

PRÁCTICA EMPRESARIAL

ELABORACIÓN DE COMPONENTES SOFTWARE, REALIZACIÓN DE PRUEBAS Y DOCUMENTACIÓN, EN EL DESARROLLO DE OPCIONES PARA LA CAPTURA DE DATOS Y GENERACIÓN DE REPORTES DEL SISTEMA SIPA, EMPLEANDO TERMINALES MÓVILES

OLMER JOVANI SAAVEDRA ORTEGA

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2006

PRÁCTICA EMPRESARIAL

ELABORACIÓN DE COMPONENTES SOFTWARE, REALIZACIÓN DE PRUEBAS Y DOCUMENTACIÓN, EN EL DESARROLLO DE OPCIONES PARA LA CAPTURA DE DATOS Y GENERACIÓN DE REPORTES DEL SISTEMA SIPA, EMPLEANDO TERMINALES MÓVILES

OLMER JOVANI SAAVEDRA ORTEGA

Código: 1981955

Proyecto de grado presentado como requisito
para optar al título de Ingeniero de Sistemas

Tutor

Ing. Gerardo Armando Garavito

Director

Mg. José Cárcamo Sepúlveda

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2006

DEDICATORIA

A Dios.

A mis padres, Efraín y Martha, por ayudarme a ser lo que soy, por el apoyo que me han brindado durante toda la vida.

A mi hermana, por confiar en mí, y a mis sobrinos.

A Maritza, por su compañía e incondicional apoyo, y por ser una gran fuerza para salir adelante.

AGRADECIMIENTOS

Al ingeniero Gerardo Garavito, por darme la oportunidad y confianza para realizar este proyecto, y por su orientación durante todo el desarrollo.

A todos mis amigos durante la carrera.

A los compañeros de VISION LTDA.

A la Escuela de Ingeniería de Sistemas e Informática y en general a la Universidad Industrial de Santander.

TÍTULO:

ELABORACIÓN DE COMPONENTES SOFTWARE, REALIZACIÓN DE PRUEBAS Y DOCUMENTACIÓN, EN EL DESARROLLO DE OPCIONES PARA LA CAPTURA DE DATOS Y GENERACIÓN DE REPORTES DEL SISTEMA SIPA, EMPLEANDO TERMINALES MÓVILES*

AUTOR:

SAAVEDRA ORTEGA, Olmer Jovani**

PALABRAS CLAVES:

Dispositivos móviles de alta resistencia, sincronización, SIPA.

DESCRIPCIÓN:

El presente trabajo de grado en modalidad práctica empresarial, tiene como objetivo principal el desarrollo de opciones software en dispositivos móviles de alta resistencia, para el apoyo a la manipulación en campo abierto de la información ambiental manejada por las entidades estatales encargadas de la administración de los recursos medioambientales de nuestro país.

Dichas opciones hacen parte de una extensión al sistema de información ambiental SIPA, propiedad de la empresa VISION LTDA, y pretenden facilitar el proceso de consulta y captura de datos y la generación de documentos de soporte en terreno por parte de los funcionarios encargados de realizar visitas técnicas a los lugares donde a alguna persona o entidad le haya sido otorgado algún derecho o impuesta cierta sanción, según sea el caso, por parte de las autoridades competentes en el tema. Además, se desarrolló la opción para sincronizar los datos recolectados por la terminal móvil con los del servidor de datos central de SIPA, asegurando de esta forma la total coherencia y confiabilidad de la información manejada, tanto en el entorno principal de escritorio, como en el móvil.

Por último se diseñaron y ejecutaron pruebas de funcionalidad a las opciones y se elaboró la documentación requerida por la empresa para ser anexada a los manuales de usuario y técnico de SIPA.

* Proyecto de grado modalidad práctica empresarial.

** Facultad de Ingenierías Físico-Mecánicas, Escuela de Ingeniería de Sistemas e Informática, Director UIS: Mg. José Cárcamo Sepúlveda, Tutor Empresa: Ing. Gerardo Garavito Flórez.

TITLE:

ELABORATION OF COMPONENT SOFTWARE, REALIZATION OF TESTS AND DOCUMENTATION IN THE DEVELOPMENT OF OPTIONS FOR THE CAPTURE OF DATA AND GENERATION OF REPORTS OF THE SYSTEM SIPA, USING MOBILE TERMINALS*

AUTHOR:

SAAVEDRA ORTEGA, Olmer Jovani**

KEY WORDS:

Mobile devices of high resistance, synchronization, SIPA.

DESCRIPTION:

The present grade work in managerial practical modality has as main objective the development of options software in mobile devices of high resistance, for the support to the manipulation in field open of the environmental information managed by the state entities in charge of the administration of the environmental resources of our country.

This options make part from an extension to the system of environmental information SIPA, property of VISION LTDA, and they seek to facilitate the consultation process and capture of data and the generation of support documents in land on the part of the officials in charge of carrying out technical visits to the places where to some person or entity has been granted him some right or imposed certain sanction, as it is the case, on the part of the competent authorities in the topic. Also, the option was developed to synchronize the data gathered by the mobile terminal with those of the central servant of data of SIPA, assuring this way the total coherence and dependability of the managed information, so much in the main environment of desk, like in the mobile.

Lastly tests of functionality to the options were designed and executed and required documentation was elaborated by the company to be annexed user's manuals and technician of SIPA.

* Graduation Project enterprise practical modality.

** Physical-Mechanical Engineering's Department, Systems and Informatics Engineering School, Director UIS: Mg. José Cárcamo Sepúlveda, Tutor Company: Ing. Gerardo Garavito Flórez.

TABLA DE CONTENIDO

INTRODUCCIÓN	19
1. DESCRIPCIÓN DEL PROYECTO.....	2
1.1. OBJETIVOS.....	2
1.1.1. Objetivo General	2
1.1.2. Objetivos Específicos	3
1.1.3. Descripción de Objetivos.....	4
1.2. JUSTIFICACIÓN	5
1.2.1. Definición del problema	5
1.2.2. Impacto.....	6
1.2.2.1. Impacto Técnico	6
1.2.2.2. Impacto Económico	6
1.2.2.3. Impacto social.....	7
1.3. CONOCIMIENTO DE LA EMPRESA	7
1.3.1. Información Corporativa	7
1.3.1.1. Misión.....	7
1.3.1.2. Visión.....	8
1.3.1.3. Política de calidad.....	8
1.4. DESARROLLO DE LAS OPCIONES.....	8
1.4.1. Hardware	8
1.4.1.1. Equipo para desarrollo	8
1.4.1.2. Terminal Móvil.....	9
1.4.2. Software.....	9
1.4.2.1. Equipo para desarrollo	9
1.4.2.2. Terminal Móvil.....	10
2. FUNDAMENTOS TEÓRICOS	11
2.1. NORMATIVIDAD DE TRÁMITES AMBIENTALES EN COLOMBIA	11
2.1.1. Corporaciones Autónomas Regionales.....	11
2.1.1.1. Naturaleza Jurídica	12
2.1.1.2. Objeto	12
2.1.1.3. Funciones.....	12
2.2. DETALLES DE LA TECNOLOGÍA EMPLEADA	15
2.2.1. Sistemas Operativos Windows para dispositivos Móviles	15
2.2.1.1. Windows CE.....	15
2.2.1.2. Windows CE .NET	15
2.2.2. Software de desarrollo.....	17
2.2.2.1. Microsoft Visual Studio .NET	18

2.2.2.2. Microsoft .NET Framework	19
2.2.2.3. Microsoft .NET Compact Framework	21
2.2.2.4. C# o Visual Basic?.....	23
2.2.2.5. Microsoft Visual Basic .NET.....	25
2.2.3. Conceptos básicos sobre Bases de Datos.....	26
2.2.3.1. Componentes de una Base de Datos	27
2.2.3.2. Modelo Entidad-Relación	28
2.2.4. Oracle	29
2.2.5. Oracle Lite.....	30
3. MARCO METODOLÓGICO.....	34
3.1. METODOLOGÍAS ÁGILES DE DESARROLLO.....	34
3.2. EXTREME PROGRAMMING	37
3.2.1. Historias de usuario	38
3.2.2. Proceso	38
3.2.3. Prácticas.....	40
4. REQUERIMIENTOS INICIALES.....	43
4.1. AMBIENTACIÓN	43
4.2. DEFINICIÓN DE LAS OPCIONES	45
4.3. DEFINICIÓN DEL MODELO DE DATOS A UTILIZAR	48
4.4. RELACIÓN CON EL SISTEMA PRINCIPAL SIPA	51
4.4.1. Sistema principal SIPA.....	51
4.4.1.1. Principal Expedientes	51
4.4.1.2. Derechos	52
4.4.1.3. Recursos.....	53
4.4.1.4. Productos ambientales	53
4.4.1.5. Obligaciones	54
4.4.1.6. Visitas ambientales	55
4.5. DESCRIPCIÓN DE INTERFACES.....	55
4.5.1. Inicio.....	55
4.5.2. Principal de Expedientes	56
4.5.3. Consulta de Expedientes.....	57
4.5.4. Visitas ambientales.....	57
4.5.4.1. Detalles de seguimiento a obligaciones	58
4.5.5. Sincronización	59
5. DESARROLLO DE LAS OPCIONES	60
5.1. ÍTEMS PARA EL DISEÑO DE INTERFACES EN DISPOSITIVOS MÓVILES.....	60
5.2. SINCRONIZACIÓN CON ORACLE LITE.....	64
5.2.1. Conceptos generales	65
5.2.2. Proceso de sincronización	66

5.2.3. Métodos de sincronización	68
5.2.3.1. Refrescamiento rápido	69
5.2.4. Errores y conflictos de sincronización.....	71
5.2.5. Usuarios concurrentes	72
5.3. ENTREGAS PRELIMINARES.....	73
5.4. IMPRESIÓN DE SOPORTES DE VISITA.....	74
5.5. PRUEBAS.....	75
5.5.1. Pruebas unitarias.....	76
5.5.2. Pruebas funcionales	77
5.6. DOCUMENTACIÓN.....	81
6. CONCLUSIONES	82
7. RECOMENDACIONES.....	83
BIBLIOGRAFÍA.....	84
ANEXO 1. DEFINICIÓN DE APLICACIONES CON ORACLE LITE.....	86
ANEXO 2. MOBILE SERVER.....	100

LISTA DE TABLAS

TABLA 1. VERSIONES DEL SISTEMA OPERATIVO WINDOWS CE.....	17
TABLA 2. METODOLOGÍAS ÁGILES Y TRADICIONALES	35
TABLA 3. DEFINICIÓN DE REQUISITO	47
TABLA 4. ACTIVIDADES ESPECÍFICAS DEL REQUISITO	48
TABLA 5. MÉTODOS DE SINCRONIZACIÓN DE ORACLE LITE.....	68
TABLA 6. CASOS DE PRUEBA DE LA OPCIÓN DE SINCRONIZACIÓN	78
TABLA 7. PROCEDIMIENTOS DE PRUEBA PARA LA SINCRONIZACIÓN	80
TABLA 8. ESTRUCTURA DE LA VISTA DE CLIENTES	101
TABLA 9. ESTRUCTURA DE LA VISTA ERRORES DE SINCRONIZACIÓN ...	101
TABLA 10. ESTRUCTURA DE LA VISTA DE PUBLICACIONES	101
TABLA 11. ESTRUCTURA DE LA VISTA DE SUSCRIPCIONES	102
TABLA 12. ESTRUCTURA DE LA VISTA DE SECUENCIAS.....	102
TABLA 13. ESTRUCTURA DE LA VISTA PARTICIONES DE SECUENCIAS ..	103
TABLA 14. ESTRUCTURA DE LA VISTA ÍTEMS ADICIONADOS.....	103
TABLA 15. ESTRUCTURA DE LA VISTA DE ÍTEMS DE PUBLICACIÓN	104
TABLA 16. ESTRUCTURA DE LA VISTA DE ÍNDICES	104
TABLA 17. ESTRUCTURA DE PARÁMETROS DE SUSCRIPCIÓN	105

LISTA DE FIGURAS

FIGURA 1. APLICACIÓN DE BASES DE DATOS Y UN DBMS	28
FIGURA 2. COMPONENTES DE ORACLE LITE.	32
FIGURA 3. ARQUITECTURA DE LAS APLICACIONES DE ORACLE LITE.....	67
FIGURA 4. PROCESO DE SINCRONIZACIÓN FAST REFRESH.....	69
FIGURA 5. USUARIOS CONCURRENTES EN ORACLE LITE	72
FIGURA 6. INTERFAZ INICIAL DEL ASISTENTE DE EMPAQUETADO	86
FIGURA 7. PANEL GENERAL DE LA APLICACIÓN.....	87
FIGURA 8. PANEL BASE DE DATOS.....	88
FIGURA 9. PANEL DE INSTANTÁNEAS	89
FIGURA 10. CONEXIÓN AL SERVIDOR DE DATOS	89
FIGURA 11. SELECTOR DE TABLAS PARA CREAR INSTANTÁNEAS.....	90
FIGURA 12. PANEL DE INSTANTÁNEAS	91
FIGURA 13. EDITOR DE SNAPSHOTS (SERVIDOR)	92
FIGURA 14. EDITOR DE SNAPSHOTS (CLIENTE).....	94
FIGURA 15. PANEL DE SECUENCIAS.....	96
FIGURA 16. DEFINICIÓN DE NUEVA SECUENCIA	97
FIGURA 17. PANEL DE DDL.....	98

FIGURA 18. FINALIZACIÓN DE LA DEFINICIÓN DE UNA APLICACIÓN.....	99
FIGURA 19. PUBLICACIÓN DE LA APLICACIÓN	99
FIGURA 20. INGRESO A MOBILE SERVER.....	106
FIGURA 21. SECCIÓN DE MOBILE SERVERS DEL MOBILE MANAGER.....	107
FIGURA 22. SECCIÓN DISPOSITIVOS MÓVILES DEL MOBILE MANAGER .	108
FIGURA 23. INTERFAZ DE ADMINISTRACIÓN DE MOBILE SERVER.....	109
FIGURA 24. PÁGINA DE PROPIEDADES DE UNA APLICACIÓN	110
FIGURA 25. PÁGINA DE USUARIOS DE ORACLE LITE.....	111
FIGURA 26. CREACIÓN DE UN NUEVO USUARIO	112
FIGURA 27. SUSCRIPCIÓN DE UN USUARIO A UNA APLICACIÓN	113
FIGURA 28. VALORES EN LOS PARÁMETROS DE SUSCRIPCIÓN	114
FIGURA 29. PÁGINA DE ADMINISTRACIÓN DE ORACLE LITE	115
FIGURA 30. PÁGINA DE SINCRONIZACIÓN DE LOS DATOS	116
FIGURA 31. PLANIFICADOR DE MGP	117
FIGURA 32. CICLOS DE MGP.....	118

GLOSARIO

ACTIVEX¹: ActiveX es un conjunto de tecnologías de Microsoft que integran componentes Software sin tener en cuenta el lenguaje en el que fueron creados. Un control ActiveX es un componente que puede ser integrado de manera sencilla en cualquier aplicación que los soporte, principalmente en páginas Web. ActiveX ha sido adoptado por fabricantes de Software y es usado hoy en día por millones de desarrolladores.

ACTIVEX DATA OBJECTS (ADO)²: Es un objeto programable para acceder y procesar datos en una base de datos. ADO.NET es la más reciente tecnología de acceso a datos que ahora es parte del .NET Framework. Proporciona acceso a datos relacionales, datos XML y datos de aplicaciones.

API: Abreviatura inglesa para Application Programming Interface, que en español significa Interfaz de programación de aplicaciones. Es el método específico prescrito por un sistema operativo o por cualquier otra aplicación mediante el cual un programador que escribe una aplicación puede hacer solicitudes al sistema operativo o a otra aplicación. Una API puede contrastarse con una interfaz gráfica de usuario (GUI) o una interfaz de comando (ya que ambas son interfaces directas del usuario) como formas de interactuar con un sistema operativo o un programa.

BLUETOOTH³: Es la norma que define un estándar global de comunicación inalámbrica, que posibilita la transmisión de voz y datos entre diferentes equipos

¹ Disponible en <<http://support.microsoft.com>> y <<http://desarrollosweb.com/articulos/993.php?manual=15>>

² Disponible <en <http://www.sqlmax.com/asp1.asp>> y <http://www.programacion.com/bbdd/tutorial/arq_adonet/1/>

³ Disponible en <<http://es.wikipedia.org/wiki/Bluetooth>>

mediante un enlace por radiofrecuencia. La frecuencia de radio con la que trabaja está en el rango de 2.4 a 2.48Ghz. Define un canal de comunicación de máximo 720Kb/seg con rango óptimo de 10 metros. Se utiliza para establecer una vía de interconexión inalámbrica entre diversos aparatos que tengan dentro de si esta tecnología, como celulares, computadoras de mano (Palm, Pocket PC), cámaras, computadoras portátiles, impresoras entre otros.

CUENCA⁴: Una cuenca es el área geográfica conformada por varias fuentes de agua superficial o subterránea que corren hacia una quebrada o río principal, que a la vez desemboca en una corriente de agua mayor, como un río, ciénaga o el mar.

HTML⁵: Es el lenguaje con el que se definen las páginas Web. Básicamente se trata de un conjunto de etiquetas que sirven para definir la forma en la que se presenta el texto y otros elementos de la página.

IP54⁶: Grado industrial de sellamiento. El primer dígito (5) indica que el polvo no afectará la operación del dispositivo. El segundo dígito (6) indica que cualquier proyección de agua desde cualquier dirección no tendrá efectos dañinos sobre la máquina.

JDBC⁷: Abreviatura inglesa utilizada para Java Database Connectivity, que en español significa Conectividad Java para bases de datos. Es una especificación de la interfaz de aplicación de programa (Application Program Interface, API) para conectar programas a los datos en bases de datos populares como Oracle, Informix, SyBase etc. y prácticamente a cualquier dato de tipo tabular. La interfaz

⁴ Disponible en <http://www.col.ops-oms.org/DIAA/2002/MINDES_1_2_consumo.htm>

⁵ Disponible en <<http://www.desarrolloweb.com/articulos/194.php?manual=>>>

⁶ Tomado del manual de usuario Hardware Reference Guide For The WORKABOUT PRO.

⁷ Disponible en <<http://www.mysql-hispano.org/page.php?id=24>> y <<http://www.gamarod.com.ar/recursos/glosario/j.asp>>

de aplicación de programa permite codificar órdenes de solicitud de acceso en lenguaje estructurado de solicitud (Structured Query Language, SQL) que luego pasan al programa que administra la base de datos.

OC4J⁸: Abreviatura inglesa utilizada para Oracle Container For Java. Es un contenedor de objetos rápido, ligero, escalable y de fácil uso, compatible y certificado con la versión 1.3 de Java 2 Enterprise Edition (J2EE). OC4J es 100% puro Java y se ejecuta bajo entornos estándar de Java Development Kit (JDK). Estas características permiten que OC4J se ejecute sobre cualquier plataforma que posea Máquinas Virtuales Java certificadas, sin necesidad de instalar distintas versiones del software.

ODBC⁹: Abreviatura inglesa utilizada para Open Database Connection, que en español significa Conexión abierta a Bases de Datos. Es un estándar de Microsoft para el acceso a bases de datos, independientemente del lenguaje o programa que se esté utilizando. Cada fabricante provee su propia librería con las características de conexión a las bases de datos.

PDA: Abreviatura inglesa utilizada para Personal Digital Assistant, que en español significa Asistente Personal Digital. Son también llamados dispositivos de cómputo móviles o computadores de mano. Se dividen en dos grandes familias, de acuerdo con el sistema operativo que utilizan: Palm y Windows CE. Aunque tienen una apariencia y un funcionamiento diferentes, son afines en muchos aspectos. Ofrecen funcionalidades básicas como agenda, calculadora y directorio telefónico y otras más especializadas como procesador de textos, hoja de cálculo y manejo de bases de datos. Son ideales para el trabajo fuera de las oficinas.

⁸ Disponible en <http://www.oracle.com/pls/wocprod/docs/page/ocom/global/es/expert_services/oracle%20expert%20knowledge%20transfer/oracle%20kt%20j2ee%20oc4j.pdf>

⁹ Disponible en <<http://www.gamarod.com.ar/recursos/glosario/o.asp>>

PL/SQL¹⁰: Abreviatura inglesa de Procedural Language SQL, que en español significa lenguaje SQL procedimental. Es un lenguaje que extiende SQL mediante la incorporación de construcciones que se encuentran en los lenguajes procedurales, tales como variables y tipos, estructuras de control, procedimientos y funciones. A través de PL/SQL se pueden emplear las estructuras de SQL para manipular datos en ORACLE, y estructuras de flujo para procesar los datos. Además se pueden declarar variables y constantes, definir subprogramas y atrapar los errores de ejecución.

POCKET PC: Es un ordenador de bolsillo basado en la plataforma Windows. Permite realizar prácticamente cualquier cosa que se haga con un ordenador de sobremesa (editar textos, hojas de cálculo, retocar imágenes, jugar, conectarse a Internet, jugar, usarlo como navegador), aunque hay que tener en cuenta limitaciones como el tamaño de la pantalla, memoria entre otras.

SERVIDOR: Computador compartido por múltiples usuarios en una red.

SINCRONIZACIÓN: Término que se refiere al intercambio de información entre el dispositivo móvil y un PC, para disponer de información actualizada en ambos equipos.

SODA¹¹: Abreviatura inglesa de Simple Object Data Access. Es una interface optimizada y orientada a objetos para permitir el acceso a datos y funcionalidad SQL de bases de datos Oracle Lite utilizando C++.

SQL¹²: Abreviatura inglesa utilizada para Structured Query Language, que en español significa Lenguaje de consulta estructurado. Es un lenguaje estándar de

¹⁰ Disponible en <<http://www.lania.mx/biblioteca/seminarios/basedatos/plsql/intro/archi01.html>>

¹¹ Tomado de la documentación de Oracle Lite. Oracle Database Lite Developer's Guide 10g (10.0.0). Parte No. B13788-01. Sección 2.2.1.5.

¹² Disponible en <<http://mx.geocities.com/sqlconceptos/usos.htm>>

acceso a datos, utilizado por muchos motores de bases de datos. Las peticiones sobre los datos se expresan mediante sentencias, que deben escribirse de acuerdo con unas reglas sintácticas y semánticas de este lenguaje. Es un lenguaje que se ha visto consolidado por el Instituto Americano de Normas (ANSI) y por la Organización de Estándares Internacional (ISO).

UML¹³: Es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir. UML capta la información sobre la estructura estática y el comportamiento dinámico de un sistema. Un sistema se modela como una colección de objetos discretos que interactúan para realizar un trabajo que finalmente beneficia a un usuario externo.

USB: Puerto de transmisión de datos, mucho más veloz que el puerto serie o el de infrarrojos. A través de este puerto se puede realizar la conexión entre el dispositivo móvil y un ordenador de escritorio.

XML¹⁴: Abreviatura inglesa utilizada para eXtensible Markup Language, que en español significa Lenguaje de marcado extensible. Es una especificación multiplataforma que permite crear etiquetas personalizadas y obtener la funcionalidad que no proporcionan otros lenguajes de marcado utilizados para aplicaciones Web como HTML. Es especialmente útil en el manejo de datos.

¹³ Disponible en < <http://www.creangel.com/uml/intro.php>>

¹⁴ MARUYAMA, Hirochi, TAMURA, Kent y URAMOTO, Naohiko. Creación de sitios Web con XML y JAVA: Prentice may, 2000. p. 3.

INTRODUCCIÓN

En la actualidad muchas organizaciones en el mundo entero se encuentran implementando la aplicación de las nuevas tecnologías de la computación, a los procesos propios de su actividad, especialmente en el manejo de la información. La computación móvil es un nuevo panorama de la tecnología informática, que cada día se desarrolla más. Si las tendencias se mantienen las terminales móviles serán el punto de acceso más utilizado por los usuarios de los servicios de Internet.

La computación móvil se adapta a cualquier entorno empresarial, especialmente a aquellos que requieren la realización de trabajo en campo, pues permite obtener acceso a la información en tiempo real para su procesamiento, en el lugar donde esta se porte. Esta tecnología cuenta con un gran panorama para su aplicación, como por ejemplo el manejo de relaciones con el cliente, fuerza de ventas, manejo de inventarios, retenes, medicina, servicios públicos, facturación en sitio entre otros. En definitiva, la computación móvil da mayor uso a la información residente en los servidores empresariales y se integra fácilmente con las aplicaciones software existentes en una entidad, generando así, valor agregado en las operaciones de las organizaciones.

El propósito de esta práctica empresarial ha sido el desarrollo de opciones software para la captura de datos y manejo de información en campo abierto, mediante la utilización de dispositivos móviles de alta resistencia. Específicamente, el campo de aplicación esta basado en el manejo de ciertos procesos requeridos en los trámites ambientales seguidos por las entidades encargadas del manejo de los recursos medioambientales en nuestro país.

1. DESCRIPCIÓN DEL PROYECTO

En este primer capítulo se presentan las ideas fundamentales de este proyecto en modalidad de práctica empresarial. Se definen aspectos de primordial importancia como son los objetivos que se plantearon al inicio, la justificación del trabajo desarrollado y el impacto esperado por éste.

Los objetivos esenciales de esta primera parte son:

- Formular y describir los objetivos del proyecto.
- Presentar la justificación del proyecto.
- Analizar el impacto del proyecto.
- Formular un marco teórico necesario para el entendimiento de este trabajo de grado.

1.1. OBJETIVOS

1.1.1. Objetivo General

Elaborar componentes software, realizar pruebas y documentar, el desarrollo de opciones para la captura de datos y generación de reportes del sistema SIPA de VISION, empleando terminales móviles, bases de datos Oracle, XML¹⁵, metodología Extreme Programming y siguiendo los lineamientos descritos en el sistema de gestión de calidad de la empresa.

¹⁵ XML. Ver Glosario

1.1.2. Objetivos Específicos

- Adaptar para la base de datos móvil, las tablas a emplear en las opciones software a elaborar, partiendo del análisis y comprensión del modelo existente en el sistema SIPA de VISION.

- Elaborar opciones software utilizando dispositivos móviles de alta resistencia con sistema operativo Windows CE .NET y empleando bases de datos Oracle y Visual Basic .NET, que permitan:
 - Recolectar datos en terreno durante el desarrollo de las actividades propias de la oferta y la demanda ambiental manejadas en el sistema SIPA.
 - Generar e imprimir los reportes necesarios en terreno utilizando una impresora móvil y conexión inalámbrica vía Bluetooth¹⁶ con el dispositivo móvil.
 - Sincronizar¹⁷ la información del dispositivo móvil, con la base de datos Oracle principal de SIPA, en doble sentido¹⁸.

- Realizar y documentar pruebas de operatividad a las diferentes opciones software desarrolladas, según escenarios establecidos.

- Generar los anexos técnicos y de usuario que complementen los manuales correspondientes de SIPA.

¹⁶ Bluetooth. Ver glosario

¹⁷ Sincronizar. Ver Glosario.

¹⁸ El término doble sentido se refiere al intercambio de información desde el dispositivo móvil hacia el servidor de datos y viceversa.

1.1.3. Descripción de Objetivos

Partiendo del análisis y comprensión del modelo existente en el sistema SIPA de VISION, se hace una adaptación de las tablas necesarias para la base de datos móvil, que permitan el almacenamiento de los datos capturados en terreno, la generación de reportes y la correcta sincronización entre el dispositivo móvil y la base de datos principal del sistema SIPA.

Mediante la elaboración de opciones software utilizando dispositivos móviles de alta resistencia con sistema operativo Windows CE .NET y bases de datos Oracle, se busca lograr la recolección en terreno de datos generados por eventos asociados a trámites ambientales. Además, se da la opción de generar e imprimir los reportes necesarios en terreno, tomando como base la información contenida en el dispositivo móvil, y utilizando una impresora móvil conectada vía Bluetooth con el dispositivo. Finalmente, se desarrollan opciones que permiten la correcta sincronización entre el dispositivo móvil y la base de datos Oracle principal del sistema SIPA en doble sentido, garantizando así, la confiabilidad y coherencia de la información manejada.

Mediante la realización de pruebas de operatividad a las diferentes opciones software desarrolladas según escenarios establecidos, se revisa y valida el correcto funcionamiento de dichas opciones en situaciones usuales a manejar en el ámbito del sistema.

Por último la generación de los anexos técnicos y de usuario, permiten complementar los manuales existentes del sistema SIPA, incluyendo ahora la información referente a la extensión en dispositivos móviles de éste.

1.2. JUSTIFICACIÓN

1.2.1. Definición del problema

La empresa VISION LTDA certificada ISO9001:2000, cuenta con aplicaciones para diferentes tipos de empresas en el sector estatal y privado. En el área ambiental, cuenta con el sistema SIPA, el cual está orientado al manejo del inventario y de la oferta y demanda ambiental. Además, facilita el manejo de expedientes ambientales asociados a sanciones o derechos y permite a las autoridades gestionar los traslados de recursos de acuerdo al reglamento vigente. Uno de los procesos que se debe cubrir es la toma de datos en campo sin que se pierda la sincronización y vigencia de los datos centrales. A la fecha este sistema opera en varias entidades ambientales en entorno Web y cliente servidor con base de datos Oracle y un componente para trabajo fuera de línea, que opera con sincronización de información con la base de datos central. El paso siguiente es llevar la información de los diccionarios ambientales al campo y tomar datos de eventos que en el mismo sucedan, y luego se lleven a la base central para su actualización.

La toma de datos en campo requiere de características especiales en los dispositivos empleados, tales como: movilidad, capacidad en almacenamiento de información, una adecuada autonomía, generación de reportes y resistencia a condiciones ambientales adversas tales como lluvia, polvo, temperaturas extremas y golpes fortuitos. Para el trabajo con estos dispositivos móviles mencionados, se requiere de múltiples tareas que van desde el diseño de los componentes, selección de herramientas adecuadas de desarrollo, trabajo en grupos con diferentes fortalezas técnicas y la integración de módulos que compartan datos y complementen funcionalidades. Esto demanda actividades orientadas a la asistencia en el desarrollo de dichos componentes, revisión de operatividad, desarrollo de pruebas y generación de escenarios de operación. La empresa

VISION, dentro de su grupo de profesionales, ha dado inicio a un proyecto que tiene como objetivo lograr la utilización de dichos dispositivos móviles de alta resistencia, como una extensión del sistema SIPA, esperando solventar así, las necesidades descritas anteriormente en el trabajo en campo abierto. Por lo tanto, se hace indispensable el apoyo a este proceso, mediante el desarrollo de opciones software para la toma de datos y generación de reportes en campo, producción de la debida documentación y revisión con base en escenarios de pruebas ya establecidos, de la operatividad de componentes desarrollados por el grupo de ingenieros de la empresa VISION.

1.2.2. Impacto

1.2.2.1. Impacto Técnico

Mediante el desarrollo de los objetivos planteados, se espera obtener un nuevo recurso para el manejo de información ambiental en terreno trayendo esto beneficios tales como información oportuna, agilidad, confiabilidad y comodidad en las tareas descritas, a través del uso de dispositivos diseñados especialmente para el trabajo que demanda gran movilidad en condiciones ambientales adversas.

1.2.2.2. Impacto Económico

Obtener una optimización en los procesos de trámites ambientales directamente en terreno, aumentando así la productividad de los funcionarios encargados de este proceso y disminuyendo o eliminando el riesgo de daños en equipos no aptos para el trabajo en condiciones ambientales adversas.

1.2.2.3. Impacto social

Ofrecer una atención más oportuna a las personas que requieran información, o algún trámite ambiental directamente en el campo.

1.3. CONOCIMIENTO DE LA EMPRESA

La empresa VISION Ltda., certificada ISO 9001:2000, está dedicada a suministrar soluciones informáticas en diferentes tipos de empresa. Cuenta con un sistema de gestión de calidad (SGC) guiado por dicha norma, el cual ha sido estudiado, para comprender en mayor detalle la actividad y políticas que guían a la empresa. Algunos de los aspectos más relevantes de este SGC, son descritos a continuación:

1.3.1. Información Corporativa

1.3.1.1. Misión

Proveer soluciones informáticas y de gestión tecnológica con parámetros de calidad, siguiendo estándares nacionales e internacionales, con recurso humano capacitado, calificado y comprometido, garantizando resultados excelentes, efectivos y eficientes que generen valor agregado a nuestros clientes, mediante la asimilación e innovación tecnológica, que aseguren la sostenibilidad y desarrollo de la empresa.

1.3.1.2. Visión

Para el año 2008, VISION tendrá presencia y establecimiento de sus marcas a nivel nacional e internacional, fortaleciendo su posicionamiento en estos mercados.

1.3.1.3. Política de calidad

Generar soluciones tecnológicas de software con calidad e innovación, siguiendo metodologías de clase mundial y utilizando tecnologías abiertas, cumpliendo con los tiempos pactados y acompañando al cliente en su operación y uso, buscando consolidar la imagen de solidez tecnológica de los productos y la empresa, ofreciendo valor agregado, ampliando la cobertura del mercado y la solidez financiera de la organización.

1.4. DESARROLLO DE LAS OPCIONES

Para el desarrollo de las opciones Software planteadas fueron utilizados los siguientes recursos de Hardware y Software, con estas especificaciones:

1.4.1. Hardware

1.4.1.1. Equipo para desarrollo

- Procesador Pentium 4 de 2.40 GHz.
- Memoria RAM de 1 GB.
- Disco Duro de 80 GB.

- Tarjeta de red.
- Puerto USB¹⁹.

1.4.1.2. Terminal Móvil

- Procesador Intel PXA 255 X-Scale corriendo a 400 MHz.
- 64 MB de memoria Flash.
- 128 MB de SDRAM.
- Pantalla de 3.5" QVGA (240 x 320) a color.
- Teclado externo alfanumérico de 52 teclas.
- Batería Principal: recargable de polímero de Litio de 1800 mAh, 4.2V (tiempo de duración aproximado de 8 horas).
- Batería de Backup: Para mantener el estado del sistema cuando la batería principal se descarga completamente.
- Radio Bluetooth integrado.
- Grado de sellamiento IP54²⁰.
- Resistencia a impactos desde 130 centímetros sobre concreto.

1.4.2. Software

1.4.2.1. Equipo para desarrollo

- Sistema Operativo Windows XP.
- Visual Studio .Net 2003.
- Oracle 9i Database.

¹⁹ USB. Ver Glosario.

²⁰ IP54. Ver Glosario

- Oracle Lite 10g.

1.4.2.2. Terminal Móvil

- Sistema Operativo Windows CE 4.2.
- Microsoft .Net Compact Framework.
- Cliente respectivo de Oracle Lite.

2. FUNDAMENTOS TEÓRICOS

En este capítulo se plasman los aspectos relevantes a tener en cuenta para una buena comprensión de la teoría necesaria en el desarrollo de este proyecto. Este marco tiene el propósito de ofrecer un sistema coordinado y coherente de conceptos que permitan comprender en mayor detalle el problema y establecer las bases conceptuales que definen el Software. Se expone cierta parte de la normatividad existente en el área ambiental de nuestro país, con el objetivo de comprender en mayor grado a quien va dirigido el desarrollo del proyecto y sus necesidades. También se describe en mayor detalle aspectos de la tecnología utilizada.

2.1. NORMATIVIDAD DE TRÁMITES AMBIENTALES EN COLOMBIA

Para comprender mejor el significado de los trámites ambientales, primero hay que conocer un poco de la normatividad establecida en el área por el gobierno nacional. A continuación, se hace una breve descripción de las leyes y los respectivos artículos de mayor relevancia, que se tuvieron en cuenta para el desarrollo del tema de la práctica, relacionados con la administración de los recursos naturales en nuestro país.

2.1.1. Corporaciones Autónomas Regionales

Tomando como base la Ley 99 de 1993 se pueden enunciar las siguientes definiciones de importancia para el proyecto:

2.1.1.1. Naturaleza Jurídica

Según el artículo 23 las Corporaciones Autónomas Regionales son entes corporativos de carácter público, creados por la ley, integrado por las entidades territoriales que por sus características constituyen geográficamente un mismo ecosistema o conforman una unidad geopolítica, biogeográfica o hidrogeográfica, dotados de autonomía administrativa y financiera, patrimonio propio y personería jurídica, encargados por la ley de administrar, dentro del área de su jurisdicción, el medio ambiente y los recursos naturales renovables y propender por su desarrollo sostenible, de conformidad con las disposiciones legales y las políticas del Ministerio del Medio Ambiente.

2.1.1.2. Objeto

Según el artículo 30 todas las Corporaciones Autónomas Regionales tendrán por objeto la ejecución de las políticas, planes, programas y proyectos sobre medio ambiente y recursos naturales renovables, así como dar cumplida y oportuna aplicación a las disposiciones legales vigentes sobre su disposición, administración, manejo y aprovechamiento, conforme a las regulaciones, pautas y directrices expedidas por el Ministerio del Medio Ambiente.

2.1.1.3. Funciones

Según el artículo 31 se ha seleccionado ciertas funciones de las Corporaciones Autónomas Regionales las cuales son de interés:

- Ejecutar las políticas, planes y programas nacionales en materia ambiental definidos por la ley aprobatoria del Plan Nacional de Desarrollo y del Plan Nacional de Inversiones o por el Ministerio del Medio Ambiente, así como

los del orden regional que le hayan sido confiados conforme a la ley, dentro del ámbito de su jurisdicción.

- Ejercer la función de máxima autoridad ambiental en el área de su jurisdicción, de acuerdo con las normas de carácter superior y conforme a los criterios y directrices trazadas por el Ministerio del Medio Ambiente.
- Otorgar concesiones, permisos, autorizaciones y licencias ambientales requeridas por la Ley para el uso, aprovechamiento o movilización de los recursos naturales renovables o para el desarrollo de actividades que afecten o puedan afectar el medio ambiente. Otorgar permisos y concesiones para aprovechamientos forestales, concesiones para el uso de aguas superficiales y subterráneas y establecer vedas para la caza y pesca deportiva.
- Fijar en el área de su jurisdicción, los límites permisibles de emisión, descarga, transporte o depósito de sustancias, productos, compuestos o cualquier otra materia que puedan afectar el medio ambiente o los recursos naturales renovables y prohibir restringir o regular la fabricación, distribución, uso disposición o vertimiento de sustancias causantes de degradación ambiental. Estos límites, restricciones y regulaciones en ningún caso podrán ser menos estrictos que los definidos por el Ministerio del Medio Ambiente.
- Ejercer las funciones de evaluación, control y seguimiento ambiental de las actividades de exploración, explotación, beneficio, transporte, uso y depósito de los recursos naturales no renovables, incluida la actividad portuaria con exclusión de las competencias atribuidas al Ministerio del Medio Ambiente, así como de otras actividades, proyectos o factores que

generen o puedan generar deterioro ambiental. Esta función comprende la expedición de la respectiva licencia ambiental.

- Ejercer las funciones de evaluación, control y seguimiento ambiental de los usos del agua, el suelo, el aire y los demás recursos naturales renovables, lo cual comprenderá el vertimiento, emisión o incorporación de sustancias o residuos líquidos, sólidos y gaseosos, a las aguas en cualquiera de sus formas, al aire o a los suelos, así como los vertimientos o emisiones que puedan causar daño o poner en peligro el normal desarrollo sostenible de los recursos naturales renovables o impedir u obstaculizar su empleo para otros usos, estas funciones comprenden expedición de las respectivas licencias ambientales, permisos concesiones, autorizaciones y salvoconductos.
- Ejercer el control de la movilización, procesamiento y comercialización de los recursos naturales renovables en coordinación con las demás Corporaciones Autónomas Regionales, las entidades territoriales y otras autoridades de policía, con conformidad con la ley y los reglamentos; y expedir los permisos, licencias y salvoconductos para la movilización de recursos naturales renovables.
- Imponer y ejecutar a prevención y sin perjuicio de las competencias atribuidas por la ley a otras autoridades, las medidas de policía y las sanciones previstas en la ley, en caso de violación a las normas de protección ambiental y de manejo de recursos naturales renovables y exigir, con sujeción a las regulaciones pertinentes, la reparación de daños causados.

2.2. DETALLES DE LA TECNOLOGÍA EMPLEADA

2.2.1. Sistemas Operativos Windows para dispositivos Móviles

2.2.1.1. Windows CE

Es el Sistema Operativo que Microsoft ha desarrollado a partir de Windows 95, para dispositivos móviles, y sirve de base para el desarrollo de los sistemas específicos de cada dispositivo. Lo que los usuarios finales disfrutaban, no es Windows CE tal y como ha sido desarrollado. En cada tipo de dispositivo se implementa, las posibilidades que permite la versión de Windows CE disponible, una interfaz y las funcionalidades requeridas. Así, el **Pocket PC**²¹ 2000, 2002 y 2003 se han desarrollado específicamente para los PDAs. Windows CE nació como un sistema operativo de fácil programación, sólido, transparente y que podía implantarse desde un ordenador a una lavadora, nevera, microondas incluso videoconsolas (DreamCast). De hecho, se pensó en integrarlo en todo lo que no fuera un PC.

2.2.1.2. Windows CE .NET

Es la evolución de Windows CE 3.0 bajo la filosofía distribuida de .NET. Es pues, un escenario de trabajo que es adaptado a cada dispositivo. Esta nueva versión tiene muchas ventajas, que pueden ser aplicadas a cada uno de los sistemas operativos derivados. Windows CE .NET, incorpora la posibilidad de manejar las conexiones Bluetooth, Microsoft Internet Explore, Windows Media y DirectX y es

²¹ Pocket PC. Ver Glosario

compatible con un amplio rango de procesadores como Xscale, ARM, MIPS, SH o x86. Cada sistema operativo derivado, tomará las propiedades que le competan.

Los dispositivos PDA²² que disponen de Windows CE son dispositivos con una magnífica pantalla de 240 x 320 píxeles a todo color. Son muy potentes, con procesadores de entre 133 - 400 Mhz y 16 - 64 Mbytes de RAM, por lo que son capaces de reproducir vídeo o música y ejecutar aplicaciones multimedia con gran rapidez. También disponen de altavoz y salida de audio para auriculares. Además incluyen diversos tipos de ranuras o slots de expansión, que permiten insertar tarjetas de diversos formatos (Multimedia, CompactFlash o PCMCIA) para aumentar memoria o incorporar módems, discos duros, tarjetas de red, etc.

En general, para la construcción de los sistemas operativos Windows CE, se tuvieron en cuenta los siguientes principios de diseño:

- Memoria Pequeña.
- Modularidad.
- Procesadores portables.
- Compatibilidad con aplicaciones Windows de 32 bits.
- Conectividad.
- Procesamiento en tiempo real.

La modularidad de Windows CE permite que este pueda ser dividido en aproximadamente 300 partes más pequeñas, cada una con una función definida. La más pequeña de estas partes solamente necesita 200Kb de memoria RAM para ser ejecutada.

²² PDA. Ver Glosario

Microsoft ha lanzado nuevas versiones del sistema operativo Windows CE, cada una con ciertas mejoras respecto a la anterior. Estas versiones pueden ser observadas en la siguiente tabla:

Versión	Fecha	Características
2.0	1997	Soporte para ActiveX ²³ , menús en cascada, conexión a redes Ethernet.
2.1	1998	Soporte para USB, criptografía y pantallas a color.
3.0	2000	Soporte para tarjetas inteligentes y funcionalidades ASP.
CE .NET 4.0	2002	Configuración automática para redes IEEE 802.11, Bluetooth, elementos de interfaz de usuario personalizables y otros.
CE .NET 4.1	2002	Soporte para IPv6 y exploradores de archivos.
CE .NET 4.2	2003	Procesamiento en tiempo real, capacidades multimedia y desempeño mejorado.
CE .NET 5.0	2005	Soporte extendido de Bluetooth para audífonos y manos libres, capacidad de descubrimiento de dominios, soporte nativo para Wi-Fi.

Tabla 1. Versiones del Sistema Operativo Windows CE

2.2.2. Software de desarrollo

Para el desarrollo de las opciones Software se eligió trabajar sobre la plataforma de desarrollo de Visual Studio .Net 2003 y el lenguaje de programación Visual Basic .Net. En los siguientes numerales se hace una breve descripción de los aspectos más importantes sobre las tecnologías utilizadas.

²³ ActiveX. Ver glosario

2.2.2.1. Microsoft Visual Studio .NET

Visual Studio .NET es un conjunto completo de herramientas de desarrollo para la construcción de aplicaciones Web ASP, servicios Web XML, aplicaciones para escritorio y aplicaciones en dispositivos móviles. Visual Basic .NET, Visual C++ .NET, Visual C# .NET y Visual J# .NET utilizan el mismo entorno de desarrollo integrado (IDE), que les permite compartir herramientas y facilita la creación de soluciones en varios lenguajes. Asimismo, dichos lenguajes aprovechan las funciones de .NET Framework (para mayor detalle ver numeral 2.2.2.2), que ofrece acceso a tecnologías clave para simplificar el desarrollo de aplicaciones Web ASP y servicios Web XML. Visual Studio .NET 2003 contiene una versión mejorada de Windows .NET Framework. Windows .NET Framework 1.1 incluye nuevas funciones, mejoras y actualizaciones de la documentación. Gracias a la compatibilidad integrada con .NET Compact Framework, Visual Studio .NET 2003 incorpora a .NET los dispositivos móviles e incrustados, como Pocket PC, y otros dispositivos que utilizan el sistema operativo Microsoft Windows CE .NET.

Los programadores pueden utilizar Visual Studio .NET para:

- Crear aplicaciones basadas en Windows rápidas y eficaces.
- Crear aplicaciones rápidas y eficaces para dispositivos basados en Windows CE.
- Crear aplicaciones Web sofisticadas y seguras.
- Crear aplicaciones Web inteligentes, sofisticadas y seguras para dispositivos móviles.
- Utilizar servicios Web XML en cualquiera de las aplicaciones mencionadas.

- Evitar conflictos entre archivos .DLL.
- Eliminar los costosos problemas de implementación y mantenimiento de las aplicaciones.

2.2.2.2. Microsoft .NET Framework

.NET Framework es una plataforma informática que simplifica el desarrollo de aplicaciones en un entorno altamente distribuido como es el Internet. El diseño de .NET Framework está enfocado a cumplir los objetivos siguientes:

- Proporcionar un entorno coherente de programación orientada a objetos, en el que el código de los objetos se pueda almacenar y ejecutar de forma local, ejecutar de forma local pero distribuida en Internet o ejecutar de forma remota.
- Proporcionar un entorno de ejecución de código que reduzca lo máximo posible la implementación de software y los conflictos de versiones.
- Ofrecer un entorno de ejecución de código que garantice la ejecución segura del mismo, incluso del creado por terceras personas desconocidas o que no son de plena confianza.
- Proporcionar un entorno de ejecución de código que elimine los problemas de rendimiento de los entornos en los que se utilizan secuencias de comandos o intérpretes de comandos.
- Ofrecer al programador una experiencia coherente entre tipos de aplicaciones muy diferentes, como las basadas en Windows o en Web.

- Basar toda la comunicación en estándares del sector para asegurar que el código de .NET Framework se pueda integrar con otros tipos de código.

Vale la pena aclarar la diferencia entre Visual Studio .NET y .NET Framework, pues existe cierta confusión en el tema.

.NET Framework es el marco de trabajo con el cuál se desarrollan las aplicaciones. En él se incluyen las diferentes partes del lenguaje (clases, objetos, tipos, etc) que permiten desarrollar soluciones informáticas.

Visual Studio .NET es el entorno RAD (Rapid Application Development o Desarrollo Rápido de Aplicaciones), que permite utilizar .NET Framework para desarrollar aplicaciones de una forma rápida y visual. Visual Studio .NET utiliza por tanto a .NET Framework.

.NET Framework contiene dos componentes principales: Common Language Runtime (CLR) y la biblioteca de clases de .NET Framework. Common Language Runtime es el fundamento de la tecnología. El motor de tiempo de ejecución se puede considerar como un agente que administra el código en tiempo de ejecución y proporciona servicios centrales, como la administración de memoria, la administración de subprocesos y la interacción remota, al tiempo que aplica una seguridad estricta a los tipos y otras formas de especificación del código que garantizan su seguridad y solidez. De hecho, el concepto de administración de código es un principio básico del motor de tiempo de ejecución. El código destinado al motor de tiempo de ejecución se denomina *código administrado*, a diferencia del resto de código, que se conoce como *código no administrado*.

La biblioteca de clases, el otro componente principal de .NET Framework, es una completa colección orientada a objetos de tipos reutilizables que se pueden

emplear para desarrollar aplicaciones que abarcan desde las tradicionales herramientas de interfaz gráfica de usuario (GUI) o de línea de comandos hasta las aplicaciones basadas en las innovaciones más recientes proporcionadas por ASP.NET, como los formularios Web Forms y los servicios Web XML.

2.2.2.3. Microsoft .NET Compact Framework

.NET Compact Framework es un entorno independiente del hardware para la ejecución de programas en dispositivos de computación con limitaciones de recursos, entre los que se encuentran los asistentes de datos personales (PDA) como Pocket PC, teléfonos móviles, decodificadores de televisión, dispositivos de computación para automóviles y dispositivos incrustados de diseño personalizado, que están integrados en el sistema operativo Windows CE .NET.

.NET Compact Framework es un subconjunto de la biblioteca de clases .NET Framework y también contiene clases diseñadas expresamente para él. Hereda la arquitectura .NET Framework completa de Common Language Runtime y la ejecución de código administrado.

.NET Compact Framework ofrece las siguientes funciones principales:

- Ejecuta programas independientemente del hardware y el sistema operativo.
- Admite protocolos de red comunes y se conecta perfectamente con servicios XML Web.

- Proporciona a los desarrolladores un modelo para orientar sus aplicaciones y componentes ya sea a una amplia gama de dispositivos o a una categoría específica de éstos.
- Facilita el diseño y la optimización de los recursos de sistema limitados.
- Obtiene un rendimiento óptimo en la generación de código nativo cuando se utiliza compilación Just-In-Time (JIT).

.NET Compact Framework tiene varias similitudes con .NET Framework completo entre las que se destacan:

- *Common Language Runtime*: Los Common Language Runtime de los dos entornos Framework se benefician de la ejecución de código administrado, compilación de código Just-In-Time (JIT) y recolección de elementos no utilizados. El Common Language Runtime de .NET Compact Framework ocupa aproximadamente un 12% del tamaño del Common Language Runtime de .NET Framework completo.
- *Formatos de archivo y ensamblados*: Las aplicaciones de los dos entornos Framework utilizan ensamblados. Ambos obtienen acceso a archivos ejecutables (PE), que contienen el Lenguaje intermedio de Microsoft (MSIL) y metadatos que definen una aplicación de .NET Framework. Un archivo PE puede hacer referencia a un espacio de nombres de programación definido y compartido por otros archivos de ensamblado.
- *Sistema de tipos comunes*: Los dos entornos Framework tienen integrados tipos primitivos así como otros tipos que pueden utilizarse y derivarse cuando se genera una aplicación.

- *Compatibilidad con estándares:* La tecnología básica implementada en los dos entornos Framework es compatible con el estándar Common Language Infrastructure de European Computer Manufacturers Association (ECMA).

Por último hay que tener en cuenta que aunque los formatos de archivo de los dos entornos son idénticos, un componente binario compilado en .NET Compact Framework no puede ejecutarse en .NET Framework completo. Tampoco es posible lo contrario, ejecutar en .NET Compact Framework un componente binario compilado en .NET Framework completo. Esto se debe a que las reglas de enlace son muy fuertes. No obstante, sí es posible volver a compilar una aplicación para ejecutarla en el otro entorno Framework.

2.2.2.4. C# o Visual Basic?

Al inicio de la práctica, se encontró que la plataforma Visual Studio .NET 2003, ofrecía dos lenguajes de programación para hacer desarrollos en dispositivos móviles: C# .NET y Visual Basic .NET. Por consiguiente, se presentó la incógnita de saber cual de los dos lenguajes era más aconsejable para realizar el desarrollo de las opciones, teniendo en cuenta aspectos tales como rendimiento, facilidad de aprendizaje, documentación disponible entre otros.

Investigando se pudo concluir que la decisión acerca de cuál lenguaje utilizar depende más de la experiencia previa con otros lenguajes tales como el C, el C++ ó el Visual Basic, ya que la curva de aprendizaje será mucho menor si se es un desarrollador con experiencia en C y se decide utilizar C#; de igual manera, si se es un desarrollador con experiencia en Visual Basic, esta curva de aprendizaje será menor si se elige Visual Basic .NET, ya que esto permite centrarse en aprender los aspectos nuevos y específicos del .NET *Framework* (ya sea en su

versión completa o compacta) y no centrarse en aprender aspectos específicos del lenguaje tales como su sintaxis, por ejemplo.

En la plataforma .NET todos los lenguajes se compilan a un mismo lenguaje intermedio (aún cuando los archivos resultantes de esta compilación sean .EXE ó .DLL), al cual normalmente se hace referencia como MSIL ó IL; además, la plataforma cuenta con un componente conocido como CLR (el *Common Language Runtime* mencionado anteriormente) el cual se encarga, entre otras cosas, de convertir estas instrucciones en IL hacia código de máquina justo antes de su ejecución haciendo uso de un compilador justo a tiempo (JIT). Ahora, el punto importante es que “todos los lenguajes se compilan a un mismo lenguaje intermedio”, lo cual en ningún momento da pie a suponer que una aplicación escrita en C# nos dará como resultado un mejor IL que la misma aplicación escrita en Visual Basic .NET, o viceversa.

La posición de Microsoft al respecto es: *“Debido a las diferencias del pasado entre Microsoft Visual Basic, Microsoft Visual C, y Microsoft Visual C++, muchos desarrolladores tienen la impresión de que Microsoft Visual C# .NET es un lenguaje mucho más poderoso que Microsoft Visual Basic .NET. Algunos desarrolladores asumen que muchas cosas que son posibles en Visual C# .NET son imposibles en Visual Basic .NET; de igual forma en que muchas cosas que son posibles en Microsoft Visual C 6.0 ó Microsoft Visual C++ 6.0 son imposibles en Microsoft Visual Basic 6.0. Asumir esto es incorrecto. Si bien existen diferencias entre Visual Basic .NET y Visual C# .NET, ambos son lenguajes de programación de primera clase basados en el Microsoft .NET Framework, y ambos son igual de poderosos.”*²⁴

Como conclusión se puede decir que ya que ambos lenguajes están basados en la misma plataforma (.NET), y hacen uso de los mismos recursos (.NET Framework

²⁴ White Paper: Differences Between Microsoft Visual Basic .NET and Microsoft Visual C# .NET.

o *.NET Compact Framework*), se puede obtener los mismos resultados con uno y otro; lo que da la libertad de seleccionar el lenguaje que más se acomode a las necesidades particulares y experiencia previa sin sacrificar la potencia o la funcionalidad que el lenguaje nos ofrece, permitiendo centrarse más en aprender los aspectos nuevos de la plataforma, que a fin de cuentas son comunes a todos los lenguajes.

Teniendo en cuenta lo expuesto anteriormente y que particularmente se contaba con experiencia previa en programación con Visual Basic 6.0, se decidió escoger como lenguaje de desarrollo a Visual Basic .NET.

2.2.2.5. Microsoft Visual Basic .NET

Visual Basic .NET es la última generación del lenguaje Visual Basic que presenta un modo rápido y fácil de crear aplicaciones .NET como servicios Web XML, aplicaciones de escritorio y aplicaciones para dispositivos móviles. Es un lenguaje de programación diseñado para crear aplicaciones que trabajen con Microsoft .NET Framework. Ofrece características como herencia, interfaces y sobrecargas, que lo convierten en un eficaz lenguaje de programación orientado a objetos. Algunas de sus características más importantes son:

- Su semántica y sintaxis son simples, sencillas y fáciles de comprender. Es muy fácil de utilizar.
- Proporciona a los programadores las características principales de .NET Framework y es coherente con las convenciones del marco de trabajo.
- Es compatible con versiones anteriores de Visual Basic. Cuando es posible, Visual Basic .NET tiene la misma sintaxis, la misma semántica y el mismo comportamiento en tiempo de ejecución que sus predecesores.

- Permite un desarrollo rápido de programas sin que se vea afectada la confiabilidad.
- Produce código predecible y eficiente.

El acceso a datos se hace posible con gran facilidad gracias a ADO .NET²⁵. La flexibilidad de ADO .NET permite el enlace de datos con cualquier base de datos, así como clases, colecciones y matrices, y proporciona una auténtica representación XML de los datos. Con ADO.NET se puede obtener acceso a alta velocidad a Microsoft SQL Server, Oracle (incluyendo la versión Lite), DB2, Microsoft Access y muchos más.

Visual Basic .NET permite el desarrollo rápido de aplicaciones para dispositivos móviles. Utilizando un subconjunto de factores de .NET Framework se pueden crear, compilar, depurar y distribuir aplicaciones que se ejecutan en .NET Compact Framework en equipos PDA, teléfonos móviles y otros dispositivos con limitaciones de recursos.

2.2.3. Conceptos básicos sobre Bases de Datos

Una base de datos puede ser definida como una colección automatizada de datos operacionales almacenados, que sirven a las necesidades de múltiples usuarios dentro de una o más organizaciones y que varían con el tiempo, tanto en tamaño como en valores o dicho en otras palabras es la fuente central de datos que está pensada para que sea compartida por muchos usuarios con una diversidad de aplicaciones. Una Base de Datos contiene datos para varios usuarios, donde cada uno puede tener acceso a todo el conjunto o solo alguna parte de ellos,

²⁵ ADO .NET. Ver Glosario.

dependiendo de la necesidad, lo cual implica que los datos son compartidos y se tiene un control sobre ellos.

2.2.3.1. Componentes de una Base de Datos

Un sistema de Bases de Datos se compone de cuatro elementos principales: datos, hardware, software y usuarios²⁶.

Datos. Cada uno de los procesos que constituyen una organización que genera datos que son registrados en algún medio de almacenamiento que puede ser impreso, fílmico, electrónico o magnético.

Partiendo de lo particular a lo general, estos datos se pueden clasificar de la siguiente forma jerárquica:

- Campo: Es la unidad más pequeña de información que se almacena en una Base de Datos. Permite definir una característica (edad, peso) sobre un elemento objeto de estudio. Puede estar en formato carácter, fecha, número u otro formato y ser opcional u obligatorio.
- Registro: Es una colección de campos asociados que permiten agrupar características acerca de un elemento objeto de estudio.²⁷
- Tabla: es una colección de registros que contienen la información de un elemento objeto de estudio.

²⁶ CÁRCAMO SEPÚLVEDA, José. Bases de Datos Relacionales: Un Enfoque Práctico de Diseño. Bucaramanga, 1997. Pág. 11.

²⁷ WHITTEN, Jeffrey L. Análisis y Diseño de Sistemas de Información. Illinois, 1996. Pág. 607.

- Base de Datos: El conjunto de estas tablas o entidades relacionadas de una forma lógica.

Hardware. Este se refiere a los medios de almacenamiento (discos duros, disquetes, CDs, cintas magnéticas entre otros) en los cuales reside la Base de Datos y los dispositivos con los que se manejan tales medios.

Software. Entre la Base de Datos física y los denominados usuarios, existe una interfaz conocida con el nombre de D.B.M.S. (Sistema manejador de Base de Datos), quien es el encargado de atender los accesos de los usuarios a la Base de Datos (Ver figura 1), es decir, es la parte medular de la Base de Datos que permite la creación, modificación y actualización de la misma, la recuperación de datos y la generación de reportes.

Usuarios. El D.B.M.S. soporta múltiples usuarios, permite accesos concurrentes, y traduce en órdenes sobre la Base de Datos todos los requerimientos que el usuario posee para el manejo de la información.

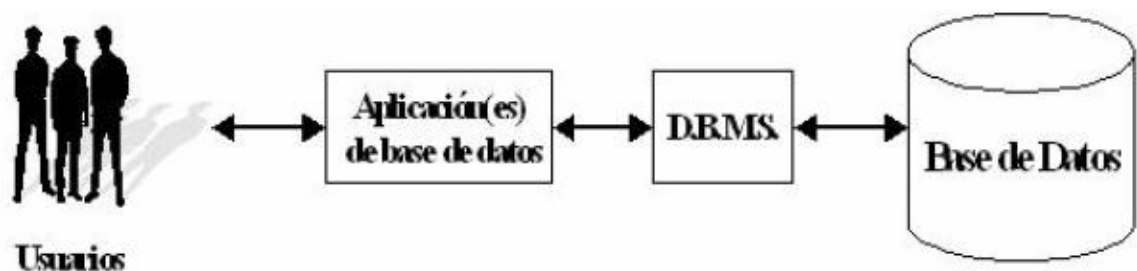


Figura 1. Aplicación de bases de datos y un DBMS

2.2.3.2. Modelo Entidad-Relación

El modelo Entidad-Relación es una técnica que se utiliza en la mayoría de las metodologías de desarrollo de Sistemas de Información. La organización de la

Base de Datos considera entidades, constituyéndose cada una en la relación de diferentes datos de la base. Allí se representan los objetos de datos, sus características y las interrelaciones que definen los requerimientos y las necesidades de información.

2.2.4. Oracle

Oracle es básicamente un herramienta cliente/servidor para la gestión de bases de datos. Es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hacen que solo se vea en empresas muy grandes y/o multinacionales, por norma general.

Para utilizar Oracle primero es necesario la instalación de la herramienta servidor (Oracle8i, Oracle9i u Oracle 10g) para posteriormente poder manipular a la base de datos desde otros equipos con herramientas de desarrollo como Oracle Designer y Oracle Developer, que son las herramientas de programación sobre Oracle.

Para desarrollar en Oracle se utiliza PL/SQL²⁸. También es posible manipular a la base de datos a través del "SQL Plus" incorporado en el paquete de programas Oracle para poder realizar consultas, utilizando el lenguaje SQL.

El Developer es una herramienta que permite crear formularios en local, es decir, mediante esta herramienta se pueden crear formularios, compilarlos y ejecutarlos, pero si se quiere que otras personas trabajen sobre este formulario, se debe copiar regularmente en una carpeta compartida para todos, de modo que, cuando se quiera realizar un cambio, se copiará de dicha carpeta para luego volverlo a subir.

²⁸ PL/SQL. Ver Glosario.

Este sistema como se puede observar es bastante engorroso y poco fiable pues es bastante normal que las versiones se pierdan con frecuencia. La principal ventaja de esta herramienta es que es bastante intuitiva y dispone de un entorno que permite componer el formulario, tal y como se haría en otros lenguajes tradicionales como Visual Basic o Delphi.

Los problemas anteriores quedan totalmente resueltos con Designer que es una herramienta que se conecta a la base de datos y por tanto se crean los formularios en ella; de esta manera todo el mundo se conecta mediante Designer a la aplicación que contiene todos los formularios y no hay problemas de diferentes versiones. Esto es muy útil para evitar dañar el trabajo de otros.

La empresa Oracle Corporation cuenta entre sus logros con la construcción del primer sistema comercial de base de datos relacional. También vendió el primer producto que empleaba SQL, hoy en día un estándar en la industria de bases de datos. Descubrió el potencial de un sistema cliente/servidor de bajo costo, en lugar de las tradicionales mainframes propietarias. Fue pionera en software portable que hoy día corre sobre prácticamente cualquier hardware.

2.2.5. Oracle Lite

Para el manejo de los datos en la terminal portátil, se ha escogido a Oracle Lite, una nueva tecnología que ofrece grandes ventajas en el almacenamiento y gestión de información entre dispositivos móviles y PC's tales como tamaño, seguridad, facilidades de administración y sobre todo funciones de sincronización.

La base de datos Oracle Lite es una adición a las bases de datos Oracle, usada para el desarrollo y despliegue de aplicaciones en dispositivos móviles. Oracle Lite ofrece sincronización de datos de forma confiable y segura entre una base de datos Oracle y un ambiente remoto. Los trabajadores pueden utilizar la

información de la empresa y realizar ciertas funciones, mientras están desconectados de la base de datos empresarial.

Básicamente Oracle lite ofrece:

- Un pequeño sistema administrador de bases de datos relacionales (RDBMS) Oracle Lite.
- Una pequeña base de datos relacional.
- Sincronización de datos en ambas direcciones entre el dispositivo móvil y una base de datos central.
- Sistema administrador de aplicaciones y usuarios.
- Herramientas de desarrollo y despliegue de aplicaciones.

Oracle Lite puede dividirse en dos áreas principalmente: Las herramientas de desarrollo para obtener aplicaciones productivas (Mobile Development Kit) y el Servidor Móvil (Mobile Server) para el despliegue, seguimiento y administración de aplicaciones, usuarios y dispositivos. La siguiente figura ilustra esta distribución, contemplando además, los principales componentes de cada área:

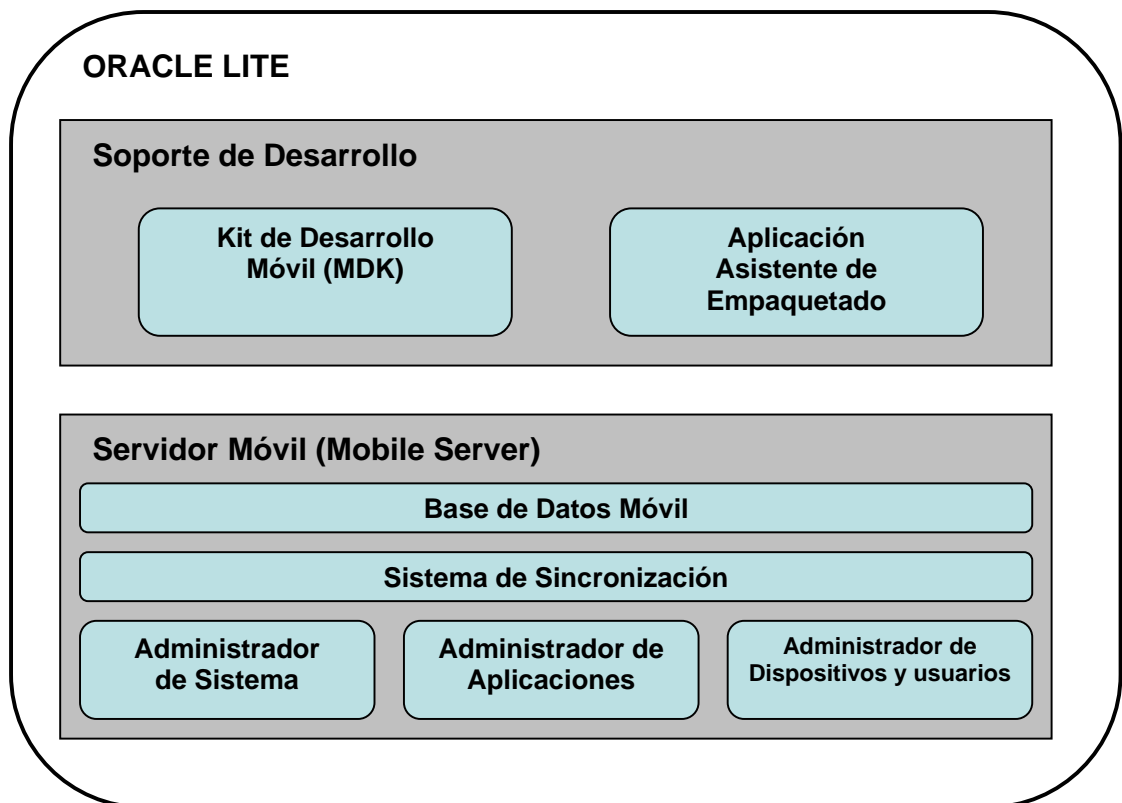


Figura 2. Componentes de Oracle Lite.

Oracle Lite usa un servidor de aplicaciones para la comunicación entre los clientes y la base de datos principal. Se debe usar uno de los siguientes servidores de aplicaciones:

- OC4J²⁹. Recomendado para ambientes con pocos usuarios. Es el modo instalado automáticamente con Oracle Lite.
- Oracle Application Server. Recomendado para ambientes de producción con un gran número de usuarios. Debe instalarse antes que Oracle Lite.

Soporta a los proveedores de datos JDBC³⁰, ODBC³¹, ADO .NET y SODA³². La programación de aplicaciones que utilicen Oracle Lite en dispositivos Pocket PC o

²⁹ OC4J. Ver Glosario.

³⁰ JDBC. Ver Glosario.

Windows CE, puede ser realizada en Visual Basic .NET o C# utilizando el proveedor de datos ADO .NET.

³¹ ODBC. Ver Glosario

³² SODA. Ver Glosario.

3. MARCO METODOLÓGICO

Una de los aspectos más importantes para que un proyecto sea llevado a feliz término, es la elección adecuada de la metodología de trabajo que se va a utilizar, ya que es este elemento el que define qué actividades se van a realizar, quién o quienes son los encargados de desarrollarlas y cuánto tiempo es necesario para llevarlas a cabo, todo dentro del contexto del proceso de desarrollo de software.

La empresa VISIÓN, en el proceso de producción definido en su sistema de gestión de calidad, tiene adoptada la metodología Extreme Programming para el desarrollo de sus proyectos, pues proporciona una forma ligera, eficiente, flexible y predecible de generar software. Por consiguiente, esta metodología fue aplicada en el desarrollo de la práctica, debido a que se dedicó gran parte del tiempo al desarrollo de las opciones software (proceso de producción).

3.1. METODOLOGÍAS ÁGILES DE DESARROLLO

La metodología seguida en el desarrollo de la práctica fue Extreme Programming, una de las llamadas metodologías ágiles de desarrollo de software. Estas metodologías se basan en valores y principios que deben permitir a los equipos desarrollar software rápidamente y responder a los cambios que puedan surgir a lo largo del proyecto. Pretenden ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas. En la tabla 2 se enuncian algunas de las principales diferencias de estas metodologías respecto a las tradicionales.

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código.	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.
Especialmente preparadas para cambios durante el proyecto.	Cierta resistencia a los cambios.
Impuestas internamente (por el equipo De desarrollo).	Impuestas externamente.
Proceso menos controlado, con pocos principios.	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible.	Existe un contrato prefijado.
El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio.	Grupos grandes y posiblemente Distribuidos.
Pocos roles.	Más roles.
Menos énfasis en la arquitectura del Software.	La arquitectura del software es esencial Y se expresa mediante modelos.

Tabla 2. Metodologías ágiles y tradicionales

Todas las metodologías ágiles más populares coinciden con los principios enunciados anteriormente aunque cada una tiene características propias y hace hincapié en algunos aspectos más específicos. A continuación se hace un corto resumen de dichas metodologías y en la siguiente sección se describe en mayor detalle a Extreme Programming.

SCRUM. Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas *sprints*, con una duración de 30 días. El resultado de cada *sprint* es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo proyecto. Éstas son las verdaderas protagonistas,

especialmente la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.

Crystal Methodologies. Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo (de ellas depende el éxito del proyecto) y la reducción al máximo del número de artefactos producidos. El desarrollo de software se considera un juego cooperativo de invención y comunicación, limitado por los recursos a utilizar. El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo en equipo definidas. Estas políticas dependerán del tamaño del equipo, estableciéndose una clasificación por colores, por ejemplo *Crystal Clear* (3 a 8 miembros) y *Crystal Orange* (25 a 50 miembros).

Dynamic Systems Development Method (DSDM). Define el marco para desarrollar un proceso de producción de software. Nace en 1994 con el objetivo de crear una metodología ágil unificada. Sus principales características son: es un proceso iterativo e incremental y el equipo de desarrollo y el usuario trabajan juntos. Propone cinco fases: estudio viabilidad, estudio del negocio, modelado funcional, diseño y construcción, y finalmente implementación. Las tres últimas son iterativas, además de existir realimentación a todas las fases.

Adaptive Software Development (ASD). Sus principales características son: iterativo, orientado a los componentes software más que a las tareas y tolerante a los cambios. El ciclo de vida que propone tiene tres fases esenciales: especulación, colaboración y aprendizaje. En la primera de ellas se inicia el proyecto y se planifican las características del software; en la segunda se desarrollan las características y finalmente en la tercera se revisa su calidad, y se entrega al cliente. La revisión de los componentes sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo.

Feature-Driven Development (FDD). Define un proceso iterativo que consta de 5 pasos. Las iteraciones son cortas (hasta 2 semanas). Se centra en las fases de diseño e implementación del sistema partiendo de una lista de características que debe reunir el software.

Lean Development (LD). Definida por Bob Charette's a partir de su experiencia en proyectos con la industria japonesa del automóvil en los años 80 y utilizada en numerosos proyectos de telecomunicaciones en Europa. En LD, los cambios se consideran riesgos, pero si se manejan adecuadamente se pueden convertir en oportunidades que mejoren la productividad del cliente. Su principal característica es introducir un mecanismo para implementar dichos cambios.

3.2. EXTREME PROGRAMMING

Extreme Programming es la metodología ágil más difundida y utilizada en el mundo. Está centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. Se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y disposición para enfrentar los cambios. Se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

En los siguientes párrafos se hace una breve descripción de la metodología Extreme Programming, basándose en sus características esenciales que son: Historias de usuario, proceso y prácticas.

3.2.1. Historias de usuario

Las historias de usuario (algo muy similar a los casos de uso de UML³³) son la técnica utilizada en Extreme Programming para especificar los requisitos del Software. Se trata de tarjetas de papel en las cuales se describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento historias de usuario pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas.

No hay que preocuparse si en un principio no se identifican todas las historias de usuario. Al comienzo de cada iteración estarán registrados los cambios en las historias de usuario y según eso se planificará la siguiente iteración. Las historias de usuario son descompuestas en tareas de programación y asignadas a los programadores para ser implementadas durante una iteración.

3.2.2. Proceso

El proceso de Extreme Programming consiste de seis fases: Exploración, Planificación de la Entrega, Iteraciones, Producción, Mantenimiento y Muerte del Proyecto.

- *Exploración:* En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se

³³ UML. Ver Glosario

exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.

- *Planificación de la entrega:* En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días.
- *Iteraciones:* Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo las historias que fueren la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el cliente quien decide qué historias se implementarán en cada iteración (para maximizar el valor de negocio). Al final de la última iteración el sistema estará listo para entrar en producción.
- *Producción:* La fase de producción requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase.
- *Mantenimiento:* Mientras la primera versión se encuentra en producción, el proyecto debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para realizar esto se requiere de tareas de

soporte para el cliente. De esta forma, la velocidad de desarrollo puede bajar después de la puesta del sistema en producción. La fase de mantenimiento puede requerir nuevo personal dentro del equipo y cambios en su estructura.

- *Muerte del proyecto:* Es cuando no se tiene más historias para ser incluidas en el sistema. Esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo.

3.2.3. Prácticas

La principal suposición que se realiza en Extreme Programming es la posibilidad de disminuir la mítica curva exponencial del costo del cambio a lo largo del proyecto, lo suficiente para que el diseño evolutivo funcione. Extreme Programming apuesta por un crecimiento lento del costo del cambio y con un comportamiento asintótico. Esto se consigue gracias a las tecnologías disponibles para ayudar en el desarrollo de software y a la aplicación disciplinada de las siguientes prácticas.

- *El juego de la planificación:* Hay una comunicación frecuente entre el cliente y los programadores. El equipo técnico realiza una estimación del esfuerzo requerido para la implementación de las historias de usuario y los clientes deciden sobre el ámbito y tiempo de las entregas de cada iteración.
- *Pequeñas entregas:* Producir rápidamente versiones del sistema que sean operativas, aunque no cuenten con toda la funcionalidad del sistema. Esta

versión ya constituye un resultado de valor para el negocio. Una entrega no debería tardar más 3 meses.

- *Metáfora*: El sistema es definido mediante una metáfora o un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo. Una metáfora es una historia compartida que describe cómo debería funcionar el sistema.
- *Diseño simple*: Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto.
- *Pruebas*: La producción de código está dirigida por las pruebas unitarias. Éstas son establecidas por el cliente antes de escribirse el código y son ejecutadas constantemente ante cada modificación del sistema.
- *Refactorización*: Es una actividad constante de reestructuración del código con el objetivo de remover duplicación de código, mejorar su legibilidad, simplificarlo y hacerlo más flexible para facilitar los posteriores cambios.
- *Programación por parejas*: Esto conlleva ventajas implícitas (menor tasa de errores, mejor diseño, mayor satisfacción de los programadores,...).
- *Propiedad colectiva*: Cualquier programador puede cambiar cualquier parte del código en cualquier momento.
- *Integración continua*: Cada pieza de código es integrada en el sistema una vez que esté lista. Así, el sistema puede llegar a ser integrado y construido varias veces en un mismo día.

- *40 horas semanales*: Se debe trabajar un máximo de 40 horas por semana. No se trabajan horas extras en dos semanas seguidas. Si esto ocurre, probablemente está ocurriendo un problema que debe corregirse.
- *Cliente en casa*: El cliente conduce constantemente el trabajo hacia lo que aportará mayor valor de negocio y además, los programadores podrán resolver de manera inmediata cualquier duda asociada. La comunicación oral es más efectiva que la escrita.
- *Estándares de codificación*: Es indispensable que se sigan ciertos estándares de programación para mantener el código legible.

4. REQUERIMIENTOS INICIALES

El presente capítulo tiene como objetivo presentar una perspectiva concreta sobre el entorno en que se trabajó y la definición de lo que se quería obtener como resultado final. Para esto, primero se hace una breve ambientación sobre el tema en el que se basa el desarrollo de las opciones. Teniendo clara esta situación, se enuncian las opciones que se determinó desarrollar. Luego, se describe el modelo de datos utilizado para el manejo de la información relacionada con la situación planteada y las opciones requeridas. Por último se describe la relación de la funcionalidad móvil, con el sistema principal de SIPA, el cual es un producto de VISION LTDA con derechos de autor registrados, y se hace una breve descripción de las interfaces utilizadas.

4.1. AMBIENTACIÓN

Las opciones Software desarrolladas en la terminal móvil, están basadas en el Sistema Integrado de Información Ambiental SIPA. SIPA es un sistema orientado al manejo de inventario y de la oferta y demanda ambiental. Además, facilita el manejo de expedientes ambientales asociados a sanciones o permisos, y permite a las autoridades ambientales gestionar los traslados de recursos de acuerdo al reglamento vigente. Cierta información manejada en SIPA, es requerida en campo abierto por los técnicos ambientales encargados, los cuales además, requieren registrar algunos datos tomados en terreno, para luego ser sincronizados con la base de datos principal de SIPA.

Dentro del sistema SIPA, se manejan Expedientes Ambientales. Un Expediente puede ser de dos tipos:

- Uno para cuando una persona o entidad solicita a las autoridades ambientales encargadas, algún tipo de licencia o permiso para explotar o aprovechar recursos naturales.
- El otro tipo de Expediente se genera cuando una persona o entidad comete una infracción contra el medioambiente.

En los Expedientes se almacena diferente información general tal como, los datos de la persona titular o entidad, fechas de expedición y de vigencia, documentos asociados al trámite etc. Además de esto, un expediente contiene otra información asociada, como los Derechos Ambientales otorgados (por ejemplo una licencia ambiental o un permiso de tala), predios involucrados, los recursos afectados (agua, bosque etc...), las especies y productos autorizadas, salvoconductos para la movilización de especies y productos entre otros datos. Al otorgarse un derecho ambiental o cometerse una infracción, se pueden generar ciertas obligaciones, las cuales deben ser cumplidas en un plazo dado. Por ejemplo, en el caso de los derechos, al darse permiso de talar una cantidad determinada de árboles en una región específica, se puede generar la obligación de sembrar otros árboles en dicha región, con el fin de compensar el efecto sobre el medio ambiente. Estas obligaciones son registradas en el sistema, y requieren de las visitas periódicas de los funcionarios ambientales a los predios definidos, para verificar el cabal cumplimiento de dichas obligaciones adquiridas. La omisión de estos deberes, puede acarrear sanciones aplicables a las personas o empresas que figuran como responsables en el Expediente asociado.

En este proceso, se hace útil y necesaria, la utilización de la terminal móvil, como medio de almacenamiento de información de interés requerida en las visitas ambientales, teniendo además en cuenta, las ventajas ofrecidas por las características propias de resistencia a condiciones ambientales adversas, de dicho dispositivo. Es necesario, que los técnicos ambientales encargados,

cuenten con la información básica en terreno de los Expedientes vigentes manejados en el sistema SIPA. Además de esta utilidad de consulta, la Terminal móvil ofrece la posibilidad de registrar en terreno cierta información derivada de la realización de las visitas, como el cumplimiento de las obligaciones asignadas junto con recomendaciones u observaciones de los técnicos, entre otros datos. A partir de esta información es posible expedir un documento de soporte a la visita para entregarlo a la persona o entidad involucrada. Luego, toda la información que haya sido recogida, requiere ser sincronizada con la base de datos principal de SIPA disponiendo así, de información confiable y oportuna para las autoridades ambientales.

4.2. DEFINICIÓN DE LAS OPCIONES

Esta sección es declarada como absolutamente confidencial y no se puede divulgar por la universidad sin previo consentimiento de VISION.

Las opciones Software a desarrollar han sido definidas teniendo en cuenta los requisitos establecidos en el manejo de los expedientes y visitas ambientales y la información de mayor importancia requerida en terreno, además de la funcionalidad brindada por las herramientas de desarrollo a utilizar. También se debe tener en cuenta, las limitaciones propias de los dispositivos móviles tales como espacio en memoria, tamaño de la pantalla, funciones software ofrecidas por las herramientas de desarrollo etc. En conjunto con VISION se definieron las siguientes opciones software para su desarrollo:

- a)** Una opción para cargar a la terminal, los datos referentes a los expedientes que aún no estén vencidos, según el funcionario encargado de realizar las

visitas y la zona que se vaya a visitar. Estos datos serán tomados desde el servidor central del Sistema Integrado Ambiental SIPA.

- b)** Una opción que ofrezca la posibilidad de seleccionar solo ciertos expedientes de los que se hayan cargado, según el nombre de la persona titular de estos (beneficiario o infractor según sea el caso).

- c)** Una opción de consulta de datos de los expedientes cargados tal como titular, fechas de vigencia, obligaciones asociadas, recursos y productos otorgados o afectados entre otros. Para facilitar este procedimiento se contará con opciones de búsqueda por los diferentes parámetros de mayor importancia que definen a un expediente.

- d)** Una opción que permita el registro en la terminal de la información recolectada en las visitas técnicas ambientales tal como la fecha de visita, observaciones, persona que atendió, cumplimiento de obligaciones entre otros.

- e)** Una opción para generar e imprimir desde la terminal un documento de soporte a la visita realizada, mediante la utilización de una impresora portátil con Bluetooth.

- f)** Una opción para descargar toda la información recolectada en la terminal hacia el servidor de datos de SIPA.

Luego, cada una de las opciones planteadas anteriormente ha sido descompuesta en actividades específicas para llevar a cabo en el desarrollo las cuales son incluidas en los planes de trabajo. En forma general se asigna el tiempo estimado, fechas de inicio y fin, porcentaje de desarrollo y observaciones. Para esto se utilizan hojas de Excel en donde se consigna esta y otra información, según los

formatos establecidos por el sistema de gestión de calidad de VISION. Esto guarda estrecha relación con la técnica de historias de usuario planteada en la metodología Extreme Programming.

Por ejemplo, la opción del numeral a) puede ser planteada inicialmente como una historia de usuario general, y luego descomponerse en un conjunto de actividades que conlleven al cumplimiento del requisito planteado.

REQUISITO No. 1	
Descripción:	Opción para cargar a la terminal, los datos referentes a los expedientes que aún no estén vencidos, según el funcionario encargado de realizar las visitas y la zona que se vaya a visitar. Estos datos serán tomados desde el servidor central del Sistema Integrado Ambiental SIPA.
Tipo:	Nuevo
Fecha:	01/10/2005
Tiempo Estimado:	10 días con dedicación de 8 horas diarias.
Prioridad:	Alta
Referencia:	
Observación:	

Tabla 3. Definición de requisito

En la anterior tabla se observa el número de requisito, la descripción general de dicho requisito, su tipo (por ejemplo Nuevo, Corrección, Mejora) según se defina, fecha de planteamiento, tiempo estimado de desarrollo, grado de prioridad asignada, referencia a otro requisito (por ejemplo cuando se plantea una mejora debe ir el número del requisito inicial al cual se referencia) y observaciones. Luego, dicho requisito se descompone en actividades o tareas más específicas tales como:

No.	Actividad	Tiempo estimado [Días]	Fecha Inicial	Fecha Final	% Desarrollo	Observaciones
1	Diseño de la interfaz de usuario	1	02/10/2005	02/10/2005	0	
2	Prueba de las consultas SQL para recuperar la información necesitada	1	03/10/2005	03/10/2005	0	
3	Implementación del código en VB .NET	5	04/10/2005	08/10/2005	0	
4	Pruebas con el servidor de datos	1	09/10/2005	09/10/2005	0	
5	Ajustes	2	10/10/2005	11/10/2005	0	

Tabla 4. Actividades específicas del requisito

Cada tarea puede ser modificada si las circunstancias lo ameritan en cualquiera de sus campos. A medida que se avanza en el desarrollo, se elaboran nuevos planes de trabajo para terminar la elaboración de las actividades planteadas, incluir modificaciones o nuevas actividades surgidas por diversas causas. Esto permite hacer un seguimiento al trabajo realizado y tener un control sobre el estado en que se encuentra el desarrollo para replantear las situaciones si es necesario.

4.3. DEFINICIÓN DEL MODELO DE DATOS A UTILIZAR

El modelo de datos utilizado en la terminal móvil, es un subconjunto del modelo del Sistema ambiental SIPA. Para definirlo, se tuvieron en cuenta las entidades de SIPA que manejan o tienen relación con la parte de Expedientes ambientales. Además, no todas estas entidades ni sus atributos son tenidos en cuenta en la

base de datos móvil por cuestiones de limitación de memoria y procesamiento en la terminal. Por esta razón, se hizo una depuración de dicha información, definiendo cual era la de mayor importancia para el manejo en las visitas técnicas ambientales asociadas a los expedientes, y cual es absolutamente necesaria para el correcto desarrollo de las opciones planteadas. Luego del análisis del modelo de datos SIPA se determinó trabajar con las siguientes tablas:

- **Expediente:** Contiene toda la información general perteneciente a los expedientes.
- **Resolución:** Tabla que almacena la información general de una resolución, la cual estará relacionada con los derechos otorgados a un expediente.
- **Obligación Expediente:** Tabla relacionada con Resolución, y contiene los datos de las obligaciones asociadas a un expediente.
- **Seguimiento:** En esta tabla se almacena la información general de una visita ambiental, tal como la fecha, funcionario que realizó la visita, persona que atendió entre otros datos.
- **Detalle Seguimiento:** Tabla relacionada con Obligación Expediente y Seguirimientos. Almacena información tal como Recomendaciones dadas a cierta obligación en una visita y estado de una obligación (cumplida o no).
- **Producto Expediente:** Tabla encargada de guardar los datos de los productos asociados a un expediente tales como cantidad, diámetro, volumen etc.
- **Tipo de derecho:** Tabla que almacena los tipos de derecho en general que se pueden otorgar. Por ejemplo licencia, concesiones entre otros.

- **Clase de derecho:** Almacena las clases de derechos o tipos específicos de derechos que se otorgan para su aprovechamiento o explotación. Por ejemplo concesión para aguas superficiales, aprobación de obras hidráulicas entre otros.
- **Clase de derecho Expediente:** Tabla encargada de guardar los datos de las clases y tipos de derechos relacionados con un Expediente.
- **Producto Ambiental:** Define los diferentes productos ambientales manejados, que pueden derivarse de cierta especie dada, por ejemplo un producto puede ser los tablones obtenidos de cierta especie de árbol.
- **Nombre común de la especie:** Esta tabla contiene todas las especies de los recursos naturales renovables que han sido definidas en un Expediente.
- **Tipo recurso:** Contiene los tipos de recurso que están determinados para definir a los recursos y/o productos otorgados o vulnerados. Por ejemplo bosque, flora, fauna, hidrobiológico entre otros.
- **Unidad de Medida:** Define las unidades de medida de los productos y recursos utilizados en los Expedientes.
- **Recurso afectado expediente:** En esta tabla se almacena que recursos y en que cantidad, están relacionados a un Expediente ya sea por otorgamiento o infracción cometida.
- **Empleado:** En esta tabla se almacena los nombres de los funcionarios encargados de realizar las visitas técnicas.

- **Área:** Define las diferentes zonas a donde pertenecen los funcionarios y Expedientes existentes.
- **Contratista:** Contiene los datos de las personas o entidades a las cuales se les otorga derechos o permisos.
- **Cuenca**³⁴: Contiene los datos de todas las cuencas definidas en las cuales se otorgan permisos, o son víctimas de infracciones.
- **Fuente:** Contiene los datos de todas las fuentes de agua pertenecientes a una cuenca en las cuales se otorgan permisos, o son víctimas de infracciones.

4.4. RELACIÓN CON EL SISTEMA PRINCIPAL SIPA

Una breve descripción de las interfaces a utilizar en el dispositivo móvil será enunciada a continuación, luego de la respectiva descripción de la interfaz relacionada del sistema principal SIPA de escritorio.

4.4.1. Sistema principal SIPA

4.4.1.1. Principal Expedientes

En la interfaz principal de los Expedientes ambientales en el sistema SIPA, se encuentra la información general de estos, tal como: Número de Expediente,

³⁴ Cuenca. Ver Glosario.

vigencia, fecha de solicitud, solicitante, apoderado, descripción del proyecto, observaciones etc.

Teniendo en cuenta las limitaciones propias de la Terminal móvil, tales como espacio en memoria, velocidad de procesamiento y tamaño reducido de la pantalla, fue necesario definir qué información es completamente necesaria para salir al cumplimiento de las actividades en campo abierto. Tomando como base la información general presentada en los Expedientes, se ha definido que en terreno es necesario tener:

- Número del Expediente.
- Fecha Inicio.
- Fecha Fin.
- Vigencia.
- Solicitante.
- Observación.

Del resto de información asociada a un Expediente, la cual está incluida en una sección llamada "Detalles del Expediente", se hace necesario contar con la lista de derechos otorgados o afectados, los recursos y productos ambientales permitidos para explotación o que hayan sido afectados y las obligaciones adquiridas. Además, se debe ofrecer la posibilidad de registrar la información levantada en las visitas ambientales, concerniente con el cumplimiento de las obligaciones mencionadas.

4.4.1.2. Derechos

En el sistema SIPA, se presenta la información de los derechos otorgados a un expediente en un formulario con los siguientes campos: tipo de derecho

(licenciamiento ambiental, permisos, autorizaciones etc...), clase de derecho y recurso afectado (flora, fauna, hidrobiológico etc...).

Para la terminal móvil, se ha definido como necesaria la siguiente información:

- Tipo de Derecho.
- Clase de derecho.
- Recurso Afectado.

4.4.1.3. Recursos

La información de los recursos otorgados o afectados contiene datos tales como: nombre del recurso, cantidad, unidad, cuenca a la que pertenece y fuente.

Para la terminal móvil se ha determinado cargar la siguiente información de recursos:

- Recurso.
- Cantidad.
- Unidad.
- Cuenca.
- Fuente.

4.4.1.4. Productos ambientales

Los productos ambientales permitidos contienen información tal como: el tipo de recurso, nombre del producto ambiental, especie ambiental, cantidad y unidad de medida.

Para la terminal móvil, se cargan todos los campos descritos anteriormente en su base de datos local.

- Tipo de recurso.
- Producto ambiental.
- Especie.
- Cantidad.
- Unidad.

4.4.1.5. Obligaciones

Las obligaciones adquiridas en un expediente dado, están relacionadas con una resolución o acto administrativo, pudiendo cada acto tener una o varias obligaciones relacionadas junto con su duración y fechas inicial y final.

En la terminal móvil se cuenta con los siguientes datos que definen a las obligaciones:

- Numero de acto administrativo.
- Fecha de vencimiento del acto.
- Fecha de notificación del acto.
- Descripción de la obligación.
- Duración.
- Fecha inicial.
- Fecha final.

4.4.1.6. Visitas ambientales

Para el registro y seguimiento de las visitas ambientales realizadas y asociadas a un Expediente dado, se cuenta con un formulario en el sistema principal SIPA, en la cual se manejan datos tales como: la fecha de la visita, la persona que atendió a la visita, funcionario que realizó la visita, observaciones entre otros datos.

En la Terminal móvil, se ofrece la posibilidad de registrar estos mismos datos, durante la realización de las visitas, con el fin de ser sincronizados luego con la base de datos principal. Se contará con:

- Fecha de la visita.
- Resolución.
- Persona que atiende.
- Funcionario.
- Recomendaciones generales.

Además se puede registrar detalles del seguimiento a obligaciones específicas, especificando si ésta ya fue cumplida o no, junto con su respectiva justificación.

4.5. DESCRIPCIÓN DE INTERFACES

A continuación se hace una breve descripción de las interfaces utilizadas, mencionando su estructura básica y su funcionamiento.

4.5.1. Inicio

Cuando la aplicación es iniciada, aparece un formulario el cual contiene cajas de texto en las que se especifica que datos se encuentran cargados en la base de datos de la terminal, es decir, de que Zona y Técnico. También se muestra la fecha y hora de la última sincronización realizada.

Se cuenta con una sección llamada “Datos a cargar”, en donde aparecen dos listados con las zonas y técnicos disponibles en la base de datos principal. Cuando se quiera cargar nuevos datos a la terminal se debe escoger la zona y técnico deseados y oprimir un botón llamado “Sincronizar con el servidor” el cual invocará al formulario de sincronización de la aplicación (Ver sección 4.5.5).

También existe un botón de selección de Expedientes, el cual invoca al formulario encargado de dicha tarea. El botón “Continuar”, da el acceso a la información requerida por los técnicos ambientales, activando al formulario principal de Expedientes (Ver sección 4.5.2).

4.5.2. Principal de Expedientes

En el dispositivo móvil se cuenta con una interfaz, en la cual se permite la visualización de los datos generales y detallados de un Expediente.

Se cuenta con cinco pestañas o fichas. En la primera se visualizan los datos generales del Expediente (número, vigencia, fechas de inicio y fin, observaciones y beneficiario o infractor según sea el caso). En la segunda, llamada Derechos, se cuenta con una lista de los derechos asignados al Expediente seleccionado. La siguiente ficha llamada Obligaciones, muestra un listado de los actos administrativos o resoluciones del Expediente, junto con otro listado de las obligaciones asociadas a cada resolución. La ficha Recursos contiene un listado de todos los recursos otorgados o afectados en el Expediente. La última ficha,

llamada Productos, muestra una lista de las especies y productos permitidos o afectados en el Expediente.

En la parte superior se cuenta con una barra de acceso rápido con botones de desplazamiento por los diferentes Expedientes cargados. También existe un menú de opciones que ofrecen una funcionalidad equivalente a ciertos botones de la barra. Hay un botón de búsqueda (o la opción de menú Buscar) que invoca un nuevo formulario, en el cual se presenta la opción de buscar Expedientes por diferentes parámetros establecidos (Ver sección 4.5.3). También existe un botón que llama al formulario de visitas ambientales (o la opción de Menú Visitas) encargado de visualizar o registrar visitas al Expediente seleccionado (Ver sección 4.5.4).

4.5.3. Consulta de Expedientes

En este formulario se pueden realizar búsquedas de Expedientes que cumplan con los diferentes parámetros ingresados (número, vigencia, fecha de inicio, fecha de fin y solicitante). Al escribir cierto(s) valor(es) y pulsar el botón “Buscar”, se mostrará en el listado inferior los resultados que cumplieron con los valores ingresados. Al tener un expediente seleccionado en el listado de resultados y pulsar un botón llamado “Aceptar”, aparecerá nuevamente el formulario “Expedientes” situado en el Expediente escogido anteriormente.

4.5.4. Visitas ambientales

En este formulario se manejan tres fichas. La llamada General, contiene datos generales tales como fecha de la visita, persona que atendió, funcionario que realizó la visita, resolución y recomendaciones. En la segunda pestaña, llamada Observaciones, se tendrá el listado de las obligaciones asociadas a la resolución

seleccionada en la ficha General, incluyendo por cada una: la descripción de la obligación, fechas de inicio y fin y duración. Existe un botón llamado “Registrar Detalle” que activa a un nuevo formulario para registrar un detalle de seguimiento a una Obligación seleccionada (Ver sección 4.5.4.1). La tercera ficha llamada Detalles, contiene un listado de los seguimientos registrados a las obligaciones listadas en la ficha Obligaciones. Posee botones que permiten eliminar o modificar alguno de los detalles de seguimiento a obligaciones registrados.

En la parte superior se cuenta con una barra de acceso rápido con botones de desplazamiento por las diferentes visitas registradas. También existe un menú de opciones que ofrecen una funcionalidad equivalente a ciertos botones de la barra. Hay un botón para Guardar (o la opción de menú Guardar) una nueva visita o cambios en una existente según sea el caso. Hay un botón para eliminar la visita (o la opción de menú Eliminar). Existe otro botón que dispone al formulario para el registro de una nueva visita (o la opción de menú Nuevo). Por último existe una opción de menú llamada “Generar Recibo”, que presenta en pantalla el documento de soporte a entregar en la visita, listo para ser impreso.

Este formulario, dependiendo del caso dado, puede servir para visualizar las visitas ambientales realizadas sobre un expediente, o para registrar la realización de una visita, junto con el cumplimiento de las obligaciones relacionadas.

4.5.4.1. Detalles de seguimiento a obligaciones

En este formulario se registra el cumplimiento o no de una obligación dada. Contiene la descripción de la obligación, una lista para escoger los posibles valores de cumplimiento (SI, NO, PARCIAL) y una caja de texto para escribir la justificación. Presenta botones para guardar el detalle de seguimiento en la base de datos o para eliminarlo.

4.5.5. Sincronización

Este formulario tiene la función de iniciar un proceso de sincronización con el servidor a partir de los parámetros ingresados. Contiene tres cajas de texto en donde se ingresa el nombre de usuario, password y URL del servidor móvil. Al haber ingresado los anteriores parámetros, se elige el tipo de sincronización deseado (“Sincronización completa” o “Solo descargar datos recolectados”) y se pulsa el botón Iniciar.

5. DESARROLLO DE LAS OPCIONES

Este capítulo presenta una descripción del orden de desarrollo y ciertos aspectos técnicos tenidos en cuenta en el desarrollo de las opciones software planteadas. Para comenzar se hace un análisis de los ítems a tener en cuenta en el desarrollo de interfaces de usuario en dispositivos móviles. Luego se describe la arquitectura y conceptos más importantes de la sincronización de datos con Oracle Lite. Se sigue con la descripción del modelo piloto que sirvió como base para el desarrollo inicial y refinamiento de las opciones software. Luego se menciona brevemente los aspectos tenidos en cuenta para la generación e impresión de reportes desde la terminal móvil. También se expone cierta parte de la metodología y plan de pruebas utilizado. Por último se menciona el contenido en general de la documentación elaborada como anexo a la del sistema SIPA.

5.1. ÍTEMS PARA EL DISEÑO DE INTERFACES EN DISPOSITIVOS MÓVILES

El diseño de la interfaz de usuario en un dispositivo móvil, es un aspecto de gran importancia en el desarrollo de la aplicación, puesto que debe contar con la aceptación por parte del cliente. El diseño de interfaz en una aplicación para dispositivos móviles, determina con que facilidad, rapidez y eficiencia puede llevar a cabo sus tareas un usuario. Al optimizar la interacción del usuario con la interfaz de la aplicación, se puede sacar mayor provecho de las capacidades ofrecidas por estos dispositivos.

Al diseñar la aplicación para la terminal móvil, se tuvieron en cuenta las diferencias existentes en varios aspectos, entre el dispositivo mencionado y un equipo de

escritorio común. Estas diferencias ofrecen ciertas limitantes con las que hay que lidiar, pero también algunas entregan ventajas que se deben aprovechar al máximo. Algunas diferencias son:

- *Tamaño:* Obviamente los dispositivos son mucho más pequeños que los computadores de escritorio. Esto presenta una ventaja, que se traduce en portabilidad y movilidad disponible para el usuario.
- *Funciones y tamaño de Pantalla:* Los dispositivos trabajan bajo un pequeño número de colores, y sus resoluciones se encuentran en los 320x240 píxeles. Además su tamaño de pantalla normalmente es de 3.5" diagonal.
- *Métodos de entrada y salida:* Los dispositivos móviles pueden usar métodos de interacción con el cliente, como botones de hardware, lápices ópticos, voz o teclado. En el caso especial, la terminal utilizada para el desarrollo utiliza lápiz óptico o hasta el dedo, para tocar objetos en la pantalla táctil. Se cuenta con un teclado alfanumérico externo, que facilita visiblemente el ingreso de datos al dispositivo, pues libera al usuario del uso del teclado virtual, el cual no es muy cómodo, y menos en las condiciones de trabajo donde se espera que opere la aplicación.
- *Limitaciones de Memoria:* Es evidente que los dispositivos cuentan con mucha menos memoria que los equipos de escritorio. En especial, se cuenta con 64 MB de RAM Flash, de los cuales alrededor de 20 MB son utilizados por el Sistema Operativo y sus aplicaciones. Aparte se cuenta con 128 MB de RAM para el almacenamiento de otros datos y la ejecución de programas.

- *Utilización y ergonomía:* Los dispositivos móviles ofrecen la gran ventaja de poder utilizarse en la mano, mientras se camina, se habla con otra persona, o se realizan otras tareas que requieren de constante movilidad.

- *Mayor fragilidad:* Normalmente los dispositivos móviles comunes presentan un riesgo de daño, por diferentes factores inherentes a las condiciones existentes en su funcionamiento normal. Al aprovecharse la movilidad ofrecida, se cae también en el riesgo de dañar el dispositivo por golpes, caídas, polvo, agua etc. En el caso especial de desarrollo de la aplicación planteada esto no presenta inconveniente alguno, pues este aspecto fue tenido muy en cuenta al comienzo del desarrollo del proyecto. Por tal razón se aseguró de conseguir una terminal que no presentara dichas limitaciones, y por tanto, aumentara en gran proporción la funcionalidad y capacidad de adaptación a los ambientes reales de operación, en donde se presentan condiciones medioambientales hostiles, que muy seguramente no resistirían las terminales portátiles comunes. El cumplimiento de la norma industrial IP54 por parte del dispositivo, garantiza el sellamiento contra el polvo y contra proyecciones o inmersión en el agua. Además su tecnología de ensamblaje, permite que resista múltiples caídas desde aproximadamente 1.3 metros sobre concreto. En el caso de la impresora móvil también se tuvo en cuenta este aspecto, y por tanto se utilizó un modelo que igualmente ofrece resistencia a múltiples caídas sobre concreto.

Además, para el diseño de interfaces de usuario en dispositivos móviles, se deben seguir criterios tales como:

- *Simplicidad:* Los usuarios finales esperan aplicaciones sencillas orientadas a la tarea específica de visualización y captura de datos. Si una aplicación presenta muchos menús, vistas y controles, puede confundir al usuario que desee realizar cierta tarea con rapidez.

➤ *Coherencia*: La coherencia consigue que una interfaz sea familiar y predecible, reduciendo así la confusión y aumentando la productividad. Una aplicación debe tener coherencia por motivos tales como:

-Facilidad de aprendizaje: Los usuarios pueden aplicar las habilidades aprendidas en una aplicación, a otras que se ejecuten en el mismo dispositivo o entorno.

-Mayor comodidad y confianza en el uso del dispositivo: Los usuarios valoran la sencillez y familiaridad en la ejecución de tareas.

-Mayor productividad: Con un tiempo de aprendizaje más breve, los usuarios requieren menos tiempo para aprender a realizar una tarea, y por tanto, pueden empezar a trabajar más rápidamente.

➤ *Control de usuario*: Los usuarios valoran los productos que pueden controlar, y con los que se sienten más cómodos. Las personas usan una aplicación, porque desean realizar tareas concretas con facilidad y con el menor número de dificultades posibles. Debe alcanzarse un equilibrio entre la sencillez de la interfaz y el nivel del control del usuario y las opciones que se le suministran.

➤ *Estética*: La estética se refiere a la imagen de una aplicación y a su forma de comunicar la información al usuario. Una interfaz atractiva y bien diseñada, ayuda al usuario a seleccionar la información apropiada. Sin embargo, todos los elementos visuales de la pantalla compiten por la atención del usuario. Por esto, hay que asegurarse de que todos ellos contribuyan a ofrecer un valor máximo para que el usuario pueda realizar sus tareas.

Hay que tener en cuenta la resolución de la pantalla. Recordando que está en 240x320 píxeles, los controles se han agrupado en páginas con pestañas o fichas. Estos controles son compactos para ahorrar espacio y evitar sobrecargar las formas. Además son familiares al usuario, pues son comunes a los utilizados normalmente en las aplicaciones de escritorio. Se utilizan elementos de presentación estándar. La organización de los elementos debe presentar una secuencia lógica para que el usuario pueda desplazarse entre los controles con facilidad y sin esfuerzo. Para la entrada de datos se utilizan cajas de despliegue, controles de selección y algunas cajas de texto. Además se debe permitir la navegación entre controles, mediante la utilización de teclas (por ejemplo la tecla TAB), como en las aplicaciones de escritorio, para aprovechar el teclado externo del dispositivo, y reducir el uso del lápiz óptico, ofreciendo así, posibilidades de interacción entre el cliente y la terminal, las cuales serán adoptadas según la preferencia de cada usuario.

5.2. SINCRONIZACIÓN CON ORACLE LITE

En el desarrollo de la práctica, se dedicó cierto tiempo a estudiar y comprender el proceso de sincronización de datos utilizado en Oracle Lite. Fue preciso tener claro estos conceptos para el desarrollo de las opciones Software planteadas, teniendo en cuenta la gran importancia de manejar información de una manera segura, confiable y actualizada constantemente. Toda la información descrita a continuación ha sido traducida desde la documentación de Oracle Lite 10g (10.0.0) en las secciones *Getting Started Guide* y *Developers Guide (Synchronization)*.

5.2.1. Conceptos generales

En Oracle Lite los datos son sincronizados invocando a Mobile Sync (lo cual es una pequeña aplicación que reside en el dispositivo móvil desde la instalación del respectivo cliente de Oracle Lite y que sincroniza los ítems definidos en una aplicación publicada) que a su vez interactúa con el Mobile Server (Servidor de aplicaciones utilizado por Oracle Lite para publicar, provisionar, sincronizar y administrar aplicaciones). Este último utiliza objetos de sincronización tales como usuarios, publicaciones, ítems de publicación y suscripciones, para procesar cambios en los datos del cliente y del servidor. Esto se refiere o es conocido como el modelo de Publicación y Suscripción, manejado por Oracle Lite.

Algunos conceptos necesarios para el entendimiento del proceso de sincronización son los siguientes:

- **Ítem de publicación:** Es un objeto del Mobile Server, que contiene una consulta SQL definida sobre una tabla, vista o sinónimo de una Base de datos Oracle, y que especifica a que subconjunto de datos puede acceder un usuario móvil.
- **Publicación:** Es un objeto del Mobile Server que sirve como contenedor de un grupo de ítems de publicación. También contiene información específica tal como la plataforma en que funcionará la aplicación.
- **Aplicación:** Todas las aplicaciones tienen asociada una publicación que define los datos necesarios para su funcionamiento.
- **Suscripción:** Una suscripción asocia a un usuario con una publicación. Cuando un usuario sincroniza por primera vez con el Mobile Server, una base de datos Oracle Lite (con extensión ODB) es creada para cada

suscripción. Cada ítem de publicación de la Publicación, se convierte en un Snapshot en esta base de datos. Un Snapshot es pues, un subconjunto del servidor de datos, que se representa como una tabla en la base de datos de la aplicación móvil. Son los objetos que se sincronizan con el servidor de datos. Solo pueden ser creados por el Mobile Server como una parte de la instalación de la aplicación.

- **Parámetro de suscripción:** Son como variables contenidas en los ítems de publicación, que sirven para especificar valores específicos que definen una suscripción individual de un cliente a una publicación. Permiten que solo el subconjunto de datos requerido sea cargado en el dispositivo del cliente.

5.2.2. Proceso de sincronización

Cuando una aplicación ha sido desarrollada, se procede a desplegarla (mediante el Asistente de Empaquetado instalado con el *Mobile Development Kit* de Oracle Lite. Para mayor detalle ver Anexo 1). Para esto, cada aplicación define unos requerimientos de datos usando una publicación. Una publicación es vinculada a un esquema de Base de Datos, el cual, contiene uno o más ítems de publicación. Una suscripción define la relación entre un usuario y una publicación. Una vez se suscribe, se comienza a recibir información asociada con la publicación. Con Oracle Lite se reciben Snapshots (estos capturan los cambios realizados en los datos desde el dispositivo móvil). La aplicación y toda su información asociada son almacenadas en un repositorio móvil, el cual es instalado con el servidor móvil y reside en la Base de Datos principal Oracle.

El proceso de sincronización es iniciado por el usuario, mediante la aplicación Mobile Sync (mSync³⁵), la cual se comunica con el Servidor Móvil y sube los

³⁵ Mobile Synchronization Application (mSync). Aplicación que reside en el dispositivo móvil y se encarga del proceso de sincronización.

cambios hechos en la máquina cliente. Este a su vez, baja los cambios para el cliente hechos en el Servidor Móvil.

En el Servidor Móvil existe un proceso oculto llamado MGP (*Message Generator Process* o en español Proceso Generador de Mensajes) que periódicamente colecta los cambios subidos por todos los clientes y los aplica a las tablas de la Base de Datos central. Este también compone nuevos datos que son bajados a cada cliente durante la siguiente sincronización, basándose en las suscripciones predefinidas.

La figura mostrada a continuación, expone el modelo y arquitectura de sincronización manejado por Oracle Lite:

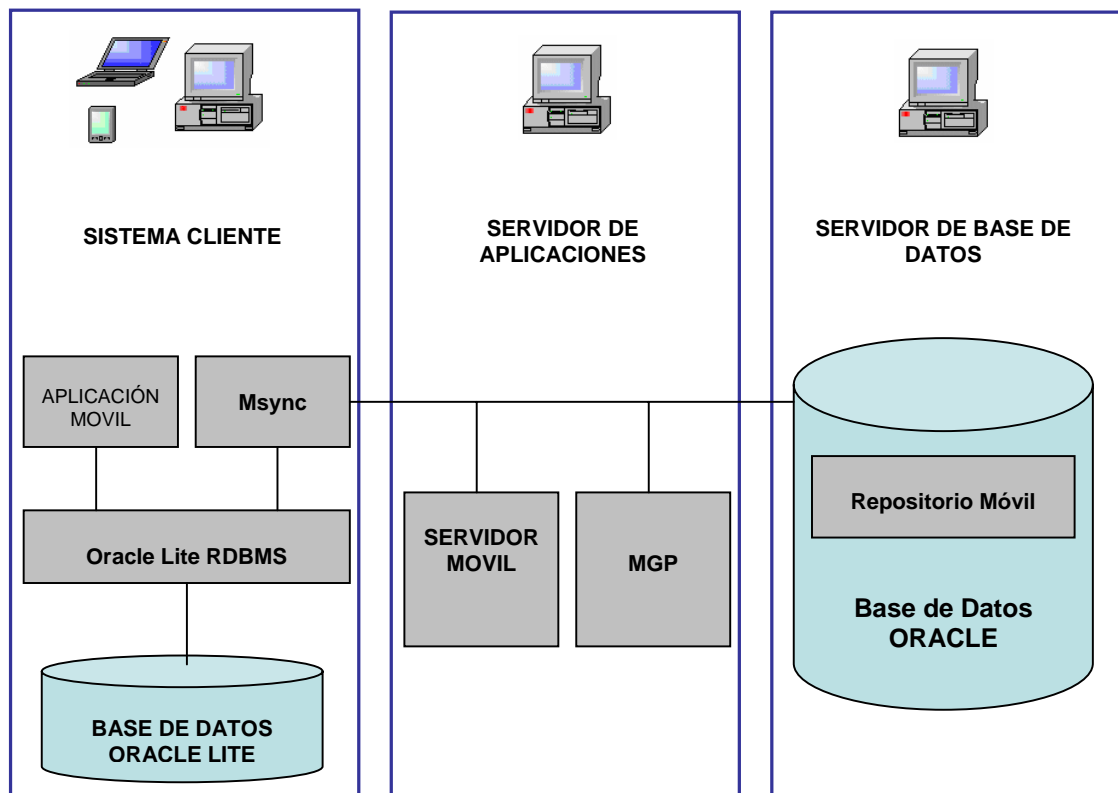


Figura 3. Arquitectura de las aplicaciones de Oracle Lite.

5.2.3. Métodos de sincronización

Básicamente existen tres tipos de ítems de publicación que pueden ser usados para definir la sincronización. Estos son: fast refresh, complete refresh, y queue based.

Método de Sincronización	Características
Complete Refresh	Todos los datos para una publicación, son siempre bajados al cliente, por lo tanto este método requiere una gran cantidad de tiempo
Queue Based	Forma más básica de sincronización. No existe lógica de sincronización; por tanto, no se cuenta con detección de conflictos, información del cliente o actualizaciones del lado del servidor ³⁶
Fast Refresh	Es el más utilizado normalmente. En este, los cambios del cliente son subidos al servidor, y los cambios del servidor son bajados al cliente. Este método utiliza un proceso corriendo oculto y periódicamente llamado MGP (Message Generator Processor), el cual recolecta los cambios subidos por todos los clientes y los aplica a las tablas del servidor de datos. Además compone nuevos datos que son bajados a cada cliente durante la siguiente sincronización, basándose en las suscripciones predefinidas.

Tabla 5. Métodos de Sincronización de Oracle Lite

36

5.2.3.1. Refrescamiento rápido

La siguiente figura ilustra el proceso de sincronización Fast Refresh (refrescamiento rápido) el cual usa un método asíncrono para sincronizar los datos de los clientes con los del servidor principal. Esto quiere decir que el Mobile Sync opera independientemente del MGP.

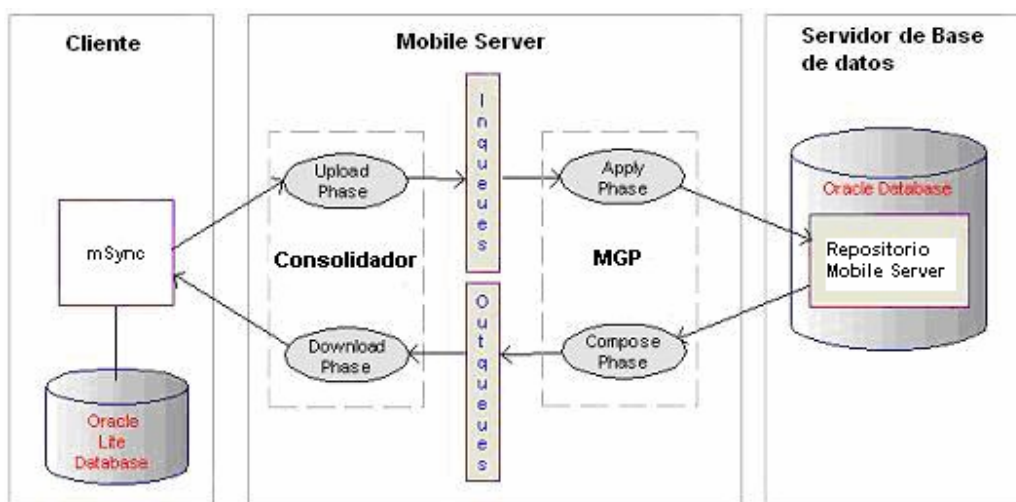


Figura 4. Proceso de sincronización Fast Refresh

A continuación se explicará brevemente el funcionamiento del proceso tanto en el cliente como en el servidor.

Operaciones del cliente

Los cambios hechos a la Base de Datos Oracle Lite del cliente son acumulados y definidos con códigos para el tipo de DML (*Data Manipulation Language* o en español Lenguaje de manipulación de datos) realizados como son las operaciones de Inserción (*Insert*), Actualización (*Update*) o Eliminación (*Delete*).

Cuando se realiza el proceso de sincronización, los datos son encriptados, comprimidos y enviados a el Mobile Server para poblar a *In queues* (ver figura 4). Los *In queue* y *Out queue* son objetos de Base de datos persistentes creados para guardar datos temporalmente durante la sincronización. Durante la misma sesión de sincronización los Snapshots en el cliente son actualizados con los datos que están en *Out queue*.

Operaciones del servidor

Para cada usuario existente y suscrito a alguna aplicación, el MGP toma todo el contenido de él, que se encuentre en *In queue* y aplica estos cambios a las tablas del servidor de datos Oracle. Cualquier conflicto es detectado y resuelto en ese tiempo. La fase *Apply* es completada cuando los cambios subidos por todos los usuarios son procesados.

Después de la fase *Apply* el MGP interviene las tablas de la Base de datos. Este proceso compone y almacena cualquier cambio en *Out queue* para luego ser bajado al cliente en la próxima sincronización.

En este proceso de refrescamiento rápido las fases *Upload* y *Download* son realizadas independientemente de las fases *Apply* y *Compose*. Durante cualquier sesión de sincronización, el *Download* ocurre después del *Upload*, y *Compose* ocurre después de *Apply*.

En el desarrollo de esta práctica empresarial se utilizó una combinación de los métodos Fast Refresh y Complete Refresh para sincronizar los datos.

5.2.4. Errores y conflictos de sincronización

En Oracle Lite un *error de compatibilidad* ocurre cuando el cliente actualiza una fila que es eliminada al mismo tiempo por el servidor. Todos los otros errores como violaciones de nulidad, llaves foráneas o violación de restricciones son *errores de sincronización*.

El Mobile Server no resuelve errores de sincronización automáticamente. Cuando estos se presentan se deshace la transacción y se mueve esta operación a *Error Queue* (cola de errores). Luego, el Administrador de la base de datos podrá cambiar estas transacciones y reejecutarlas o simplemente borrarlas de *Error Queue*.

Los conflictos de sincronización, si son resueltos automáticamente por el Mobile Server. Estos ocurren cuando:

- El cliente y el servidor actualizan la misma fila.
- El cliente y el servidor crean una fila con el mismo valor de llave primaria.
- El cliente borra la misma fila que el servidor actualizó.

Estos conflictos son resueltos teniendo en cuenta la regla de resolución de conflictos fijada en la definición de la aplicación (Ver Anexo 1), en la cual se determina si se fijan los valores existentes en el servidor o en el cliente. También se puede personalizar este proceso mediante funciones y procedimientos almacenados en el servidor de base de datos.

5.2.5. Usuarios concurrentes

La arquitectura de Oracle Lite permite que varios usuarios realicen operaciones de sincronización de datos al mismo tiempo, a través del Mobile Server. Mediante equipos que tengan acceso al PC donde se encuentre instalado el Mobile Server, se puede tener acceso concurrente de usuarios que quieran cargar y descargar datos con el servidor principal. Esto se hace posible porque Oracle Lite internamente maneja una serie de hilos con cada uno de los usuarios que realiza peticiones de sincronización. Por tanto, Oracle Lite es de gran ayuda en entornos donde existen varios usuarios que necesitan sincronización al mismo tiempo, por ejemplo en una empresa de distribución de artículos donde todos sus trabajadores encomendados deben cargar en sus terminales la lista de paquetes y rutas de entrega a las ocho de la mañana, antes de salir a realizar su trabajo.

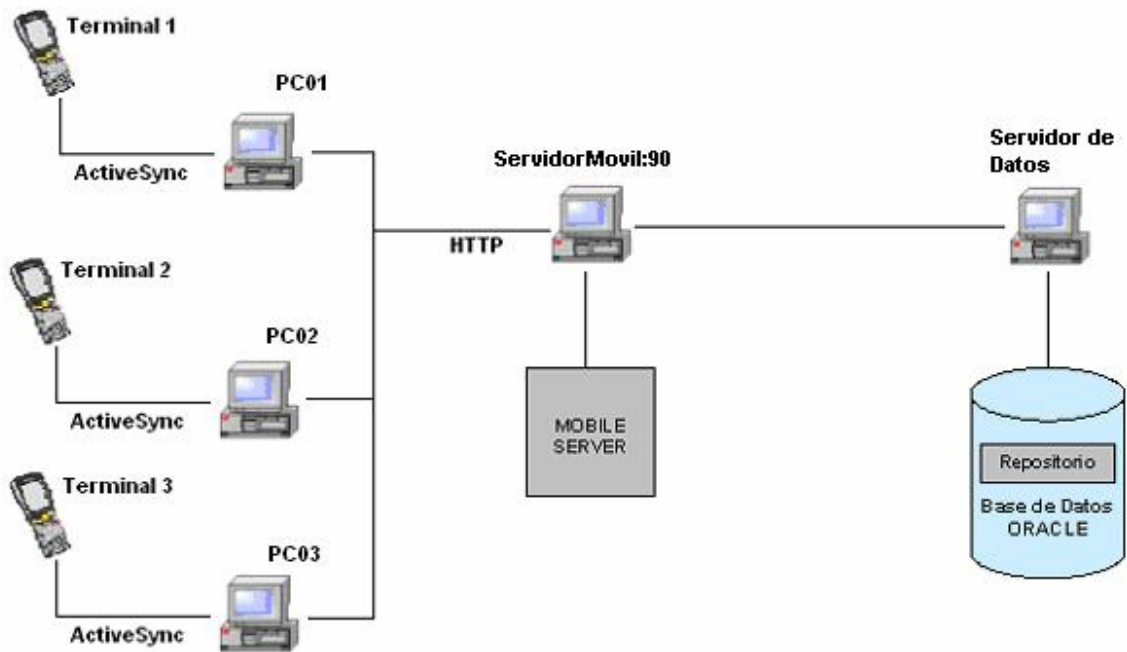


Figura 5. Usuarios concurrentes en Oracle Lite

5.3. ENTREGAS PRELIMINARES

Cabe recordar que la metodología de desarrollo Extreme Programming, seguida en la práctica, se clasifica dentro de las metodologías evolutivas, pues uno de sus principios fundamentales es el de hacer pequeñas entregas constantemente. Las entregas deben hacerse cuanto antes, porque entre más se tarde en introducir una parte esencial, menos tiempo habrá luego para trabajar en ella. De esta forma, un error en una parte esencial del sistema se encontrará pronto y por tanto, se podrá arreglar antes.

Por esta razón, se construyó un prototipo inicial básico de la aplicación (sin validaciones ni estructura definida), con algunas de las opciones planteadas, para servir como base en el desarrollo y luego ser refinado a través de cada iteración, basándose en las evaluaciones realizadas según la funcionalidad requerida, y en los diferentes aspectos surgidos para tener en cuenta, fruto de nuevos requisitos o determinaciones de los ingenieros líderes del proyecto. Además, el desarrollo básico de las opciones sirvió como prueba para poner en práctica y familiarizarse con los conceptos estudiados previamente sobre las tecnologías de desarrollo utilizadas. Las actividades más relevantes que se realizaron fueron:

- Creación de una base de datos Oracle Lite de prueba (extensión .odb), a partir de las entidades y atributos definidos como necesarios para la aplicación móvil, basándose en el modelo y los datos existentes en SIPA, para evaluar el comportamiento y tamaño aproximado que tendría la base de datos móvil.

- Implementación básica de la interfaz de visualización de los expedientes cargados a la base de datos Oracle Lite de la terminal, mostrando los datos generales del expediente, derechos, obligaciones, recursos y productos afectados para comparar con la información de la base de datos central.

- Opción de consulta de expedientes cargados a la base de datos Oracle Lite por los diferentes parámetros planteados.
- Implementación básica de las opciones para el registro, modificación y eliminación de los datos generales recolectados en una visita técnica ambiental, tales como fecha, funcionario, persona que atendió, recomendaciones etc.
- Implementación básica de la opción de sincronización con la base de datos principal, utilizando solamente las funciones propias ofrecidas por Oracle Lite.

5.4. IMPRESIÓN DE SOPORTES DE VISITA

Como se planteó desde el inicio de la práctica, debe existir la opción de generar e imprimir ciertos reportes o recibos de soporte desde la terminal portátil en una impresora móvil vía Bluetooth.

En este sentido, se encontraron varias limitantes para lograr el objetivo planteado. Como se cita en la página de Microsoft : *“.NET Compact Framework no incorpora compatibilidad nativa con Bluetooth”, “Teniendo en cuenta que las aplicaciones móviles apenas imprimen, no es ningún problema eliminar la familia completa de controles relacionados con la impresión. Por tanto, se han eliminado los controles CrystalReportViewer, PageSetupDialog, PrintDialog, PrintDocument, PrintPreviewControl y PrintPreviewDialog.”*³⁷. Teniendo en cuenta estas premisas, se descartó el control directo de la impresión desde la aplicación desarrollada con

³⁷ <http://www.microsoft.com/spanish/msdn/articulos/archivo/050503/voices/netcfgetstarted.asp>

Visual Basic .NET 2003, aunque existen en el mercado ciertas utilidades y SDK que permiten realizar dicha tarea, pero que requieren el pago de licencias para su utilización (el más utilizado es **PrinterCE.NetCF SDK**³⁸ de la empresa FieldSoftware, pero su licencia es vendida mínimo para 100 instalaciones con un costo de 499 dólares)

Por esto se planteó un método de impresión, aprovechando la capacidad nativa que tienen los dispositivos con Windows CE .NET 4.2 de imprimir desde sus aplicaciones (wordpad, Internet Explorer etc.) a diferencia de los Pocket PC (*“los Pocket PC no implementan por defecto la opción de imprimir en ninguna de las aplicaciones. De este modo resulta que no se puede imprimir ni con Bluetooth, ni por infrarrojos y ni siquiera por cable”*). Por tanto la idea es que desde el código de la aplicación, se genere un archivo tipo HTML (porque es un formato de presentación fácil de construir y formatear), se escriba y configure su contenido según sea la necesidad. De esta forma, luego cuando se requiera la impresión, dicho archivo será llamado desde la aplicación mediante la utilización de la función de la API **CreateProcess**. Este archivo aparecerá en pantalla listo para imprimir normalmente como en cualquier aplicación de escritorio (es decir desde File->Print).

5.5. PRUEBAS

En Extreme Programming existen dos tipos de pruebas. Unas son las *pruebas de aceptación*, las cuales son fijadas en compañía del cliente y deben ser superadas completamente, contando así con la aceptación total del producto por parte del cliente. Además, en esta metodología se incluye un concepto de *pruebas unitarias*, las cuales deben ser implementadas a la vez que el código de

³⁸ Tomado de la página http://www.fieldsoftware.com/PrinterCE_NetCF_download.htm

producción. De hecho cada vez que se va a implementar una pequeña parte se escribe una prueba sencilla y el código suficiente para que la pase. Esto ayuda a priorizar y comprobar la evolución del desarrollo y ofrece realimentación inmediata. Se argumenta que encontrar un error puede ser cien veces más caro que realizar las pruebas unitarias.

5.5.1. Pruebas unitarias

Durante el desarrollo de las opciones descritas se han ido realizando pequeñas pruebas unitarias, las cuales, efectivamente han detectado ciertos errores en el funcionamiento de las opciones, dando evidencia de algunos ajustes que debieron ser realizados, pero sobre el código implementado recientemente, evitando así, el avance en generación de código y posteriormente, la mayor dificultad para encontrar la causa de una falla y su respectiva solución. Por ejemplo en la opción de consulta de Expedientes se han ido realizando pruebas unitarias, las cuales han ido siendo superadas satisfactoriamente. Ejemplos de algunas pruebas son:

- Probar el código que valida que haya valores en por lo menos uno de los campos de búsqueda especificados.
- Prueba y código para validar el correcto tipo de datos ingresado en el parámetro de búsqueda. Por ejemplo si se espera una fecha, validar que el valor ingresado tenga un formato de fecha válido.
- Probar la correcta construcción de la cadena de consulta SQL, incluyendo los parámetros definidos por el usuario.
- Verificar directamente en la base de datos principal, que el resultado arrojado por la consulta de la aplicación sea coherente con el esperado.
- Probar la correcta captura de errores por diferentes motivos en la aplicación.

- Probar el código que activa o desactiva los botones de Búsqueda y Aceptación (botón que toma el expediente seleccionado por el cliente en los resultados de la consulta, para ser visualizado junto con toda su información asociada en el formulario principal de Expedientes), según los parámetros ingresados por el cliente y/o los resultados obtenidos de la consulta.

5.5.2. Pruebas funcionales

Además, se plantea una serie de pruebas funcionales para evaluar el comportamiento general de la aplicación. La estrategia a seguir es observar el comportamiento del sistema ante la ejecución de las funcionalidades ofrecidas por la aplicación. Por ejemplo, a continuación se mostrarán algunos de los escenarios de prueba planteados para la opción de *sincronización con el servidor de datos*, junto con la entrada requerida y el resultado esperado.

ESCENARIO DE PRUEBA	ENTRADA	RESULTADO
Iniciar un proceso de sincronización sin estar conectado a un Equipo.	Ingresar a la opción de sincronización, completar los datos requeridos y dar la orden de iniciar el proceso estando desconectado.	La aplicación devuelve un mensaje de error notificando al usuario sobre la ausencia de conexión con un PC.
Iniciar un proceso de sincronización con un nombre de usuario o contraseña inválido.	Ingresar a la opción de sincronización, completar los datos requeridos y dar la orden de iniciar el proceso, teniendo un usuario y/o contraseña inválidos.	La aplicación se conecta con el Mobile Server, verifica y devuelve un mensaje de error diciendo que el nombre de usuario y/o contraseña con inválidos.
Iniciar un proceso de sincronización especificando una	Ingresar a la opción de sincronización, completar los datos requeridos y dar	La aplicación trata de conectarse con el Mobile Server pero devuelve un

ESCENARIO DE PRUEBA	ENTRADA	RESULTADO
dirección URL inválida para el Mobile Server.	la orden de iniciar el proceso, teniendo una URL inválida.	mensaje de error diciendo que la URL especificada no está activada o no existe.
Subir al servidor de datos central de SIPA las visitas registradas en el día en la terminal móvil.	Tener nuevas visitas registradas en la base de datos de la terminal móvil, e iniciar el proceso de sincronización con parámetros válidos.	Al terminar el proceso de aparece un mensaje de "Sincronización finalizada correctamente" y al revisar en el servidor de datos, se encuentran los nuevos datos registrados en las respectivas tablas listos para ser usados por otra aplicaciones.
Cargar en la terminal nuevos datos que pertenezcan solamente a los Expedientes e información asociada de un técnico ambiental específico para una zona determinada.	En el formulario inicial escoger una zona y técnico de las listas desplegadas, ingresar a la opción de sincronización, completar los datos requeridos y dar la orden de iniciar el proceso.	Al terminar el proceso de aparece un mensaje de "Sincronización finalizada correctamente" y al consultar en la terminal se encontrará solamente la información existente del técnico y zona especificados, agilizando así las búsquedas y ahorrando espacio en memoria.

Tabla 6. Casos de prueba de la opción de sincronización

También se presenta el procedimiento necesario para llevar a cabo cada escenario de prueba.

OPCIÓN DE SINCRONIZACIÓN CON EL SERVIDOR DE DATOS
Escenario de Prueba: Iniciar un proceso de sincronización sin estar conectado a un Equipo.
1. Asegurarse de que la terminal no esté conectada a un PC.

OPCIÓN DE SINCRONIZACIÓN CON EL SERVIDOR DE DATOS
<ol style="list-style-type: none"> 2. Ingresar al formulario inicial de la aplicación “Visitas Ambientales” y presionar el botón “Sincronizar con el servidor”. 3. Estando en el formulario “Sincronización”, ingresar parámetros en los campos Usuario, Password y Servidor. 4. Presionar el botón iniciar. 5. Verificar que la aplicación despliegue un mensaje de error indicando que la red no está activa o el nombre host es inválido.
<p>Escenario de Prueba: Iniciar un proceso de sincronización con un nombre de usuario o contraseña inválido.</p> <ol style="list-style-type: none"> 1. Conectar la terminal móvil con el PC mediante la estación base y el cable USB. Esperar a que Microsoft ActiveSync reconozca al dispositivo. 2. Ingresar al formulario inicial de la aplicación “Visitas Ambientales” y presionar el botón “Sincronizar con el servidor”. 3. Estando en el formulario “Sincronización”, ingresar parámetros inválidos para los campos Usuario y/o Password. Escribir la dirección correcta del servidor. 4. Presionar el botón iniciar. 5. Verificar que la aplicación despliegue un mensaje de error indicando que el usuario y/o password son inválidos.
<p>Escenario de Prueba: Subir al servidor de datos central de SIPA las visitas registradas en el día en la terminal móvil.</p> <ol style="list-style-type: none"> 1. Asegurarse de tener nuevas visitas registradas en la terminal móvil con respecto a la última sincronización con el servidor. 2. Conectar la terminal móvil con el PC mediante la estación base y el cable USB. Esperar a que Microsoft ActiveSync reconozca al dispositivo. 3. Ingresar al formulario inicial de la aplicación “Visitas Ambientales” y presionar el botón “Sincronizar con el servidor”. 4. Estando en el formulario “Sincronización”, ingresar parámetros válidos para los campos Usuario, Password y Servidor.

OPCIÓN DE SINCRONIZACIÓN CON EL SERVIDOR DE DATOS
<ol style="list-style-type: none"> 5. Presionar el botón iniciar. 6. Verificar que la aplicación despliegue cuadros de progreso del proceso y al final muestre el mensaje de sincronización finalizada correctamente. 7. Revisar en el servidor de datos principal que las nuevas visitas existentes en la terminal hayan sido descargadas correctamente.
<p>Escenario de Prueba: Cargar en la terminal nuevos datos que pertenezcan solamente a los Expedientes e información asociada de un técnico ambiental específico para una zona determinada.</p>
<ol style="list-style-type: none"> 1. Conectar la terminal móvil con el PC mediante la estación base y el cable USB. Esperar a que Microsoft ActiveSync reconozca al dispositivo. 2. Ingresar al formulario inicial de la aplicación “Visitas Ambientales”, escoger una Zona y Técnico de las respectivas listas desplegables y presionar el botón “Sincronizar con el servidor”. 3. Estando en el formulario “Sincronización”, ingresar parámetros válidos para los campos Usuario, Password y Servidor. 4. Presionar el botón iniciar. 5. Verificar que la aplicación despliegue cuadros de progreso del proceso y al final muestre el mensaje de sincronización finalizada correctamente. 8. Regresar al formulario inicial y revisar que en el recuadro “Datos Cargados Actualmente” se encuentre la zona y técnico escogidos inicialmente en el servidor de datos principal que las nuevas visitas existentes en la terminal hayan sido descargadas correctamente.

Tabla 7. Procedimientos de prueba para la Sincronización

5.6. DOCUMENTACIÓN

Una parte sencilla pero de gran importancia en la documentación ha sido realizada en el código, mediante la utilización de los comentarios, los cuales han servido para describir la funcionalidad de ciertos procedimientos, el significado de algunas variables y en general para explicar brevemente la lógica de la codificación.

Al terminar el desarrollo de las opciones y probar el correcto funcionamiento de estas, se procedió a elaborar los anexos a los manuales técnicos y de usuario del sistema SIPA según los formato establecidos por VISION.

En el manual técnico se especifica el funcionamiento de la aplicación junto con el listado de tablas utilizadas, consultas SQL implementadas y ciertos detalles de código. Además se enuncia los requerimientos iniciales para el funcionamiento de la terminal y la impresora, los requisitos de hardware y Software para la utilización de Oracle Lite y la correcta configuración de este.

En el manual de usuario se hace una descripción más detallada del funcionamiento y operación de las opciones por parte del usuario, eliminando todos los detalles técnicos que no son de interés para él.

6. CONCLUSIONES

El manejo de trámites ambientales en nuestro país es un asunto que demanda gran atención y esfuerzos, con el fin de tener un buen control sobre los recursos naturales que se ven afectados por diferentes actividades comerciales y productivas. Las opciones que se propuso desarrollar, ofrecen la posibilidad de obtener un mejor control sobre estas situaciones, facilitando el trabajo de las autoridades ambientales en el manejo de los recursos de nuestro medio ambiente.

La computación móvil cada día demuestra mucho más su amplio panorama de aplicación y diversidad de posibilidades, que aún faltan por explorar. Es una tecnología que debe ser aplicada cuando se de el caso, pues ofrece una forma efectiva de atacar los problemas inherentes al funcionamiento de las empresas.

El uso de Oracle Lite ofrece grandes ventajas en el manejo de información corporativa en las aplicaciones móviles. Al contarse con un servidor de datos central Oracle, la utilización de la versión Lite en aplicaciones móviles (ya sea en Windows CE o Palm OS), permite obtener una correcta y confiable interacción entre estos dos entornos, al tener a disposición una pequeña pero potente base de datos relacional en el dispositivo móvil que puede ser manipulada con instrucciones SQL igual que sus “hermanos mayores”, y unas herramientas de sincronización que garantizan la coherencia de la información empresarial.

Se pudo conocer y aplicar una nueva metodología para la producción de software: *Extreme Programming*. Esta se basa en la realización de varias iteraciones durante el desarrollo, lo cual permite ver el avance del proyecto y recibir realimentación oportuna a cada uno de los procesos y al surgimiento de nuevas necesidades.

7. RECOMENDACIONES

Elegir la modalidad de trabajo de grado en práctica empresarial, pues es una excelente alternativa, ya que permite al estudiante aplicar todos los conocimientos adquiridos durante la carrera a un entorno real de trabajo, y adquirir una experiencia inicial para desenvolverse en su vida laboral.

Desarrollar aplicaciones móviles para múltiples plataformas al mismo tiempo, en especial, para Windows CE y Palm OS, pues esto le daría a cualquier aplicación una independencia del sistema operativo, lo cual es muy conveniente para acceder a un rango más amplio de usuarios finales.

Incluir en otras aplicaciones móviles en PDAs, opciones de impresión de recibos y soportes necesarios en terreno, mediante la utilización de impresoras portátiles, avanzando de esta forma un poco más en la automatización de los procesos empresariales en campo abierto.

Teniendo en cuenta la gran evolución de las telecomunicaciones, se debe avanzar al desarrollo de sistemas móviles inalámbricos cuando sea posible técnica y económicamente, pues con esto se conseguirían aplicaciones que obtengan y transmitan datos en tiempo real, lo cual generaría información actualizada en todo momento.

Experimentar con otras metodologías de desarrollo de software, como son las denominadas “ágiles”, pues ofrecen un marco de trabajo diferente al tradicional, que puede ser de gran ayuda en proyectos donde no se requiere de mucha formalidad durante el desarrollo.

BIBLIOGRAFÍA

BOEMH, Barry. Software engineering economics. Prentice Hall. Englewood Cliffs, New Jersey. 1981.

CANTOR, Murray. Object-Oriented, Project Management with UML. Wiley. 1998

CÁRCAMO SEPÚLVEDA, José. Bases de Datos Relacionales: Un enfoque práctico de diseño. Universidad Industrial de Santander. 1994

GÓMEZ FLOREZ, Luis Carlos. Planeación de proyectos Informáticos. Grupo de investigación en sistemas y tecnología de la información (STI), Escuela de Ingeniería de Sistemas e Informática, Universidad Industrial de Santander. Bucaramanga, Colombia, 2001.

GRECH, Pablo. Introducción a la Ingeniería. Santa Fé de Bogotá, Colombia. 2001. Ed. Prentice Hall. Primera edición.

INSTITUTO COLOMBIANO DE NORMAS TÉCNICAS Y CERTIFICACIÓN. Compendio: tesis y otros trabajos de grado. Santa Fe de Bogotá: ICONTEC, 2002.

LETELIER, Patricio, PENADÉS, M^a Carmen. “Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)”. Departamento de Sistemas Informáticos y Computación. Facultad de Informática. Universidad Politécnica de Valencia. www.willydev.net/desacargas/masyxp.pdf. 2003.

CASTRO MORA Giovanni Andrés, LEAL GÓMEZ José Luis. *MOVILCRED - APLICACION EN COMPUTACION MOVIL PARA LA CAPTURA DE DATOS EN CAMPO COMO SOPORTE AL CATASTRO DE REDES DE ACUEDUCTO*. Universidad Industrial de Santander, 2004.

PETROUTSOS, Bilgin. *Visual Basic .NET Database programming*. Sybex, 2000.

LEY 99 DE 1993. Disponible en http://www.secretariasenado.gov.co/leyes/L0099_93.HTM.

EXTREME PROGRAMMING: A GENTLE INTRODUCTION. Disponible en línea en <http://www.extremeprogramming.org/>.

NORMATIVIDAD AMBIENTAL. Disponible en <http://www.ideam.gov.co:8080/legal>

En línea < <http://msdn.microsoft.com/library/spa/>>.

Oracle Database Lite InDepth. Documentación técnica de Oracle Lite, 2003.

Oracle Database Lite 10g (10.2.0) Documentation, 2005.

En línea <<http://www.todopocketpc.com>>

ANEXO 1. DEFINICIÓN DE APLICACIONES CON ORACLE LITE

Este anexo ha sido adaptado desde la documentación de Oracle Lite 10g (10.0.0) en las secciones *Developers Guide (Synchronization)* y *Tools And Utilities Guide (Using The Packaging Wizard)*.

Después de haber terminado la implementación del código de una aplicación, se deben definir los comandos SQL que recuperen los datos para los Snapshots de los usuarios. Para esto se utiliza la herramienta "Packaging Wizard", la cual es instalada con el Mobile Development Kit.

Esta aplicación (llamada *wtgpack.exe*) es instalada en la ubicación *Oracle_Home\Mobile\Sdk\BIN*. Al lanzarla aparece en pantalla la siguiente interfaz:

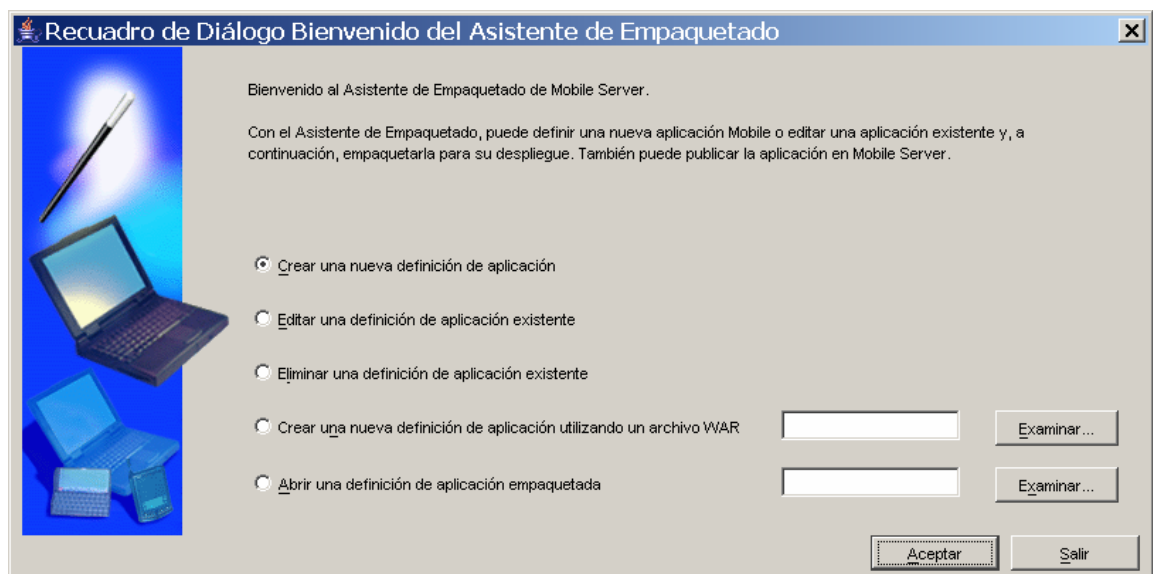


Figura 6. Interfaz inicial del asistente de empaquetado

La herramienta permite crear nuevas definiciones de aplicaciones, editar definiciones ya existentes (aplicaciones ya publicadas), eliminar definiciones, crear nuevas definiciones a partir de archivos WAR, o abrir definiciones empaquetadas (archivos .JAR). Al elegir crear una nueva aplicación, el primer paso a seguir es elegir de una lista desplegable la plataforma soportada por la aplicación. Luego se debe ingresar la información general de la aplicación tal como nombre, ruta de acceso virtual (nombre con el que se publicará en el Mobile Server), descripción entre otros.

Asistente de Empaquetado

Utilice el panel Aplicación para asignar un nombre y describir la aplicación Mobile.

Nombre de la Aplicación:

Ruta de Acceso Virtual:

Descripción:

Directorio de la Aplicación Local:

Nombre de la Publicación:

Figura 7. Panel general de la Aplicación

Luego aparece el panel de archivos para especificar los archivos que se deseen incluir (por ejemplo los instaladores de la aplicación desarrollada en Visual Basic .Net, imágenes a utilizar etc.). En seguida aparece el panel Base de Datos, en donde se especifica el nombre que tendrá la Base de datos en el cliente.

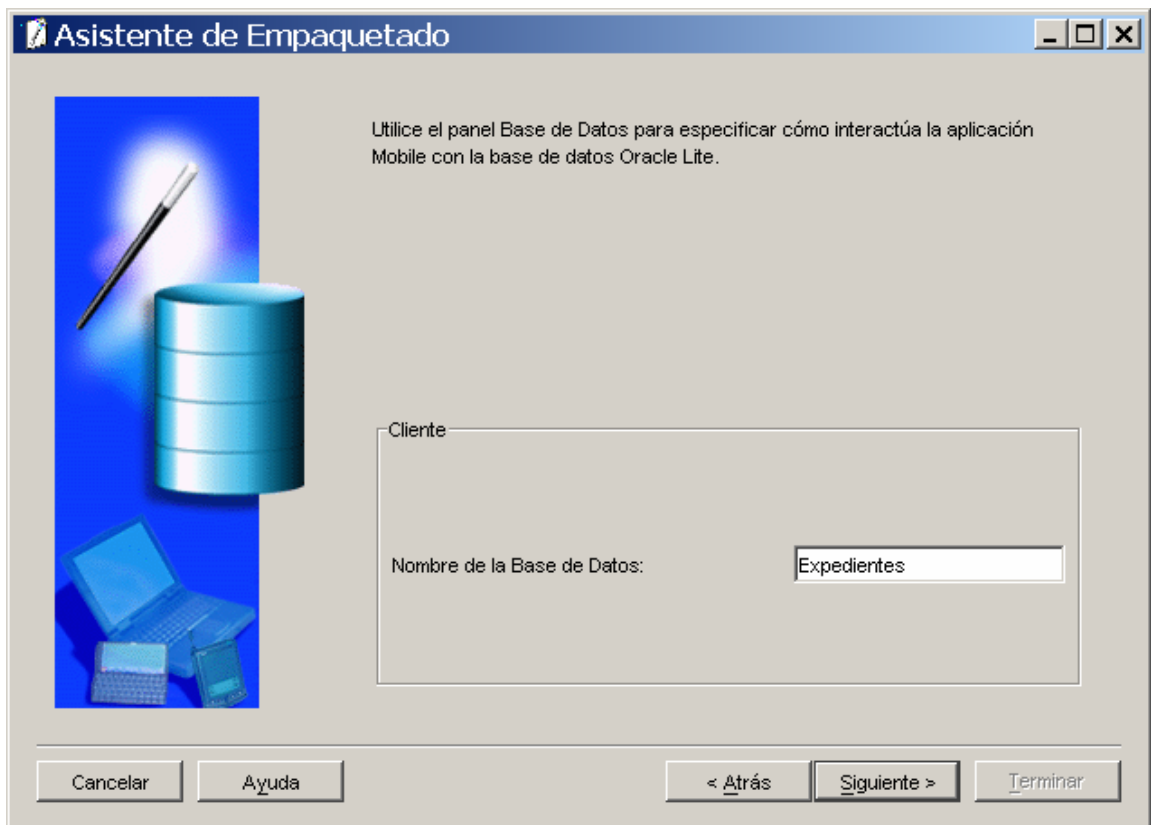


Figura 8. Panel Base de Datos

El siguiente paso es considerado el más importante y el que requiere más tiempo en la definición de una aplicación, pues es donde se especifica que subconjunto de tablas y campos son necesarios trasladar al dispositivo cliente. Aparece el panel de instantáneas (snapshots) con el siguiente aspecto:



Figura 9. Panel de instantáneas

En este punto se presiona el botón “Importar” el cual muestra una ventana en donde se debe ingresar los datos de conexión a la base de datos con la cual se quiere definir una aplicación cliente.

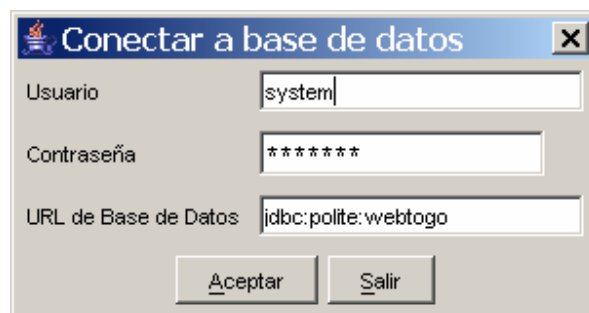


Figura 10. Conexión al servidor de datos

La URL utilizada normalmente tiene la forma:

jdbc:oracle:thin:@<Host>:<Puerto>:<Nombre del Servicio>

- Host: Nombre del equipo en donde se encuentra el servidor de datos Oracle.
- Puerto del servidor Oracle (por lo general 1521).
- Nombre del servicio de base de datos.

Al conectar correctamente con la base de datos aparece un listado con los esquemas disponibles y sus tablas, vistas o sinónimos correspondientes. En esta instancia se va seleccionando las tablas que se necesitan en la aplicación pulsando el botón “Agregar”. Al cerrar la ventana Tablas, estarán en el panel de instantáneas todos los objetos agregados anteriormente.

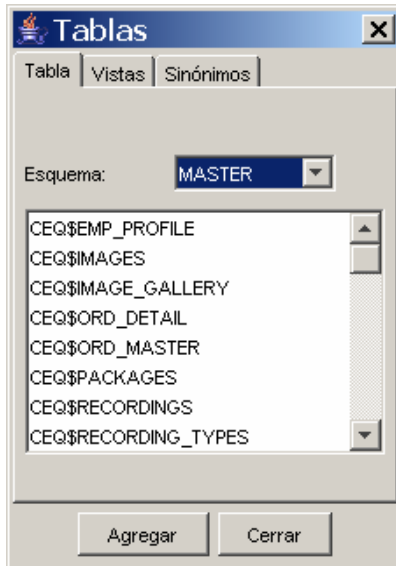


Figura 11. Selector de tablas para crear instantáneas

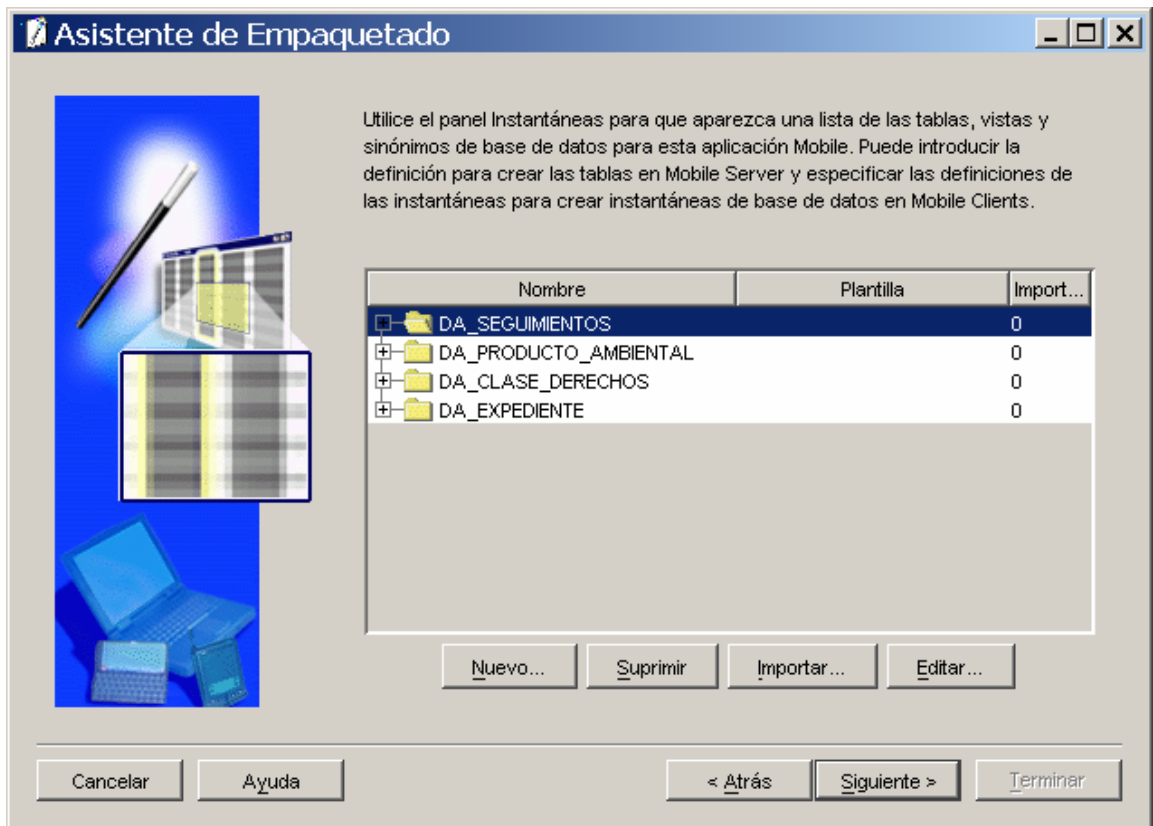


Figura 12. Panel de instantáneas

Al seleccionar un elemento de la lista se presiona el botón “Editar”, el cual despliega una ventana para definir y editar la configuración del snapshot elegido.

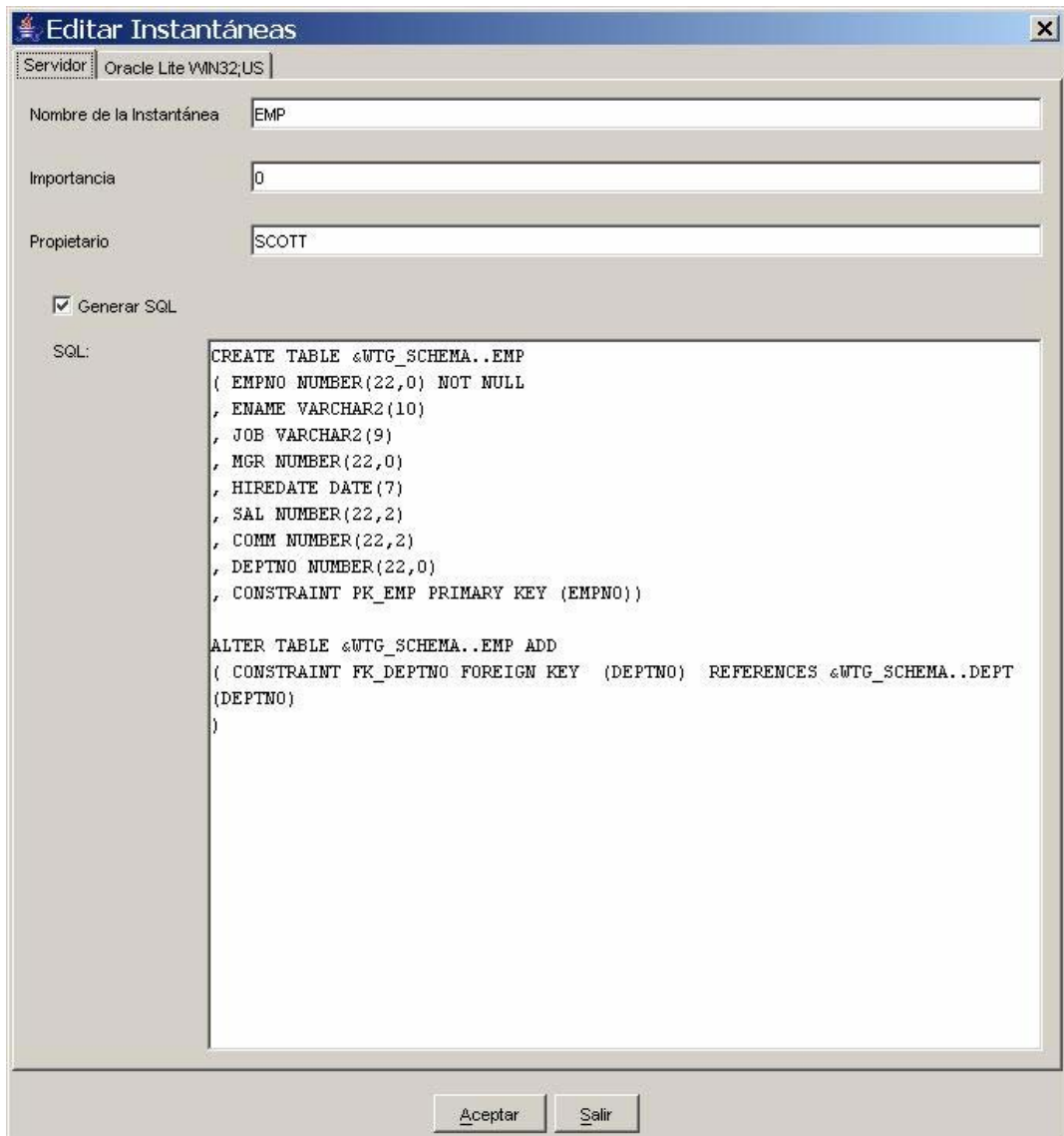


Figura 13. Editor de Snapshots (Servidor)

En la ficha “Servidor”, se encuentra la información de la tabla: Nombre, Importancia, Propietario y SQL. El campo “Importancia”, es un número entero que puede estar entre 1 y 1023 y es el más significativo pues define el orden en que deben ser actualizados los snapshots en los procesos de sincronización con el servidor. Se debe tener en cuenta que los objetos con números más bajos se sincronizan primero. Además, las instrucciones se ejecutan así:

1. Las operaciones INSERT del cliente son ejecutadas primero, desde el más bajo hasta el más alto grado de importancia de las tablas.
2. Las operaciones DELETE del cliente son ejecutadas primero, desde el más alto hasta el más bajo grado de importancia de las tablas.
3. Las operaciones UPDATE del cliente son ejecutadas primero, desde el más bajo hasta el más alto grado de importancia de las tablas.

Es muy importante fijar correctamente estos valores de importancia para evitar errores de sincronización. Los valores más bajos corresponderán a las tablas padre. Pueden existir varias tablas con el mismo número de importancia, es decir, se actualizan al mismo tiempo.

En la ficha del dispositivo (en este caso Oracle Lite PPC2003 ARMV4;US) se define la configuración para la creación y sincronización del objeto en el cliente.

Servidor: Oracle Lite WIN32,US

Crear en Cliente ¿Actualizable?

Tipo de Objeto Base: Tabla Vista

Resolución de Conflictos: Prioridad del Servidor Prioridad del Cliente Procedimiento DML:

Tipo de Refrescamiento: Refrescamiento Rápido Refrescamiento Completo

Indicación Principal: Nombre de la Tabla Principal:

Indicación Primaria Virtual: Nombre del Objeto Base: Nombre de Columna de Objeto Base:

Plantilla:

```
SELECT * FROM SCOTT.EMP
```

Indicación de Clave Primaria:

Nombre de la Tabla	Nombre de Columna de Tabla	Nombre de Columna de Vista

Índices

Nombre	Tipo	Columnas

Figura 14. Editor de Snapshots (Cliente)

Hay que asegurarse que el campo “Crear en Cliente” este seleccionado. El campo “Actualizable” indica si el snapshot tendrá sincronización bi-direccional. Si está desactivado, todos los cambios que se hagan en el cliente no serán tenidos en cuenta, es decir, solo se dispondrá de sincronización en un solo sentido (desde el servidor hacia el cliente). Luego se escoge el “Tipo de Objeto” (*Tabla* o *Vista*). En la “Resolución de conflictos” la opción *Prioridad del Servidor* aplica los datos

existentes en el servidor y deshace los del cliente, en caso de haber un conflicto. *Prioridad del Cliente* hará lo contrario. En el “Tipo de Refrescamiento” la opción *Refrescamiento Rápido* utilizará sincronización “Fast Refresh” basada en el proceso MGP, es decir, solo se subirán al cliente los registros que hayan sufrido cambios (inserción, actualización, eliminación) desde la última sincronización, y por tanto se encuentren en la “Cola de Salida” del Mobile Server. La opción *Refrescamiento Completo* actualiza todos los registros en el cliente y por tanto requiere de un tiempo mayor para su ejecución.

En el campo “Plantilla”, se escribe la instrucción SELECT que define los campos solicitados para manejar en el cliente. Es decir, se pueden cargar solo ciertos campos de la tabla y definir condiciones WHERE para limitarse a ciertos registros solamente. Cuando se tienen varios clientes de la aplicación, es común que cada uno necesite un conjunto diferente de registros dados por la condición WHERE. Para esto se crean los parámetros de suscripción, los cuales son variables en la condición WHERE que cambian según lo requiera el cliente. Por ejemplo:

```
SELECT ENAME, SAL, JOB FROM EMP WHERE DEPTNO = :Variable
```

En la instrucción anterior se puede observar que la condición está restringida por el campo DEPTNO según un valor definido como **:Variable**. Por tanto, cada usuario obtendrá un subconjunto de datos dado por el valor **:Variable**, el cual podrá ser fijado por el administrador del sistema mediante el administrador del Mobile Server (ver anexo 2), o si se tiene conocimiento, podrá ser manipulado accediendo directamente al esquema de datos creado por Oracle Lite en el servidor de datos (método seguido en el desarrollo de las opciones de este proyecto).

En la sección “Índices” se definen las columnas que serán indexadas al crear la base de datos en el cliente pudiendo ser estas de tipo *primario, regular o único*.

Luego, aparece el panel de secuencias. Oracle Lite define las secuencias como un mecanismo para controlar que en un entorno con varios usuarios móviles insertando datos en las mismas tablas, no se choquen o se generen conflictos a causa de que se usen los mismos valores en llaves primarias de las tablas. Por esto se definen rangos de valores para dichos campos, los cuales serán únicos, para poder ser asignados a cada usuario.

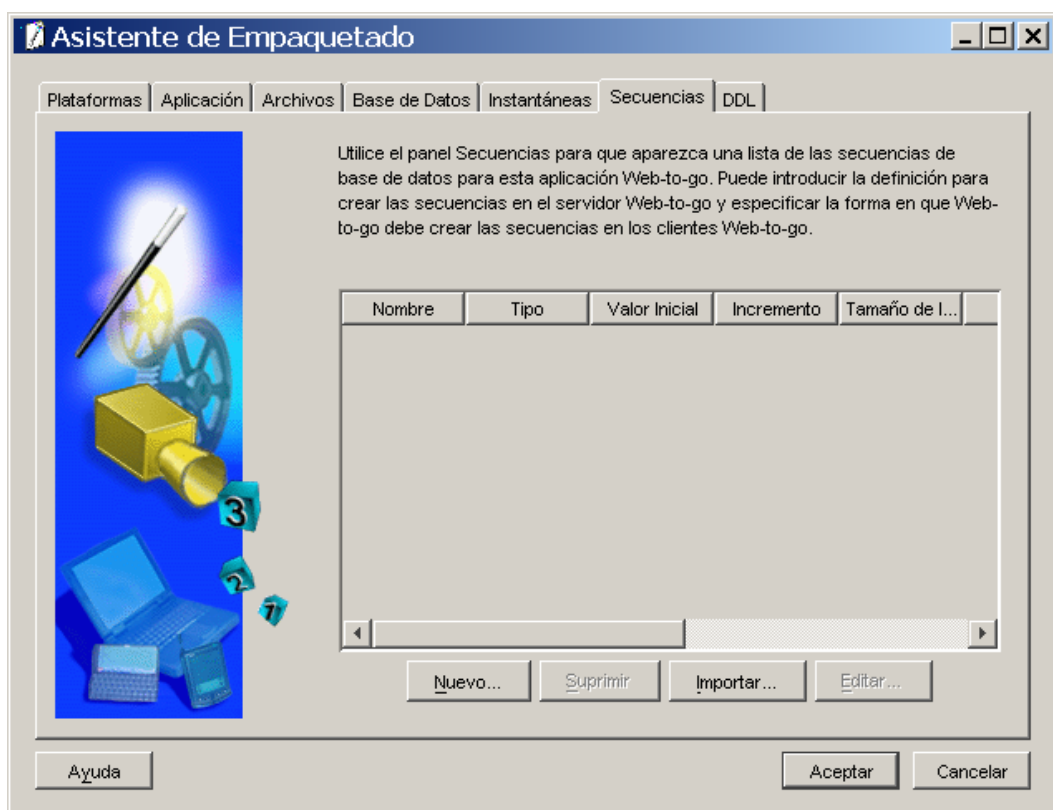


Figura 15. Panel de secuencias

Nueva Secuencia

Nombre:

Secuencia del servidor:

Generar SQL

Valor Inicial: Incremento:

Mínimo: Máximo:

Secuencia del cliente:

Crear en Cliente

Tipo:

Valor Inicial: Incremento:

Tamaño de la Ventana: Umbral:

Figura 16. Definición de nueva secuencia

Por último se llega al panel DDL, en el cual se pueden definir sentencias de lenguaje de definición de datos de SQL que el cliente ejecutará la primera vez que se sincronice con el servidor, es decir, cuando se crea la base de datos.

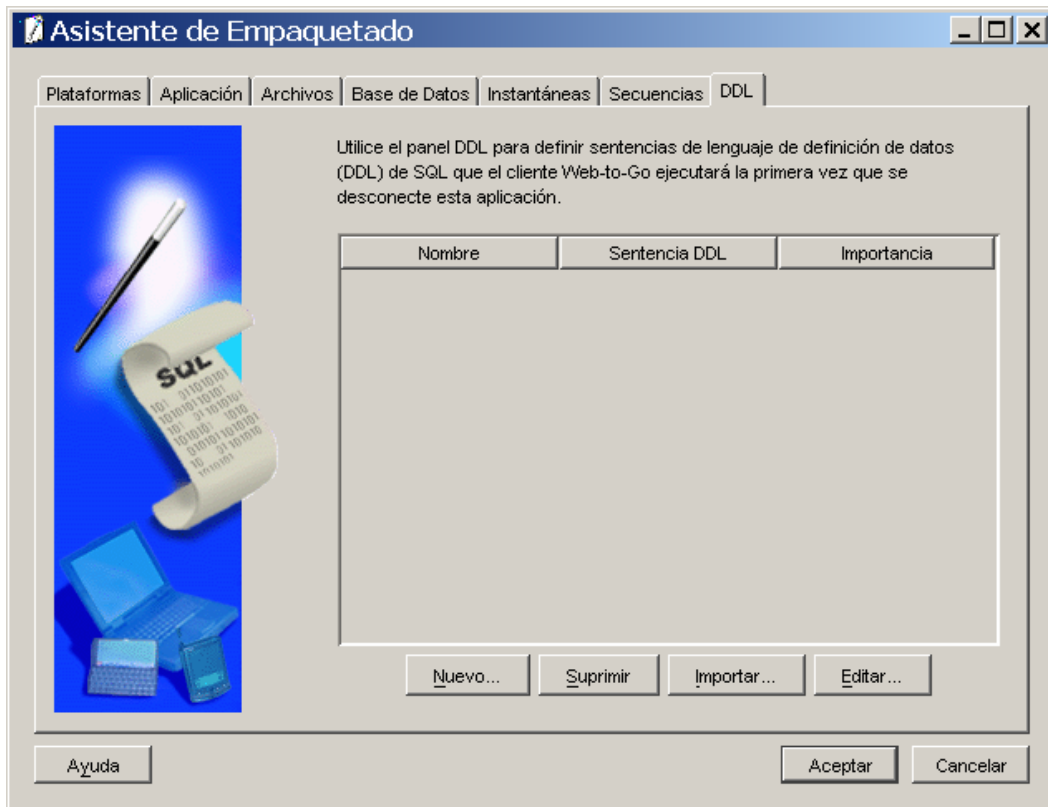


Figura 17. Panel de DDL

En este punto se oprime “Terminar” apareciendo así una ventana en donde se da la opción de publicar la aplicación o crear los archivos relacionados (por ejemplo empaquetarla en un archivo .JAR).

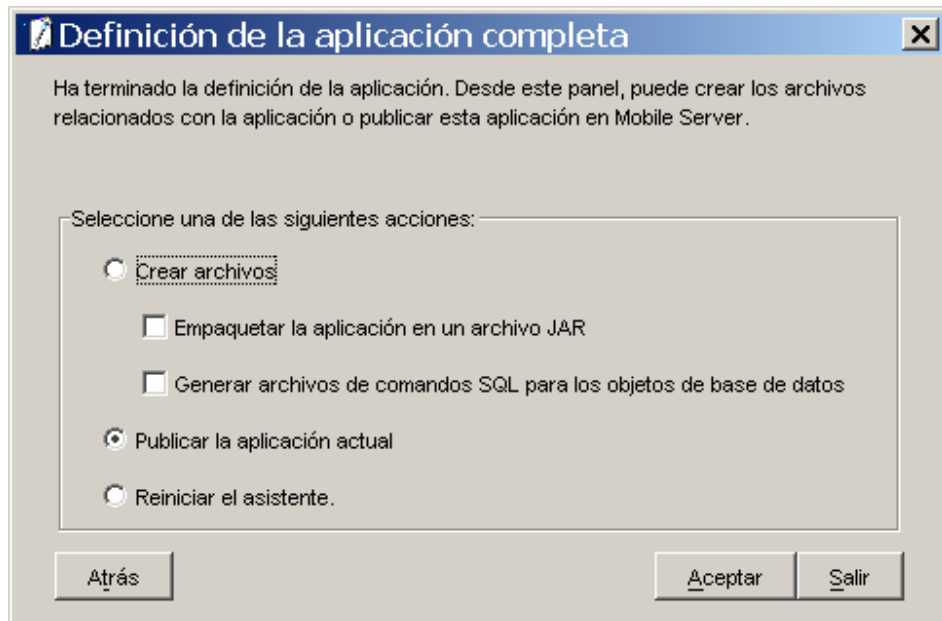


Figura 18. Finalización de la definición de una aplicación

Si se elige la opción de “Publicar la aplicación actual” aparecerá la siguiente interfaz en donde se debe ingresar la información de la URL del Mobile Server y el nombre de usuario y contraseña de un usuario con derechos de administrador.

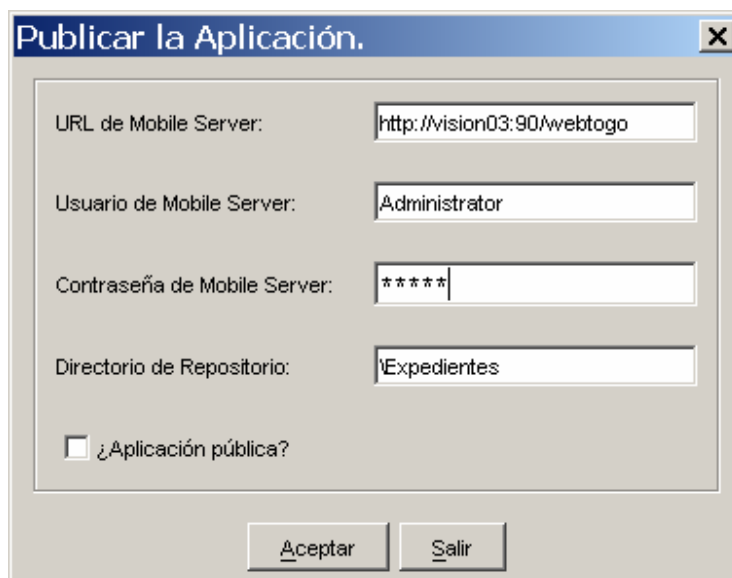


Figura 19. Publicación de la aplicación

ANEXO 2. MOBILE SERVER

Este anexo ha sido adaptado desde la documentación de Oracle Lite 10g (10.0.0) en la parte *Administration and Deployment Guide*.

Esquema del Mobile Server

Cuando la instalación del Mobile Server de Oracle Lite se ha realizado correctamente, se crea un esquema con tablas y vistas en las cuales se guardan y consultan todos los datos referentes a las aplicaciones, usuarios, dispositivos registrados, datos de sincronización, fallos de sincronización, estadísticas de sincronización etc. No es aconsejable manipular los datos de las tablas de este esquema si no se tiene el conocimiento suficiente sobre el funcionamiento interno de Oracle Lite. Las vistas son de solo lectura y por tanto pueden ser consultadas sin ningún problema para observar ciertos datos registrados en el repositorio móvil. A continuación se hace una breve descripción de algunas de las vistas que ofrecen información de interés para el administrador del sistema.

MOBILEADMIN".CV\$ALL_CLIENTS: Esta vista provee información sobre los clientes del Mobile Server. Su estructura está dada en la siguiente tabla:

Columna	Tipo	Nulo?	Descripción
CLIENT	VARCHAR(30)	SI	Cliente del Mobile Server
LASTREFRESH_STARTTIME	VARCHAR(19)	SI	Tiempo de inicio de la última sesión de sincronización.
LASTREFRESH_ENDTIME	VARCHAR(19)	SI	Tiempo de conclusión de la última sesión de

Columna	Tipo	Nulo?	Descripción
			sincronización.

Tabla 8. Estructura de la vista de clientes

MOBILEADMIN.CV\$ALL_ERROR: Esta vista provee información sobre las transacciones fallidas del cliente en las sincronizaciones realizadas. Su estructura está dada en la siguiente tabla:

Columna	Tipo	Nulo?	Descripción
CLIENT	VARCHAR(30)	NO	Cliente al que pertenece la transacción fallida.
TRANSACTION_ID	NUMBER(10)	NO	Identificador de la transacción fallida.
ITEM_NAME	VARCHAR2(30)	NO	Nombre del ítem de publicación de la falla.
MESSAGE_TEXT	VARCHAR2(2048)	NO	Texto de error asociado con la transacción fallida y el ítem de publicación.

Tabla 9. Estructura de la vista errores de sincronización

MOBILEADMIN.CV\$ALL_PUBLICATIONS: Provee información sobre las publicaciones del Mobile Server.

Columna	Tipo	Nulo?	Descripción
NAME	VARCHAR2(30)	SI	Nombre de la publicación.
TYPE	VARCHAR2(40)	SI	Tipo de publicación.
NAME_TEMPLATE	VARCHAR2(30)	SI	Nombre de la plantilla del Snapshot.
ENFORCE_RI	CHAR(1)	NO	Reservado.

Tabla 10. Estructura de la vista de publicaciones

MOBILEADMIN.CV\$ALL_SUBSCRIPTIONS: Provee información sobre las suscripciones al Mobile Server.

Columna	Tipo	Nulo?	Descripción
CLIENT	VARCHAR2(30)	SI	El cliente de suscripción.
PUBLICATION	VARCHAR2(30)	SI	La publicación de suscripción.
INSTANTIATED	CHAR(1)	SI	Un valor boolean que indica si la suscripción está instanciada.

Tabla 11. Estructura de la vista de suscripciones

MOBILEADMIN.CV\$ALL_SEQUENCES: Provee información sobre las secuencias definidas en el Mobile Server.

Columna	Tipo	Nulo?	Descripción
NAME	VARCHAR2(30)	SI	Nombre de la secuencia.

Tabla 12. Estructura de la vista de secuencias

MOBILEADMIN.CV\$ALL_SEQUENCE_PARTITIONS: Provee información sobre las particiones de secuencias del Mobile Server.

Columna	Tipo	Nulo?	Descripción
CLIENT	VARCHAR2(30)	SI	El cliente al que la secuencia es asignada.
NAME	VARCHAR2(30)	SI	Nombre de la secuencia
CURR_VALUE	NUMBER(38)	SI	Actual valor de la secuencia.

Columna	Tipo	Nulo?	Descripción
INCREMENT_BY	NUMBER(38)	SI	Valor de incremento de la secuencia. La secuencia se incrementa basada en este número.

Tabla 13. Estructura de la vista particiones de secuencias

MOBILEADMIN.CV\$ALL_PUBLICATION_ITEMS_ADDED: Esta vista provee información sobre los ítems de publicación existentes en el Mobile Server en relación con las publicaciones.

Columna	Tipo	Nulo?	Descripción
PUB_NAME	VARCHAR2(30)	SI	El Nombre de la publicación.
ITEM_NAME	VARCHAR2(30)	SI	El nombre del ítem de publicación
OWNER	VARCHAR2(30)	NO	Propietario del objeto base.
OBJECT_NAME	VARCHAR2(30)	NO	Nombre del objeto base.
TEXT	VARCHAR2(2048)	NO	Instrucción SELECT.
UPDATABLE	VARCHAR2(1)	SI	Opción de actualización.
REFRESH_METHOD	CHAR(1)	NO	Método de refrescamiento. Las opciones incluyen <i>fast refresh</i> y <i>complete refresh</i> .
WINNING_RULE	VARCHAR2(30)	SI	Regla para resolver conflictos de sincronización. Las opciones incluyen <i>client wins</i> y <i>server wins</i> .

Tabla 14. Estructura de la vista ítems adicionales

MOBILEADMIN.CV\$ALL_PUBLICATION_ITEMS: Esta vista provee información sobre los ítems de publicación del Mobile Server.

Columna	Tipo	Nulo?	Descripción
NAME	VARCHAR2(30)	SI	Nombre del ítem de publicación.
OWNER	VARCHAR2(30)	NO	Propietario del objeto base del ítem de publicación.
OBJECT_NAME	VARCHAR2(30)	NO	Nombre del objeto base.
TEXT	VARCHAR2(2048)	NO	Instrucción SQL que define al ítem de publicación
REFRESH_METHOD	CHAR(1)	NO	Método de refrescamiento. Las opciones incluyen <i>fast refresh</i> y <i>complete refresh</i> .

Tabla 15. Estructura de la vista de ítems de publicación

MOBILEADMIN.CV\$ALL_PUBLICATION_ITEM_INDEXES: Esta vista provee información sobre los índices definidos en los ítems de publicación del Mobile Server.

Columna	Tipo	Nulo?	Descripción
NAME	VARCHAR2(30)	SI	Nombre del índice.
PUB_ITEM	VARCHAR2(30)	NO	Nombre del ítem de publicación.
INDX_TYPE	CHAR(1)	NO	Tipo de índice.
COLUMN_LIST	VARCHAR2(2048)	NO	Lista de columnas.

Tabla 16. Estructura de la vista de índices

MOBILEADMIN.CV\$ALL_SUBSCRIPTION_PARAMS: Esta vista provee información sobre los parámetros de suscripción definidos en el Mobile Server.

Columna	Tipo	Nulo?	Descripción
NAME	VARCHAR2(30)	SI	Nombre de la publicación

Columna	Tipo	Nulo?	Descripción
CLIENT	VARCHAR2(30)	SI	Nombre del cliente.
PARAM_NAME	VARCHAR2(30)	SI	Nombre del parámetro.
PARAM_VALUE	VARCHAR2(30)	SI	Valor del parámetro.

Tabla 17. Estructura de parámetros de suscripción

Manejo de Aplicaciones y Usuarios en el Mobile Server

El Mobile Server de Oracle Lite, se puede iniciar de la siguiente forma:

- En entorno Windows ir a: Inicio->Todos los programas->Oracle Database Lite 10g->Mobile Server
- O desde DOS: `cd Oracle_Home/Mobile_Server/bin/runmobileserver` (en modo OC4J solamente)

El Mobile Server cuenta con una interfaz para su administración y configuración en entorno web, la cual es activada desde un navegador de la siguiente manera:

Acceder a la URL: `http://<Nombre_máquina>[:Puerto]/webtogo`

- Nombre_máquina: es el nombre del computador en el cual se instaló el Mobile Server.
- Puerto: Durante la instalación se define este puerto para el funcionamiento del OC4J.

Aparece en el navegador la siguiente interfaz:

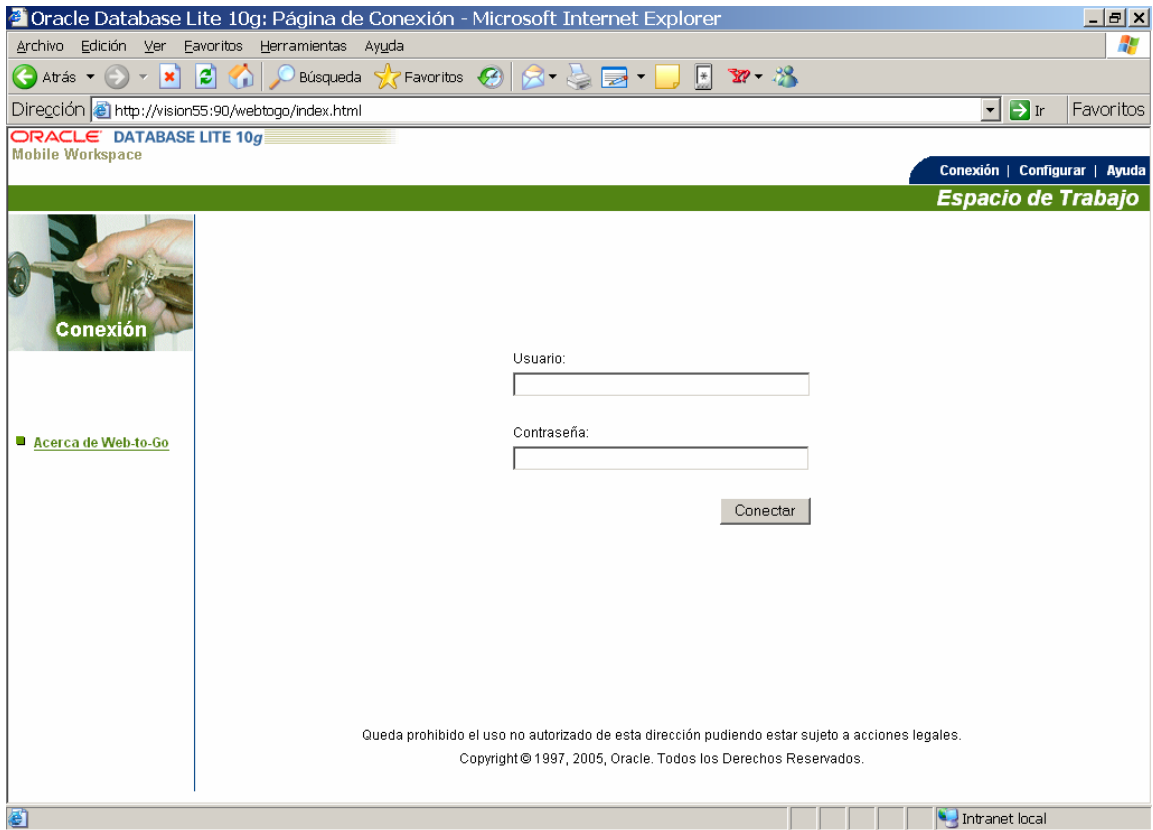


Figura 20. Ingreso a Mobile Server

Por defecto Oracle Lite instala el usuario “Administrator” con contraseña “Admin”. Al introducir estos valores y presionar el botón “Conectar”, se ingresa al espacio de trabajo “Aplicaciones” en donde se tiene acceso al “Mobile Manager” o administrador móvil, como se muestra a continuación.

Oracle Database Lite 10g Mobile Manager - Mobile Servers - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Atrás Búsqueda Favoritos

ORACLE Database Lite 10g
Mobile Manager

Mobile Servers | Dispositivos Móviles

Mobile Servers

Página Refrescada 12/01/2006 02:59:16

Buscar Ir

Nombre del Host	Puerto	SSL	Estado	MGP	Hora de Inicio	Versión
vision55	90		✓	✓	12/01/2006 02:46:02 PM	10.2.0.1.0

[Ayuda](#)

Copyright © 1997, 2005, Oracle. Todos los Derechos Reservados.

Figura 21. Sección de Mobile Servers del Mobile Manager

En este espacio se puede observar y tener acceso a una lista de todos los equipos en donde haya sido instalado y configurado un Mobile Server sobre el mismo repositorio en el servidor principal de base de datos. Se muestra el nombre del Equipo donde está instalado cada Mobile Server, el puerto, estado (activado o desactivado), estado del MGP (activado o desactivado) entre otros. Además en la parte superior se tiene acceso a la opción “Dispositivos Móviles” la cual muestra el listado de dispositivos registrados en el repositorio, las plataformas activas y ciertas opciones de Administración de dispositivos.

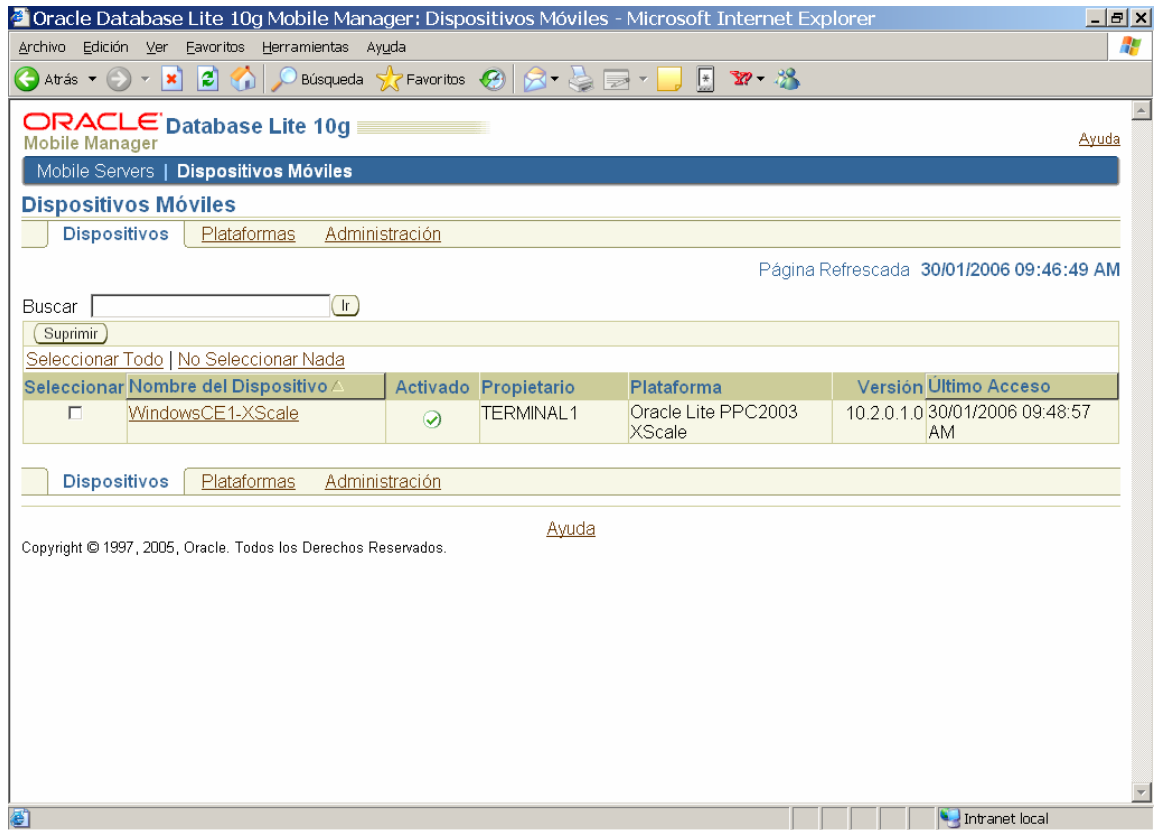


Figura 22. Sección Dispositivos Móviles del Mobile Manager

Desde el Mobile Manager al dar click sobre el nombre del host de alguno de los Mobile Server activos, se ingresará a la página principal de aplicaciones, usuarios y opciones de administración de dicho Mobile Server. Esta es la página más importante pues desde esta se publica, distribuyen y administran las diferentes aplicaciones desarrolladas para los usuarios, y además, se dispone de varias herramientas para configurar y hacer seguimiento a la sincronización de los diferentes usuarios con el servidor central.

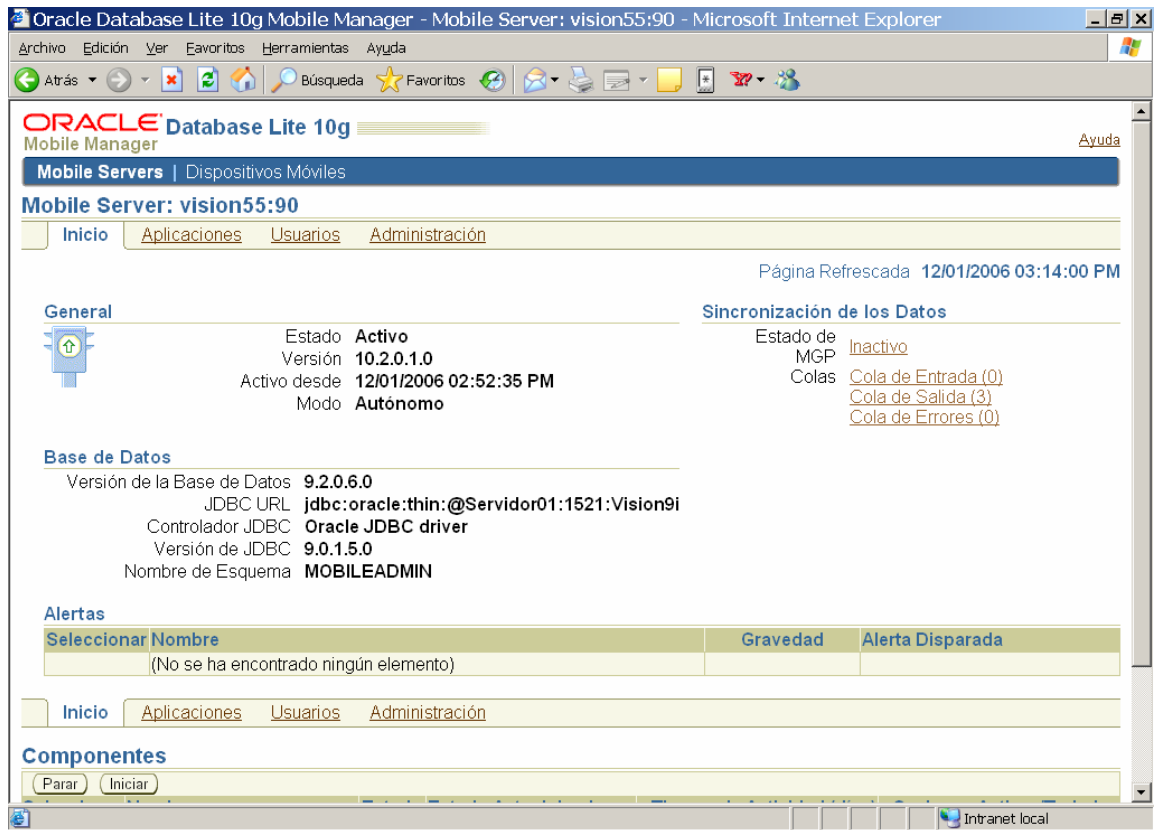


Figura 23. Interfaz de administración de Mobile Server

Después de haberse definido y publicado alguna aplicación mediante la herramienta *Packagin Wizard* o en español Asistente de Empaquetado del MDK (para mayor detalle ver el Anexo 1), se puede ingresar a la ficha “Aplicaciones”, en donde se observa en el listado la aplicación creada anteriormente. Al dar click sobre el nombre, se muestra en pantalla la información perteneciente a la aplicación tal como: estado (en ejecución o suspendida), fecha de publicación, plataforma, descripción entre otros.

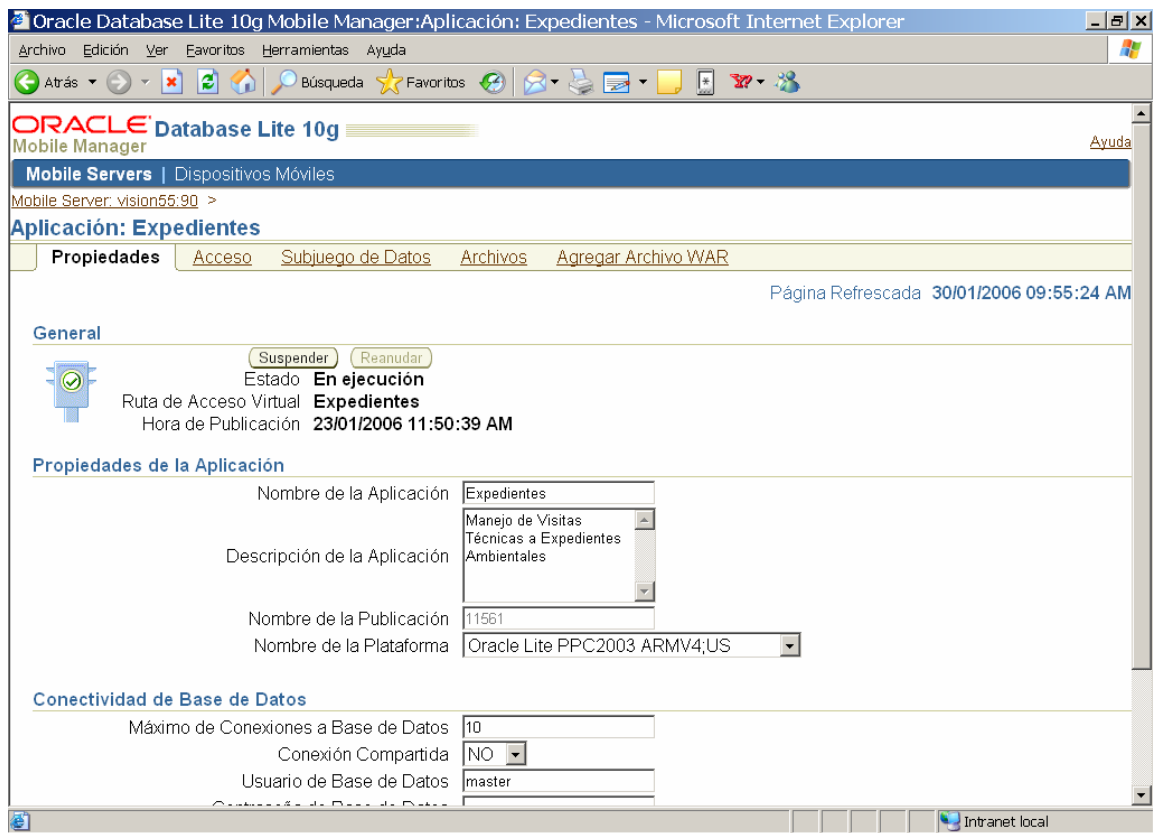


Figura 24. Página de propiedades de una aplicación

Para crear un usuario se debe ingresar a la ficha “Usuarios” del Mobile Server. En esta se puede observar un listado de todos los usuarios registrados. Para agregar uno nuevo se presiona el botón “Agregar Usuario”.

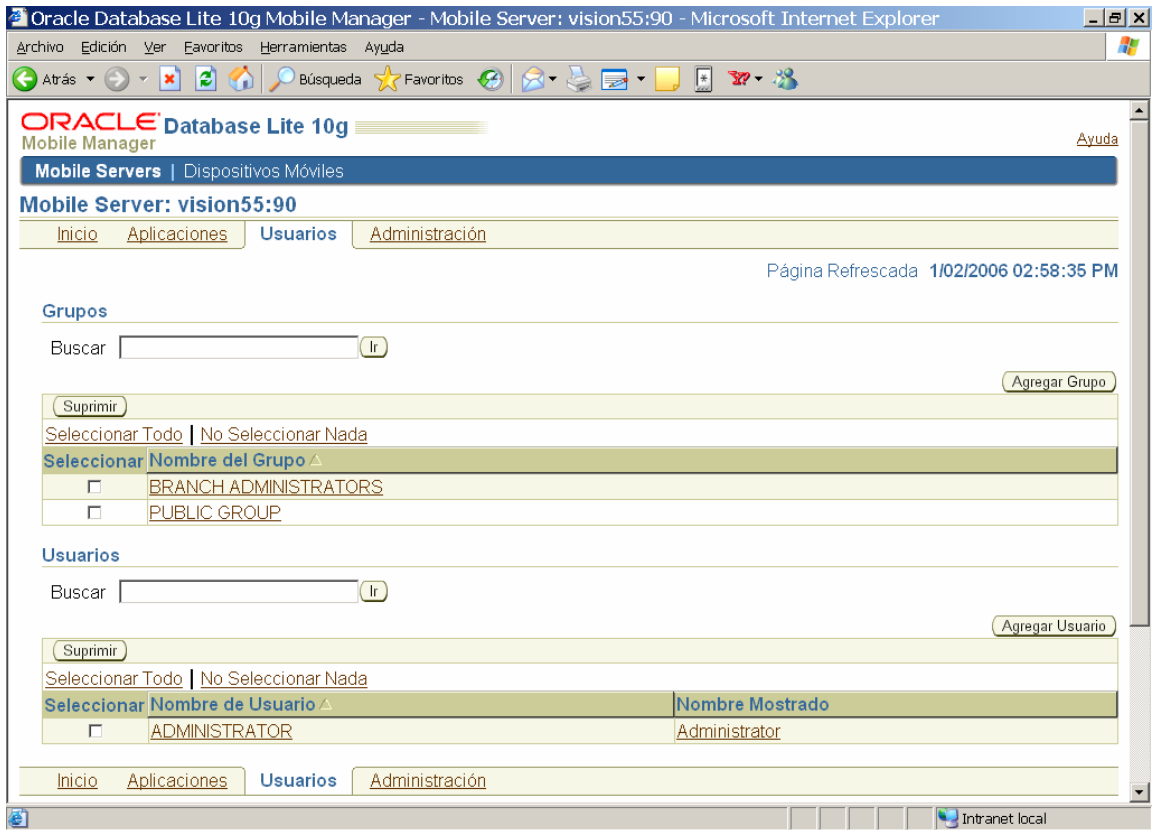


Figura 25. Página de usuarios de Oracle Lite

En este punto se ingresa la información referente al nuevo cliente (Nombre de usuario, contraseña, y privilegio = USER) y se presiona el botón Aceptar.

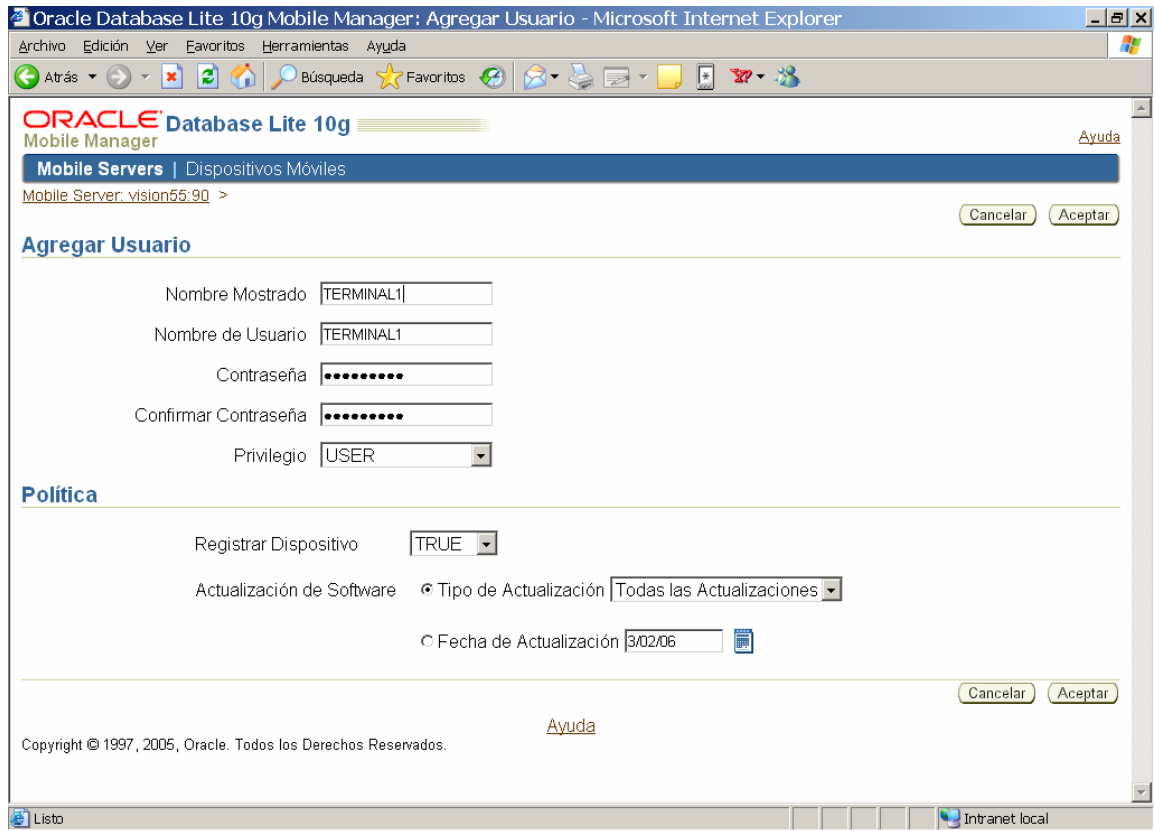


Figura 26. Creación de un nuevo usuario

Se regresa a la pantalla principal de Usuario en donde aparece en el listado el nuevo usuario creado.

Para suscribir a un usuario a cierta aplicación, se da click sobre el nombre del usuario, lo cual lleva a la pantalla de propiedades del usuario escogido. En esta interfaz se da click sobre la ficha "Acceso", donde aparece el listado de todas las aplicaciones disponibles en el Mobile Server. Se selecciona la aplicación a la cual se quiera suscribir al usuario y se presiona el botón "Guardar".

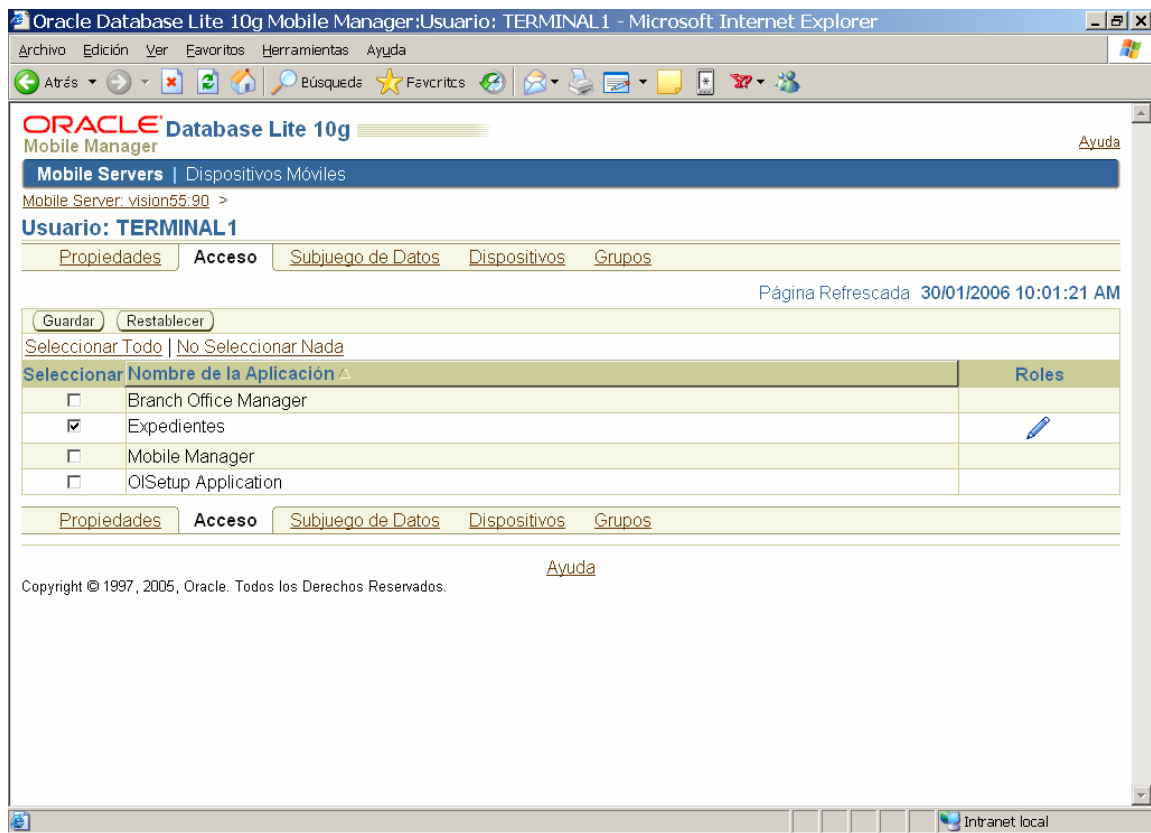


Figura 27. Suscripción de un usuario a una aplicación

Después de seguir este proceso, el usuario es un cliente de la aplicación y por consiguiente pueda cargar el subconjunto de tablas y datos que se hayan definido, y realizar sincronizaciones con el servidor cuando sea necesario.

Si durante la definición de la aplicación se fijaron parámetros de suscripción para restringir el conjunto de datos que cargará cada usuario, al ingresar a la ficha "Subjuego de datos", se podrán fijar dichos valores para el usuario seleccionado.

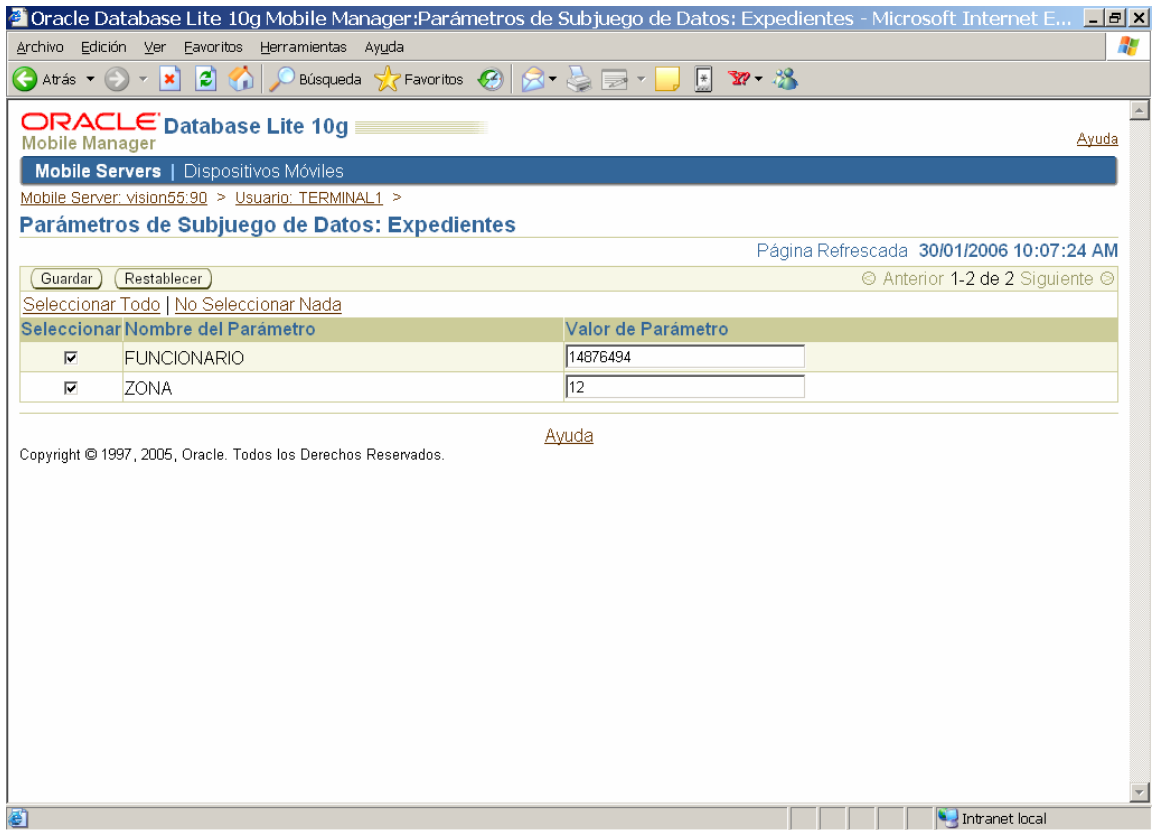


Figura 28. Valores en los parámetros de suscripción

Además, Oracle Lite ofrece una serie de herramientas para la administración y seguimiento de los procesos de sincronización, a los cuales se puede tener acceso desde la ventana principal dando click sobre la ficha "Administración".

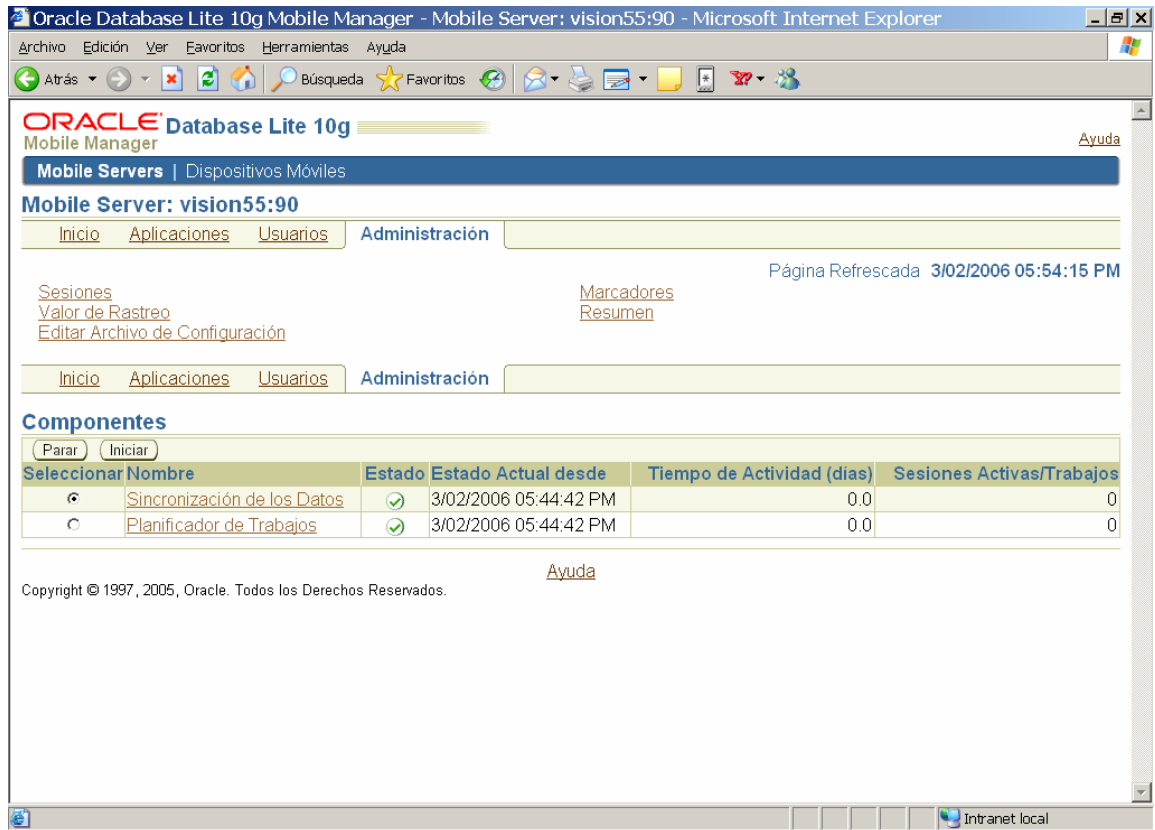


Figura 29. Página de administración de Oracle Lite

En la parte superior se observa vínculos a opciones de configuración propias del Mobile Server, tales como, sesiones, edición del archivo de configuración y marcadores. En la sección “Componentes” se puede acceder a la administración de todo lo relacionado con la sincronización de datos. El vínculo “Sincronización de los Datos” visualiza una ventana en donde se presenta toda la información configurada referente a la sincronización.

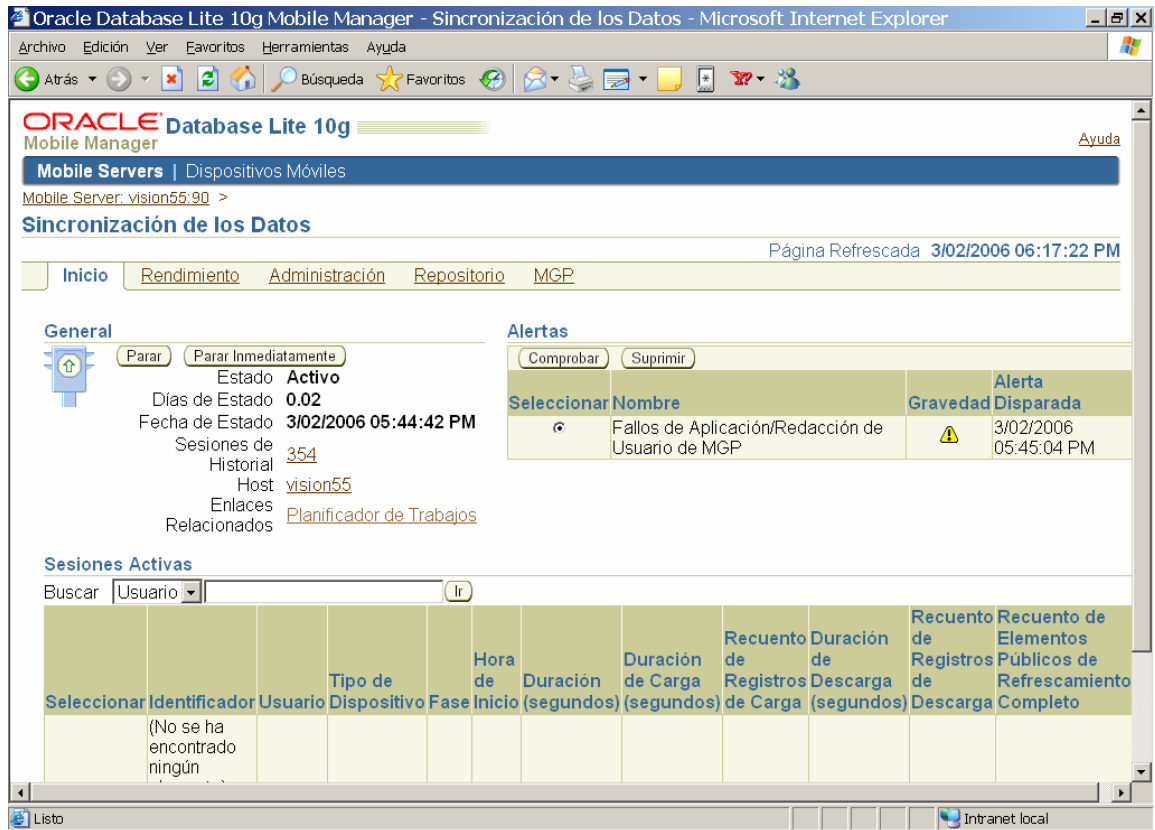


Figura 30. Página de sincronización de los datos

Desde la ficha “Administración” en el vínculo “Planificador de Trabajos”, se pueden programar diversas tareas para que se ejecuten cada cierto periodo de tiempo según se establezca. Por ejemplo, en esta sección se encuentra la configuración del ciclo de aplicación y redacción de datos del MGP. Este trabajo se ejecuta por defecto cada 60 segundos, pero puede ser personalizado según las necesidades, o pueden crearse nuevos trabajos en donde solo se ejecute la fase de aplicación por ejemplo cada 20 segundos.

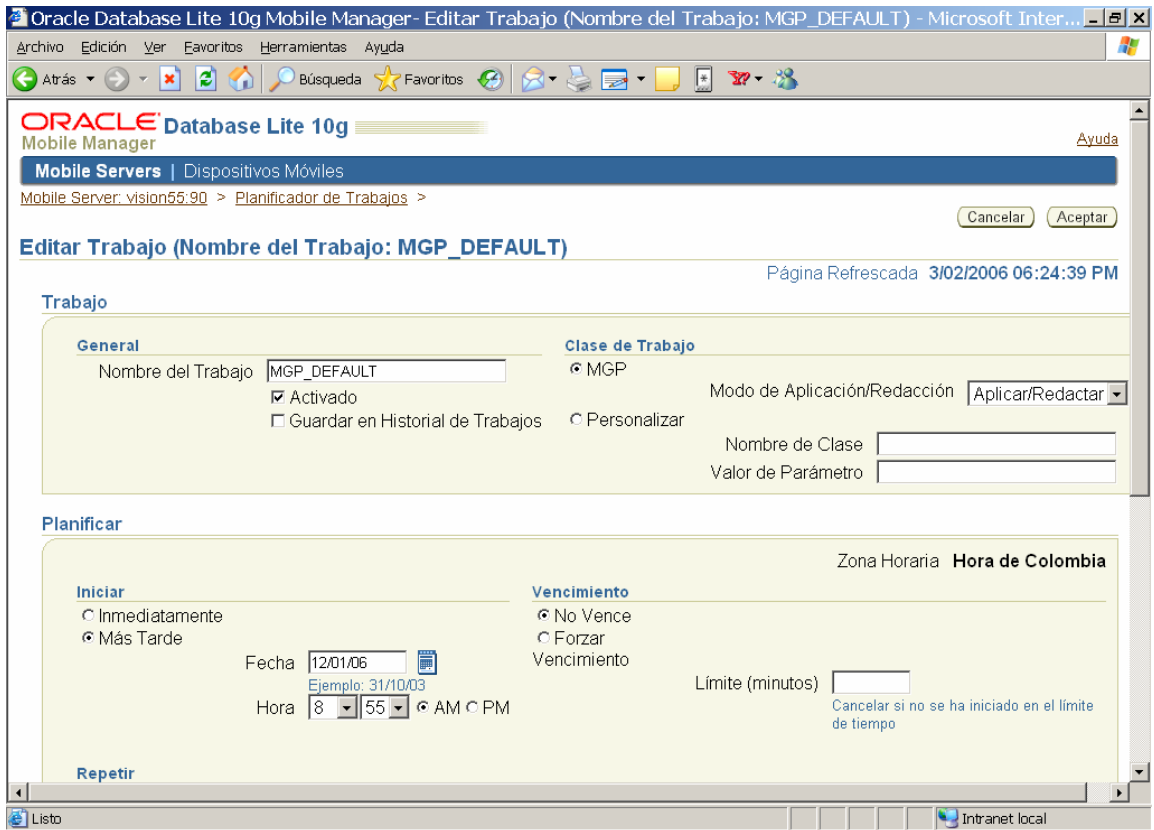


Figura 31. Planificador de MGP

Por último, desde la ventana de “Sincronización de los datos” se pueden consultar los resultados obtenidos del proceso MGP. Por ejemplo, al seleccionar la opción “Ciclos de Aplicación/Redacción de MGP” se puede observar algunas estadísticas de los procesos de sincronización realizados entre los clientes y el servidor.

Oracle Database Lite 10g Mobile Manager - Ciclos de Aplicación/Redacción de MGP - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

ORACLE Database Lite 10g Mobile Manager

Mobile Servers | Dispositivos Móviles

Mobile Server: vision55.90 > Sincronización de los Datos >

Ciclos de Aplicación/Redacción de MGP

Página Refrescada 3/02/2006 06:28:20 PM

Buscar

Tipo: Todo De: A

Resultado: Todo Fecha: 27/01/06 Hora: 6:25 AM Zona Horaria: Hora de Colombia

Fecha: 10/02/06 Hora: 6:25 AM

Buscar Buscar y Suprimir

Resultados

Detalles Suprimir Anterior 1-25 de 97 Siguientes 25

Seleccionar	Identificador	Tipo	Resultado	Hora de Finalización	Duración (segundos)	Recuento de Registros de Aplicación	Duración de Aplicación (segundos)	Recuento de Registros de Log de Proceso	Duración de Log de Proceso (segundos)	Recuento de Registros de Redacción	Duración de Redacción (segundos)
<input checked="" type="radio"/>	8807	APPLY_COMPOSE	×	2006-02-03 18:27:57.0	0	0	0	0	.031	0	0
<input type="radio"/>	8806	APPLY_COMPOSE	×	2006-02-03 18:27:27.0	0	0	0	0	.032	0	0
<input type="radio"/>	8805	APPLY_COMPOSE	×	2006-02-03 18:26:57.0	0	0	0	0	.047	0	0
<input type="radio"/>	8804	APPLY_COMPOSE	×	2006-02-03 18:26:27.0	0	0	0	0	.031	0	0
<input type="radio"/>	8803	APPLY_COMPOSE	×	2006-02-03 18:25:57.0	0	0	.016	0	.031	0	0
<input type="radio"/>	8802	APPLY_COMPOSE	×	2006-02-03 18:25:27.0	0	0	.015	0	.032	0	0
<input type="radio"/>	8801	APPLY_COMPOSE	×	2006-02-03 18:24:57.0	0	0	0	0	.046	0	0

Intranet local

Figura 32. Ciclos de MGP