

## Sincronización de Bases de Datos móviles basada en servicios web bajo una aproximación multiplataforma

Alejandro Astudillo, Gustavo Álvarez y Francisco Martínez

{joastudillo, gaalvarez, fomarti}@unicauca.edu.co

Universidad del Cauca, Popayán - Colombia

**Resumen:** *Un sistema de replicación y sincronización de datos móviles que permite la operación en modo desconectado, facilita las tareas de recolección y tratamiento de información en campo. Sin embargo, enfrenta desafíos como la convergencia, consistencia y seguridad de los registros, la gestión de dispositivos y aplicaciones, las múltiples plataformas y herramientas de desarrollo, la variedad de dispositivos clientes y aplicaciones empresariales, la integración con sistemas legados, la alta fragmentación que existe en el mercado de los dispositivos móviles, entre otros. Además, algunas soluciones existentes evidencian inconvenientes como el alto grado de acoplamiento entre sus componentes, y dependencia con la plataforma móvil, la aplicación, protocolo de transporte o sincronización. En el presente trabajo, realizado al interior del Grupo de Interés en el Desarrollo de Aplicaciones para Dispositivos Móviles – W@Pcolombia, se aborda la adaptación de un Framework de sincronización de bases de datos móviles basado en servicios Web, enfocado principalmente en el problema de la convergencia y consistencia de los registros, por medio de la definición e implementación de métodos para el tratamiento de conflictos de datos.*

**Palabras clave:** Sincronización, replicación de datos, bases de datos móviles, servicios web, multiplataforma, resolución de conflictos.

**Abstract:** *A replication and data synchronization mobile system, which includes support for disconnected mode operation, eases the tasks of collection and processing of information in the field. However, it faces challenges such as: the convergence, records consistency and security, devices and applications management, the multiple platforms and development tools, the variety of client devices and business applications, the integration with legacy systems, the high fragmentation that exists in the industry of mobile devices, among others. Moreover, some existing solutions have drawbacks such as the high coupling between its components, and its dependence on the mobile platform, application, transport or specific sync protocol. As part of the work performed by the Interest Group on Mobile Devices Applications Development – W@Pcolombia, this work ports a Framework of mobile data synchronization based on Web services, focused mainly on the problem of convergence and consistency of records, through the definition and implementation of methods for data conflicts treatment.*

**Keywords:** synchronization, data replication, mobile databases, web services, platform, conflict resolution.

### 1 Introducción

La evolución de las tecnologías móviles en los últimos años ha facilitado su masificación a nivel global, al punto de contar hoy con más de cinco billones de usuarios alrededor del mundo, cerca del 80% de la población mundial [1]. Esto ha impulsado el aprovisionamiento de nuevas y sorprendentes características en los dispositivos móviles [2][3] y el desarrollo de una gran variedad de aplicaciones y servicios para todo tipo de usuarios, incluyendo entre ellos al sector empresarial, donde los dispositivos móviles se están convirtiendo en una plataforma popular para las aplicaciones de negocios (e-bussines) [4][5], permitiendo a las empresas movilizar sus procesos y aplicaciones (m-bussines) [32], y aumentando la disponibilidad de los datos. La implementación de este tipo de soluciones, optimiza los procesos en diferentes niveles de las organizaciones, reduce tiempos, costes, y a la vez aumenta la productividad y competitividad [6][7].

Sin embargo, las aplicaciones móviles empresariales presentan grandes desafíos a los desarrolladores, entre ellos están la seguridad, la gestión de dispositivos y aplicaciones, las múltiples herramientas y plataformas de desarrollo, la variedad de dispositivos cliente y aplicaciones empresariales, y por último la integración con los sistemas existentes [8][9][10].

En un entorno replicado los datos pueden ser generados y modificados por dispositivos móviles cliente con diferentes características y funcionalidades, incluso cuando están desconectados, para luego ser sincronizados a un sistema de información central [11][7] [8]. Cuando una aplicación soporta el trabajo desconectado o fuera de línea, se presentan contradicciones inevitables de información entre la base de datos central y la base de datos de los móviles [12]. La sincronización de información permite a los datos replicados ser coherentes entre sí y permanecer actualizados [13], de esta manera se resuelven las inconsistencias, se garantiza la integridad, y se mantiene la misma versión de los datos entre múltiples dispositivos cliente, para una o varias bases de datos centrales [14].

No obstante, algunas soluciones actuales de sincronización de datos móviles presentan diversos inconvenientes, por ejemplo, el alto grado de acoplamiento entre sus componentes, dependencia con la plataforma móvil, la aplicación, protocolo de transporte o sincronización específico, además de la alta fragmentación que existe en el mercado de los dispositivos móviles. Una nueva aproximación, pretende facilitar la sincronización de bases de datos móviles bajo estándares Web, ofreciendo la posibilidad de replantear la lógica de sincronización como servicio, específicamente a

través de servicios Web, lo que facilita su integración y despliegue en las arquitecturas modernas orientadas a servicios.

En el presente trabajo, se definen algunos lineamientos con el objetivo de guiar y facilitar el desarrollo de este tipo de soluciones; además, se ejemplifica su uso y pertinencia mediante la adaptación de un Framework para la sincronización de datos móviles basada en servicios Web, abordando principalmente el problema del tratamiento de conflictos de datos que se presentan en un entorno de replicación asíncrono con desconexiones frecuentes y trabajo en modo desconectado.

El paper se organiza de la siguiente manera: en la sección 2, se presenta el marco teórico y el estado del arte; en la sección 3, se define un conjunto de lineamientos para la construcción de este tipo de soluciones; en la sección 4, se describe la adaptación realizada al Framework; en la sección 5, se realizan las pruebas a la solución desarrollada y, finalmente, en la sección 6, se presentan las conclusiones.

## 2 Base teórica

### 2.1 Bases de datos móviles

Las bases de datos móviles se pueden definir como una base de datos portátil, físicamente separada del servidor central de información, con la propiedad de conectarse a él para intercambiar datos y mantenerse actualizada[15]. Uno de los mecanismos usados para implementarlas es la replicación de datos móviles. En la Figura 1, se observa un escenario genérico de sincronización de bases de datos móviles.

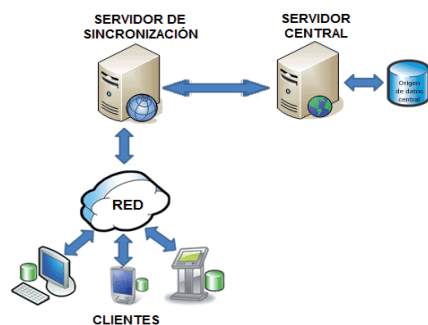


Figura 1: Arquitectura general de un sistema de sincronización.

### 2.2 Sincronización y replicación de bases de datos

La replicación de datos consiste en distribuir y mantener actualizadas múltiples copias de datos llamadas réplicas en equipos independientes [16][17], en diferentes ubicaciones, mediante redes locales o de área extensa [18][19][20]. Los cambios realizados en la réplica de cualquier sitio se capturan y se almacenan localmente, antes de ser transmitidos y aplicados a cada uno de los sitios remotos restantes del entorno de sincronización. Este proceso es conocido como sincronización de datos [21][22][23][24], ofrece ventajas como una mayor disponibilidad, un mejor rendimiento [25], soporte a la computación desconectada, reducción de la carga en la

red y facilita el despliegue en masa. En un entorno de sincronización, existen principalmente dos tipos de sitios: el sitio maestro que aloja la base de datos central, así como los objetos de replicación principales y los sitios clientes que disponen de una réplica local parcial o total de los datos.

#### 2.2.1 Replicación Asíncrona y los conflictos de datos

En la replicación asíncrona, los cambios hechos en un sitio de replicación ocurren en un tiempo posterior en los otros sitios participantes del sistema; esto quiere decir que los cambios de un sitio se pueden almacenar localmente y propagarse a otros sitios en un tiempo posterior, permitiendo así la computación desconectada [26][27].

Por esta razón, se pueden presentar situaciones en las que varios usuarios intentan modificar un dato al mismo tiempo, siendo necesario establecer controles para impedir que las modificaciones de un usuario influyan negativamente en las de otros, a lo que se denomina tratamiento de conflictos de datos [28]. Los conflictos de datos más comunes, que se pueden presentar en un entorno de replicación, están divididos en las siguientes categorías [29][30][31]: Conflictos de Actualización, de Eliminación, de Unicidad, de Orden y de Negocio.

#### 2.2.2 Replicación Sincrónica

En la replicación sincrónica, los cambios hechos en un sitio ocurren inmediatamente en todos los sitios participantes del entorno de sincronización. Bajo este enfoque, solo puede existir una versión de los datos; aunque minimiza la posibilidad de conflictos de datos, se requiere un ambiente estable para operar correctamente, con un mecanismo de comunicación permanente entre las réplicas cliente y el sitio central.

#### 2.2.3 Sincronización de bases de datos móviles basada en servicios Web

En una solución de sincronización de datos móviles tradicional, existen muchas formas de almacenar los datos en cada cliente, diversos protocolos de comunicación para transferirlos, diferentes mecanismos de sincronización, varias plataformas móviles y empresariales, y diversas tecnologías de bases de datos. Esta heterogeneidad se convierte en un problema cada vez más complejo de manejar, que implica sobrecostos en tiempo y recursos necesarios para desplegar un sistema de este tipo.

Como respuesta a este reto, una perspectiva SOA (Service Oriented Architecture) propone la implementación de servicios Web para el acceso a los datos centrales con fines de sincronización [11]. Algunas de las principales ventajas que brinda al proceso de sincronización de bases de datos móviles son [11][33]: Interoperabilidad, desacoplamiento, clientes ligeros, agrupación de conexiones, desarrollo multiplataforma y multilinguaje. RhoSync, RhoConnect y Microsoft Sync Framework, son algunos proyectos que consideran la perspectiva basada en servicios Web para la sincronización de bases de datos móviles.

### 2.3 Trabajos relacionados

En [34], se describe una arquitectura de sincronización definida en tres partes, la aplicación empresarial, el servidor de sincronización y los clientes móviles. Este trabajo presenta un aporte importante en cuanto a la arquitectura de referencia para un servicio de sincronización que usa servicios web. Sin embargo, las tecnologías usadas para el desarrollo están limitadas a ciertas plataformas, como .NET o PHP y Windows Phone.

En [12], el algoritmo definido para la sincronización permite un total grado de movilidad a los clientes, gracias a la ejecución de un gran número de tareas en modo desconectado, reduciendo considerablemente la carga sobre la red. Sin embargo, presenta el problema del acoplamiento y no considera el tratamiento de conflictos.

Así mismo, en [35], se considera la distribución de una base de datos central, usando una arquitectura que facilita la comunicación con diferentes almacenes y modelos de datos, desplegando un adaptador específico para cada caso. Aunque soluciona la dependencia con los motores y estructuras de datos, la lógica de negocio está fuertemente ligada a la lógica de sincronización.

En [36], se describe un Framework basado en Java con tres componentes principales: los almacenes de sincronización, los sincronizadores, y los gestores de almacén en los clientes, define un protocolo que usa XML para el intercambio de datos, e incluye un método para el manejo de versiones de los datos con el objetivo de garantizar la disponibilidad y la consistencia, sin embargo, esto puede generar redundancia en los datos.

En [48], se presenta una estrategia de replicación que permite mantener la consistencia de los datos en sistemas distribuidos a gran escala, que implican un gran número de cambios simultáneos implementando una arquitectura de tres niveles, un protocolo de propagación de actualizaciones, y un método de replicación.

En [49], se propone un nuevo modelo de sincronización y replicación para optimizar el tratamiento de conflictos de datos con un mecanismo de preprocesamiento basado en información del usuario en cuestión y las transacciones relacionadas. Este es ejecutado antes de la detección y resolución de conflictos, ahorrando recursos en el cliente móvil y ancho de banda, reduce las cargas de trabajo, disminuye la complejidad del algoritmo de detección y resolución de conflictos y utiliza el protocolo SyncML.

Otros trabajos, como [37], [38], [39], [40], [41], [42] y [43], no consideran el uso de servicios web, sin embargo representan una base conceptual importante para entender la filosofía de la replicación de datos en temas como la identificación y resolución de conflictos, arquitecturas de referencia, protocolos de sincronización, seguridad, gestión de conexiones, optimización de operaciones, gestión de modelos de datos dinámicos y diferentes recomendaciones sobre las bases de datos móviles.

### 3 Definición de los lineamientos

En este trabajo, se definen un conjunto de lineamientos para la construcción de aplicaciones de sincronización de

bases de datos móviles basadas en servicios Web bajo una aproximación multiplataforma tomando como referencia:

- La adaptación al mundo móvil de conceptos, recomendaciones y especificaciones sobre sincronización de datos de algunas de las empresas con mayor reconocimiento a nivel mundial en el sector de las tecnologías de la información y las comunicaciones, como ORACLE, IBM y Microsoft.
- Un análisis realizado al estado actual del conocimiento referente a la sincronización de bases de datos móviles y las diferentes tecnologías involucradas en el proceso.
- La realimentación práctica y experimental obtenida durante la implementación de la adaptación del mecanismo de sincronización de bases de datos móviles basado en servicios Web, realizados en este trabajo.

La información detallada del proceso de definición de los lineamientos y su respectiva justificación puede ser consultada en el documento relacionado con este artículo [47].

Fijando este conjunto de lineamientos como punto de partida, se busca optimizar y facilitar el proceso de desarrollo de este tipo de soluciones. En la figura 2, se presenta la estructura de los lineamientos.



Figura 2: Estructura general de los lineamientos.

### 3.1 Lineamientos para la arquitectura del sistema

La arquitectura se refiere principalmente al diseño y especificación de la estructura general del sistema [44][45][46]. Se define una arquitectura de tres niveles conformada por el Back-End, el servidor de sincronización y los clientes móviles. Se recomienda usar un solo sitio maestro, el modo de replicación asíncrona, y se definen algunas funciones importantes que deben ejecutarse en el mediador.

### 3.1.1 Lineamientos para las aplicaciones Back-End

Constituyen una guía sobre las características generales que debe tener una aplicación empresarial o Back-End, para integrarse correctamente en un entorno de sincronización de datos móviles basado en servicios web, como son el tipo de interfaces y las notificaciones que se deben enviar al mediador y los clientes móviles.

### 3.1.2 Lineamientos para las aplicaciones móviles

Los lineamientos para el desarrollo de las aplicaciones móviles cliente constituyen una guía sobre las características generales que debe tener una aplicación móvil cliente para integrarse correctamente y consumir los servicios de un entorno de sincronización de datos móviles basado en servicios web. Estos, hacen referencia al desarrollo multiplataforma, el uso de una base de datos réplica local en cada móvil, el almacenamiento cifrado de datos en los dispositivos, el uso de SmartPhone, Tabletas y un método de eliminación suave de registro para evitar conflictos de eliminación.

### 3.1.3 Lineamientos para el servidor de sincronización

Los lineamientos para el servidor de sincronización, constituyen una guía sobre las características generales que éste debe cumplir para soportar correctamente un entorno de sincronización de datos móviles, basado en servicios web. Algunos de ellos son: Proveer un mecanismo de des-aprovisionamiento de datos y aplicaciones en los dispositivos, contar diferentes tipos de sincronización, exponer la lógica de sincronización como una API para facilitar el desarrollo y extensión de las aplicaciones, además de modos de sincronización, como la Push y Bulk o mayor, que permiten optimizar el proceso al consumir menos recursos de red y de procesamiento.

### 3.1.4 Lineamientos para el tratamiento de conflictos de datos

En un entorno de sincronización de datos móviles basado en servicios Web, en el que se permite la manipulación de datos en modo desconectado y las actualizaciones bidireccionales por parte de los clientes móviles, es inevitable que se presenten inconsistencias como los conflictos de datos. Para dar un tratamiento adecuado a éstos, se definieron algunos mecanismos que permiten identificar los conflictos de eliminación, actualización y unicidad. Además, se definen métodos para evitar y resolver conflictos de datos en un entorno de replicación. Éstos son presentados en la figura 3.

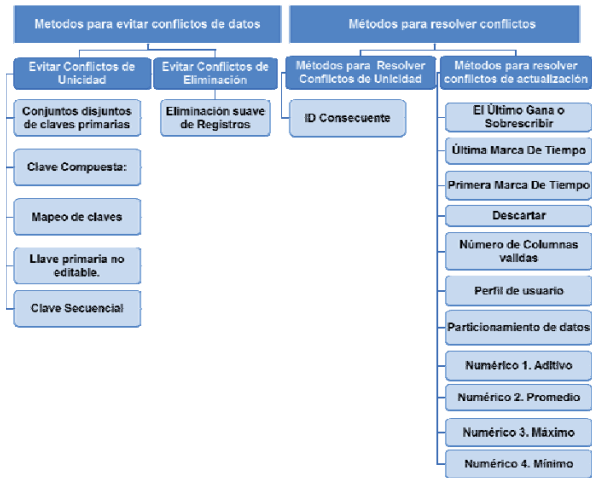


Figura 3: Estructura de los métodos para evitar y resolver conflictos.

Los mecanismos para evitar y resolver conflictos definen políticas que el sistema debe aplicar para mantener la coherencia de los datos cuando una o más acciones conflictivas se han llevado a cabo en diferentes clientes o en el sitio maestro. Por ejemplo, la eliminación de registros puede ser manejada a través de un borrado suave que consiste en marcar los registros a eliminar para que sean borrados posteriormente en una operación de purga, de manera que no se eliminan directamente de la base de datos central, y estarán disponibles si otro cliente realiza una operación sobre dicho registro y así evitar conflictos. Los detalles de las diferentes técnicas definidas en este trabajo pueden ser consultados en [47].

Por último, los conflictos de datos que no se puedan evitar, identificar o resolver, deben ser registrados en una bitácora o log que permita a un usuario administrador resolver estas inconsistencias manualmente. La lógica para el tratamiento de conflictos debe ser implementada en el servidor de sincronización, ya que es el encargado de procesar las operaciones de sincronización.

## 3.2 Lineamientos para la gestión de conexiones

La conexión se refiere a los enlaces o relaciones entre los distintos componentes de la arquitectura general del sistema de sincronización de bases de datos móviles basada en servicios Web. Se recomienda el uso de servicios Web REST para el intercambio de información entre los componentes, el manejo de sesiones de usuario para el acceso a la lógica de sincronización de datos, y se establecen algunas recomendaciones de seguridad, por ejemplo el uso de protocolos seguros como HTTPS.

## 3.3 Arquitectura de referencia

La Figura 4 ilustra la arquitectura de referencia de un sistema de sincronización de bases de datos móviles basado en servicios Web, dividida en 3 capas, en cada una de éstas se muestra las características más importantes que debe tener un sistema de sincronización de datos móviles de acuerdo con los lineamientos planteados.

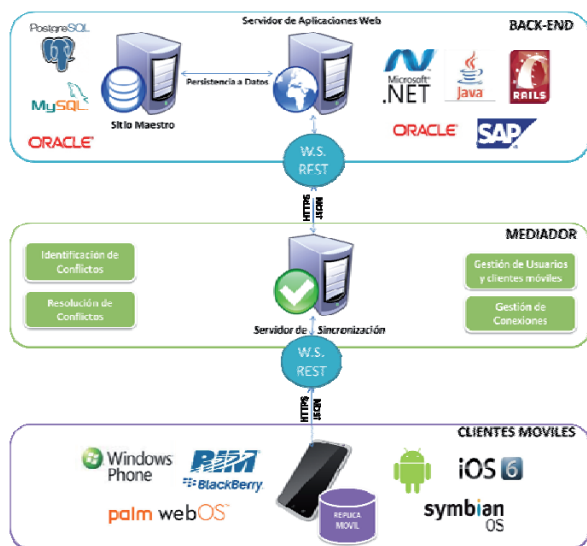


Figura 4: Arquitectura de referencia.

#### 4 Adaptación de un Framework

El proceso de adaptación de un Framework de sincronización de bases de datos móviles se desarrolló de acuerdo con los lineamientos planteados en la Sección 3, y se describe en los siguientes pasos.

##### 4.1 Selección del Framework

Se realizó el proceso de selección para cada una de las tres capas de acuerdo con los lineamientos planteados.

###### 4.1.1 Servidores Empresariales - BackEnd

En teoría, es indiferente el lenguaje de programación y la plataforma de software que se utilice para desarrollar las aplicaciones de esta capa. La única condición que se debe cumplir, es exponer un servicio Rest o SOAP que permita el acceso a lógica de negocio y a la base de datos central.

###### 4.1.2 Mediador - Servidor de Sincronización

Para la selección del Framework en la capa del servidor de sincronización, se compararon tres proyectos existentes: RhoSync, RhoConnect y Microsoft Sync Framework.

Tabla 1: Framework's de sincronización móvil.

CRITERIO	RhoSync	RhoConect	Microsoft Sync Framework
Código Abierto	SI	NO	NO
Software Libre	GNU General Public License	Comercial	Libre / Comercial
Entorno de desarrollo	No Tiene	Rhostudio	VisualStudio
Lenguaje de programación	RUBY	RUBY	C# - VB.NET
Sistemas Operativos compatibles	Linux, Mac, Windows	Linux, Mac, Windows	Windows

Debido a los requerimientos de este trabajo, el Framework seleccionado es RhoSync dando prioridad al criterio de código abierto y en segunda instancia a la

licencia de distribución, ya que permiten realizar la adaptación.

##### 4.1.3 Aplicaciones Cliente

Para las aplicaciones cliente, se tiene en cuenta principalmente la compatibilidad e integración con el servidor de sincronización seleccionado. Por este motivo, se elige a Rhodes como el framework de desarrollo móvil, que además cuenta con soporte multiplataforma permitiendo desarrollo para Android, iOS, RIM y Windows Phone.

#### 4.2 Diagnóstico del Framework

En la tabla 2, se presenta un resumen de la confrontación de los lineamientos definidos con las características de los Framework's seleccionados.

Tabla 2: Resumen de la evaluación.

LINEAMIENTO		PRESENTE	
		SI	NO
<b>Arquitectura General del Sistema</b>	Arquitectura de Tres Niveles def tres niveles	X	
	Funciones del Mediador		X*
	Sitio Maestro Único	X	
	Replicación Asíncrona – Modo desconectado	X	
<b>Gestión de las Conexiones</b>	Seguridad en el transporte de datos	X	
	Uso de servicios Web REST	X	
	Manejo de Sesión para la Sincronización	X	
<b>Aplicaciones del Back-End</b>	Interface de acceso a datos y lógica de negocio		
	Notificaciones		
<b>Aplicaciones Multiplataforma</b>	Desarrollo Multiplataforma	X	
	Uso de Smartphone y Tablet	X	
	Base de datos local	X	
	Almacenamiento cifrado de los datos	X	
	Manejo de Sesión		X**
<b>Servidor de Sincronización</b>	Des-aprovisionamiento		X
	Tipos de Sincronización	X	
	API de Sincronización	X	
	Sincronización push	X	
	Sincronización Mayor	X	
	Sincronización Stream	X	
<b>Tratamiento de conflictos de datos</b>	Identificación de Conflictos		X
	Metodos para evitar Conflictos		X
	Métodos de Resolución de conflictos		X
	Registro de los conflictos o Cola de Error		X

\*Cumple solo algunas funciones.

\*\*Depende del método que se requiera implementar en el entorno de sincronización.

### 4.3 Definición de los requisitos de adaptación

A partir del diagnóstico, se describen los requerimientos de la adaptación enfocados principalmente al tratamiento de conflictos de datos, como se presentan a continuación:

**Requerimiento 1:** Implementar un mecanismo que permita identificar los diferentes conflictos de datos.

**Requerimiento 2:** Implementar un mecanismo que contribuya a evitar los posibles conflictos de datos.

**Requerimiento 3:** Implementar un mecanismo que permita dar un tratamiento a los conflictos de datos y resolverlos de acuerdo con el método previamente establecido.

**Requerimiento 4:** Implementar una cola de error que permita registrar los conflictos que no se puedan resolver.

**Requerimiento 5:** Implementar un entorno de desarrollo para facilitar la creación, depuración y despliegue de las aplicaciones Rhosync.

### 4.4. Implementación de la adaptación

A partir de los requisitos planteados, se realizó una adaptación al framework seleccionado denominada cRhosync, disponible en: <http://goo.gl/DgbxbY> y algunos cambios a RhoDes, debido a que el trabajo se centra en el tratamiento de conflictos. La figura 5, resume el esquema general de la adaptación que se describe a continuación.



Figura 5 : Esquema de la adaptación.

#### 4.4.1 Identificación de conflictos

En la adaptación, se implementaron métodos para la identificación de conflictos de unicidad, actualización y eliminación, en cada uno de los modelos o “tablas” replicadas, gracias a una base de datos desplegada en el servidor de sincronización y la implementación de un método de control de versiones de los datos que usa marcas de tiempo agregadas a cada registro en el cliente.

#### 4.4.2 Resolución conflictos

La adaptación cuenta con los siguientes métodos para la resolución de conflictos de unicidad y actualización:

- Conflictos de unicidad: Id Consecuente.
- Conflictos de actualización: Marca De Tiempo Más Reciente, Marca Más Antigua De Tiempo, Usuario Propietario, Sobre-Escritura, Descarte, Perfil De Usuario y Cantidad De Información valida; y los métodos numéricos: Promedio, Máximo, Mínimo y Suma.

Debido a que los conflictos de eliminación no pueden ser resueltos, en estos casos se registra el conflicto en Log o bitácora implementados y se notifica al cliente.

En la tabla 3 se describe el proceso de identificación y resolución de conflictos de unicidad que ejecuta cRhoSync.

Tabla 3 : Identificación y Resolución conflictos de unicidad.

	SITIO MAESTRO	CLIENTE A	CLIENTE B	Tabla Conflictos cRhoSync
U1		R1={PK=1}(Offline)	(Offline)	
U2		R1={PK=1}(Offline)	R2={PK=1}(Offline)	
U3	R1={PK=1}	R1={PK=1}(Online)	R2={PK=1}(Offline)	R1={PK=1}
U4	R1={PK=1}	R1={PK=1}(Offline)	R2={PK=1}(Online)	R1={PK=1}
U5	R1={PK=1}	R1={PK=1}(Offline)	R2={PK=1}(Online)	R1={PK=1}
U6	R1={PK=1}	R1={PK=1}(Offline)	R2={PK=1}(Online)	R1={PK=1} R2={PK=2}
U7	R1={PK=1} R2={PK=2}	R1={PK=1} Se sincronizará en la siguiente conexión. (Offline)	R1={PK=1} R2={PK=2} (Offline)	R1={PK=1} R2={PK=2}

La descripción de las acciones consignadas en la primera columna de la Tabla 3 se presentan a continuación.

- **U1:** Cliente A crea un registro R1 con clave primaria PK=1.
- **U2:** Cliente B crea un segundo registro R2 con la misma clave primaria PK=1.
- **U3:** Cliente A Sincroniza exitosamente sus datos con el sitio maestro.
- **U4:** Cliente B envía solicitud de sincronización de datos.
- **U5:** El servidor de sincronización detecta un conflicto de unicidad, debido a que ya existe un registro con la clave primaria igual a 1 en la tabla de conflictos de cRhoSync.
- **U6:** El servidor de sincronización resuelve el conflicto de acuerdo con el método definido para este fin (en este caso ID consecuente).
- **U7:** El servidor de sincronización inserta el registro en el sitio central e informa al cliente de los cambios realizados.

La tabla 4 muestra el proceso de identificación y resolución de un conflicto de actualización.

Tabla 4: Identificación y resolución de conflictos de actualización.

	SITIO MAESTRO	CLIENTE A	CLIENTE B	Tabla de Conflictos cRhoSync
A1	R1= Carlos	R1= Carlos (Offline)	R1= Carlos (Offline)	R1= Carlos
A2	R1= Carlos	R1= Juan (Offline)	R1= Carlos (Offline)	R1= Carlos

A3	R1= Carlos	R1= Juan (Offline)	R1 = Pedro (Offline)	R1= Carlos
A4	R1= Juan	R1= Juan (Online)	R1 = Pedro (Offline)	R1 = Juan
A5	R1 = Juan	R1= Juan (Offline)	R1 = Pedro (Online)	R1 = Juan
A6	R1 = Pedro	R1= Juan (Offline)	R1 = Pedro (Online)	R1 = Pedro

La descripción de las acciones consignadas en la primera columna de la Tabla 3, se presentan a continuación.

- A1: Convergencia de datos, en todos los sitios en el valor: R1= Carlos.
- A2: Cliente A actualiza a R1= Juan.
- A3: Cliente B actualiza a R1 = Pedro.
- A4: Cliente A sincroniza exitosamente los datos con el Sitio Maestro.
- A5: Cliente B envía una solicitud de sincronización de datos con el Sitio Maestro. Como la versión anterior del cliente B (R1=Carlos) no coincide con la versión actual de la tabla de conflictos (R1=Juan), se detecta un conflicto de actualización.
- A6: El servidor de sincronización resuelve el conflicto con el método de resolución definido, en este caso (última marca de tiempo) como  $T3 > T2$  gana el registro modificado por el cliente B.

#### 4.4.3 Evitar Conflictos

La adaptación cuenta con mecanismos para evitar conflictos de unicidad y eliminación. En el caso de los conflictos de eliminación, se dispone del método de borrado suave de registros. Para los de unicidad, cuenta con el método de clave secuencial. Los conflictos de actualización no se pueden evitar.

#### 4.4.4 Registro de Conflictos

En caso de detectar un conflicto de datos y que por alguna razón no pueda ser resuelto, como es el caso de los de eliminación, se registra en el log o bitácora de error de manera que un administrador del sistema los pueda resolver manualmente.

#### 4.4.5 Plug-in para cRhosync

Las principales funcionalidades del IDE implementado para el desarrollo de aplicaciones cRhoSync, son:

- Creación de proyectos cRhoSync a través de un asistente.
- Creación de adaptadores fuente a través de un asistente.
- Editar y depurar códigos ruby.
- Despliegue de aplicaciones cRhoSync.

Disponible en: <http://goo.gl/4IUgTQ>.

## 5 Pruebas a la adaptación

### 5.1 Pruebas de confidencialidad

Con el ánimo de poner a prueba el Framework adaptado, se realizó un piloto en el contexto de las actividades de recolección de información en campo de la empresa

UTEN seccional Cauca, encargada de la operación del sistema eléctrico del departamento del Cauca-Colombia, en los procesos de generación, distribución y comercialización de energía. Para el cual se utilizó una arquitectura general, como se muestra en la figura 6.



Figura 6: Arquitectura general del piloto.

Se ejecutó un caso de prueba para cada tipo de conflicto con el ánimo de confirmar el funcionamiento de los métodos desarrollados en la adaptación cRhoSync.

Los resultados obtenidos confirman el correcto funcionamiento de los procedimientos de detección, resolución y registro de conflictos implementados, tareas que no ejecuta formalmente el Framework original RhoSync. Para una descripción más detallada, se puede consultar la monografía relacionada con este trabajo [47].

### 5.2 Pruebas de desempeño del prototipo

Para la ejecución de las pruebas de desempeño a la adaptación, se plantearon dos objetivos iniciales que dieron pie a los escenarios y tipos de pruebas. Estos son:

- Demostrar que el proceso de adaptación del Framework de sincronización, no influyó en los tiempos de respuesta respecto con el Framework original.
- Realizar un análisis comparativo del ancho de banda consumido, entre la representación con XML y Json para los datos intercambiados entre el servidor de sincronización y el Back-End, con el ánimo de enfatizar en la ventaja de usar Json como representación de los datos en ambientes móviles distribuidos.

Para cumplir con estos objetivos, se ejecutaron dos pruebas. La primera consta de una prueba de carga al servidor de sincronización desde clientes simulados con la herramienta JMeter, comparando cRhosync y Rhosync. La segunda prueba compara el ancho de banda utilizado por la conexión entre un servicio de sincronización cRhoSync y el Back-End haciendo una confrontación en la representación de los datos entre xml y json.

#### 5.2.1 Resultados obtenidos de la prueba 1 – Carga al servidor de sincronización

Las tablas presentadas a continuación, resumen los resultados de las pruebas en los escenarios 1 y 2, correspondientes a RhoSync y cRhoSync.

Las pruebas se ejecutaron en tres contextos para cada escenario, con 250, 500 y 1000 usuarios, respectivamente. Para este documento, se presentan los datos promedios de las tres repeticiones por prueba que se ejecutaron. El dato de rendimiento representa las peticiones por segundo y los demás datos son tiempos en ms (milisegundos).

**Resultados para el Escenario 1.**

Tabla 5: Resultados Contexto 1, escenario 1.

Muestras	Media	Mín.	Máx.	Desviación estándar.	Rendimiento
250	58	35	249	21,91268078	0,83631899

Tabla 6: Resultados Contexto 2, escenario 1.

Muestras	Media	Mínimo	Máximo	Desviación estándar.	Rendimiento
492	61	30	326	23,8183776	1,557317

Tabla 7: Resultados Contexto 3, escenario 1.

Muestras	Media	Mínimo	Máximo	Desviación estándar.	Rendimiento
964	68	29	241	28,6154035	3,216537

**Resultados para el Escenario 2.**

Tabla 8: Resultados Contexto 1, escenario 2.

Muestras	Media	Mínimo	Máximo	Desviación estándar.	Rendimiento
250	73	47	234	21,1462019	0,836213

Tabla 9: Resultados Contexto 2, escenario 2.

Muestras	Media	Mín.	Máx.	Desviación estándar.	Rendimiento
490	77	34	791	45,00979436	1,63524151

Tabla 10: Resultados Contexto 3, escenario 2.

Muestras	Media	Mín.	Máx.	Desviación estándar.	Rendimiento
957	82	32	736	60,62473668	3,19098231

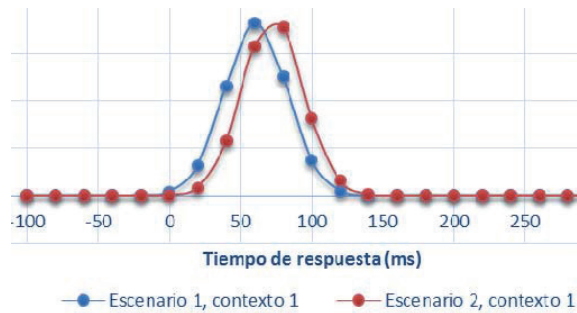


Figura 7: Distribución de probabilidad de tiempos de respuesta contexto 2.

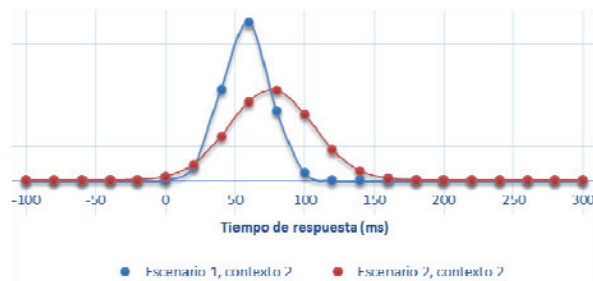


Figura 8: Distribución de probabilidad de tiempos de respuesta contexto 2.

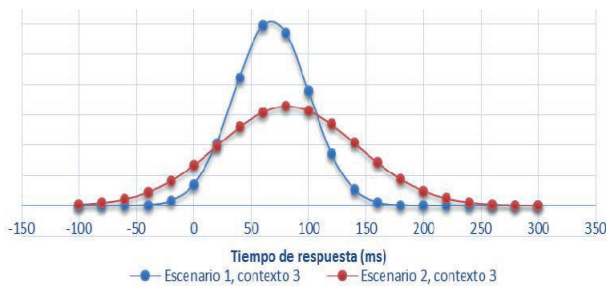


Figura 9: Distribución de probabilidad de tiempos de respuesta contexto 3.

Nota: En la figura 9, se presentan probabilidades para tiempos de respuesta negativos, debido a que la prueba está hecha con una muestra de datos limitada que arroja un comportamiento aproximado de la distribución de probabilidad. Pero hay que aclarar que no se pueden tener tiempos de respuesta negativos en la implementación real.

Se puede observar que el rendimiento es similar comparando los dos escenarios. Por lo tanto, el número de peticiones por segundo que puede atender un servicio desarrollado con cRhoSync es similar al que puede atender un servicio RhoSync.

Otra conclusión importante tiene que ver con los tiempos de respuesta para los escenarios. Se puede notar que para el caso de cRhosync, hay un aumento proporcional entre el número de usuarios y los tiempos de respuesta, sin embargo, las diferencias de las medias entre contextos de los dos escenarios no son grandes la mayor de ellas en el contexto 2 es de 16ms.

**5.2.2 Resultados de la prueba 2 – Ancho de banda**

Los datos consignados en figura 10, representan los promedios de las tres repeticiones que se ejecutaron para cada escenario, usando una representación de datos en XML y Json, respectivamente.

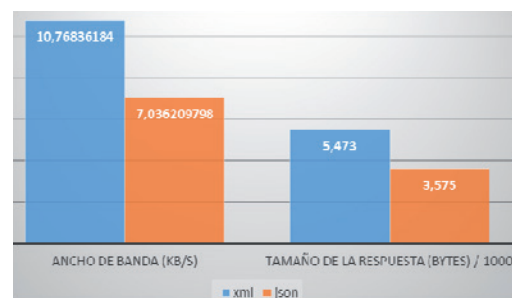


Figura 10: Resumen prueba ancho de banda.

En la figura 10, se observa la ventaja de usar Json para un menor consumo de ancho de banda. Se puede calcular que tanto el tamaño de las respuestas y el ancho de banda consumido con Json, es aproximadamente un 34,7% menor que el representado por los datos en XML.

**6 Conclusiones y trabajos futuros**

Los lineamientos de sincronización definidos en este trabajo representan una guía para optimizar y facilitar la construcción de sistemas de sincronización de datos en ambientes móviles basados en servicios web, no solo con el uso del framework cRhoSync, sino para la



implementación de este tipo de servicios con otras características y haciendo uso de diversas tecnologías.

En un sistema de sincronización de datos móviles con soporte a operaciones desconectadas, es importante contar con mecanismos de detección y resolución de conflictos de datos con el objetivo de evitar inconsistencias y pérdidas de información. En el desarrollo de este trabajo, se evidenció que la implementación de este tipo de mecanismos garantiza la convergencia en los datos y un tratamiento confiable y consistente de información.

La configuración de los métodos de resolución de conflictos de datos depende directamente de las características particulares de cada entorno de sincronización; por esta razón, en el presente trabajo, se definen diferentes métodos, de manera que el usuario administrador pueda definir el mecanismo más adecuado para cada escenario.

El uso de un mediador en una arquitectura de sincronización de datos móviles, facilita el despliegue de este tipo de soluciones, ya que permite separar la lógica de negocio, acceso y datos de la lógica de sincronización. Además, facilita la implementación de procesos como: la gestión de usuarios y conexiones, el tratamiento de conflictos y genera clientes más ligeros.

El uso de servicios Web Rest con datos en formato Json optimiza el consumo de ancho de banda en la transferencia de información, reduce los tiempos de respuesta y, además, reduce complejidad al proceso de interpretación de los datos; por lo tanto, sistemas como cRhosync, que hacen uso de estas tecnologías, son muy eficientes en estos aspectos, ideal en un ambiente con clientes móviles.

Los sistemas de sincronización de bases de datos en ambientes móviles plantean diferentes retos en términos de seguridad de los datos, fragmentación de las plataformas móviles, heterogeneidad en las fuentes de datos, el acceso y los conflictos de datos en ambientes fuera de línea. Por lo tanto, es importante definir y seguir políticas como las planteadas en los lineamientos de este trabajo.

Mediante la adaptación del framework y la implementación del piloto, se evidenció la funcionalidad de aplicar los lineamientos a un sistema de sincronización móvil basada en servicios web para un entorno real por la facilidad y versatilidad que aportan al proceso de desarrollo y despliegue.

### Trabajos futuros

Adaptar el Framework cRhoSync para soportar aplicaciones cliente nativas, o desarrolladas en otros Framework's multiplataforma diferentes a Rhodes.

Adaptar el Framework cRhoSync para soportar las tareas de des-aprovisionamiento de datos y aplicaciones teniendo en cuenta los requerimientos de seguridad de las aplicaciones empresariales.

Construcción de una herramienta basada en meta-información que permita parametrizar un entorno de sincronización, y obtener así información que ayude en

una implementación adecuada del servicio de sincronización.

### Referencias bibliográficas

- [1] M. Lennighan, «Number of phones exceeds population of world», 20-may-2011. [En línea]. Disponible en:  
<http://www.totaltele.com/view.aspx?ID=464922>. [Accedido: 13-mar-2012].
- [2] A. B. Waluyo, B. Srinivasan, y D. Taniar, «Research in mobile database query optimization and processing», *Mobile Information Systems*, vol. 1, n.o 4, pp. 225-252, ene. 2005.
- [3] Mi-Seon Choi, Young-Kuk Kim, y Juno Chang, «Transaction-centric split synchronization mechanism for mobile e-business applications», en *International Workshop on Data Engineering Issues in E-Commerce*, 2005. *Proceedings*, 2005, pp. 112-118.
- [4] Y. Natchetoi, V. Kaufman, y A. Shapiro, «Service-oriented architecture for mobile applications», en *Proceedings of the 1st international workshop on Software architectures and mobility*, New York, NY, USA, 2008, pp. 27-32.
- [5] *E-business 2.0: Roadmap for Success*. Addison-Wesley Professional, 2001.
- [6] S. Mall, T. Stefanov, y S. Stadelman, *Mobilizing Your Enterprise with SAP*. SAP Press, 2012.
- [7] H. H., Eric Lai, Lori Jo Piquet y Sybase, *Enterprise Mobility Guide 2011*. Sybase.
- [8] *Yankee Guide to Successfully Deploying Enterprise Mobile Applications pdf free ebook download*.
- [9] C. Kleisath y O. Rege, *The Business Value Roadmap to Mobilized Software Solutions*.
- [10] R. C. Basole, «The Emergence of the Mobile Enterprise: A Value-Driven Perspective», en *Management of Mobile Business*, 2007. *ICMB 2007. International Conference on the*, 2007, pp. 41-41.
- [11] E. Jamil, «SOA in Asynchronous Many-to-One Heterogeneous Bi-Directional Data Synchronization for Mission Critical Applications», *WeDoWebSphere*, Jul, 2009. Springer Berlin / Heidelberg, 2000, pp. 287-298.
- [12] Mi-Young Choi, Eun-Ae Cho, Dae-Ha Park, Jong-Youn Bae, Chang-Joo Moon, y Doo-Kwon Baik, «A Synchronization Algorithm of Mobile Database for Ubiquitous Computing», en *Fifth International Joint Conference on INC, IMS and IDC*, 2009. *NCM '09*, 2009, pp. 416-419.
- [13] A. Laganà, *Computational science and its applications: ICCSA 2004, international conference*, Assisi, Italy, May 14-17, 2004: *proceedings*. Springer, 2004.
- [14] M.-S. Choi y Young-Guk Kim, «Introduction of mobile database and research status», *Journal of Database Research*, vol. 17, n.o 3, pp. 3-16.
- [15] J. H. Abawajy, M. M. Deris, y M. Omar, «A novel data replication and management protocol for

- mobile computing systems», *Mobile Information Systems*, vol. 2, n.o 1, pp. 3-19, ene. 2006.
- [16] Y. Saito y M. Shapiro, «Optimistic replication», *ACM Comput. Surv.*, vol. 37, n.o 1, pp. 42–81, mar. 2005.
- [17] J.-L. Li y J. Li, «Data synchronization protocol in mobile computing environment using SyncML and Huffman coding», en *2012 International Conference on Wavelet Active Media Technology and Information Processing (ICWAMTIP)*, 2012, pp. 260-262.
- [18] «Replicación de SQL Server». [En línea]. Disponible en: <http://msdn.microsoft.com/es-es/library/ms151198.aspx>. [Accedido: 18-abr-2013].
- [19] N. Preguiça, M. Shapiro, y C. Matheson, «Semantics-Based Reconciliation for Collaborative and Mobile Environments», en *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, R. Meersman, Z. Tari, y D. C. Schmidt, Eds. Springer Berlin Heidelberg, 2003, pp. 38-55.
- [20] J. Cho y H. Garcia-Molina, «Synchronizing a database to Improve Freshness», 1999. [En línea]. Disponible en: <http://ilpubs.stanford.edu:8090/396/>. [Accedido: 06-jul-2013].
- [21] «Sincronizar datos». [En línea]. Disponible en: <http://msdn.microsoft.com/es-es/library/ms151793.aspx>. [Accedido: 18-abr-2013].
- [22] M. Wiesmann y A. Schiper, «Comparison of database replication techniques based on total order broadcast», *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, n.o 4, pp. 551-566, 2005.
- [23] E. Cecchet, G. Candea, y A. Ailamaki, «Middleware-based database replication: the gaps between theory and practice», en *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, New York, NY, USA, 2008, pp. 739–752.
- [24] M. Wiesmann, F. Pedone, A. Schiper, B. Kemme, y G. Alonso, «Understanding replication in databases and distributed systems», en *20th International Conference on Distributed Computing Systems*, 2000. *Proceedings*, 2000, pp. 464-474.
- [25] «Introduction to Advanced Replication». [En línea]. Disponible en: [http://docs.oracle.com/cd/E14072\\_01/server.112/e10706/repoverview.htm#i19713](http://docs.oracle.com/cd/E14072_01/server.112/e10706/repoverview.htm#i19713). [Accedido: 18-abr-2013].
- [26] N. Schiper, R. Schmidt, y F. Pedone, «Optimistic Algorithms for Partial Database Replication», en *Principles of Distributed Systems*, M. M. A. A. Shvartsman, Ed. Springer Berlin Heidelberg, 2006, pp. 81-93.
- [27] R. G. Guy, G. J. Popek, T. W. Page, y Jr, «Consistency Algorithms for Optimistic Replication», en *In Proceedings of the First International Conference on Network Protocols*. IEEE, 1993.
- [28] «Control de concurrencia en ADO.NET». [En línea]. Disponible en: [http://msdn.microsoft.com/es-es/library/22hhd212\(v=vs.80\).aspx](http://msdn.microsoft.com/es-es/library/22hhd212(v=vs.80).aspx). [Accedido: 19-abr-2013].
- [29] Joseph Debuzna - Product Manager, Oracle y Bharath Aleti - Software Development Manager, Oracle, «Best Practices for Conflict Detection and Resolution in Oracle GoldenGate for Active/Active», presentado en Oracle OpenWorld.
- [30] F. D. Borda Luciani, E. O. Gagliardi, y G. Hernández Peñalver, «Sincronización de datos entre aplicaciones sobre redes móviles», presentado en VIII Workshop de Investigadores en Ciencias de la Computación, 2006.
- [31] John Garmany y Robert Freeman, «Multi-Master Replication Conflict Avoidance and Resolution». Rampant TechPress. 2004.
- [32] Chris Kleisath y Ojas Rege, «The Developer's Roadmap», en *MOBILIZED SOFTWARE SOLUTIONS*.
- [33] Ejaz Jamil, «Emerging SOA Appliances».
- [34] Krishnaa G, «Light Weight Architecture for Remote DB Synchronization on Windows Mobile Devices using Web Services», SRM University, may 2008.
- [35] W. Kurschl, S. Mitsch, y R. Prokop, «SmartDOTS - A Framework for Efficient Data Synchronization on Mobile Devices», en *Third International Conference on Information Technology: New Generations*, 2006. *ITNG 2006*, 2006, pp. 300-305.
- [36] N. Cohen, «A Java Framework for Mobile Data Synchronization», en *Cooperative Information Systems*, vol. 1901, P. Scheuermann y O. Etzion, Eds. Springer Berlin / Heidelberg, 2000, pp. 287-298.
- [37] H. Larkin, «Applying Concurrent Versioning to Serverless Mobile Device Synchronisation», en *6th IEEE/ACIS International Conference on Computer and Information Science*, 2007. *ICIS 2007*, 2007, pp. 157-162.
- [38] Y. Saito y M. Shapiro, «Replication: Optimistic Approaches», 2002.
- [39] V. . Zuikevi\veit\le y F. Pedone, «Conflict-aware load-balancing techniques for database replication», en *Proceedings of the 2008 ACM symposium on Applied computing*, New York, NY, USA, 2008, pp. 2169–2173.
- [40] S. C. Choi y H. Y. Youn, «Dynamic hybrid replication effectively combining tree and grid topology», *J. Supercomput.*, vol. 59, n.o 3, pp. 1289–1311, mar. 2012.
- [41] Y. Xia y A. (Sumi) Helal, «A dynamic data/currency protocol for mobile database design and reconfiguration», en *Proceedings of the 2003 ACM symposium on Applied computing*, New York, NY, USA, 2003, pp. 550–556.
- [42] D. R. Selvarani y T. N. Ravi, «A survey on data and transaction management in mobile databases», arXiv:1211.5418, nov. 2012.

- [43] Z. Ding, X. Meng, y S. Wang, «A novel conflict detection and resolution strategy based on TLRSP in replicated mobile database systems», en Seventh International Conference on Database Systems for Advanced Applications, 2001. Proceedings, 2001, pp. 234-240.
- [44] E. J. Braude, Software engineering: an object-oriented perspective. John Wiley & Sons, Inc., 2000.
- [45] L. Bass, P. Clements, y R. Kazman, Software architecture in practice. Addison-Wesley Professional, 2012.
- [46] A. Rollings y D. Morris, «Game architecture and design: a new edition», 2003.
- [47] Alejandro Astudillo - Gustavo Álvarez, «Sincronización de Bases de Datos Móviles Basada en Servicios Web Bajo una Aproximación Multiplataforma», Universidad del Cauca, Popayán, 2013. Disponible en: <http://goo.gl/20RnsH>, <http://goo.gl/B1gPw6>.
- [48] A. Ashraf, P. Dominic, D. Durai, A. Azween, y I. Hamidah, «A New Optimistic Replication Strategy for Large-scale Mobile Distributed Database Systems», *International Journal of database management systems*, vol. 2, n.º 4, pp. 86-105, 2010.
- [49] M. Jin, X. Zhou, J. Zhou, X. Gao, y C. Gong, «Strategy of Conflict Preprocessing and Reconciliation for Mobile Databases», en *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM'08. 4th International Conference on*, 2008, pp. 1-6.