho/2,nuAlto);
matrizGanar[2][1]=0;
jg_j2++;
mensaje=new Ticker("Puntaje J1: " + jg_j1+ " J2:"+jg_j2+");
this.setTicker(mensaje);
}
c8=false;
}
}
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
if (c9){
if(turnoJugador==1){
g.setColor(0,0,255);
g.drawImage(f, 220, 225, 10);
matrizGanar[2][2]=1;
int j=verificarGanador(1);
if (j == 1){
g.setColor(255, 0, 0);
g.drawLine(0,nuAlto-45,nuAncho,nuAlto-45);
matrizGanar[2][2]=0;
jg_j1++;
mensaje=new Ticker("Puntaje J1: " + jg_j1+" J2:"+jg_j2+""");
this.setTicker(mensaje);
}
int j2=verificarGanador2(1);
if(j2==1){
g.setColor(255, 0, 0);
g.drawLine(nuAncho-45,0,nuAncho-45,nuAlto);
matrizGanar[2][2]=0;
jg_j1++;
mensaje=new Ticker("Puntaje J1: " + jg_j1+" J2:"+jg_j2+""");
this.setTicker(mensaje);
}
String t=verificarGanador3(1);
if (t.equals("! GANASTE¡"))
{
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
g.setColor(255, 0, 0);
g.drawLine(1, 1, nuAncho, nuAlto);
matrizGanar[2][2]=0;
jg_j1++;
mensaje=new Ticker("Puntaje J1: " + jg_j1+"   J2:"+jg_j2+""");
this.setTicker(mensaje);
    }
c9=false;
}
if(turnoJugador==2){
g.setColor(0,0,255);
g.drawImage(f2, 220, 225, 10);
matrizGanar[2][2]=2;
int j=verificarGanador(2);
if (j == 1){
    g.setColor(255, 0, 0);
g.drawLine(0,nuAlto-45,nuAncho,nuAlto-45);
matrizGanar[2][2]=0;
jg_j2++;
mensaje=new Ticker("Puntaje J1: " + jg_j1+"   J2:"+jg_j2+""");
this.setTicker(mensaje);
}
int j2=verificarGanador2(2);
if(j2==1){
    g.setColor(255, 0, 0);
g.drawLine(nuAncho-45,0,nuAncho-45,nuAlto);
matrizGanar[2][2]=0;
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
jg_j2++;
mensaje=new Ticker("Puntaje J1: " + jg_j1+"   J2:"+jg_j2+""");
this.setTicker(mensaje);
}
    String t=verificarGanador3(2);
    if (t.equals("! GANASTE¡"))
    {
        g.setColor(255, 0, 0);
        g.drawLine(1, 1, nuAncho, nuAlto);
        matrizGanar[2][2]=0;
        jg_j2++;
        mensaje=new Ticker("Puntaje J1: " + jg_j1+"   J2:"+jg_j2+""");
        this.setTicker(mensaje);
    }
    c9=false;
}
}

g.setColor(0,0,255);
g.drawLine(nuDAncho,0,nuDAncho,nuAlto);
g.drawLine(2*nuDAncho,0,2*nuDAncho,nuAlto);
g.drawLine(0,nuDAlto,nuAncho,nuDAlto);
g.drawLine(0,2*nuDAlto,nuAncho,2*nuDAlto);
//lineas triki
g.setColor(255,0,0);
//          g.drawLine(1, 1, nuAncho, nuAlto);//iz-der
//          g.drawLine(nuAncho,1, 1, nuAlto);//der-iz
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
//          g.drawLine(nuAncho/2,0,nuAncho/2,nuAlto);//ver
////          g.drawLine(0,nuAlto/2,nuAncho,nuAlto/2);//hor
}

protected void keyReleased(int codigo) {
switch(codigo){
    case Canvas.KEY_POUND:
turnoJugador=2;
break;
    case Canvas.KEY_STAR:
turnoJugador=1;
break;
    default:
break;
}

if(turnoJugador!=0){
switch (codigo) {
case Canvas.KEY_NUM1:
c1=!c1;
break;
case Canvas.KEY_NUM2:
c2=!c2;
break;
case Canvas.KEY_NUM3:
c3=!c3;
break;
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
case Canvas.KEY_NUM4:
c4=!c4;
break;
case Canvas.KEY_NUM5:
c5=!c5;
break;
case Canvas.KEY_NUM6:
c6=!c6;
break;
case Canvas.KEY_NUM7:
c7=!c7;;
break;
case Canvas.KEY_NUM8:
c8=!c8;
break;
case Canvas.KEY_NUM9:
c9=!c9;
break;
default:
break;
}
}
this.repaint();
}
public static int ganofila(int fila, int jugador){
int cont2 = 0;
for (int x = 0; x < 3; x++){
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
if (matrizGanar[filas][x] == jugador){
    cont2 = cont2 + jugador;
}
}
return (cont2);
}
public static int verificarGanador(int jugador){
int temp = 0;
int v = 0;
for (int y = 0; y < 3;y++ )
{
temp = ganofila(y,jugador);
if (temp == 3){
v = 1;
temp = 0;
}
if (temp == 6){
v = 1;
temp = 0;
}
}
return v;
}
public static int ganoColumna(int columna, int jugador){
int cont2 = 0;
for (int x = 0; x < 3; x++){
if (matrizGanar[x] [columna] == jugador){
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
        cont2 = cont2 + jugador;}
    }
    return (cont2);
}

public static int verificarGanador2(int jugador){
    int v = 0;
    int temp = 0;
    for (int y = 0; y < 3; y++){
        temp = ganoColumna(y, jugador);
        if (temp == 3){
            v = 1;
            temp = 0;
        }
        if (temp == 6){
            v = 1;
            temp = 0;
        }
    }
    return v;
}

public static int ganoDiagonal1(int jugador){
    int cont2 = 0;
    for (int x = 0; x < 3; x++){
        if (matrizGanar[x][ x] == jugador){
            cont2 = cont2 + jugador;
        }
    }
}
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
return (cont2);
}
public static int ganoDiagonal2(int jugador){
int cont2 = 0;
int y2 = 2;
for (int x = 0; x < 3; x++){
if (matrizGanar[x] [y2] == jugador){
    cont2 = cont2 + jugador;
}
y2--;
}

return (cont2);
}
public static String verificarGanador3(int jugador)
{
String temp2 = "";
int temp = 0;
int temp5 = 0;
for (int y = 0; y < 1; y++)
{
temp = ganoDiagonal1( jugador);
temp5 = ganoDiagonal2(jugador);
if (temp == 3)
{
temp2 = "! GANASTE¡";
temp = 0;

```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
}  
if (temp == 6)  
{  
  
    temp2 = "! GANASTE1";  
    temp = 0;  
}  
if (temp5 == 3)  
{  
    temp2 = "! GANASTE2i";  
    temp5 = 0;  
}  
if (temp5 == 6)  
{  
    temp2 = "! GANASTE2j";  
    temp5 = 0;  
}  
  
}  
return temp2;  
}  
public void commandAction(Command c, Displayable arg1) {  
// TODO Auto-generated method stub  
if(c==juegoNuevo){  
for(int i=0;i<matrizGanar.length;i++){  
    for(int j=0;j<matrizGanar.length;j++){  
        matrizGanar[i][j]=0;  
    }  
}
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
}  
}  
mid.destroyApp(true);  
mid.notifyDestroyed();  
dibuja d=new dibuja(mid);  
teclasv1.pantalla.setCurrent(d);  
}  
}  
}
```

### **Archivo teclasv1.java**

```
ackage Hellow;  
import javax.microedition.lcdui.*;  
import javax.microedition.midlet.MIDlet;  
public class teclasv1 extends MIDlet {  
    public static Display pantalla;  
    private dibuja triki;  
    public teclasv1(){  
        pantalla=Display.getDisplay(this);  
        triki=new dibuja(this);  
    }  
  
    protected void startApp(){  
        // TODO Auto-generated method stub  
        pantalla.setCurrent(triki);  
    }  
}
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
}  
protected void pauseApp() {  
    // TODO Auto-generated method stub  
}  
protected void destroyApp(boolean arg0) {  
}  
public void salir() {  
    destroyApp(false);  
    notifyDestroyed();  
}  
}
```

## CÓDIGO FUENTE TRANSMISIÓN DE DATOS VIA HTTP

**E**l código fuente que se muestra a continuación representa la transmisión de datos vía Http



Figura No D.1. Formulario para el Envío de datos vía Http

## COMPUTACIÓN MÓVIL, Principios y técnicas

### Código fuente httpconnection.java

Los comentarios inician con los símbolos “//”

```
import com.sun.midp.security.ImplicitlyTrustedClass;
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import javax.microedition.io.*;
import java.io.*;/**
 * @author Victor Viera Balanta
 * Julio -2010
 */
public class httpconnection extends MIDlet implements CommandListener
{
//creacion de variables de visualizacion, formularios y objetos
    private Display display;    // Reference to Display object
    private Form fmMain;    // The main form
    private Command cmGET;    // Request method GET
    private Command cmExit;    // Command to exit the MIDlet
    private TextField dato1;    // Get account number
    private TextField dato2;    // Get password
    private TextField resultado;    // Get password

    private String errorMsg = null;
    private Thread t;
    public httpconnection()
    {
        display = Display.getDisplay(this);
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

### **// Crea los comandos**

```
cmGET = new Command("GET", Command.SCREEN, 2);
```

```
cmExit = new Command("Exit", Command.EXIT, 1);
```

### **// Textfields**

```
dato1 = new TextField("código:", "", 5, TextField.NUMERIC);
```

```
dato2 = new TextField("nombre:", "", 10, TextField.ANY );
```

```
resultado= new TextField("", "", 50, TextField.ANY );
```

### **// crea el formulario**

```
fmMain = new Form("Informacion");
```

```
fmMain.addCommand(cmExit);
```

```
fmMain.addCommand(cmGET);
```

```
fmMain.append(dato1);
```

```
fmMain.append(dato2);
```

```
fmMain.append(resultado);
```

```
fmMain.setCommandListener(this);
```

```
}
```

```
public void startApp()
```

```
{
```

```
    display.setCurrent(fmMain);
```

```
}
```

```
public void pauseApp() {
```

```
}
```

```
public void destroyApp(boolean unconditional) {
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
}  
public void commandAction(Command c, Displayable s)  
{  
  
    try  
    {  
        if (c == cmGET)  
            //peticionGET();  
            //crea un hilo para que el dispositivo  
            //no se bloquee cuando envia los datos  
            t = new Thread(){  
                public void run()  
                {  
                    //llama el método para el envio  
                    envioGet();  
                }  
            };  
            t.start();  
        }  
        catch (Exception e)  
        {  
            System.err.println("Msg: " + e.toString());  
        }  
        if (c == cmExit)  
        {  
            destroyApp(false);  
            notifyDestroyed();  
        }  
    }  
}
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
}  
}  
/*-----  
* Access servlet using GET  
*-----*/  
  
public void envioGet(){  
    boolean ret = false;  
    try{  
        //direccion del servidor para el envio de datos por el método get  
        //se envian los datos codigo y nombre  
        String url = "http://localhost/recibir.php?codigo=" + dato1.getString() + "&" +  
        "nombre=" + dato2.getString();  
        //se establece la comunicación  
        HttpURLConnection hc = (HttpURLConnection)Connector.open(url);  
        hc.setRequestMethod(HttpURLConnection.GET);  
  
        //flujo de datos de recibidos  
        InputStream in = hc.openDataInputStream();  
        //se inicializa la cadena de datos  
        String cadena= "";  
        int ch=0;  
        //procesa caracter por caracter  
        for(int i=0;i<hc.getLength();i++)  
        {  
            //lee un caracter  
            ch= in.read();  
            //si es fin de la cadena para el for
```

## COMPUTACIÓN MÓVIL, Principios y técnicas

```
        if(ch==-1){
            break;
        }
        //va adicionando los caracteres leídos a la cadena
        cadena+= ((char)ch);

    }
    //coloca el resultado en un campo de texto
    resultado.setString(cadena);

    //cierra la conexión
    hc.close();
}catch(Exception e){
    e.printStackTrace(); //Muestra Toda la traza del error
}
}
}
```

Para la transmisión de datos de manera satisfactoria se debe contar con un programa en el servidor Web que procese los datos enviados y en muchas ocasiones devuelva una respuesta. A continuación se presenta el código en el lenguaje de programación PHP <sup>22</sup>

---

<sup>22</sup> Sitio oficial del lenguaje de programación Php

<http://php.net/index.php>

## COMPUTACIÓN MÓVIL, Principios y técnicas

### Código fuente recibir.php

Los comentarios inician con los caracteres //

```
<?PHP
//autor Victor Viera
//Julio-2010
//se reciben los datos enviados
$codigo=$_GET['codigo'];
$nombre=$_GET['nombre'];

//conexion base de datos
$conexion = mysql_connect("localhost", "root", "");
//seleccionamos la base de datos

if (!$conexion)
{
    echo "Error al intentar conectarse con el servidor MySQL";
    exit();
}else{

    //selecciono la base de datos
    mysql_select_db("nueva", $conexion);
    //se hace el insert, desde el formulario solo falta lo del celular
    $cadena = "insert into ciudad(codigo,nombre)
values('$codigo','$nombre)";
    //se ejecuta el Quero y devuelve un mensaje al dispositivo en caso
de //error
    $resultado = mysql_query($cadena, $conexion) or
die(mysql_error());
}
?>
```

## **BIBLIOGRAFIA**

---

**G**ran parte de los conocimientos abordados corresponden a la experiencia del autor en el campo empresarial, pocos textos abordan la computación móvil desde esta perspectiva. Aquí se presentan textos importantes que pueden servir para ampliar algunos temas en el área de la programación con J2me.

### Libros

Froufe, Agustin Quintasy Cardenas Patricia Jorge

J2ME Manual de Usuario y Tutorial

Editorial Alfa-Omega,Rama, 2003

Galvez, Sergio Rojas y Ortega Diaz Lucas

Java a Tope, J2ME, Java 2 Micro Edition Edicion Electronica

Sun Microsystems, 2003

Lopez , Angel y Novo Alejandro

Protocolos de Internet Diseño e Implementación en sistema Unix

Editorial Alfa-Omega,Rama, 2000

## **COMPUTACIÓN MÓVIL, Principios y técnicas**

Prieto Martin, Manuel Jesus

Desarrollo de Juegos con J2ME

Alfa-Omega,Rama, 2007

### **Recursos en Internet**

<http://www.mailxmail.com/curso-tecnicas-desarrollo-computacion-movil-orientado-pda>

Curso en línea de técnicas para el desarrollo de Computación Móvil orientado a Pda (Víctor Viera) , descargable en formato Pdf

<http://download-llnw.oracle.com/javame/>

Sitio oficial de J2me, Oracle Corporation

<http://www.symbian.org/>

Sitio sistema operativo Symbian

<http://www.roseindia.net/j2me/>

Pagina con ejemplos variados con J2me, desde el ejemplo más sencillo holamundo hasta el ejemplo complejo de comunicación por medio del protocolo http.

<http://www.mailxmail.com/curso-programacion-juegos-moviles-j2me>

Curso en Línea de programación con Dispositivos inalámbricos, se puede bajar la versión en formato Pdf del mismo.

<http://www.mailxmail.com/curso-programacion-juegos-moviles-j2me>

Curso en línea de programación de videojuegos con J2me,descargable en formato Pdf.

<http://httpd.apache.org/>

Pagina oficial del proyecto del servidor apache utilizado en para la atención de clientes vía http.

<http://www.easyphp.org/>

## **COMPUTACIÓN MÓVIL, Principios y técnicas**

Sitio oficial del proyecto que integra Servidor Apache, Base de Datos Mysql y el lenguaje de programación Php

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

Otro proyecto que integra la tecnología de Servidor Apache, Base de Datos Mysql y el lenguaje de programación Php.