

# PROTOTIPO DE COMPONENTE SOFTWARE PARA LA GESTIÓN DE BASES DE DATOS EN DISPOSITIVOS MOVILES UTILIZANDO JAVA 2 MICROEDITION

Deyner Paul Reinoso Rosado

**Resumen-** Este artículo pretende mostrar la implementación de un componente de software para gestión de bases de datos usando archivos al interior de un dispositivo móvil; Este componente va dirigido a los desarrolladores de dispositivos móviles que usan J2ME ya que la base de funcionamiento del componente es la clase manejadora de archivos embebidos RMS (Record Management System) que permite usar la memoria interna del dispositivo y almacenar colecciones de datos en bytecodes de tamaño variable.

A pesar de las facilidades que tiene RMS es complejo de manejar los datos con este, El componente resultado de esta investigación permite utilizar el lenguaje estándar SQL (Structured Query Language) y la teoría de bases de datos relacionales para poder acceder a estos datos haciendo posible la utilización de estos tal como lo haría cualquier Motor de bases de datos.

## I. Introducción

El trabajo con dispositivos móviles ha permitido a la humanidad acceder a la información en cualquier lugar y momento, por lo que en las empresas se ha requerido cambiar infraestructura tanto de hardware como de software que permita la movilidad. J2ME es la apuesta de Java en el desarrollo de software para dispositivos móviles. La gestión de datos a través de los años se ha venido haciendo usando plataformas cliente/servidor. J2ME introdujo para el tratamiento de datos RMS que es un administrador de registros embebidos en el móvil,

pero no permite tratamiento de datos usando SQL ni usando el modelo relacional de bases de datos.

## II. Administradores de bases de datos

El almacenamiento de datos en aplicaciones que incluyen dispositivos móviles comprende la utilización de bases de datos tanto en los dispositivos para permitir la continua movilidad y bases de datos en servidores o computadores fijos que permitan la sincronización de ambas bases de datos.

El almacenamiento de datos dentro de dispositivos móviles ha sido un problema complicado de solucionar por los desarrolladores de software pero a la vez un 'mal' necesario ya que muchas de las aplicaciones que se dan a las tecnologías móviles requieren un pre-almacenamiento antes de la conexión de estos con los datos del servidor de la aplicación; dado que la utilización de aplicaciones móviles en línea requieren una conexión continua al servidor lo que aumenta el costo en infraestructura de funcionamiento de estas bases de datos hace que las bases de datos embebidas sean una opción muy rentable y recomendable.

Los sistemas de gestión de bases de datos para dispositivos móviles requieren un espacio de memoria básico para ejecutarse sobre el dispositivo móvil, por ejemplo: SQL Anywhere S. Ultralite Requiere un mínimo de memoria de entre 75 y 175 KB; SQL Server Compact Edition requiere entre 800 KB y 1.5 MB, solo por nombrar algunos, esto hace que el dispositivo en

donde se ejecuta el aplicativo sea bastante lento al momento de consultar o registrar los datos esto con el fin de permitir la sincronización de los datos en forma bidireccional.

### III. RMS

La mayoría de los aplicativos requieren la consulta de datos lo que hace indispensable el almacenamiento en estos; Java Micro Edition contiene una clase muy importante que permite el tratamiento de datos en dispositivos que lo soportan.

El nombre de esta clase es Record Management System (RMS); RMS provee un grupo de métodos que permiten el almacenamiento, modificación, eliminación, filtración y búsqueda de datos.

RMS trata a los datos igual que un arreglo de bytes de tamaño variable y se diferencian con un identificador único.

RMS permite la manipulación de los datos dentro de aplicaciones JME, utilizando unos objetos especiales llamados Recordstores que es usado en dispositivos como handhelds, smartphones, PDAs, teléfonos celulares, etc

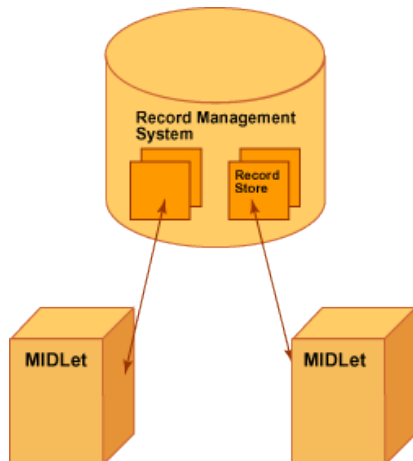
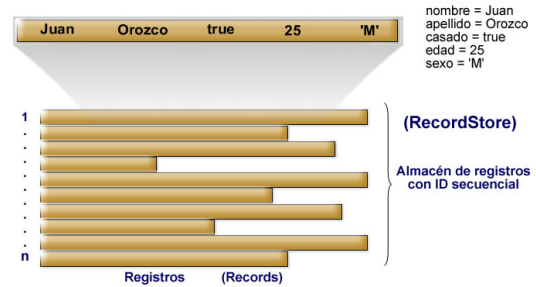


Imagen tomada de blog <http://www.gotocosmik.com> (Persistencia en aplicaciones MIDP)

RMS posee algunas limitaciones, la primera y bastante importante son a las limitaciones de memoria del teléfono móvil ya que RMS utiliza el espacio de almacenamiento del móvil por lo que en dispositivos de poca memoria no es posible almacenar una cantidad muy amplia de datos.

La segunda limitante de RMS es que no trabaja los datos como lo plantea el modelo relacional usando bases de datos, tablas y campos o atributos de tal manera que es imposible utilizar instrucciones SQL para consultar los datos almacenados.

El siguiente es un diagrama que muestra el funcionamiento de RMS

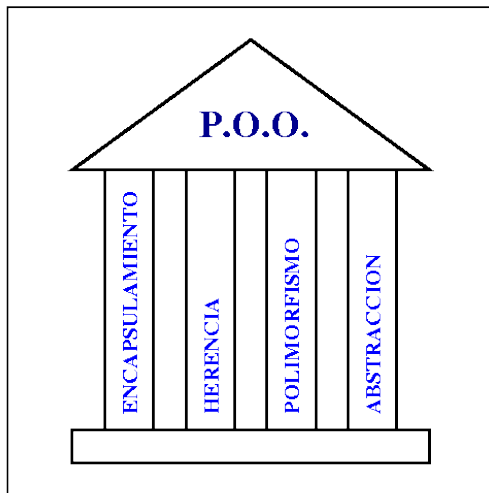


### IV. Programación orientada a componentes

La programación orientada a componentes es una excelente práctica de programación de software que permite a los desarrolladores crear, modificar y adicionar comportamientos a un software más robusto, lo que hace mucho más fácil la reutilización de estos tanto en su programación, depuración y testeado de software.

La programación orientada a componentes habilita la posibilidad de crear funcionalidades de software completamente independientes y luego ensamblarlas en un sistema más complejo permitiendo la fácil escalabilidad del software y el continuo mejoramiento de un sistema sin alterar el funcionamiento de este ni sus demás componentes.

Esta es una disciplina de la programación orientada a objetos y respeta los pilares de esta mostrados en la siguiente figura.



Pilares de la programación orientada a objetos, tomada del curso online programación orientada a objetos <http://elvex.ugr.es>

Los componentes desarrollados en java son muy fáciles de integrar a los proyectos ya que java utiliza la base de la programación orientada a componentes que es la programación orientada a objetos.

### V. Arquitectura del Componente

Este componente de software utiliza como base de aplicación RMS y aplica el modelo relacional de bases de datos.

Éste funciona como sistema de administración o gestión de bases de datos (DBMS) y compatible con el lenguaje SQL Standard separando el nivel físico y lógico de los datos manejando a su vez 3 de las reglas básicas de el modelo relacional de bases de datos.

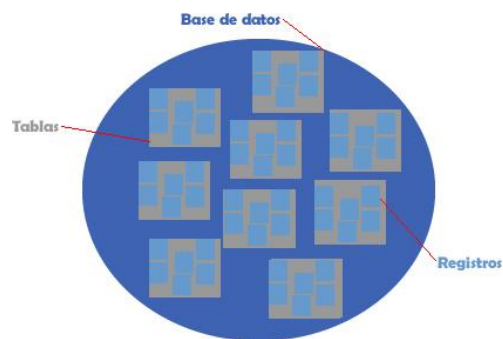
Éste componente respeta los siguientes aspectos relacionales de las bases de datos:

1. **Aspecto estructural**, de tal forma que se pueda percibir el ordenamiento de los datos en estructuras de tablas.
2. **Aspecto de integridad**, de esta manera se garantiza al usuario que las tablas respetan ciertos parámetros.

3. **Aspecto de manipulación**, que son las sentencias SQL que se le dictan al motor para el tratamiento de los datos y estructura del almacenamiento.

Este componente de software implementa un conjunto de características del modelo relacional permitiendo al usuario utilizar los datos como cualquier sistema de bases de datos.

Los modelos de bases de datos relacionales permiten llevar a cabo operaciones un poco más complejas con los datos almacenados en bases de datos.



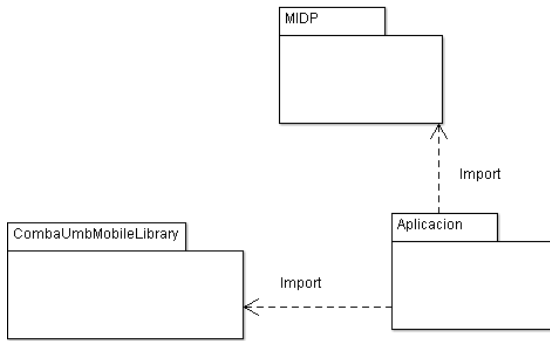
Modelo teórico del componente.

Uniendo la programación por componentes, la teoría de bases de datos relacionales y la clase RMS de Java Microedition se ha conseguido realizar un componente que admite el tratamiento de datos dentro de dispositivos móviles utilizando el lenguaje estándar de consultas.



Funcionamiento del componente sobre J2ME.

De tal forma que el diagrama de clases básico de una aplicación que use el componente se vería como en la siguiente figura.



## VI. Pruebas del componente

Para realizar las pruebas de este componente se diseñó y desarrolló una aplicación en JME que utilizó el paquete de componentes y que permite al usuario escribir las sentencias SQL que desea ejecutar y una “display” que muestra el resultado que retorna la el componente a la acción ejecutada.

Este aplicativo se compiló para 4 perfiles de dispositivos incluyendo celulares de gama media y alta y para PDAs.

Al instalar la aplicación de pruebas en los dispositivos cargándose el archivo .Jar ejecutable de esta se procedió a probar las sentencias de creación de bases de datos la selección de una en particular, la creación de tablas con sus respectivos atributos, la inserción de datos respetando la estructura de las tablas y por último la búsqueda y eliminación de estos. Todo esto usando sentencias SQL.

## VII. Conclusiones

Este componente es una herramienta que puede ser utilizada por los desarrolladores de software tanto para el trabajo utilizando dispositivos móviles como para la investigación continua de sistemas de administración de datos en dispositivos móviles e incluso la creación de un equipo interdisciplinar de investigación que permita la evolución de este y el conocimiento.