

**SISTEMA INTEGRADO DE RECOLECCION Y PROCESAMIENTO DE DATOS A  
TRAVES DE DISPOSITIVOS MOVILES BAJO LA PLATAFORMA PALM**

Francisco Orlando Martínez Pabón

Franco Arturo Urbano Ordoñez

**Universidad del Cauca  
Facultad de Ingeniería Electrónica y Telecomunicaciones  
Departamento de Telemática  
Popayán  
2003**

**SISTEMA INTEGRADO DE RECOLECCION Y PROCESAMIENTO DE DATOS A  
TRAVES DE DISPOSITIVOS MOVILES BAJO LA PLATAFORMA PALM**

FRANCISCO ORLANDO MARTÍNEZ PABÓN

FRANCO ARTURO URBANO ORDOÑEZ

Monografía de Grado

Director: Ing. Juan Carlos Corrales Muñoz

**Universidad del Cauca  
Facultad de Ingeniería Electrónica y Telecomunicaciones  
Departamento de Telemática  
Popayán  
2003**

A Dios.  
A mi Madre y a mi Abuela, por guiarme con su sabiduría.  
A mi Hermana y a mis Tíos, por estar siempre conmigo.  
A mi Novia, por su gran amor y apoyo incondicional.  
A mi Padre, por tener siempre para mí una luz desde el cielo.

*Francisco Orlando Martínez Pabón*

A mi Mamá  
por apoyarme incansablemente  
durante mis estudios,  
por creer en mí para alcanzar  
este título profesional.  
A mis Hermanas  
por su comprensión y cariño.

*Franco Arturo Urbano Ordoñez*

## **AGRADECIMIENTOS**

Queremos agradecer al Ing. Diego Andrés Acosta por su valiosa colaboración en el inicio de este proyecto. Igualmente al Ing. Juan Carlos Corrales, director del proyecto, por su apoyo y gran colaboración durante el desarrollo de este trabajo. Finalmente, queremos agradecer al Ing. Juan Carlos Diago del Instituto de Investigación e Información Geocientífica, Minero-Ambiental y Nuclear (INGEOMINAS) y a los representantes de cada una de las áreas, por su disposición y valiosos aportes para el desarrollo del proyecto.

## CONTENIDO

INTRODUCCION .....	1
1 TECNOLOGÍAS MOVILES .....	3
1.1 LA PLATAFORMA PALM .....	3
1.1.1 Introducción.....	3
1.1.2 Características generales .....	4
1.1.3 Sistema operativo Palm.....	5
1.1.4 Gestión de la memoria del dispositivo .....	6
1.1.5 Arquitectura de la memoria.....	6
1.1.6 Data Manager.....	7
1.1.7 Tipos de archivos .....	7
1.1.8 Campos de la cabecera de base de datos.....	8
1.1.9 Filosofía Palm.....	9
1.2 Java 2 Micro Edition .....	14
1.2.1 Arquitectura J2ME .....	15
1.2.2 KVM (Kilobyte Virtual Machine) .....	16
1.2.3 Configuraciones .....	16
1.2.4 Perfiles .....	17
1.2.5 Librerías de clases MIDP.....	18
1.2.6 Ciclo de vida de una aplicación J2ME .....	19
2 SINCRONIZACION .....	20
2.1 Introducción.....	20
2.2 Estructura de un sistema de sincronización.....	21
2.3 La tecnología de sincronización HotSync .....	22
2.3.1 Componentes de una sincronización HotSync.....	23
2.3.2 Conductos .....	24
2.3.3 Flujo de sincronización HotSync.....	25
2.3.4 JSync Suite .....	26
2.3.5 Arquitectura de un conducto basado en Java .....	26
2.4 SyncML (Synchronization Markup Language) .....	27
2.4.1 Ventajas de SyncML.....	27
2.4.2 Características generales de SyncML .....	28
2.4.3 Estructura de SyncML .....	30
2.4.4 Tipos de sincronización .....	33
2.4.5 Arquitectura genérica de Sincronización.....	33
3. TECNOLOGIAS INVOLUCRADAS EN EL DESARROLLO WEB .....	40
3.1 Generalidades.....	40
3.2 Modelo Cliente/Servidor .....	41
3.2.1 Arquitectura en Tres Niveles .....	41
3.2.2 Beneficio de la Arquitectura en Tres Niveles .....	42
3.3 Servlets Java.....	43
3.3.1 Que es un Servlet?.....	43
3.3.2 Características de los Servlets .....	44
3.3.3 Arquitectura del Paquete Servlet .....	45

3.3.4 Ciclo de Vida de un Servlet .....	46
3.3.5 Servlets y JDBC .....	48
3.4 JSP (Java Server Pages) .....	49
3.4.1 Qué es JSP? .....	49
3.4.2 Comparación JSP con ASP .....	51
3.4.3 JSP o Servlets .....	52
3.4.4 Arquitectura JSP .....	52
3.5 Sistemas de Gestión de Bases de Datos (DBMS) .....	54
3.5.1 Plataformas y procesadores .....	55
3.5.2 MySQL .....	56
4. SISTEMA GEOS - ESPECIFICACIÓN DE REQUISITOS .....	58
4.1 Esencia del Sistema .....	58
4.2 Descripción General del Sistema .....	59
4.3 Glosario .....	59
4.4 Modelo de la Organización .....	60
4.5 Arbol de funciones .....	65
4.6 Casos de Uso de Alto Nivel .....	67
4.6.1 Diagrama de Casos de Uso de Alto Nivel .....	67
4.6.2 Descripción de los Casos de Uso de Alto Nivel .....	67
5. SISTEMA GEOS - ANÁLISIS DEL SOFTWARE .....	72
5.1 Casos de Uso Extendidos .....	72
5.2 Diagrama de clases del sistema .....	108
5.3 División en Clases Específicas .....	110
5.4 Diagrama de Implantación .....	114
CONCLUSIONES Y RECOMENDACIONES .....	115
BIBLIOGRAFÍA .....	116
ANEXOS .....	117
CONTENIDO CD ROM .....	118

## LISTA DE TABLAS

Tabla 1. Necesidad del usuario vs. Características del dispositivo.....	11
Tabla 2. Comparación JSP con ASP.....	51
Tabla 3. Relación sitio general-sitio específico-área.....	60
Tabla 4. Tipos de pregunta que conforman un Formato Simple .....	72
Tabla 5. Tipos de pregunta que conforman un Formato Tipo Encuesta .....	73
Tabla 6. Casos de Uso Principales .....	73
Tabla 7. Relación área-sitio general-visita.....	88

## LISTA DE FIGURAS

Figura 1. Palm m125.....	4
Figura 2. Arquitectura del sistema operativo Palm .....	5
Figura 3. Estructura básica de una base de datos .....	8
Figura 4. Estructura de la cabecera .....	8
Figura 5. Características vs. Experiencia del usuario .....	10
Figura 6. DateBook .....	12
Figura 7. Menú para opciones avanzadas.....	13
Figura 8. Ediciones Java con sus respectivos mercados .....	14
Figura 9. Arquitectura J2ME.....	15
Figura 10. Relación entre Configuración y Perfil .....	15
Figura 11. Compatibilidad entre CDC y CLDC .....	17
Figura 12. Perfiles J2ME.....	17
Figura 13. Ciclo de vida de un MIDlet .....	19
Figura 14 Esquema simple de sincronización .....	20
Figura 15. Estructura de un sistema de sincronización .....	21
Figura 16. Componentes de una sincronización HotSync .....	23
Figura 17. Uso de PC ID y User ID .....	25
Figura 18 Selección de los conductos a ejecutar .....	25
Figura 19. Estructura de funcionamiento de un conducto Java .....	26
Figura 20. SyncML: Un mundo de interoperabilidad.....	28
Figura 21. Relación entre GUID y LUID .....	30
Figura 22. Arquitectura genérica de sincronización.....	34
Figura 23. Arquitectura de Sincronización Utilizada en el proyecto .....	34
Figura 24. Inicialización.....	35
Figura 25. Sincronización y mapeo .....	37
Figura 26. Arquitectura Web en Tres Niveles.....	42
Figura 27. Ejemplo básico donde trabaja un Servlet .....	43
Figura 28. Interfaz Servlet.....	45
Figura 29. Ciclo de Vida de un Servlet.....	47
Figura 30. Arquitectura JDBC Servlet .....	49
Figura 31. Fase de traducción de una página JSP .....	53
Figura 32. Modelo Página Céntrica.....	53
Figura 33. Modelo MVC .....	54
Figura 34. Modelo de la Organización.....	61
Figura 35. Caso de Uso Elaborar Formato.....	61
Figura 36. Caso de Uso Programar visita.....	62
Figura 37. Caso de Uso Recibir formato .....	62
Figura 38. Caso de uso Recolectar Datos.....	63
Figura 39. Caso de Uso Transcribir datos.....	63
Figura 40. Caso de Uso Generar informe.....	64
Figura 41. Diagrama General del Modelo de la Organización .....	64
Figura 42. Diagrama General de Objetos de la Organización .....	65
Figura 43. Diagrama de Casos de Uso de Alto Nivel.....	67



Figura 44. Usuario .....	68
Figura 45. Interfaz Acceso Usuario .....	75
Figura 46. Menús de Usuario .....	75
Figura 47. Cambiar password .....	77
Figura 48. Menú de Administrador .....	78
Figura 49. Interfaz Agregar Usuario .....	80
Figura 50. Interfaz Gestión para Sitios generales de visita.....	81
Figura 51. Sección de Información General del Formato .....	84
Figura 52. Adición y Eliminación de Datos Generales al Formato .....	85
Figura 53. Sección de Preguntas del Formato .....	85
Figura 54. Elaboración de una Pregunta de Selección Múltiple.....	86
Figura 55. Elaboración de una Pregunta tipo Contenido o Texto .....	86
Figura 56. Elaboración de una Pregunta Combinada .....	87
Figura 57. Elaboración de una Pregunta Lista y Texto .....	87
Figura 58. Interfaz Definir visita.....	89
Figura 59. Interfaz Formatos seleccionados para visita.....	90
Figura 60. Interfaz Consulta de información para sitios específicos .....	92
Figura 61. Interfaz Visitas .....	92
Figura 62. Consultar información utilizando Formato Simple.....	93
Figura 63. Interfaz Consultar información utilizando Formato Tipo Encuesta .....	93
Figura 64. Interfaz Preguntas procesables estadísticamente .....	95
Figura 65. Interfaz Procesamiento estadístico pregunta combinada .....	96
Figura 66. Interfaz Transferir información.....	97
Figura 67. Lista de Visitas en Curso.....	98
Figura 68. Lista de Formatos seleccionados para la Visita.....	99
Figura 69. Interfaz Principal en la PDA .....	100
Figura 70. Interfaz Datos Generales .....	100
Figura 71. Interfaz Pregunta Selección Múltiple .....	101
Figura 72. Interfaz Pregunta Combinada y Contenido.....	101
Figura 73. Interfaz para Recolección de Datos Formato Simple.....	102
Figura 74. Interfaz Objetos de Medida .....	103
Figura 75. Interfaz de Consulta .....	103
Figura 76. Interfaz de Modificación .....	104
Figura 77. Formulario de Acceso para Sincronización .....	106
Figura 78. Progreso de Sincronización.....	107
Figura 79. Diagrama de clases del sistema.....	108
Figura 80. Diagrama de clases Arquitectura en 3 niveles.....	109
Figura 81. Clases del C.U. Iniciar sesión.....	110
Figura 82. Clases del C.U. Cambiar password.....	110
Figura 83. Clases del C.U. Agregar sitios de visita.....	110
Figura 84. Clases del caso de uso Elaborar Formato.....	111
Figura 85. Clases del caso de uso Recolectar Información en Campo.....	111
Figura 86. Clases del caso de uso Consultar Información en Campo .....	112
Figura 87. Clase del C.U. Desplegar estadísticas básicas .....	112
Figura 88. Clase del CU Transferir información.....	113
Figura 89. Clases del caso de uso Sincronizar Datos .....	113

## INTRODUCCION

La recolección de datos en campo para su posterior procesamiento, es una de las principales actividades que se lleva a cabo en ambientes empresariales, académicos y de investigación. La veracidad de los resultados obtenidos depende en gran parte, de un proceso de recolección adecuado. Los formatos escritos en papel han sido el medio de recolección por excelencia, a pesar de su extrema vulnerabilidad. Este método es susceptible a errores que tienen su origen en una transcripción deficiente de la información, pérdida de los datos por daños ocasionados al formato o simplemente, datos ilegibles.

En el Departamento del Cauca, el Instituto de Investigación e Información Geocientífica, Minero-Ambiental y Nuclear (INGEOMINAS) es una de las entidades que realiza procesos de recolección y procesamiento de datos de manera exhaustiva. Su objetivo es proveer información y conocimientos sobre el subsuelo del territorio colombiano mediante el estudio científico de su evolución, composición, dinámica, recursos y amenazas de origen geológico. Debido al carácter científico de la información, un método de recolección tradicional no es adecuado, ya que además de los inconvenientes mencionados anteriormente, el campo de trabajo de INGEOMINAS está constituido por zonas de difícil acceso donde los formatos en papel no solo son inadecuados, sino incómodos para llevar a cabo el proceso. De esta manera, es claro que esta entidad necesita mejorar la técnica empleada en la recolección y procesamiento de la información

El presente proyecto tiene como objetivo principal el diseño e implementación de un sistema integrado de recolección y procesamiento de datos, que proporcione una solución efectiva a los problemas que afronta la entidad. En primer lugar, la puesta en marcha del proyecto busca eliminar los procesos de transcripción e ilegibilidad de los datos mediante la utilización de dispositivos electrónicos conocidos como PDAs (Personal Digital Assistance), los cuales ofrecen una forma natural de ingreso de información con una pantalla sensible al tacto y brindan la posibilidad de conectarse a un PC convencional. El sistema tendrá la capacidad de enviar la información recolectada en diferentes dispositivos móviles a una base de datos central, a la cual se puede acceder vía web con el objeto de consultar la información y realizar su procesamiento. Adicionalmente, el sistema brinda la posibilidad de elaborar formatos para ser descargados, definir los sitios de trabajo en campo, programar visitas, gestionar la información de los usuarios presentes y transferir información al software estadístico Excel, el cual es utilizado por la entidad para generar resultados. Con la implementación del sistema descrito, INGEOMINAS mejora considerablemente la veracidad de los resultados obtenidos y disminuye sustancialmente el tiempo transcurrido entre la recolección de información y la generación de resultados.

Desde el punto de vista técnico, el presente trabajo de grado tiene como objetivo el diseño e implementación de una aplicación web en tres niveles capaz de integrarse con tecnologías móviles a través de un protocolo de sincronización conocido como SyncML

(Synchronization Markup Language), el cual se perfila como un futuro estándar en el mundo móvil.

El presente documento se ha dividido en cinco capítulos: El primero describe los aspectos más relevantes de las tecnologías móviles involucradas en el proyecto, Palm como sistema operativo de los dispositivos y J2ME como plataforma de desarrollo. El segundo, habla sobre los aspectos fundamentales involucrados en la sincronización de datos y se realiza una descripción simple del protocolo SyncML. El tercero está enfocado en las diferentes tecnologías web que hacen parte del proyecto. Finalmente, los capítulos cuatro y cinco corresponden al análisis y diseño del software, para el cual se hace uso del Lenguaje Unificado de Modelado (UML).

## 1 TECNOLOGÍAS MOVILES

### 1.1 LA PLATAFORMA PALM

#### 1.1.1 Introducción

Hace casi dos décadas fue concebida la idea de diseñar un asistente electrónico capaz de gestionar información personal y cumplir las funciones de una secretaria. Esta idea dio origen al concepto de PDA (Personal Digital Assistance) o Asistente digital Personal. Palm en realidad, es una fase avanzada de la evolución de este concepto, que remonta su inicio hacia el año 1987 con la aparición del Sharp Wizard PA-7000 (Japón), un organizador que contaba con pantalla, teclado y 64 Kb de memoria, con un costo aproximado de 200 dólares. Desde este momento, varias empresas tomaron la iniciativa para fortalecer el mercado de las PDAs<sup>1</sup>.

En el año 1992, Sharp (Japón) y Apple (USA) desarrollaron en conjunto a "Newton", el antepasado directo de los actuales PDAs, con pantalla sensible al tacto, software para gestionar direcciones (PIM) además de capacidades gráficas (para trazar dibujos o bocetos), gran capacidad de almacenaje para esa época (1024KB), un puerto serial para conectar al PC y un lápiz plástico aunque su tamaño era un poco más grande que el actual (19,8 x 10,5 cm). Lo más destacable era un software de reconocimiento de escritura mucho más eficiente que el "Graffiti" actualmente usado en las Palm, el cual reconocía la letra particular y el estilo de escribir del usuario. El modelo tuvo aceptación en medios especializados pero nunca fue promovido con una campaña de marketing apropiada por lo cual Apple decidió sorpresivamente discontinuar su fabricación.

Luego de varios intentos fallidos, incluido el modelo Gryphon de Microsoft, US Robotics lanza en 1996, sus modelos Palm Pilot 1000 (de 128Kb) y 5000 (de 256Kb). El nombre evocaba una palma de mano y un lápiz. Estos Palms arcaicos ya disponían del 95% de lo que se conoce actualmente: un amistoso sistema operativo (basado en la arquitectura de los primeros PCs de Apple), capacidades para administrar listas de contactos, reuniones, tareas, etc, método de graffiti y teclado virtual, capacidad para soportar software creado por terceros, interface al PC (básicamente es el mismo software usado hoy) y el clásico tamaño y diseño de bolsillo. Posteriormente, 3Com adquiere los derechos del producto y gracias a una estrategia de marketing acertada enfatizada en promover más un servicio que ayudara a las personas a organizar su tiempo eficientemente, que un simple artefacto electrónico más, alcanza el 55% del mercado. Por otro lado, 3Com promovió la primera versión del kit de desarrollo SDK de forma gratuita en Internet, con la cual la producción de software creció en forma exponencial.

De dónde nace entonces el nombre de Palm vigente hoy en día? En 1998, la compañía de lápices Pilot demanda a 3Com por el uso no autorizado de su nombre, por lo cual esta

---

<sup>1</sup> J.C Sandoval.; Breve Historia de las PDAs <http://www.todoportable.cl>

última decide no solo cambiar el nombre de su modelo con el lanzamiento de la Palm III (capacidad de comunicación infrarroja y sistema operativo 3.0) sino que independiza su línea de PDAs dando origen a Palm Inc.

Desde entonces Palm se ha posicionado de una forma importante en el mercado (con más del 75% según estudios de la empresa analista de mercados IDC<sup>1</sup>), pese a los esfuerzos de Microsoft por consolidar su sistema operativo Windows CE que sirve como plataforma a las Pocket PC, competidoras directas de las Palm. Aunque Windows CE sorprende con la apariencia de tener un “sophisticado PC en el bolsillo”, el costo que se paga por ello es claro: Menor duración de la batería, mayor tamaño y por supuesto, mayor costo. Palm por otro lado, aparece como una plataforma menos sofisticada pero no menos útil, fundamentada en una visión bastante concreta: “Los PCs están diseñados para la creación de información, las PDAs para acceder y gestionar la información”<sup>2</sup>.

Para finalizar esta primera parte, se deben resaltar dos aspectos: el primero, que las Palm y Pocket PC son clases de PDAs por lo tanto, estos conceptos no deben tomarse como sinónimos; es decir, una Palm es una PDA, mas no es necesariamente correcto lo contrario. Por otro lado, Palm se proyecta como la plataforma líder para la comunidad de desarrolladores en computación móvil, gracias a su amplia aceptación en el mercado y las miles de aplicaciones disponibles, lo cual mejora los servicios prestados por este interesante dispositivo. A continuación se describirán los aspectos técnicos más relevantes de la plataforma Palm.

### 1.1.2 Características generales

Diseñar aplicaciones para Palm requiere un punto de vista diferente del que se tiene al programar para desktops con grandes pantallas a color, donde los usuarios se sientan y trabajan por muchas horas. Las handhelds Palm son dispositivos operados por batería, portables y de baja potencia, usados ocasionalmente. A continuación se muestran sus principales características:

Figura 1. Palm m125



---

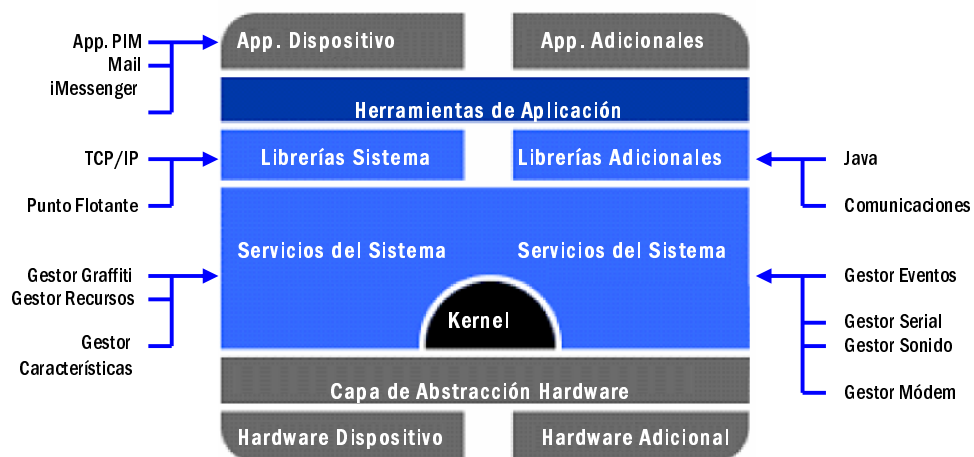
<sup>1</sup> Revista Enter; Palm se enfoca en el mercado corporativo; <http://enter.terra.com.co>. Julio 2002

<sup>2</sup> Palm Inc; Comparison: Palm Handhelds vs. Windows CE 3.0. <http://www.palmsource.com>.

- Tamaño de la pantalla: La mayoría de las handhelds Palm poseen una pantalla de 160x160 píxeles, lo cual limita la cantidad de información que puede desplegarse.
- Patrón de uso: El patrón de uso para las handhelds es comúnmente cortos periodos de tiempo en el día. No se debe considerar a las handhelds como pequeños laptops.
- Entradas limitadas: Las Palm proveen al usuario de la herramienta graffiti para introducir datos a través del stylus. También se dispone de una pequeña pantalla de teclado; estas dos herramientas no son tan eficientes como el teclado y el mouse de un desktop.
- Potencia: La velocidad de procesamiento de la mayoría de las handhelds se encuentran en rango de 16 a 33 MHz, aunque los modelos modernos ya han sobrepasado este valor.
- Conectividad con el PC: Palm proporciona una tecnología de sincronización conocida como HotSync, la cual permite a las aplicaciones intercambiar información con el PC. Adicionalmente se proporciona un kit de desarrollo para nuevas aplicaciones.
- Batería: se tienen baterías recargables y no recargables para las handhelds. Para un uso moderado del dispositivo, las primeras pueden tener un promedio de duración de un mes, mientras las segundas pueden mantener su carga por una o dos semanas.

### 1.1.3 Sistema operativo Palm

Figura 2. Arquitectura del sistema operativo Palm



En la Figura 2 se puede observar la arquitectura en capas del sistema operativo Palm<sup>1</sup>. Los aspectos más importantes de estas capas se describen a continuación:

<sup>1</sup> Palm Inc; A flexible architecture for innovative solutions. <http://www.palmos.com/platform>. Diciembre 2002.

- *Eventos*: Los eventos o señales informan a la aplicación de qué es lo que está ocurriendo en el sistema, desde una pulsación de un botón de un teclado externo, hasta el inicio de una operación de sincronización.
- *La memoria*: La RAM está dividida en dos áreas, memoria de almacenamiento (sistema de ficheros) y memoria dinámica (la que emplea la aplicación durante la ejecución). Cada una de estas es gestionada por una parte distinta del sistema operativo, en el primer caso el *Database Manager* y en el segundo por el *Memory Manager*.
- *Controles y formularios*: Por lo general, podemos definir un formulario como una pantalla para cada vista de la aplicación. Los formularios son contenedores de controles. Un control puede ser un botón, una etiqueta, una lista, una tabla, un bitmap, un campo de texto o cualquier otro objeto definido por el usuario (Gadgets).
- *Menús*: Los menús están asociados con un formulario; generalmente se definen en tiempo de compilación, aunque pueden cambiarse dinámicamente. Los menús se componen de barras de menú y opciones dentro de éstos.
- *Comunicaciones*: *Palm OS* da soporte a tres protocolos de comunicaciones: *IrDA*, serie (RS232) y *TCP/IP*. El sistema operativo proporciona una serie de funciones para trabajar con estos protocolos. *TCP/IP* está disponible sólo sobre el puerto serie o sobre una conexión con módem.

#### 1.1.4 Gestión de la memoria del dispositivo

La cantidad de memoria disponible en un dispositivo Palm varía de acuerdo al modelo. Es común encontrar capacidades entre 2 y 16 Mb. En una handheld Palm se tienen dos tipos de memoria ROM y RAM, las cuales residen en un módulo conocido como "card". Aunque este término puede confundirse con las tarjetas de expansión disponibles para este tipo de dispositivos, en realidad un "card" es simplemente una construcción lógica usada por el sistema operativo. La ROM y la RAM en cada card es dividida en una o más áreas contiguas de almacenamiento conocidas como "heaps", cuya capacidad no excede los 64Kb. Los heaps no necesariamente tienen que ser adyacentes entre sí; estos pueden ser extendidos por todo el espacio de almacenamiento, siempre y cuando residan en el mismo card. El sistema operativo y las aplicaciones proporcionadas por Palm residen en ROM y son totalmente actualizables.

#### 1.1.5 Arquitectura de la memoria

La memoria de un dispositivo Palm está diseñada bajo una arquitectura de 32 bits. El sistema operativo divide la RAM total en dos áreas lógicas: RAM dinámica y RAM de almacenamiento. La RAM dinámica es usada como un espacio de trabajo temporal, similar a la RAM empleada en un PC convencional mientras que la RAM de almacenamiento, es similar al disco duro. Los dos tipos de memoria mantienen su contenido cuando el dispositivo se apaga, gracias a que la potencia de la batería se aplica sobre las mismas. La RAM dinámica se utiliza para implementar un "heap" simple que provee memoria para localizaciones dinámicas como por ejemplo, variables globales. El

tamaño de esta memoria depende de la versión del sistema operativo, la cantidad de RAM física disponible y los requerimientos de software instalado como el stack de TCP/IP o IrDA.

### 1.1.6 Data Manager

Aunque los dispositivos Palm no tienen un sistema de archivos como los que se manejan en un PC convencional, dispone de una aplicación que gestiona las operaciones que deben llevarse a cabo en memoria conocida como DataManager. Su forma de trabajo es bastante diferente a la realizada en un sistema de archivos tradicional; En este último se localiza una porción del archivo en un buffer de memoria para posteriormente llevarlo al disco. Debido a las limitaciones de memoria, este modelo no es óptimo para gestionar los datos del usuario en Palm OS. Por esta razón, Data Manager accesa y actualiza la información en el mismo lugar. Esto reduce los requerimientos de memoria dinámica y elimina el costo de transferir los datos desde un buffer como en el sistema de archivos.

Como una mejora adicional, los datos en un dispositivo Palm son fragmentados en múltiples registros de tamaño finito que pueden ser esparcidos por todo el espacio de memoria. Así, añadir, borrar o redimensionar un registro no requiere mover otros registros alrededor en memoria.

### 1.1.7 Tipos de archivos

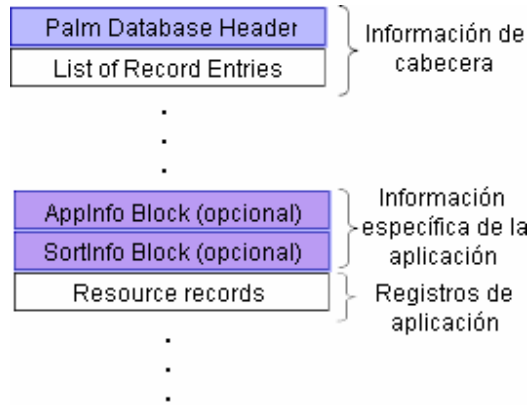
Actualmente existen tres tipos de archivos que se utilizan en la plataforma Palm: Palm Database (PDB), Palm Resource (PRC) y Palm Query Application (PQA). Los archivos correspondientes a cada tipo, se diferencian en un PC por su extensión y cumplen una función específica:

- Palm Database (PDB): Corresponde a un conjunto de registros que almacena los datos para una aplicación.
- Palm Resource (PRC): Es una base de datos que almacena los recursos de una aplicación. En esta información se incluye el código fuente y elementos de interfaz de usuario.
- Palm Query Application (PQA): Básicamente es una PDB que almacena contenido web. En una handheld Palm, este tipo se asocia a un software conocido como Web Clipping Application Viewer, una especie de browser que permite visualizar su contenido.

En términos generales, una base de datos consiste de una cabecera y una lista secuencial de registros. Adicionalmente, es posible que una base de datos utilice una o dos porciones de formato libre para almacenar datos relacionados con la aplicación. La estructura de la cabecera de un registro, se muestra en la Figura 3.

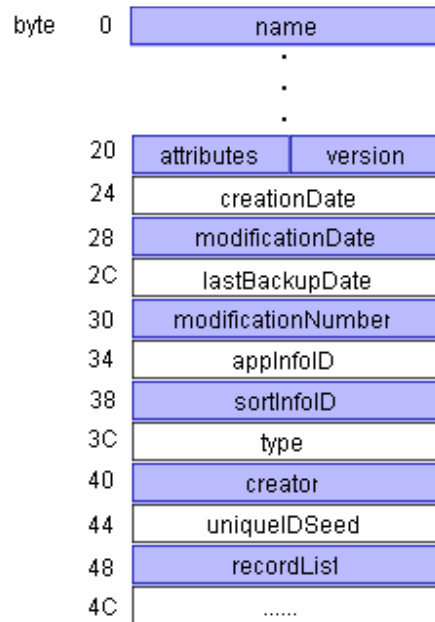


Figura 3. Estructura básica de una base de datos



### 1.1.8 Campos de la cabecera de base de datos

Figura 4. Estructura de la cabecera



En la Figura 4 se muestran los campos que componen la cabecera de una base de datos. Su significado se describe a continuación:

- Name: Contiene el nombre de la base de datos.
- Attributes: Contiene banderas de la base de datos, útiles en el proceso de sincronización.
- Version: Contiene un número específico de la aplicación para la base de datos.

- CreationDate: Fecha de creación de la base de datos<sup>1</sup>
- ModificationDate: Última fecha de modificación de la base de datos<sup>1</sup>
- LastBackupDate: Fecha del backup más reciente en la base de datos
- ModificationNumber: Se incrementa cada vez que un registro en la base de datos es borrado, añadido o modificado. Así las aplicaciones pueden determinar fácilmente si una base de datos compartida ha sido modificada por otro proceso.
- ApplInfoID: Corresponde al offset local desde el inicio de la cabecera hasta el comienzo del campo opcional applInfo, el cual puede contener información específica de la aplicación.
- SortInfoID: Corresponde al offset local desde el inicio de la cabecera hasta el comienzo del campo opcional sortInfo, que puede incluir información de ordenamiento de registros.
- Type: Identifica el tipo de base de datos. Para PDB, este identificador depende del desarrollador de la aplicación; para PRC, usualmente es app1 y para PQA corresponde a pqa en todos los casos.
- Creator: Campo de 4 bytes que identifica al creador de la base de datos. Para bases de datos PQA, este valor siempre corresponde a clpr.
- UniqueIDSeed: Usado internamente por el sistema operativo para generar identificadores únicos para los registros cuando la base de datos es cargada en el dispositivo.
- RecordList: Lista de registros de la base de datos.

Es importante tener en cuenta, que estos formatos describen la estructura de las bases de datos cuando se almacenan en el PC en forma de archivos. Los formatos son similares en la memoria del dispositivo, con algunas diferencias que no son documentadas por Palm Inc. Cuando el Gestor de memoria del dispositivo rastrea los registros, crea algún overhead, lo cual hace que el tamaño de las bases de datos sea menor en el PC<sup>2</sup>

### 1.1.9 Filosofía Palm

Cuando se trata de construir aplicaciones para handhelds, los desarrolladores se enfrentan a un nuevo desafío. No solo los limitados recursos hardware imponen una sólida barrera en este proceso, sino también una clara filosofía de desarrollo basada en tres conceptos fundamentales:

- El usuario es el centro de la solución: Los usuarios no adquieren una handheld bajo el vago concepto de tener un pequeño PC. En realidad los usuarios desean satisfacer una necesidad particular.
- Las handhelds están disponibles en cualquier lugar y a cualquier hora: Las handhelds fueron diseñadas para proporcionar funciones computacionales donde los PCs o laptops no son convenientes. De esta razón se desprende su reducido peso y tamaño.
- Las aplicaciones deben tener un enfoque específico: A diferencia de un PC, las handhelds son dispositivos especialmente diseñados para utilizarse en formas muy específicas y ampliamente enfocadas en satisfacer necesidades muy importantes para el usuario.

---

<sup>1</sup> Especificada como el número de segundos desde Enero 1 de 1904 12 A.M

<sup>2</sup> Palm Inc; Palm File Format Specification. <http://www.palmsource.com>.

Teniendo en cuenta los principios expuestos anteriormente, a continuación se describen los tres pilares de la “Filosofía de desarrollo Palm”.

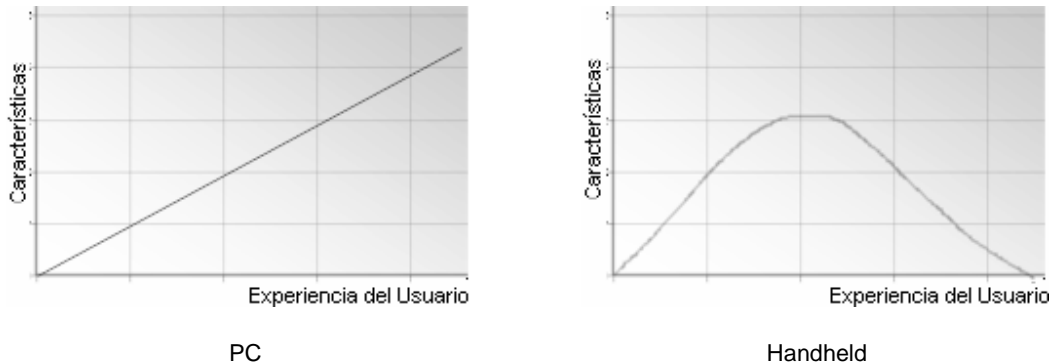
## 1. Del PC a la Handheld

Las diferencias entre un PC y una handheld van más allá del tamaño y la cantidad de recursos hardware. Para entender mejor estas diferencias es necesario comprender la esencia de cada dispositivo.

**La esencia del PC:** En el PC se mantiene una relación lineal entre características y valor. Así lo reflejan las palabras de Steve Ballmer de Microsoft, al afirmar que “el software debería crecer cada año”<sup>1</sup>. Esta es la esencia del PC: Tener muchas funciones y herramientas para hacer cada vez más cosas, aunque esto implique actualizar el sistema operativo, mejorar las características hardware del equipo y largos tutores que instruyan a los usuarios en el manejo de las aplicaciones.

**La esencia de las Handhelds:** A diferencia del PC, las Handhelds giran en torno a la agilidad y facilidad para acceder a la información. Por lo tanto, su esencia plantea aplicaciones simples, de manejo intuitivo, sin largos tutoriales y adaptables a los limitados recursos. Como lo muestra la figura 5, en el mundo de las Handhelds, “más puede ser menos”, llevando al usuario a una experiencia frustrante.

Figura 5. Características vs. Experiencia del usuario



Además de su esencia, las Handhelds y los PCs tienen patrones de uso bastante diferentes. Mientras las personas pasan horas frente al PC, con grandes pantallas y discos duros de gran capacidad, los usuarios de las Handhelds, realizan un acceso ágil a la información durante cortos periodos de tiempo.

La conclusión para este primer pilar se puede resumir en la siguiente frase: Los desarrolladores de aplicaciones para Handhelds deben estar dispuestos a dejar a un lado, su conocimiento y experiencia desarrollando productos para PC. La esencia de las Handhelds impone a los desarrolladores un nuevo desafío.

---

<sup>1</sup> S. Ballmer; Building a Platform on Web's Technology, entrevista por Karl Lillington, San Jose MercuryNews. Diciembre 1999.

## 2. Soluciones, no Características

Uno de los factores importantes a tener en cuenta en el desarrollo de aplicaciones para Handhelds, es que estas no sobresalen por sus características técnicas, con procesadores rápidos y grandes cantidades de memoria, sino mas bien por el beneficio que brindan a sus usuarios. En la Tabla 1, se muestra claramente que las necesidades del usuario en últimas, determinaron las características del dispositivo:

Tabla 1. Necesidad del usuario vs. Características del dispositivo

Necesidad del usuario	Característica del dispositivo
Tamaño de bolsillo	Alta integración hardware Despliegue reducido Estructura de datos compacta
Fácil y rápido	Software compacto orientado a tareas Herramienta para escritura Graffiti Solo una aplicación se ejecuta al tiempo
Bajo costo	Componentes de bajo costo Expansión hardware mínima Tamaño de memoria reducido
Mayor duración de la batería	RAM estática y pseudo-estática Tamaño de memoria reducido Apagado automático
Conexión con el PC	Intercambio de información entre aplicaciones del PC y la Handheld. Diseño de un Gestos de datos para el rastreo de cambios en la información de la Handheld Conexión a través de puerto serie o USB.

Es claro entonces que las aplicaciones para Handhelds, se deben centrar en proporcionar simplemente soluciones y no ser un conjunto de complicadas características. Así cuando se piensa en el desarrollo de una aplicación para este tipo de dispositivos, se deben seleccionar cuidadosamente las características más importantes y desechar aquellas que se consideran simplemente como “adicionales”. Lograr un equilibrio entre el número de características presentes en la aplicación y la satisfacción del usuario, no es una tarea sencilla. A continuación se presenta una guía simplificada para lograr este equilibrio.

*Buscar las necesidades detrás de los requerimientos del usuario:* Complacer al usuario no siempre significa seguir al pie de la letra los requerimientos, más aún si son bastante complejos. La búsqueda de alternativas juega un papel fundamental en el momento de desarrollar un aplicación para Handhelds. Mas que simplemente usar la tecnología, el propósito debe ser la innovación. Un claro ejemplo lo demuestran los creadores de Palm; cuando se planteó como requerimiento la capacidad del dispositivo para reconocer la manera natural de escritura del usuario, se dieron cuenta que para cumplirlo necesitarían un procesador más rápido, mayor cantidad de memoria y más duración de las baterías. Fue entonces cuando llegaron a la conclusión que el problema radicaba en un método eficiente para introducir texto, lo cual llevó al nacimiento del “Graffiti”, un sistema alternativo que proporciona una representación simplificada para cada carácter, fácil de aprender y muy eficiente en el manejo de los recursos del dispositivo.

En la mayoría de los casos, los clientes o usuarios finales no son conscientes del desagradable impacto que puede causar una larga lista de requerimientos en un dispositivo con características técnicas limitadas. Por esta razón, se debe indagar a fondo cada uno de los requerimientos con el usuario con el fin de determinar las necesidades reales. Una vez identificadas, es aconsejable la presentación de prototipos para conocer la reacción del cliente ante las nuevas alternativas.

*Añadir características siempre y cuando mejore la experiencia del usuario:* Añadir una característica a la aplicación implica mantener un balance entre la tecnología disponible y utilidad. Para esto, antes de adicionar una característica se deben responder las siguientes preguntas:

- ¿Qué gana el usuario?
- ¿Cuál es el costo en cuanto a dificultad y recursos hardware?

*La regla 80/20:* Una forma de anticiparse al exceso de características en una aplicación es aplicar la regla 80/20, que consiste en concentrarse en lo que los usuarios hacen el 80% del tiempo y tratar de ignorar el otro 20%. Por ejemplo, los diseñadores Palm decidieron que al presionar el botón Date Book (aplicación para programar actividades en una fecha y hora específica), se desplegara la información correspondiente al día actual, ya que los usuarios el 80% del tiempo desean conocer lo que han programado para hoy.

Figura 6. DateBook



*Descomponer las aplicaciones:* En ambiente para PC es deseable y algunas veces requerido, construir aplicaciones robustas que consideren todos los casos posibles. Sin embargo, para una Handheld no es muy práctico tratar de cubrirlo todo. Por esta razón en algunas ocasiones, es necesario descomponer una aplicación grande en aplicaciones más pequeñas, puesto que a diferencia de los usuarios de PC, los usuarios de Handhelds prefieren aplicaciones más livianas y más específicas.

*Compartir el trabajo:* Una de las características más importantes de las Handhelds es su capacidad para intercambiar información con un PC. Por lo tanto, una buena solución debería considerar la posibilidad de trasladar el trabajo pesado a un PC tomando ventaja de sus mayores recursos computacionales. Desde este punto de vista, se logra diseñar una aplicación en la Handheld como satélite de una aplicación en el PC, considerando la primera como el medio para acceder y gestionar contenido y la segunda, como el medio para crear y procesar información.

*Crear aplicaciones intuitivas:* Uno de los factores más importantes que se deben tener en cuenta al crear aplicaciones para Handhelds es la intuición como recurso primario del usuario para conocer el funcionamiento de la aplicación. A diferencia de las aplicaciones para PC, no es conveniente desarrollar largos tutores o manuales de usuario para aprender a manejar la aplicación. El usuario busca en las Handhelds mayor agilidad en el acceso a la información. Para lograrlo, fundamentalmente se deben seguir dos técnicas: Minimizar al máximo el número de elementos en la pantalla y en lo posible, utilizar componentes característicos de la interfaz del dispositivo (Botones, listas, diálogos, etc) de tal manera que el usuario se familiarice fácilmente con la aplicación.

*Reducir el número de pasos a seguir:* Una forma de conseguir mayor velocidad en la aplicación es reducir el número de acciones que debe realizar el usuario para una tarea específica. Este factor debe ser muy importante en las funciones que el usuario utiliza más frecuentemente (Regla 80/20).

*Ocultar opciones avanzadas:* En algunas ocasiones, es necesario disminuir la facilidad de acceso por parte del usuario a ciertas funciones. Los casos más comunes son aquellos que involucran la eliminación de registros o información importante. Una técnica común es el uso de diálogos para confirmar la realización de operaciones peligrosas, pero adicionalmente, es recomendable ocultar este tipo de funciones a través de un menú por ejemplo, por razones de seguridad.

Figura 7. Menú para opciones avanzadas



### 3. El Usuario tiene la última palabra

Una de las fases más importantes en el desarrollo del software es la fase de validación con el usuario. En el mundo de las Handhelds esta fase no es la excepción y constituye un paso fundamental hacia una buena solución. Para muchos usuarios el mundo de los computadores se convierte en un universo hostil, por lo cual se hace necesario obtener de ellos una buena retroalimentación de ideas y experiencias acerca de la aplicación desarrollada.

La validación no es una fase que deba realizarse al final del camino. Por el contrario, es una actividad que siempre debe estar presente. La realización de prototipos en fases tempranas del desarrollo de la aplicación es una buena costumbre que ayuda a minimizar, en gran proporción, el riesgo de impactos indeseados para el usuario final. El proceso de validación pretende resolver los siguientes interrogantes:

- ¿La aplicación logra satisfacer un requerimiento específico?
- ¿El usuario puede intuir el propósito de la aplicación a través de su nombre o interfaz?
- ¿La interfaz lleva al usuario a intuiciones erradas acerca de la funcionalidad de la aplicación?
- ¿Cuánto tiempo toma el usuario para llevar a cabo una tarea completa?
- ¿Cuánto tiempo toma el usuario para recordar la manera de llevar a cabo una tarea realizada previamente?

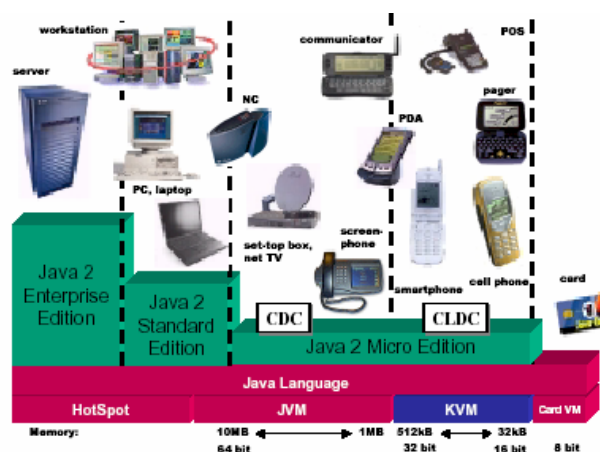
De acuerdo a las características específicas de la aplicación, las respuestas a estos interrogantes lleva a conclusiones importantes, de las cuales se deriva la aprobación de la versión actual de la aplicación o se da inicio a un nuevo proceso de búsqueda de alternativas<sup>1</sup>.

## 1.2 Java 2 Micro Edition

La primera versión de Java, J2SE (Java 2 Standar Edition)<sup>2</sup>, creó un conjunto de herramientas y APIs para desarrollar applets y aplicaciones. Al principio, esto parecía suficiente pero con el paso de los años la demanda de aplicaciones distribuidas que comparten información entre distintas ubicaciones físicas, dio origen a J2EE (Java 2 Enterprise Edition)<sup>3</sup> para aplicaciones de ambientes “enterprise” que exigen mayor capacidad de almacenamiento, memoria y por consiguiente, mayor complejidad que las aplicaciones convencionales.

Una nueva generación de aplicaciones para dispositivos móviles de escasa memoria, recursos computacionales limitados y pequeños despliegues, justificó el surgimiento de una tercera plataforma: J2ME (Java 2 Micro Edition)<sup>4</sup>. J2ME, busca resumir la funcionalidad de J2SE para adaptarse a los requerimientos mínimos de dispositivos con capacidades limitadas de procesamiento. En la Figura 8 se muestra la relación entre las tres versiones de Java.

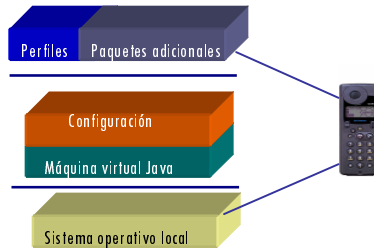
Figura 8. Ediciones Java con sus respectivos mercados



<sup>1</sup> Palm Inc; Zen of Palm. <http://www.palmsource.com>  
<sup>2</sup> Sun Microsystems; J2SE (Java 2 Standar Edition). <http://java.sun.com/j2se>. Mayo 2003.  
<sup>3</sup> Sun Microsystems; J2EE (Java 2 Enterprise Edition). <http://java.sun.com/j2ee>. Mayo 2003  
<sup>4</sup> Sun Microsystems; J2ME (Java 2 Micro Edition)". <<http://java.sun.com/j2me>>. Mayo 2003.

### 1.2.1 Arquitectura J2ME

Figura 9. Arquitectura J2ME



La arquitectura de alto nivel para J2ME<sup>1</sup> que se muestra en la Figura 9, representa una estructura en 4 capas básicas:

- Sistema operativo local: Representa el sistema operativo del dispositivo donde corre la máquina virtual de Java.
- Máquina virtual Java: Esta capa es la implementación de la máquina virtual de Java optimizada para el sistema operativo del dispositivo y soporta una configuración específica.
- Configuración: Esta capa es menos visible para los usuarios pero es muy importante para la implementación de los perfiles. Define un conjunto mínimo de clases Java y características de la máquina virtual para una categoría de dispositivos que representan un segmento “horizontal” del mercado es decir, define las características comunes a un grupo de familias de dispositivos.
- Perfil: es la capa más visible para los usuarios y desarrolladores de aplicaciones. Define un conjunto mínimo de APIs (Application Programming Interface) para una familia de dispositivos que representan un segmento “vertical” del mercado, es decir una familia de dispositivos específica (celulares, PDAs). Los perfiles se implementan sobre una configuración específica.
- Paquetes opcionales: En esta capa se agrupan aplicaciones desarrolladas por terceros para realizar funciones que aún no se encuentran dentro del estándar, por ejemplo APIs para XML, herramientas multimedia, entre otras.

Figura 10. Relación entre Configuración y Perfil



---

<sup>1</sup> G. Diez-Andino y R.M García; Java2 Micro Edition: Un primer vistazo. <http://www.it.uc3m.es/pervasive>.



La Figura 10 especifica mas claramente la relación entre las configuraciones y los perfiles a través de un ejemplo hipotético. El conjunto A representa un perfil exclusivo para PDAs. El conjunto B representa un perfil para teléfonos celulares. Si se realiza la operación  $A \cap B$ , se obtiene la configuración sobre la cual se basan los perfiles A y B. A continuación describiremos con más detalle cada una de las capas.

### 1.2.2 KVM (Kilobyte Virtual Machine)

La tecnología KVM<sup>1</sup>, es una máquina virtual portable específicamente diseñada para dispositivos de bajos recursos. El principal objetivo en esta tecnología es disponer de una máquina virtual lo más reducida posible que mantenga las características de Java (“write once run anywhere”) y corra en unos pocos kilobytes de memoria. La KVM está diseñada para correr en microprocesadores RISC/CISC de 16/32 bits con un total de memoria mínimo (menos de 128 Kb), lo cual incluye la máquina virtual, las librerías Java mínimas especificadas por la configuración y algo de espacio heap para ejecutar las aplicaciones Java.

### 1.2.3 Configuraciones

Actualmente, existen dos configuraciones definidas: CDC (Connected Device Configuration) y CLDC (Connected Limited Device Configuration).

CDC: Los requerimientos para esta configuración fueron establecidos en JSR 36<sup>2</sup>. Agrupa dispositivos con las siguientes características:

- 512 Kb de ROM disponible (mínimo)
- 256 Kb de RAM disponible (mínimo)
- Conectividad a la red
- Especificación de máquina virtual compacta
- Interfaces de usuario relativamente reducidas

En esta configuración podemos tener dispositivos como Screen phones, Set top boxes, PDAs de última generación, entre otros.

CLDC: Los requerimientos para esta configuración fueron establecidos en JSR 30<sup>3</sup>. Agrupa dispositivos con las siguientes características:

- 128 a 512 Kb de memoria total disponible
- Potencia limitada (batería)
- Conectividad a la red
- Especificación de máquina virtual reducida (KVM)
- Interfaces de usuario reducidas

---

<sup>1</sup> Sun Microsystems; Java. 2 Platform MicroEdition (J2ME.) Technology for Creating Mobile Devices White Paper. <http://java.sun.com/j2me>. Mayo 2000.

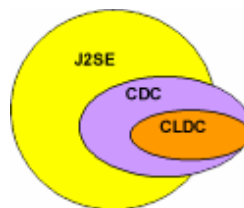
<sup>2</sup> Java Community Process ; Connected Device Configuration (JSR-036). <http://www.jcp.org/jsr/detail/36.jsp>.

<sup>3</sup> Java Community Process ; Connected Limited Device Configuration (JSR-030). <http://www.jcp.org/jsr/detail/30.jsp>.

En esta configuración se destacan dispositivos como teléfonos celulares y un amplio rango de PDAs.

Aunque la mayor parte de la funcionalidad de CDC y CLDC ha sido heredada de J2SE, estas configuraciones pueden adicionar nuevas características para adaptarse específicamente a las necesidades de los dispositivos. Por otro lado, CLDC es un subconjunto de CDC como lo muestra la Figura 11, lo cual garantiza de alguna manera la compatibilidad entre las configuraciones si se tiene en cuenta que una aplicación CLDC podrá ejecutarse bajo CDC, mas no podrá llevarse a cabo el proceso contrario.

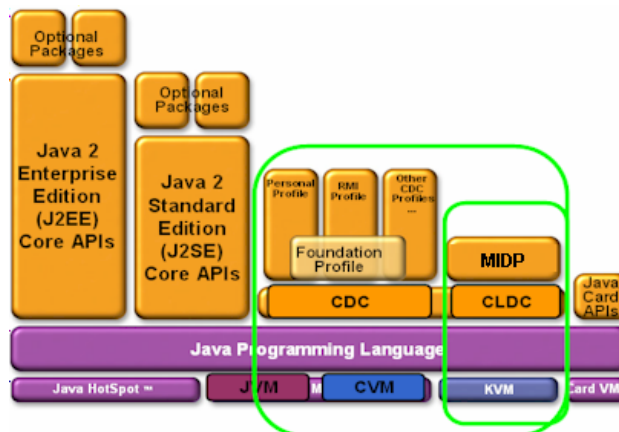
Figura 11. Compatibilidad entre CDC y CLDC



### 1.2.4 Perfiles

En la actualidad se han definido varios perfiles para implementarse sobre las configuraciones vistas anteriormente, como se muestra en la Figura 12. Algunos ya se encuentran implementados, otros se encuentran en desarrollo.

Figura 12. Perfiles J2ME



Se realizará una descripción del perfil MIDP (Mobile Information Device Profile)<sup>1</sup>, ya que es el más utilizado en el desarrollo de aplicaciones móviles actualmente. MIDP está

<sup>1</sup> Sun Microsystems; Mobile Information Device Profile (JSR-37) JCP Specification. <http://java.sun.com/j2me>. Diciembre 2000.

diseñado para operar bajo la configuración CLDC y su objetivo es establecer un conjunto de APIs que proporcionen un ambiente de desarrollo para dispositivos de información móvil o MIDs (Mobile information devices) que cumplan con las siguientes especificaciones:

- Tamaño mínimo de la pantalla: 96x54 pixels con una profundidad de despliegue de 1 bit (blanco y negro).
- Mecanismos de entrada: Teclados manejados con una mano (teléfonos celulares), teclados manejados con las dos manos (pagers) o dispositivos punteros (PDAs).
- Memoria RAM: 128 Kb para componentes MIDP, 8 Kb para datos creados por la aplicación y 32 Kb para el ambiente de ejecución Java (memoria heap).
- Conexión a red ocasional con ancho de banda limitado (típico 9.6 Kbps)

Es necesario tener en cuenta que MIDP asume algunas características acerca del sistema operativo del dispositivo con el objeto de garantizar la ejecución correcta de las aplicaciones. Entre las características que MIDP asume se encuentran las siguientes:

- Disponibilidad de un mecanismo para leer y escribir sobre memoria RAM.
- Acceso de lectura/escritura a las capacidades de red del dispositivo.
- Un mecanismo que proporcione una base de tiempo para sello de tiempo (timestamp) sobre los registros presentes en memoria.
- Capacidad mínima para escribir en un despliegue de mapa de bits.

### 1.2.5 Librerías de clases MIDP

MIDP ha definido un conjunto de librerías para adaptarse a las características de los dispositivos móviles que cumplen su especificación. Se han establecido paquetes adicionales identificados como `javax.microedition` y se han heredado algunas características de J2SE como se muestra a continuación:

- `javax.microedition.midlet`: Paquete para gestionar el ciclo de vida de la aplicación
- `javax.microedition.lcdui`: Paquete de interfaz de usuario.
- `javax.microedition.rms`: Paquete para gestionar almacenamiento persistente en el dispositivo.
- `javax.microedition.io`: Paquete para networking.
- `java.lang` y `java.util`: Un subconjunto de los paquetes J2SE para diferentes utilidades.

Hasta el momento, MIDP es el único perfil implementado sobre la configuración CLDC; sin embargo, otro perfil conocido como PDAP (PDA Profile)<sup>1</sup> se encuentra en desarrollo. Como su nombre lo indica, el objetivo de este perfil es especializarse en el mundo de las PDAs y aprovechar al máximo las mejores capacidades de estos dispositivos que de alguna manera, se encuentran subutilizadas por el MIDP. Mientras se da la transición hacia este perfil, se ha definido un paso intermedio que corresponde a una implementación del MIDP específica para Palm OS<sup>2</sup>. A través de esta especificación, el MIDP define algunas características especiales que se adecuan a las PDAs que cuentan

---

<sup>1</sup> Java Community Process ; Personal Digital Assistant Profile (JSR-075). <http://www.jcp.org/jsr/detail/75.jsp>

<sup>2</sup> Sun Microsystems ; MIDP for Palm OS, Version 1.0 FCS". <http://java.sun.com/j2me>> Octubre 2001.

con sistema operativo Palm. Por qué definir una implementación para Palm y no para otras plataformas? La razón se encuentra en el mercado.

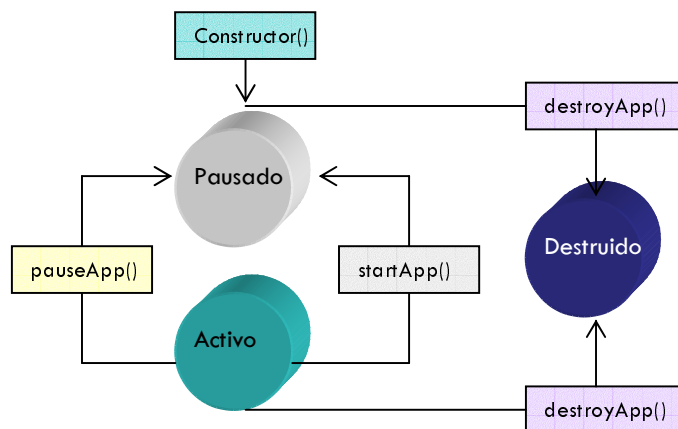
### 1.2.6 Ciclo de vida de una aplicación J2ME

Las aplicaciones implementadas bajo el perfil MIDP, reciben el nombre de MIDlets y deben agruparse en un fichero .jar para ser distribuidas a los diferentes dispositivos. En un fichero .jar se pueden incluir varios MIDlets, lo cual recibe el nombre de MIDlet Suite. Cuando se trabaja con J2SE, la máquina virtual lleva a cabo un proceso de verificación en tiempo de ejecución. Este proceso consume recursos, lo cual no es adecuado en un ambiente J2ME. En este caso, en el proceso de compilación se lleva a cabo una tarea conocida como “preverificación” que consiste en llevar a cabo parte de la verificación de la JVM antes de realizar la descarga de las clases al dispositivo. Una vez compilado, se generan dos archivos muy importantes a la hora de descargar los MIDlets:

- Manifiesto: Su finalidad es describir el contenido del fichero .JAR a través de información como el nombre, versión, fabricante, etc; también se incluye en este fichero una entrada por cada MIDlet que lo compone. Su extensión es .mf.
- Descriptor: Aunque este fichero comparte el mismo formato que el del manifiesto, su finalidad es totalmente diferente. El objetivo de este fichero es proporcionar la información requerida por el Application Management Software (programa que gestiona las descargas de aplicaciones en el móvil) para cerciorarse que el MIDlet suite es adecuado para el dispositivo en el que se va a instalar. Su extensión es .jad y al contrario que el manifiesto, éste no se incluye en el paquete.

Cuando el MIDlet entra en ejecución, inicialmente va al estado de pausa y pasa al estado activo. El estado de pausa es útil cuando se desea cambiar el MIDlet que se está ejecutando, pasando de activo a pasivo. El estado destruido hace referencia a la liberación de recursos por parte del MIDlet. La figura 13 resume el ciclo de vida de un MIDlet.

Figura 13. Ciclo de vida de un MIDlet

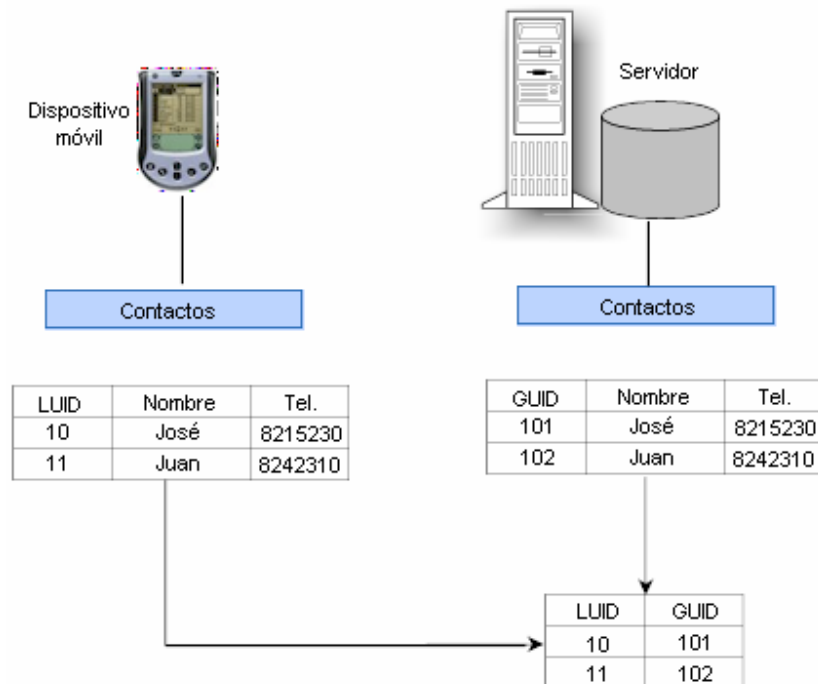


## 2 SINCRONIZACION

### 2.1 Introducción

La tecnología móvil ha extendido sus fronteras en el mundo moderno. Como consecuencia, la necesidad de transportar información en diferentes dispositivos se hace evidente; sin embargo, el verdadero beneficio que aporta un dispositivo móvil es su capacidad para intercambiar información con bases de datos corporativas a través de diferentes tecnologías de red, lo cual implica que la información que reside en este tipo de dispositivos no es estática sino dinámica. En la mayoría de los casos, disponer de una conexión permanente con una fuente de datos remota, no es solo una solución indeseable sino también impráctica debido a los requerimientos de ancho de banda y baterías que esto significa. De esta manera, es indispensable que cada usuario móvil disponga de una copia local de los datos en su dispositivo, la cual pueda acceder y manipular de acuerdo a sus necesidades. No obstante, la fuente de datos central necesita conocer las modificaciones realizadas por los usuarios móviles, así como también reportarles los cambios que hayan ocurrido en su contenido, resolviendo posibles conflictos. A este proceso se le conoce como “*Sincronización de datos*”<sup>1</sup>.

Figura 14 Esquema simple de sincronización



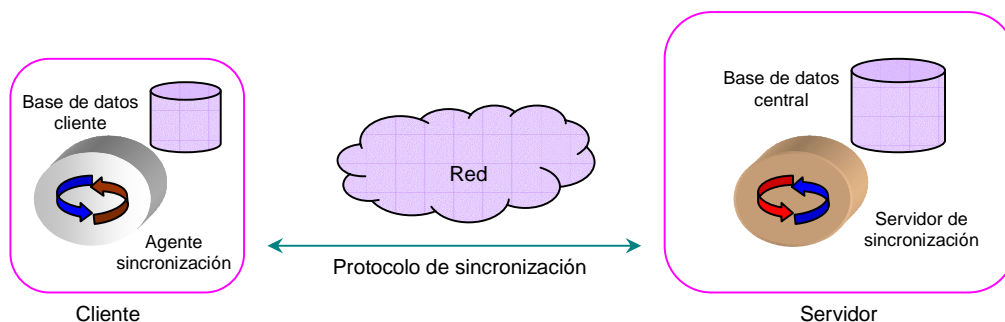
<sup>1</sup> SyncML; Building an Industry-Wide Mobile Data Synchronization Protocol. <http://www.syncml.org>. Febrero 2000.

La Figura 14 muestra un esquema simple de sincronización de datos. Después de este proceso, las dos fuentes de datos deben ser consistentes. La forma más simple de lograrlo es enviar todos los datos de un lado al otro, lo cual se conoce como una sincronización completa o lenta<sup>1</sup>. Aunque este método no es el más eficiente, no debe descartarse cuando alguna de las partes no es capaz de mantener un rastreo efectivo de los cambios realizados. Sin embargo, el proceso de sincronización es más eficiente cuando se envían solo los datos modificados; si el mismo dato fue modificado en ambas fuentes de información, ocurre un conflicto que puede ser resuelto de muchas formas, entre ellas la combinación de las dos versiones del dato o la presentación de una alerta al usuario para que éste decida el procedimiento más correcto. Este proceso se conoce como sincronización rápida<sup>1</sup>. En algunos casos, es necesario realizar algún tipo de sincronización durante intervalos de tiempo fijos debido a la necesidad de obtener información lo más actualizada posible, como es el caso de los corredores de bolsa. A este tipo de sincronización se le conoce como sincronización on-line.

Como se puede observar en la Figura 14, cada uno de los items que reside en la fuente de datos central posee un identificador conocido como GUID (Global Unique Identifier) o Identificador Unico Global y cada uno de los items que reside en las fuentes de datos móviles poseen un identificador conocido como LUID (Local Unique Identifier) o Identificador Unico Local. Durante la sincronización, se debe realizar una correspondencia entre cada LUID y cada GUID para modificar los items correctamente. Esta es una de las operaciones más importantes que deben llevarse a cabo durante el proceso de sincronización y recibe el nombre de mapeo.

## 2.2 Estructura de un sistema de sincronización

Figura 15. Estructura de un sistema de sincronización



En la Figura 15 se observa la estructura básica de un sistema de sincronización<sup>2</sup>. A continuación se describe cada uno de los componentes:

- Base de datos central: Fuente de datos principal que reside en un sistema servidor, comúnmente en un motor de base de datos (DBMS) que es accedido a través de tecnologías como JDBC u ODBC.

<sup>1</sup> D. Buchmann; SyncML and its Java Implementation sync4j. <http://diuf.unifr.ch/telecom/projects/syncml/>. Septiembre 2002

<sup>2</sup> Sybase Inc; MobiLink Synchronization User's Guide. <http://www.sybase.com>. Noviembre 2001

- Base de datos cliente: Corresponde a una copia o a un subconjunto de datos de la fuente principal que reside en la memoria del dispositivo cliente. La forma como se gestiona la información en el dispositivo, depende de las características específicas del mismo.
- Servidor de sincronización: Hace referencia a la aplicación que gestiona el proceso de sincronización es decir, realiza funciones de rastreo de cambio en los datos, mapeo de la información, resolución de conflictos y toma de decisiones acerca del tipo de sincronización requerido.
- Agente de sincronización: Corresponde a la aplicación que gestiona el proceso de sincronización en el cliente, lo cual significa que debe realizar un seguimiento de todas las modificaciones realizadas en la base de datos del cliente para ser enviadas al servidor de sincronización.
- Protocolo de sincronización: Define un flujo información que permite la comunicación entre el servidor y el agente de sincronización. El protocolo debe proporcionar métodos para identificación de registros, definición de operaciones de sincronización y posible resolución de conflictos.

En los últimos años han surgido cientos de tecnologías de sincronización propietarias entre las que se destacan RealSyncServer de Synchrologic<sup>1</sup>, ScoutSync de AetherSystems<sup>2</sup>, ActiveSync de Microsoft<sup>3</sup> y HotSync de Palm Inc<sup>4</sup>. Aunque cada tecnología posee características muy particulares, giran en torno al mismo principio. A continuación se describirá la tecnología de sincronización HotSync, puesto que juega un papel importante en la aplicación como se verá más adelante y una interesante iniciativa que busca unificar las tecnologías de sincronización en un protocolo común conocida como SyncML .

### 2.3 La tecnología de sincronización HotSync

Como se planteó en el capítulo anterior, una de las características más importantes de la plataforma Palm es la capacidad de los dispositivos para intercambiar información con aplicaciones que se ejecutan en un PC. El objetivo de esta implementación es llevar a cabo las siguientes operaciones:

- Sincronizar datos almacenados en la Handheld con datos almacenados en el PC.
- Realizar un backup o copia de seguridad de los datos de la Handheld en el PC.
- Instalar aplicaciones en la Handheld.

Las Handhelds tienen un procesador mucho más lento que la mayorías de los PCs convencionales, por lo cual es recomendable que las tareas de mayor procesamiento como la sincronización de datos se lleven a cabo en el PC. Estas tareas se llevan a cabo a través de una aplicación conocida como HotSync Manager que se instala en el PC. HotSync Manager utiliza una API llamada SyncManager que gestiona el flujo de comunicación desde y hacia la Handheld, lo cual hace a HotSync Manager independiente del tipo de conexión.

---

<sup>1</sup> Synchrologic; RealSync Server Architecture. <http://www.synchrologic.com>

<sup>2</sup> AetherSystems; Introducing the ScoutSync Client and the Scout DPA. <http://www.aethersystems.com>

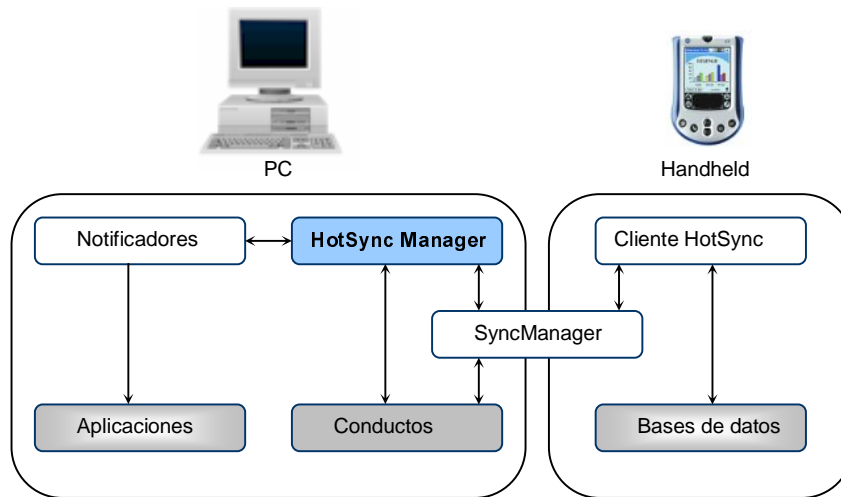
<sup>3</sup> Microsoft; ActiveSync3.7. <http://www.microsoft.com/windowsmobile/resources/downloads/pocketpc/activesync37.msp>

<sup>4</sup> A. Butinof; Introduction to Conduit Development. <http://www.palmsource.com>

### 2.3.1 Componentes de una sincronización HotSync

La Figura 16 muestra los principales componentes software que intervienen en un proceso de sincronización:

Figura 16. Componentes de una sincronización HotSync



- Aplicaciones en el PC: Son aplicaciones desarrolladas por terceros que operan sobre los datos que son enviados o recuperados de la Handheld.
- Aplicaciones Palm: Aplicaciones que se ejecutan en la Handheld.
- HotSync Manager: Aplicación que gestiona la comunicación con la Handheld.
- Cliente HotSync: Esta aplicación reside en el cliente y responde a las solicitudes que realiza HotSync Manager a través del API Sync Manager para acceder a las bases de datos del dispositivo.
- Conductos: Son aplicaciones que HotSync Manager ejecuta para interactuar con los datos de una aplicación específica en la Handheld. Un conducto solo puede interactuar con una aplicación.
- API SyncManager: Proporciona una interfaz de programación que utilizan los conductos para comunicarse con la Handheld. Esto permite que los conductos sean independientes del tipo de conexión utilizado para comunicarse con el dispositivo.
- Notificadores: Son DLLs bajo sistema operativo Windows que permiten notificar a la aplicación del PC relacionada con los datos que se están procesando, que el conducto está realizando su función. De esta manera se garantiza que los datos no son modificados por la aplicación y el conducto al mismo tiempo.

De acuerdo a lo anterior, los conductos proporcionan la lógica para el tratamiento de los datos durante la sincronización. A continuación se describen sus características más a fondo.



### 2.3.2 Conductos

En un sentido más estricto, un conducto es un módulo plug-in para HotSync Manager que transfiere un tipo específico de información entre la Handheld y el PC sin intervención del usuario, lo cual significa que los conductos carecen de interfaz gráfica o GUI. Algunas aplicaciones comparten información con programas que se ejecutan en el PC, como el caso del Date Book o libro de citas, una aplicación propia de Palm que permite la introducción de información a través del dispositivo o a través del Palm Desktop Software que reside en el PC, por lo tanto, el conducto del Date Book debe sincronizar las bases de datos en el PC y la Handheld. Otras aplicaciones simplemente requieren que se realice una copia de seguridad de los datos de la Handheld en el PC. De acuerdo a estos criterios, los conductos tienen la siguiente clasificación:

- Conducto de instalación: Instala aplicaciones y bases de datos en la memoria principal del dispositivos o en las tarjetas de expansión. HotSync Manager incluye un conducto de instalación por defecto.
- Conducto de sincronización: Realiza todas las tareas de sincronización para las que fue diseñado entre las aplicación en la Handheld y su contraparte.
- Conducto de Backup: Copia el contenido de las bases de datos de la Handheld al PC cuando así se requiera. En términos generales, los conductos de backup se utilizan cuando las aplicaciones no tienen conductos de sincronización asociados.

Las funciones de un conducto se pueden resumir en la adición, modificación o borrado de registros en el PC y la Handheld, adaptando la información a formatos específicos cuando sea requerido. Estas tareas pueden llevarse a cabo de acuerdo a las condiciones que se presenten en cada uno de los tipos de sincronización definidos por la tecnología HotSync, los cuales se describen a continuación:

- Espejo (Mirror Image): Permitir realizar modificaciones de la información en la Handheld y el PC. Su objetivo es generar dos conjuntos de datos idénticos. En este caso el conducto debe resolver posibles conflictos. De acuerdo a las condiciones, se puede llevar a cabo una sincronización lenta o rápida:

Sincronización rápida: Solo envía los registros que han sido modificados desde la última sincronización que se llevó a cabo entre la Hnadhheld y el PC.

Sincronización lenta: Todos los registros de la Handheld son enviados al PC y son comparados uno a uno con la fuente de datos correspondiente para determinar las acciones que se deben llevar a cabo. Este tipo de sincronización debe efectuarse cuando no se ha efectuado una sincronización recientemente entre las partes involucradas.

- Unidireccional (One directional): En este caso la sincronización es de una vía es decir, una de las fuentes de datos, la del PC o la Handheld se copia en la otra.

### 2.3.3 Flujo de sincronización HotSync

A continuación se hará referencia al procedimiento general de sincronización que sigue HotSync Manager. Como es lógico, debe existir una primera sincronización entre HotSync Manager y una Handheld con el objeto que intercambien su información y se reconozcan, análogo a una primera cita. Durante este primer contacto, HotSync Manager genera un par de números pseudoaleatorios; el primero se conoce como PC ID e identifica al PC, el segundo se conoce como User ID e identifica a la Handheld. Adicionalmente, cada aplicación en la Handheld tiene un identificador único llamado creator id, el cual registran los desarrolladores ante Palm Inc. para evitar posibles conflictos.

Figura 17. Uso de PC ID y User ID

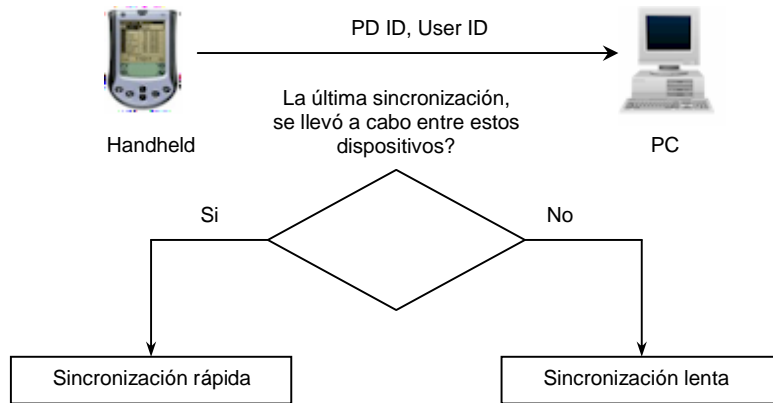
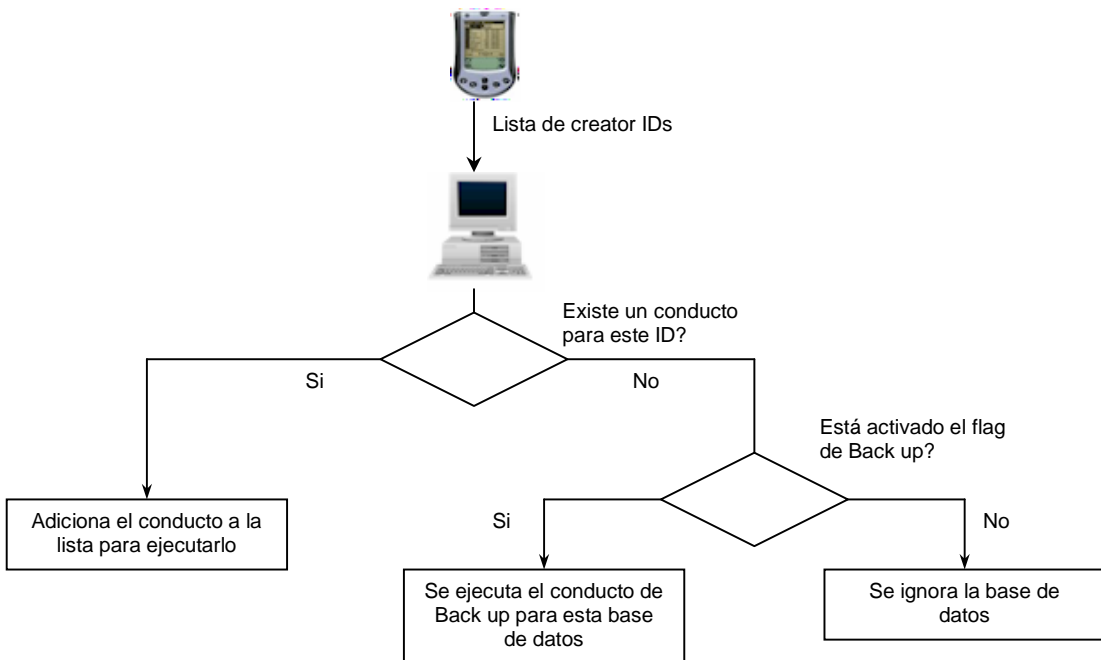


Figura 18 Selección de los conductos a ejecutar



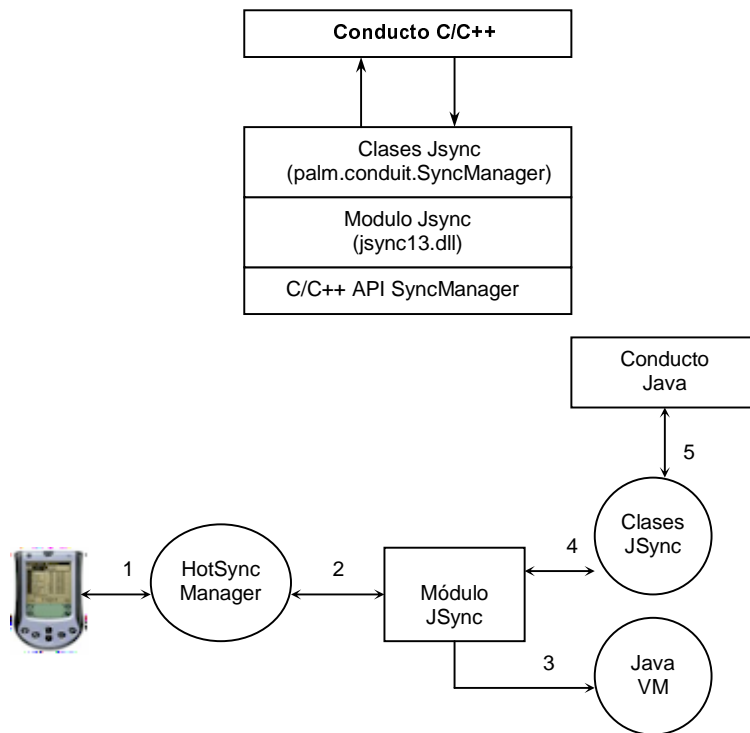
En sincronizaciones posteriores, HotSync Manager utiliza estos identificadores para decidir si es posible realizar una sincronización rápida o se debe realizar una sincronización lenta (Figura 17). Luego de completar la fase de autenticación, se sigue una serie de operaciones como se muestra en la Figura 18: Inicialmente, la Handheld envía los identificadores de cada aplicación (creator id). Para cada aplicación HotSync Manager verifica si existe un conducto, para agregarlo a la lista de los conductos que debe ejecutar. De lo contrario, verifica si la aplicación ha activado su flag de backup. En caso afirmativo, HotSync Manager ejecutará el conducto de backup para esta aplicación, de lo contrario, HotSync Manager no llevará a cabo ninguna acción para esta aplicación.

### 2.3.4 JSync Suite

Palm Inc. proporciona a los desarrolladores un kit de para desarrollo de conductos conocido como CDK (Conduit Development Kit), el cual dispone de APIs para construir los conductos en C, C++, Visual Basic y Java. JSync Suite es un conjunto de librerías y APIs que permiten el desarrollo de conductos en. A continuación se hará una breve descripción de este paquete.

### 2.3.5 Arquitectura de un conducto basado en Java

Figura 19. Estructura de funcionamiento de un conducto Java



En la Figura 19 se muestra la estructura de funcionamiento de un conducto basado en Java. El API SyncManager está basada en C y C++, por lo cual JSync Suite debe

proporcionar capas de código adicionales que permitan a los desarrolladores la utilización de SyncManager desde clases Java. El componente principal es el módulo JSync, el cual es una DLL Windows que sirve de puente entre la clase Java y el API SyncManager, haciendo una traducción en tiempo de ejecución entre los métodos Java y las funciones que residen en el API SyncManager.

En la Figura 19 se muestra el procedimiento que siguen los conductos basados en Java para comunicarse con la Handheld:

1. HotSync Manager se comunica con la Handheld a través de una conexión por puerto serie, módem, red o infrarrojo.
2. HotSync Manager se comunica con el módulo JSync para invocar los conductos basados en Java.
3. El módulo JSync carga y mantiene la máquina virtual de Java, e invoca los métodos principales del conducto (open y configure).
4. Los datos de sincronización son transportados entre el módulo JSync y las clases del paquete palm.conduit (presente en el JSync Suite) utilizando la tecnología JNI (Java Native Interface).
5. Los conductos acceden a los datos a través de las clases del paquete palm.conduit.

## **2.4 SyncML (Synchronization Markup Language)**

La necesidad de intercambiar información entre dispositivos móviles y bases de datos corporativas, causó el surgimiento de múltiples tecnologías para llevar a cabo esta tarea. Lo que fue la solución a una necesidad latente, con el tiempo hizo que la sincronización se convirtiera en el talón de Aquiles de la computación móvil debido a la diversidad de protocolos existentes, lo cual conlleva a serios problemas de interoperabilidad.

En 1999, un grupo de compañías entre las que se destacan Ericsson, IBM, Lotus, Nokia, Palm Inc., Psion, y Starfish Software fundaron la iniciativa SyncML (Synchronization Markup Language) con el objeto de desarrollar un estándar general para procesos de sincronización. La primera versión del protocolo, se publicó un año después basada en XML (Extended Markup Language). La adopción del protocolo como un estándar depende fundamentalmente de la decisión de los fabricantes de dispositivos móviles y sistemas backend de implementar SyncML en sus productos. Por otro lado, el éxito del protocolo depende no sólo de la calidad de su especificación, sino también de factores económicos y decisiones estratégicas de las compañías. Por ahora el panorama parece favorable, puesto que los modelos 6800 y 7250 de Nokia y PA800 de Ericsson, pioneros en teléfonos celulares que soportan esta tecnología, muestran la confianza y la voluntad de reconocidas compañías para lograr un ambiente de interoperabilidad.

A continuación se hará una descripción de los aspectos más importantes de este protocolo.

### **2.4.1 Ventajas de SyncML**

SyncML aparece como una interesante alternativa para unificar las tecnologías de sincronización en torno a un solo estándar, lo cual genera múltiples beneficios en todos los niveles:

- Usuarios finales: Los usuarios de hoy, utilizan diferentes procedimientos de sincronización de acuerdo al tipo de dispositivo y aplicación (calendar, email, transferencia de archivos, etc). Con SyncML, el usuario final tendrá la posibilidad de comprar dispositivos que se sincronicen con un rango más amplio de datos.
- Proveedores de servicio: Debido a la diversidad de tipos de datos y dispositivos, los proveedores de servicio se ven obligados a configurar diferentes infraestructuras para soportar los servicios. El hecho de utilizar una solución única para conectividad de datos, hace la infraestructura más fácil de mantener y menos costosa.
- Fabricantes: La preocupación actual de los fabricantes es soportar la mayor parte de las tecnologías que den a los usuarios y proveedores de servicio, un acceso eficiente a los datos. El hecho de tener un protocolo común, se traduce en dispositivos interoperables con un rango más amplio de aplicaciones, servicios y tecnologías de red.
- Desarrolladores: Soportar múltiples tecnologías, implica aplicaciones complejas, difíciles de mantener y por supuesto, costosas. Con SyncML, es posible desarrollar aplicaciones capaces de conectarse a un mayor número de dispositivos de una manera más simple y económica.

Figura 20. SyncML: Un mundo de interoperabilidad



#### 2.4.2 Características generales de SyncML

La filosofía de un protocolo de sincronización común, se basa en la capacidad para sincronizar cualquier dispositivo móvil con cualquier red de datos y viceversa. Para llevarlo a cabo, SyncML soporta las siguientes características:

- Operación efectiva sobre redes cableadas e inalámbricas.
- Soporte para varios protocolos de transporte (HTTP, WSP, Bluetooth, IrDA, TCP/IP, entre otros).
- Acceso a los datos desde varias aplicaciones
- Optimizado para dispositivos de bajos recursos.
- Construido bajo las tecnologías Web existentes (sus mensajes se representan en XML).

La representación del protocolo en mensajes XML lo hace más portable y leíble. Sin embargo, es posible que XML sea un lenguaje algo redundante en sus etiquetas, lo cual

puede ser perjudicial en el transporte de un volumen de datos considerable. La especificación del protocolo plantea la posibilidad de utilizar un formato conocido como WBXML (Wap Binary XML), el cual comprime las etiquetas de un documento XML y por consiguiente el tamaño del paquete. Este formato es utilizado en la tecnología WAP debido a las limitaciones de ancho de banda.

A continuación se muestra la estructura de un mensaje SyncML simple:

```
<SyncML>
<SyncHdr>
<VerDTD>1.1</VerDTD>
<VerProto>SyncML/1.1</VerProto>
<SessionID>1</SessionID>
<MsgID>2</MsgID>
<Target><LocURI>my_phone</LocURI></Target>
<Source>
<LocURI>http://www.myserver.com</LocURI>
</Source>
</SyncHdr>
<SyncBody>
<Sync>
<CmdID>1</CmdID>
<Target><LocURI>CalendarDB</LocURI></Target>
<Source><LocURI>samsbook.nsf</LocURI></Source>
<!--Add a new record to the CalendarDB, record ID 2021 -->
<Add>
<CmdID>4</CmdID>
<Meta><Type>text/x-vCalendar</Type></Meta>
<Item>
<Source><LocURI>2021</LocURI></Source>
<Data><!--The vCalendar data would be placed here.-->
BEGIN:VCALENDAR
VERSION:1.0
BEGIN:VEVENT
DTSTART:20010510T063000Z
DTEND:20010510T073000Z
SUMMARY:SyncML Test Checkpoint DB002021
DESCRIPTION;ENCODING=QUOTED-PRINTABLE:John =
Smith is still the presenter.=0D=0ASyncML =
is the topic.
CLASS:PUBLIC
CATEGORIES:APPOINTMENT
AALARM:20010510T061500Z
END:VEVENT
END:VCALENDAR
</Data>
</Item>
</Add>
<!--Delete a record, record ID 2022 -->
<Delete>
<CmdID>5</CmdID>
<Meta><mi:Type>text/x-vCalendar</mi:Type></Meta>
<Item>
<Source><LocURI>2022</LocURI></Source>
</Item>
</Delete>
</Sync>
</SyncBody>
</SyncML>
```

Un mensaje SyncML está compuesto de dos partes: La cabecera, identificada por la etiqueta SyncHdr y el cuerpo identificado por la etiqueta SyncBody. La cabecera SyncML especifica información general del protocolo como la versión del protocolo y la DTD que utiliza así como también información de ruteo a través de las etiquetas Target que identifica el destino y Source que identifica el origen de la información. El cuerpo SyncML

es un contenedor para uno o más comandos de sincronización que incluye información de identificación, metainformación y datos cuando se requiera. En el ejemplo anterior, a través del comando Add se adiciona un registro con id 2021 (GUID) en la base de datos CalendarDB del dispositivo destino. La información que se encuentra en la etiqueta Data es transportada en formato v-Calendar, lo cual es especificado a través de la etiqueta meta. Adicionalmente, en el mensaje se incluye el comando Delete para indicar que un registro con id 2022 (LUID) debe ser borrado del dispositivo.

### 2.4.3 Estructura de SyncML

SyncML basa su funcionamiento en dos protocolos: SyncML Representation Protocol<sup>1</sup> y SyncML Synchronization Protocol<sup>2</sup>. El primero define el formato de representación de los mensajes SyncML a través de XML y el segundo define el conjunto de operaciones que deben llevarse a cabo durante una sesión de sincronización. A continuación, observaremos los componentes más importantes involucrados en estos protocolos.

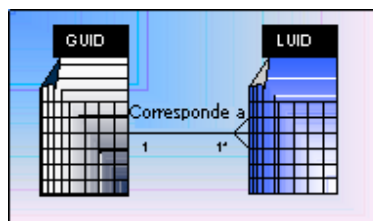
#### *Registro de cambios*

SyncML requiere que el cliente y el servidor mantengan un registro de los cambios realizados en sus bases de datos. Este mecanismo recibe el nombre de Registro de cambios (Change log). SyncML no especifica el formato del Registro de cambios sin embargo, cada dispositivo debe ser capaz de determinar qué datos han cambiado desde la última sincronización.

#### *Operación de mapeo*

SyncML parte del principio que tanto cliente como servidor poseen sus propios identificadores para los ítems que conforman sus bases de datos. El identificador del cliente se conoce como "Identificador único local (locally unique identifier) LUID" y el del servidor como "Identificador único global (globally unique identifier) GUID" como se mencionó en la introducción de este capítulo. Debido a que estos identificadores pueden ser diferentes, es necesario que el servidor disponga de una tabla de mapeo para todos los ítems intercambiados entre éste y el cliente, de tal manera que pueda conocer cual LUID y GUID apuntan al mismo ítem de datos.

Figura 21. Relación entre GUID y LUID



<sup>1</sup> SyncML Initiative; SyncML Representation Protocol. [http://www.syncml.org/docs/syncml\\_represent\\_V11\\_20020215.pdf](http://www.syncml.org/docs/syncml_represent_V11_20020215.pdf)

<sup>2</sup> SyncML Initiative; SyncML Sync Protocol. [http://www.syncml.org/docs/syncml\\_sync\\_V11\\_20020215.pdf](http://www.syncml.org/docs/syncml_sync_V11_20020215.pdf)

### *Claves de sincronización (Sync Anchors)*

Las claves de sincronización son cadenas de texto que SyncML utiliza para representar un evento de sincronización asociado a la fecha y hora de ocurrencia, lo cual es útil para realizar chequeos del proceso. Una clave de sincronización podría ser:

20010909T090909Z

Esta cadena hace referencia al sello de tiempo Octubre 9, 2001, 09:09:09 AM. Dos cadenas son enviadas siempre que se inicie una sesión de sincronización: Última clave de Sincronización (Last Sync Anchor) y Clave de sincronización actual (Next sync Anchor). La primera, representa el último evento de sincronización y la segunda el evento actual, ambas desde el punto de vista del dispositivo que las envía. De esta manera, cliente y servidor intercambian sus claves.

Si al ser comparadas estas claves coinciden, el dispositivo es capaz de determinar que no han existido fallas. De lo contrario, pueden llevarse a cabo otras operaciones como por ejemplo, una sincronización lenta. Esto brinda seguridad en el protocolo ante posibles fallas como el borrado de la memoria del dispositivo. A continuación se muestra un ejemplo donde el cliente envía sus claves de sincronización a través de la etiqueta Anchor, que a su vez especifica cada clave a través de las etiquetas Next y Last. El envío de estas claves se realiza a través de un comando Alert, cuya función es realizar peticiones o notificaciones; por ejemplo, a través de un comando Alert se puede solicitar un tipo de sincronización específico.

```
<Alert>
<CmdID>1</CmdID>
.
.
.
<Anchor xmlns='syncml:metinf'>
<Last>234</Last>
<Next>276</Next>
</Anchor>
.
.
</Alert>
```

### *Resolución de conflictos*

Un conflicto se origina cuando se modifica el mismo ítem en el cliente y en el servidor, lo cual resulta en dos versiones diferentes para el mismo dato. La resolución del conflicto está a cargo de la aplicación que implementa el protocolo de sincronización y el protocolo de representación SyncML provee funcionalidad para notificar al cliente del conflicto y definir los términos de su resolución. La administración y configuración de políticas para resolver conflictos está fuera del alcance de SyncML.

```
<Status>
<CmdID>1</CmdID>
<MsgRef>1</MsgRef>
<CmdRef>2</CmdRef>
<Cmd>Replace</Cmd>
<SourceRef>1212</SourceRef>

<Data>208</Data> <!-- Conflict, originator wins -->

</Status>
```



En el ejemplo anterior, el servidor notifica al cliente del estado de un comando Replace o de actualización de un registro específico que éste había solicitado. Los mensajes de estado se envían a través del comando Status; la etiqueta CmdRef se refiere al número único que identifica al comando Replace enviado por el cliente; la etiqueta SourceRef hace referencia al LUID enviado por el cliente para identificar el registro. Finalmente, SyncML utiliza diferentes códigos para identificar los diferentes mensajes que pueden producirse en una sesión de sincronización. Para el caso del ejemplo, el código 208 que se encuentra en la etiqueta Data, notifica al cliente que la actualización solicitada originó un conflicto que fue resuelto ignorando el valor propuesto por el servidor.

### *Seguridad*

SyncML no propone ninguna estructura de seguridad nueva, sino que proporciona mecanismos de autenticación a diferentes niveles: A nivel de servidor, a nivel de bases de datos y a nivel de objeto, de los cuales solo el primero es obligatorio, los otros son opcionales. SyncML establece que el cliente y el servidor deben soportar mecanismos de autenticación básico y MD5. A continuación se muestra un ejemplo donde el cliente incluye información de autenticación en la cabecera utilizando un mecanismo de autenticación básico. Este tipo de información se incluye en la etiqueta Cred. Dentro de la etiqueta Cred, se incluye el tipo de autenticación en la etiqueta Meta y el resultado del algoritmo en la etiqueta Data.

```
<SyncHdr>
<VerDTD>1.1</VerDTD>
<VerProto>SyncML/1.1</VerProto>
<SessionID>1</SessionID>
<MsgID>2</MsgID>
<Target><LocURI>http://www.syncml.org/sync-server</LocURI></Target>
<Source><LocURI>IMEI:493005100592800</LocURI></Source>

<Cred>
<Meta><Type xmlns='syncml:metinf'>syncml:auth-basic</Type></Meta>

<Data>QnJ1Y2UyOk9oQmVoYXZl</Data> <!-- Formato base64 de "userid:password"-->
</Cred>

</SyncHdr>
```

### *Capacidades de los dispositivos*

SyncML permite que el cliente y el servidor intercambien información acerca de sus respectivas capacidades durante una inicialización. Esta operación puede ser solicitada por el cliente o por el servidor. Si sus capacidades son compatibles se puede dar paso a una operación de sincronización. La información intercambiada es de dos tipos: información del dispositivo (tipo, número del modelo, fabricante) o información del servicio, que hace referencia al soporte para un formato de datos específico. Por ejemplo, si el cliente soporta vCalendar, el servidor debería soportar este formato para realizar sincronización.

```
<Sync>
.
.
<Meta>
<Mem xmlns='syncml:metinf'>
<FreeMem>8100</FreeMem>
<!--Memoria libre (bytes) en la base de datos del dispositivo
```

```
<Freeld>81</Freeld>  
<!--Numero de registros libres en la base de datos del dispositivo-->  
</Mem>  
</Meta>  
. .  
</Sync>
```

En el ejemplo anterior se muestra un mensaje en el cual el cliente envía información al servidor acerca de las capacidades de memoria del dispositivo; esta información se envía a través del comando Sync que contiene las operaciones de sincronización. En el ejemplo, la etiqueta FreeMem especifica la memoria libre del dispositivo y la etiqueta Freeld, el número de registros disponibles.

#### 2.4.4 Tipos de sincronización

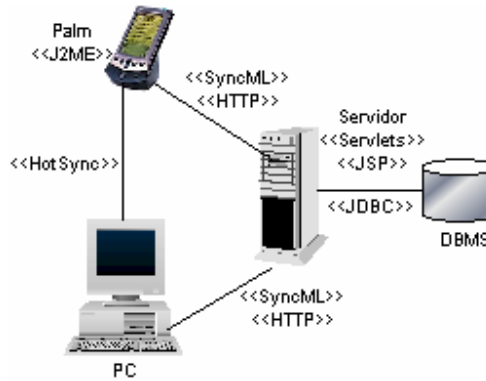
SyncML especifica siete tipos de sincronización:

- Sincronización de dos vías (Two Way Sync): Es el tipo más común, donde cliente y servidor intercambian información acerca de las modificaciones realizadas en sus bases de datos. El cliente realiza la solicitud al servidor y este procesa los datos del cliente al compararlos y unificarlos con su base de datos. Posteriormente, el servidor envía los datos que ha modificado al cliente y este actualiza su base de datos. Finalmente, el cliente envía al servidor toda la información de mapeo necesaria.
- Sincronización lenta (Slow Sync): Es un tipo de sincronización de dos vías en la cual todos los registros de la base de datos cliente son comparados con todos los registros en la base de datos del servidor. Esto puede ser necesario cuando las claves de sincronización no coinciden o el cliente o el servidor han perdido la información de su Registro de cambios.
- Sincronización de una vía desde el cliente (One Way Sync from Client Only): Es un tipo de sincronización donde el cliente envía sus modificaciones al servidor, pero el servidor no envía sus modificaciones al cliente.
- Sincronización de una vía desde el servidor (One Way Sync from Server Only): El cliente obtiene las modificaciones del servidor pero no envía las suyas.
- Sincronización de actualización desde el cliente (Refresh Sync from Client Only): El cliente exporta toda una base de datos al servidor y este reemplaza la base de datos destino con los datos del cliente.
- Sincronización de actualización desde el servidor (Refresh Sync from Server Only): El servidor exporta toda una base de datos al cliente y este debe reemplazar la base de datos destino con la información enviada.
- Sincronización iniciada por el servidor (Server Alerted Sync): El servidor informa al cliente que necesita realizar un tipo de sincronización específico.

#### 2.4.5 Arquitectura genérica de Sincronización

En la Figura 22 se muestra la arquitectura genérica para un sistema de sincronización de datos, que reúne los conceptos estudiados hasta ahora. En el lado del cliente, el dispositivo móvil considerado corresponde a una PDA con sistema operativo Palm por ser éste el tipo de dispositivo utilizado en este trabajo de grado.

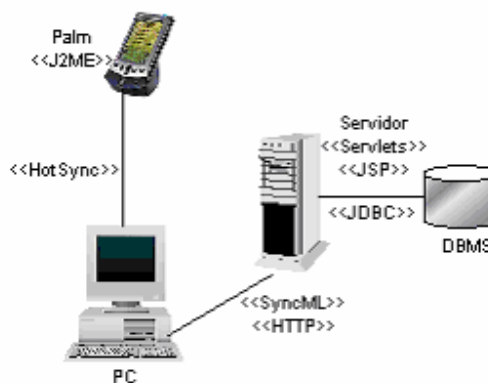
Figura 22. Arquitectura genérica de sincronización



La aplicación J2ME (Midlet) que se ejecuta en el dispositivo puede conectarse directamente al servidor a través de la red si se dispone de un módem para llevar a cabo el enlace, o puede hacerlo utilizando un PC como proxy a través de un conducto desarrollado bajo la tecnología HotSync. En el primer modelo, los datos SyncML se encapsulan dentro del paquete HTTP directamente al servidor. En el segundo modelo, los datos que van desde la PDA al PC se transportan a través del protocolo propietario HotSync y el conducto construye los paquetes SyncML que van al servidor a través de HTTP.

En el lado del servidor, se dispone de tecnologías Java como JSP y Servlets para recibir los paquetes SyncML y realizar su procesamiento. El acceso al DBMS se realiza a través del conocido estándar Java para conexión a bases de datos JDBC (Java Database Connectivity API). Para este trabajo de grado, se optó por implementar el segundo tipo de conexión, es decir utilizando HotSync como proxy ya que era más adecuado a los requerimientos del cliente como se verá más adelante. Sin embargo, el conducto que construye el protocolo fue implementado utilizando WBXML, lo cual permite que la aplicación migre más fácilmente a la arquitectura genérica en una futura versión. De acuerdo a lo anterior, la arquitectura de sincronización propuesta en este trabajo de grado, se muestra en la Figura 23.

Figura 23. Arquitectura de Sincronización Utilizada en el proyecto



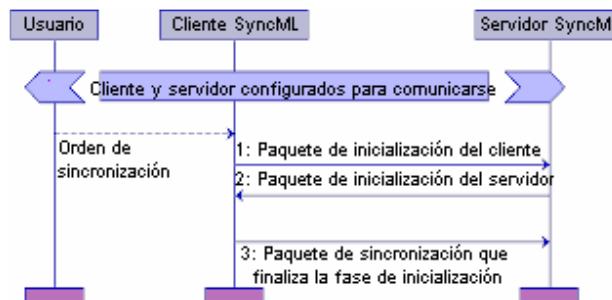
Teniendo en mente los conceptos descritos en la sección 2.4, el proceso se ha dividido en tres fases: inicialización, sincronización y mapeo.

### Inicialización

En esta fase, cliente y servidor intercambian capacidades de dispositivo y servicio, además de definir las bases de datos con las que se va a efectuar el proceso y el tipo de sincronización (Figura 24).

En el paquete 1 el cliente indica al servidor las bases de datos con las cuales se quiere sincronizar y el tipo de sincronización. Opcionalmente, el cliente puede enviar la información de autenticación en este paquete. Si no lo hace, el servidor iniciará el proceso de autenticación. Independientemente de quien inicie la autenticación, este proceso debe llevarse a cabo. El servidor chequea las bases de datos requeridas, confronta las capacidades y compara las claves de sincronización, enviando un reporte de cada base de datos en el paquete de inicialización. El cliente debe responder a los comandos enviados, pero puede hacerlo en el primer paquete de la fase de sincronización que se verá mas adelante.

Figura 24. Inicialización



A continuación, se muestran los paquetes 1 y 2 mencionados anteriormente, implementados en SyncML:

### Paquete de inicialización desde el cliente

```

<SyncML>
<SyncHdr>
<VerDTD>1.1</VerDTD>
<VerProto>SyncML/1.1</VerProto>
<SessionID>1</SessionID>
<MsgID>1</MsgID>
<Target><LocURI>http://www.syncml.org/syncserver</LocURI></Target>
<Source><LocURI>IMEI:493005100592800</LocURI></Source>

<Cred> <!-- Información de autenticación-->
<Meta><Type xmlns='syncml:metinf'>syncml:auth-basic</Type></Meta>
<Data>QnJ1Y2UyOk9oQmVoYXZl</Data> <!-- base64 formatting of "userid:password"-->
</Cred>

<Meta> <!-- en este caso, la etiqueta Meta, especifica el tamaño máximo de paquete que el cliente puede recibir-->
<MaxMsgSize xmlns='syncml:metinf'>5000</MaxMsgSize>
</Meta>
    
```

```
</SyncHdr>

<SyncBody>
<Alert>
<CmdID>1</CmdID>
<Data>200</Data> <!-- 200 = Solicita una sincronización de dos vías (TWO_WAY_SYNC) -->
<Item>
<Target><LocURI>./contacts/james_bond</LocURI></Target>
<Source><LocURI>./dev-contacts</LocURI></Source>
<Meta>
<Anchor xmlns='syncml:metinf'>
<Last>234</Last>
<Next>276</Next>
</Anchor>
</Meta>
</Item>
</Alert>
<Final/>
</SyncBody>
</SyncML>
```

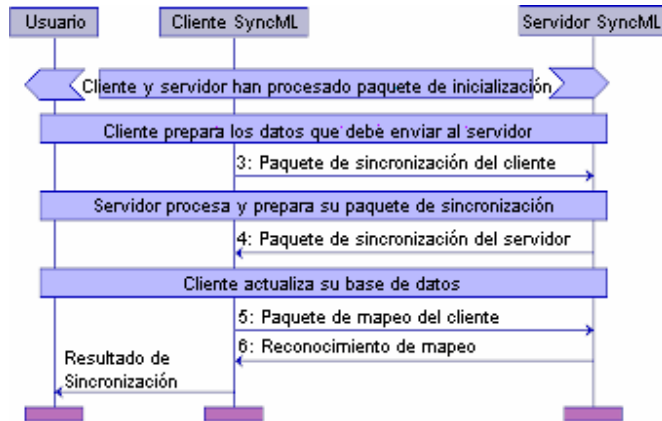
## Paquete de inicialización desde el servidor

```
<SyncML>
<SyncHdr>
<VerDTD>1.1</VerDTD>
<VerProto>SyncML/1.1</VerProto>
<SessionID>1</SessionID>
<MsgID>1</MsgID>
<Target><LocURI>IMEI:493005100592800</LocURI></Target>
<Source><LocURI>http://www.syncml.org/sync-server</LocURI></Source>
</SyncHdr>
<SyncBody>
<Status>
<CmdID>1</CmdID>
<MsgRef>1</MsgRef><CmdRef>0</CmdRef><Cmd>SyncHdr</Cmd>
<TargetRef>http://www.syncml.org/sync-server</TargetRef>
<SourceRef>IMEI:493005100592800</SourceRef>
<Data>212</Data> <!-- Código para una autenticación exitosa-->
</Status>
<Status>
<CmdID>2</CmdID>
<MsgRef>1</MsgRef><CmdRef>1</CmdRef><Cmd>Alert</Cmd>
<TargetRef>./contacts/james_bond</TargetRef>
<SourceRef>./dev-contacts</SourceRef>
<Data>200</Data> <!-- Código para indicar que se acepta una sincronización de dos vías-->
<Item>
<Data><Anchor xmlns='syncml:metinf'><Next>276</Next></Anchor></Data>
<!-- el servidor regresa al cliente la clave Next-->
</Item>
</Status>
<Final/>
</SyncBody>
</SyncML>
```

## Sincronización

El cliente utiliza la información de su Registro de cambios para extraer las modificaciones de sus bases de datos desde la última sincronización. De esta manera, construye el paquete con comandos para Adicionar(Add), Eliminar (Delete) o Actualizar (Replace) registros según sea el caso. El servidor incorpora los cambios en sus bases de datos, y sobre él, recae la responsabilidad de manejar posibles conflictos. El servidor construye y envía el paquete de sincronización en la misma forma que lo hace el cliente como se muestra en la Figura 25.

Figura 25. Sincronización y mapeo



A continuación, se muestra un paquete de sincronización simple, intercambiado entre cliente y servidor:

### Paquete de sincronización

```

<SyncML>
<SyncHdr>
<VerDTD>1.1</VerDTD>
<VerProto>SyncML/1.1</VerProto>
<SessionID>1</SessionID>
<MsgID>2</MsgID>
<Target><LocURI>http://www.syncml.org/sync-server</LocURI></Target>
<Source><LocURI>IMEI:493005100592800</LocURI></Source>
</SyncHdr>

<SyncBody>
<Sync>
<CmdID>1</CmdID>
<Target><LocURI>./contacts/james_bond</LocURI></Target>
<Source><LocURI>./dev-contacts</LocURI></Source>

<!--Comando para actualización registros -->
<Replace>
<CmdID>2</CmdID>
<Meta><Type xmlns='syncml:metinf'>text/x-vcard</Type></Meta>
<Item>
<Source><LocURI>1012</LocURI></Source>
<Data><!--Nuevos datos--></Data>
</Item>
</Replace>

<!--Comando para adición de registros -->
<Add>
<CmdID>3</CmdID>
<Meta><Type xmlns='syncml:metinf'>text/x-vcard</type></Meta>
<Item>
<Source><LocURI>1013</LocURI></Source>
<Data><!--Datos a transferir --></Data>
</Item>
</Add>

<!--Comando para eliminación de registros -->
<Delete>
<CmdID>4</CmdID>
<Meta><Type xmlns='syncml:metinf'>text/x-vcard</type></Meta>
<Item>

```

```
<Source><LocURI>1014</LocURI></Source>
</Item>
</Delete>
</Sync>
<Final/>
</SyncBody>
</SyncML>
```

## Mapeo

El cliente incorpora los cambios enviados por el servidor a sus bases de datos y construye un reporte con operaciones de mapeo para que el servidor actualice su tabla de LUIDs y GUIDs. Esto se hace a través del comando Map y las etiqueta MapItem. El cliente puede abstenerse de enviar este paquete si el servidor así lo ha solicitado; en este caso, el cliente debe guardar las operaciones de mapeo para enviarlas en la próxima operación de sincronización. Para finalizar el proceso, el servidor envía un paquete de reconocimiento al cliente, aún si las operaciones de mapeo no fueron enviadas. A continuación se muestra un paquete de estado de sincronización enviado desde el cliente, que incluye información de mapeo.

```
<SyncML>
<SyncHdr>
<VerDTD>1.1</VerDTD>
<VerProto>SyncML/1.1</VerProto>
<SessionID>1</SessionID>
<MsgID>3</MsgID>
<Target><LocURI>http://www.syncml.org/sync-server</LocURI></Target>
<Source><LocURI>IMEI:493005100592800</LocURI></Source>
</SyncHdr>
<SyncBody>
```

```
<Status>
<CmdID>1</CmdID>
<MsgRef>2</MsgRef><CmdRef>1</CmdRef><Cmd>Sync</Cmd>
<TargetRef>./dev-contacts</TargetRef>
<SourceRef>./contacts/james_bond</SourceRef>
<Data>200</Data> <!--Estado general de sincronización exitoso -->
</Status>
```

```
<Status>
<CmdID>3</CmdID>
<MsgRef>2</MsgRef><CmdRef>2</CmdRef><Cmd>Replace</Cmd>
<TargetRef>1023</TargetRef>
<Data>200</Data>
</Status>
```

```
<Status>
<CmdID>4</CmdID>
<MsgRef>2</MsgRef><CmdRef>4</CmdRef><Cmd>Delete</Cmd>
<SourceRef>10536681</SourceRef>
<Data>200</Data>
</Status>
```

```
<Status>
<CmdID>4</CmdID>
<MsgRef>2</MsgRef><CmdRef>3</CmdRef><Cmd>Add</Cmd>
<SourceRef>10536681</SourceRef>
<Data>200</Data>
</Status>
```

```
<Map>
<CmdID>5</CmdID>
```

```
<Target><LocURI>./contacts/james_bond</LocURI></Target>  
<Source><LocURI>./dev-contacts</LocURI></Source>  
<MapItem>  
<Target><LocURI>10536681</LocURI></Target>  
<Source><LocURI>1024</LocURI></Source>  
</MapItem>  
</Map>  
  
<Final/>  
</SyncBody>  
</SyncML>
```



### 3. TECNOLOGIAS INVOLUCRADAS EN EL DESARROLLO WEB

#### 3.1 Generalidades

El servicio World Wide Web (WWW) que apareció a principio de la década de los 90, se ha convertido en el medio mas común y mas utilizado para adquirir y publicar información mediante interfaces gráficas, utilizando un navegador como Netscape o Internet Explorer<sup>1</sup>.

Este medio utiliza HTML (Hipertext Markup Language) como lenguaje para formatear los documentos en páginas Web que son entregadas a los clientes por un servidor, en respuesta a una solicitud expresa en una URL (Universal Resource Locator). La comunicación entre el servidor y el navegador se realiza por medio del protocolo HTTP (Hipertext Transfer Protocol).

En sus inicios, el World Wide Web solo fue utilizado para publicación de páginas Web de naturaleza estática, es decir que los desarrolladores colocaban en ellas cierta información la cual no podía variar bajo ninguna circunstancia, a menos que los mismos desarrolladores decidieran realizar modificaciones explícitas. Esto, aunque no fue ningún impedimento para el crecimiento de la Web, sí limitaba su funcionalidad ya que las páginas venían siendo como una especie de anuncios informativos. El World Wide Web de esta forma estaba conformado solo por la unión de páginas HTML mediante hiperenlaces.

El inminente desarrollo de este campo, llevo al surgimiento de tecnologías que permitirían crear páginas Web dinámicas e interactivas. Una de las primeras tecnologías que brindó un soporte de este tipo fueron los CGIs (Common Gateway Interface), la cual permitió a los clientes acceder a muchos mas recursos dentro de la red como bases de datos, monitores de transacciones, ficheros y cualquier recurso imaginable. Los CGIs evolucionaron para dar paso a tecnologías derivadas, entre las que se encuentran los servlets, los cuales permiten el desarrollo de aplicaciones web a un nivel mas sofisticado, beneficiando finalmente a los clientes quienes disponen de aplicaciones útiles, amigables y muy fáciles de utilizar.

Este capitulo está orientado fundamentalmente a describir las tecnologías Web Servlets y JSP, de la empresa Sun Mycosystem, las cuales se convierten en el soporte para la implementación de la aplicación Web desarrollada.

---

<sup>1</sup> ACOSTA, Diego Andrés. Introducción a las Aplicaciones Web. 2000. Documento de internet introducción.pdf

## 3.2 Modelo Cliente/Servidor

Cliente/Servidor es un modelo de cooperación entre el cliente que solicita un servicio o un recurso y el servidor que ejecuta el servicio o proporciona el recurso<sup>1</sup>. La arquitectura cliente/servidor tiene una infraestructura versátil y modular que mejora la utilidad, flexibilidad, interoperabilidad y escalabilidad de los sistemas comparada con arquitecturas centralizadas como es el caso de los mainframes usados hace unas décadas.

El modelo cliente/servidor esta marcado como el mayor desarrollo en la evolución de las tecnologías de sistemas de computación, surge por el reconocimiento de las compañías de cuanto dinero podrían ahorrar proveyendo a sus usuarios de PCs en lugar de contar con un solo sistema centralizado.

Con el auge de Internet la arquitectura cliente/servidor adquirió relevancia ya que se constituyó en el principio básico del World Wide Web: un usuario que mediante un navegador (cliente) solicita un servicio (paginas HTML) a un computador que hace las veces de servidor. Actualmente se ha añadido mayor inteligencia tanto al cliente como al servidor. Los clientes han adquirido inteligencia a través de lenguajes como JavaScript y los Applets de Java mientras que los servidores lo han hecho inicialmente a través de CGIs para evolucionar hacia tecnologías populares como Servlets, ASP (Active Server Pages) y JSP (Java Server Pages).

El modelo cliente/servidor presentaba inicialmente una arquitectura en 2 niveles, en la cual la interfaz de usuario del sistema está localizada generalmente en un PC (cliente) y la administración de procesos y bases de datos se localiza en un servidor de altas prestaciones (servidor). El modelo cliente/servidor continuó su evolución hacia una arquitectura en 3 Niveles también conocida como arquitectura multinivel. Las aplicaciones Web pueden utilizar cualquiera de estas dos arquitecturas. En este proyecto se maneja la arquitectura en 3 niveles, por ser la que se ajusta al requerimiento del sistema y además por los beneficios que ofrece (Ver numeral 3.2.2).

### 3.2.1 Arquitectura en Tres Niveles

En esta arquitectura un nivel medio es adicionado entre la interfaz de usuarios (cliente) y el administrador de datos (servidor). Este nivel medio provee servicios para la administración de procesos que son compartidos por múltiples aplicaciones<sup>2</sup>. De esta forma se pueden distinguir 3 tipos de servicios: servicios de presentación, servicios de negocio y servicios de datos.

Los *servicios de presentación* generan la interfaz de usuario ofreciendo los medios para el acceso y la interacción con el sistema. Estos pueden ser por ejemplo un navegador, una plantilla electrónica o una aplicación diseñada para ello.

Los *servicios de negocio* que extienden un puente entre los servicios de presentación y los servicios de datos. Atienden las solicitudes del usuario para ejecutar una determinada función.

---

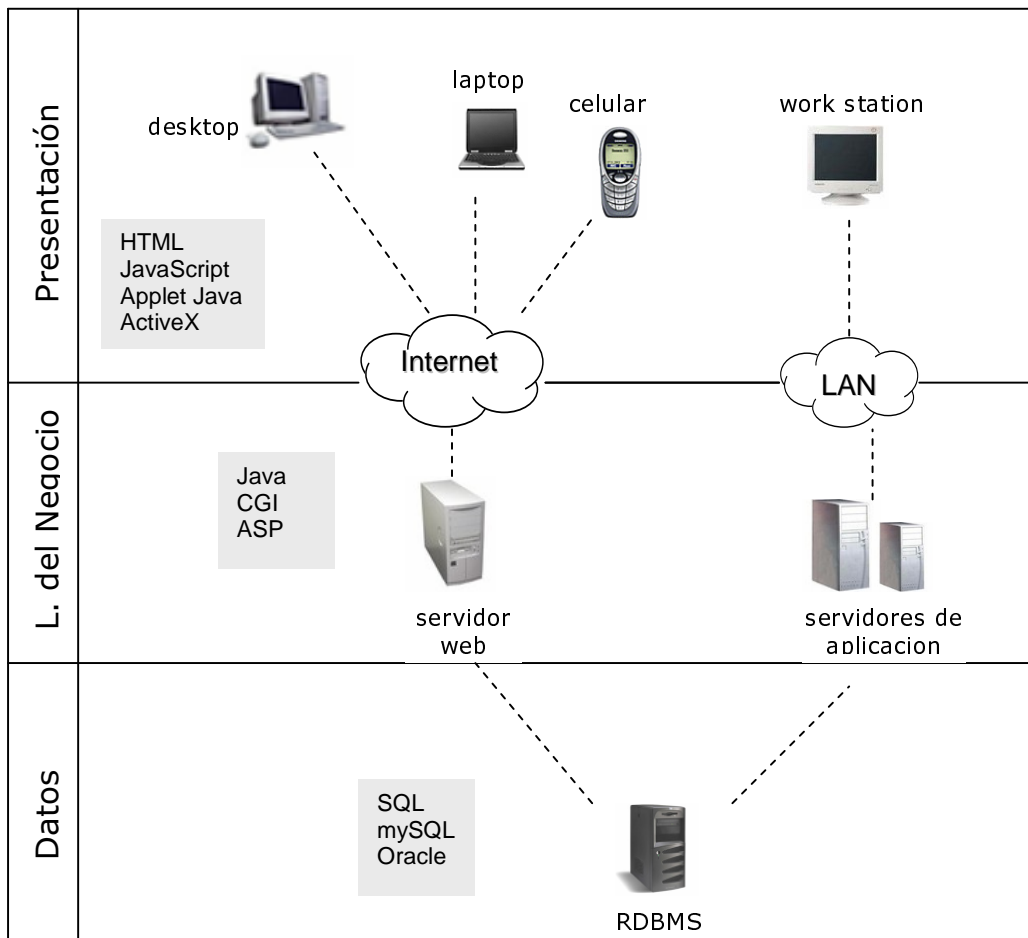
<sup>1</sup> ACOSTA, Diego Andrés. Modelo Cliente/Servidor. 1999. Documento de internet capitulo1.pdf

<sup>2</sup> Darleen Sadoski; Three Tier Software Arquitecturas <http://www.sei.cmu.edu/str/descriptions/threetier.html>

Los *servicios de datos* permiten definir, acceder y alterar los datos, atendiendo a las solicitudes de servicio de negocio. Es de destacar que los servicios de negocio y de datos pueden aparecer físicamente en la misma máquina.

La Figura 26 ilustra los lenguajes comúnmente usados en desarrollo de aplicaciones Web y su distribución según los niveles enunciados anteriormente.

Figura 26. Arquitectura Web en Tres Niveles



### 3.2.2 Beneficio de la Arquitectura en Tres Niveles

La arquitectura en tres niveles ofrece los siguientes beneficios:

- *Mantenimiento*: las reglas del negocio se centralizan en componentes específicos en los servidores y ya no están dispersas en las estaciones de los usuarios.
- *Reutilización*: la funcionalidad presente en los componentes puede ser compartida con los varios componentes del propio sistema.

- *Flexibilidad*: la carga de trabajo puede distribuirse bien entre las estaciones del usuario y los servidores con mas recursos.
- *Escalabilidad*: las aplicaciones distribuidas facilitan la expansión del sistema en escalas menores, a medida que la carga de trabajo aumenta.<sup>1</sup>

### 3.3 Servlets Java

A mediados de 1997, Javasoft saca al mercado el servidor Web en Java (Jeeves) el cual tenía dos características importantes:

- Era la primera implementación de un servidor Web basado en Java.
- Introdujo el concepto de Servlet, un nuevo plugin en Java para servidores.

Javasoft provee un Kit para el Desarrollo de Servlets Java (JSDK, Java Servlet Development Kit), que permite correr servlets en los servidores Web mas populares tales como Apache, Netscape Enterprise Server y Microsoft Internet Information Server. Este Kit incluye un conjunto de clases e interfaces que definen exactamente cómo un servidor Web interactúa con un Servlet. La ultima especificación de Sun es Java Servlet versión 2.4. El contenedor de servlets usado en la Implementación de Referencia Oficial para las tecnologías Servlet y JavaServerPages es Apache Tomcat.

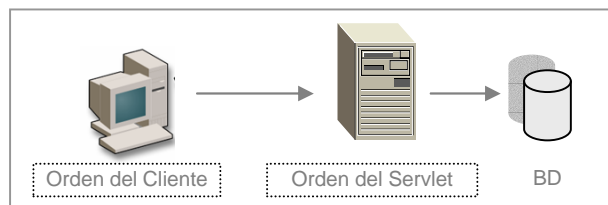
Los Servlets proveen a los desarrolladores Web de un simple pero consistente mecanismo para extender la funcionalidad de un servidor Web y permitir acceder a recursos existentes dentro del sistema de negocios u otras aplicaciones.

#### 3.3.1 Que es un Servlet?

Es una pequeña pieza de código Java que un servidor Web carga para manejar peticiones de un cliente<sup>2</sup>. Por ejemplo, un servlet podría ser responsable de tomar los datos de un formulario HTML para realización de pedidos y aplicarle la lógica de negocio necesaria para actualizar la base de datos de una compañía.

La Figura 27 ilustra el ejemplo básico descrito anteriormente donde se puede observar el entorno de trabajo de un servlet.

Figura 27. Ejemplo básico donde trabaja un Servlet



<sup>1</sup> Extraído de <http://misdb.bpa.arizona.edu/~mis696g/Reports/Yoususb/classrep.htm>

<sup>2</sup> ACOSTA, Diego Andrés. Servlets. 2000. Documento de internet Servlet.pdf

Los Servlets surgieron como respuesta de la tecnología Java a la programación CGI. Los CGIs tradicionales tenían la característica de generar procesos pesados del sistema operativo pues al llegar peticiones simultáneas sobre un programa CGI, este era cargado en memoria tantas veces como peticiones existieran. Esto perjudicaba enormemente la eficiencia de las aplicaciones construidas alrededor de los CGIs. Con los Servlets Java no se tiene este problema, la Máquina Virtual de Java permanece arrancada y cada petición es manejada por un hilo Java de peso ligero y no un pesado proceso del sistema operativo. De esta forma se pueden crear tantos hilos como peticiones existan, pero el Servlet es cargado en memoria solo una vez y permanecerá incluso aun después de que el requerimiento termine, a la espera de nuevas peticiones.

Un servlet es capaz de construir páginas Web de forma dinámica lo que resulta interesante por las siguientes razones:

- *La página Web construida está basada en datos enviados por el usuario.* Los motores de búsqueda por ejemplo generan páginas Web de esta forma al igual que los programas que procesan pedidos desde sites de comercio electrónico.
- *Los datos pueden cambiar frecuentemente.* Un informe sobre el tiempo o páginas de cabecera de noticias, podrían generar inicialmente una página previamente construida y luego actualizarle con los datos mas recientes.
- *La página Web puede usar información almacenada en bases de datos corporativas.* Se podrían generar páginas Web para una tienda on-line que liste los precios actuales y el número de artículos en inventario.

### 3.3.2 Características de los Servlets

A continuación se resumen algunas de las características más importantes de los servlets.

*Primera*, los servlets son independientes del servidor utilizado y de su sistema operativo, lo que quiere decir que a pesar de estar escritos en Java, el servidor puede estar escrito en cualquier lenguaje de programación.

*Segunda*, los servlets pueden llamar a otros servlets e incluso a métodos concretos de estos. De esta forma, se puede distribuir mas eficientemente el trabajo a realizar. Se podría tener un servlet encargado de la interacción con los clientes y que llamara a otro servlet para que a su vez se encargue de la comunicación con una base de datos. De igual forma, estos permiten redireccionar peticiones de servicio a otros servlets en la misma máquina o en una máquina remota, lo que se conoce como "Servlet Chaining"<sup>1</sup>.

*Tercera*, los servlets pueden obtener fácilmente información acerca del cliente, a través del protocolo HTTP, tal como la dirección IP, el puerto que se utiliza en la petición, el método utilizado GET o POST, entre otros.

---

<sup>1</sup> Extraído de <http://www.java.sun.com/products/servlet>

*Cuarta*, los servlets permiten la utilización de cookies y sesiones, de forma que se pueda guardar información específica acerca de un usuario determinado, personalizando de esta forma la interacción cliente-servidor.

*Quinta*, los servlets pueden actuar como enlace entre el cliente y una o varias bases de datos en arquitecturas cliente-servidor de 3 capas.

*Sexta*, los servlets por estar escritos en Java, pueden acceder a la familia entera de APIs Java, incluyendo el API JDBC para acceder a bases de datos. Así se benefician de un lenguaje maduro como lo es Java, incluyendo su portabilidad, rendimiento, reusabilidad y robustez.

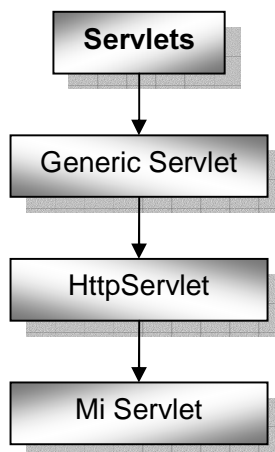
*Finalmente*, los servlets pueden realizar tareas de proxy para un applet. Debido a las restricciones de seguridad, un applet no puede acceder directamente por ejemplo a un servidor de datos localizado en cualquier máquina remota, pero el servlet sí puede hacerlo.

### 3.3.3 Arquitectura del Paquete Servlet

El JSDK (*Java Servlet Developer Kit*)<sup>1</sup> distribuido gratuitamente proporciona un conjunto de herramientas necesarias para el desarrollo de servlets. Uno de sus componentes es el API del JSDK, el cual ha sido diseñado como una extensión del JDK (*Java Development Kit*)<sup>2</sup>. Esta conformado por dos paquetes el *javax.servlet* y el *javax.servlet.http*. Todas las clases o interfaces que se pueden utilizar en la programación de servlets se encuentran en estos dos paquetes.

La abstracción central en la API Servlet, es la interfaz Servlet. Todos los servlets implementan esta interfaz ya sea directamente o extendiéndola de una clase que la implemente como HttpServlet. En la Figura 28 se puede observar la jerarquía de la interfaz Servlet.

Figura 28. Interfaz Servlet



<sup>1</sup> Extraído de <http://www.java.sun.com/products/j2sdk>

<sup>2</sup> Extraído de <http://www.java.sun.com/products/jdk>

El API Servlet es soportado por varios servidores Web entre los que se encuentran Java Web Server de Sun, Apache Web Server, Iplanet Web Sever, Internet Information Server, Iserver.

“Los Servlets implementan el interface Servlet, normalmente implementando un servlet genérico o bien una implementación específica para HTTP: La forma mas sencilla de un Servlet define un único método, el método *service()*”<sup>1</sup>.

Cuando un servlet acepta una llamada desde un cliente recibe dos objetos:

- Un *ServletRequest* que encapsula la comunicación desde el cliente al servidor. Este permite al servlet acceder a información como por ejemplo los nombres de los parámetros pasados por el cliente, el protocolo que esta siendo utilizado por el cliente, los nombres del host remoto que ha realizado la petición y del servidor que la ha recibido.
- Un *ServletResponse* que encapsula la comunicación desde el servlet hacia el cliente. Este permite seleccionar la longitud y el tipo de contenido que será enviado en la respuesta.

A continuación un ejemplo de un Servlet sencillo donde se pueden observar los objetos enunciados anteriormente.

```
import javax.servlet.*;
import javax.servlet.http;
public class SimpleServlet extends HttpServlet {
    public void service(
        HttpServletRequest request,
        HttpServletResponse response
    ) throws ServletException, IOException
    {
        ...
    }
    ...
}
```

### 3.3.4 Ciclo de Vida de un Servlet

Un servlet no es mas que una clase Java y por esto necesita correr en una máquina virtual de Java mediante un servicio llamado Servlet Engine que carga las clases la primera vez que el servlet es llamado o cuando se inicia el servidor<sup>2</sup>.

Cada servlet tiene el mismo ciclo de vida:

- Un servidor carga e inicializa el servlet
- El servlet maneja cero o mas peticiones del cliente

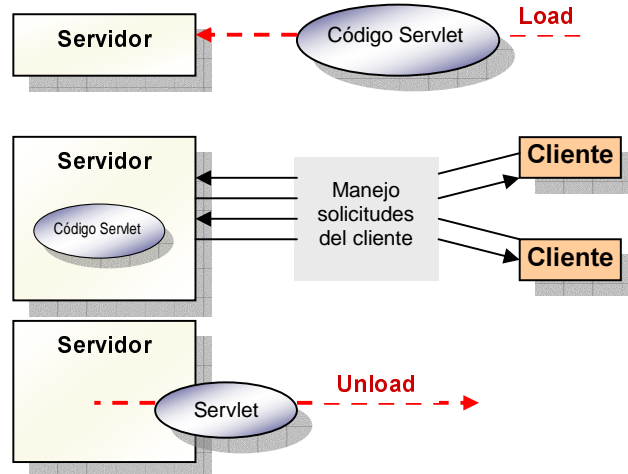
---

<sup>1</sup> HUNTER, Jason. Java Servlet Programing. Segunda edición. 2001: EU. Pag. 17

<sup>2</sup> Sun Microsystems. Especificación 2.3 de Servlet 2000. Documento de internet servlet 23.pdf

- El servidor elimina el servlet

Figura 29. Ciclo de Vida de un Servlet



Un servidor web se comunica con un servlet mediante la API Servlet en la cual se encuentran tres métodos básicos<sup>1</sup>:

*init()*, *destroy()*, *service()*

### El método *init()*

Cuando un servlet es cargado por primera vez, este método es llamado por el servidor Web y no será llamado de nuevo mientras el servlet se esté ejecutando. Esto permite la ejecución de cualquier operación de inicialización, potencialmente costosa en términos de CPU, solo una vez. Esta es una ventaja importante de los servlets frente a programas CGI. Por ejemplo, si en un día se presentan 500 consultas a una base de datos, mediante un CGI se tendría que abrir una conexión con la base de datos 500 veces, mientras que con un servlet sería necesaria una sola conexión que podría quedar abierta a la espera de recibir nuevas peticiones.

Si el servidor permite precargar los servlets, el método *init()* será llamado al iniciarse el servidor. Si el servidor no ofrece esta posibilidad, entonces será llamado la primera vez que haya una petición por parte de un cliente.

### El método *service()*

Este método es el núcleo fundamental del servlet. Una vez se inicialice el servlet, cada petición por parte de un cliente se traduce en una llamada a este método, el cual lee la solicitud y debe producir una respuesta en base a los parámetros recibidos.

---

<sup>1</sup> RODRIGUEZ, Jose y GARCIA, Javier. Aprenda Servlets de Java como si estuviera en segundo. 1999. Documento de internet Servlets.pdf



El principio de este método es sencillo, pues consiste en generar una respuesta por cada petición recibida de un cliente. Es de resaltar que las peticiones pueden ser concurrentes, por lo que pueden haber múltiples respuestas que estén siendo procesadas al mismo tiempo, ya que los servlets son multihilo.

### **El método destroy()**

Una implementación adecuada de este método permite que el servlet concluya sus tareas de manera ordenada. De esta forma, es posible liberar recursos tales como ficheros abiertos, conexiones con bases de datos, de una forma limpia y segura. Aquí se pueden liberar los resultados tomados por el servlet antes de que los procese el recolector de basura.<sup>1</sup>

Puede suceder que al llamar al método destroy() hayan peticiones de servicio que estén todavía siendo ejecutadas por el método service(), lo que podría provocar un fallo general del sistema. Por este motivo es conveniente escribir el método destroy() de forma que se retrase la liberación de recursos hasta que no hayan concluido todas las llamadas al método service().

Este método se ejecuta solo una vez y el servidor no ejecutará de nuevo el servlet, hasta haberlo cargado e inicializado de nuevo.

### **3.3.5 Servlets y JDBC**

Los sistemas de bases de datos tradicionales en el Web no eran mas que programas externos corridos por el servidor Web para acceder la base de datos y crear salidas en forma de un documento HTML. Esta es una arquitectura probada que tiene tres limitaciones:

- Dificultad para mantener el estado y la sesión de una conexión
- Cuellos de botella y bajo rendimiento
- Puede involucrar APIs propietarias

Estas limitaciones pueden convertirse en problemas significativos cuando se quiere desarrollar aplicaciones grandes basadas en Web.

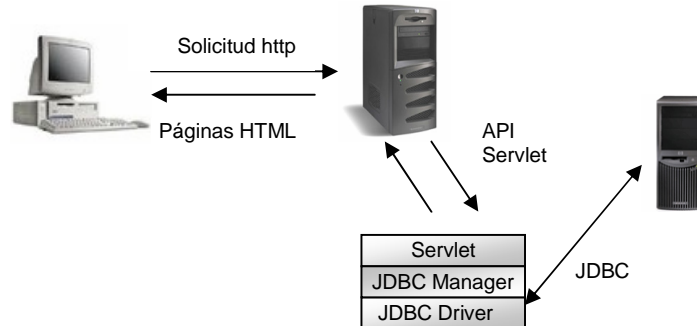
JDBC (versión Java de ODBC) es una API de Java que forma parte del estándar y esta incluida desde el JDK 1.1. Los servlets que son capaces de mantener estados, pueden usar una API estándar como JDBC.

En la Figura 30 se muestra un escenario de acceso a bases de datos con Servlet, en el que el Servlet provee la generación de HTML dinámico, comunicación con el cliente y acceso a la base de datos vía JDBC

---

<sup>1</sup> HUNTER, Jason. Java Servlet Programming. Segunda edición. 2001: EU. Pag. 43

Figura 30. Arquitectura JDBC Servlet



### 3.4 JSP (Java Server Pages)

Las primeras aplicaciones basadas en Web estaban conformadas por páginas que se enlazaban a través de hiper-enlaces para mostrar interfaces gráficas con contenido estático. Pronto los contenidos presentados en las páginas requerirían de mayor dinamismo, de tal forma que la información presentada no fuera solamente la seleccionada por los desarrolladores sino que se presentara información acorde con las solicitudes de usuario. Surgieron entonces soluciones de primera generación entre las que se encontraban los CGI, inicialmente desplegando contenido estático para evolucionar hacia la presentación de contenido dinámico en páginas Web.

Pero los CGI se vieron limitados ante el inminente desarrollo en que se encontraban las aplicaciones Web. Aparecieron las soluciones de segunda generación, algunas propietarias donde los vendedores ofrecían servidores Web junto con las APIs necesarias para desarrollo; entre ellos se encontraba Microsoft proporcionando las páginas activas del Servidor (ASP). Otras, fueron soluciones no propietarias, entre la más popular los Servlets usando tecnología Java.

Finalmente, para contribuir con estrategias innovadoras en el desarrollo de aplicaciones basadas en Web, aparece JSP como solución de tercera generación. JSP se caracteriza por ser una solución no propietaria, que puede ser combinada con soluciones de segunda generación para obtener mayores beneficios.

#### 3.4.1 Qué es JSP?

Java Server Pages es una tecnología Java en la plataforma J2EE (Java 2 Platform Enterprise Edition) para la construcción de aplicaciones con contenido dinámico como HTML, DHTML, XHTML y XML<sup>1</sup>. JSP provee una forma fácil de crear páginas Web con contenido dinámico y simplifica la tarea de construir aplicaciones basadas en Web que

---

<sup>1</sup> Sun Microsystems. Especificación 1.2 de JSP. 2000. Documento de internet jsp12.pdf

trabajan con una amplia variedad de servidores Web, servidores de aplicación, navegadores y herramientas de desarrollo<sup>1</sup>.

“En su forma básica una página JSP es simplemente una página Web HTML que contiene bits adicionales de código que ejecutan lógica de aplicación para generar contenido dinámico. Esta lógica de aplicación puede involucrar JavaBeans, objetos JDBC, Enterprise Java Beans (EJB) y objetos de Invocación Remota de Método (RMI) los cuales pueden ser fácilmente accesados desde una página JSP”<sup>2</sup>.

Por ejemplo una página JSP puede contener código HTML que despliega texto estático y gráficos, como también un método que llama a un objeto JDBC que accesa a una base de datos. Cuando la página sea desplegada en el navegador presentará tanto el contenido HTML estático como la información dinámica recuperada de la base de datos.

Esta tecnología ofrece los siguientes beneficios:

- *Sigue el paradigma “Write Once, Run Anywhere”*  
La tecnología JSP es independiente de la plataforma tanto para sus páginas Web con contenido dinámico como para los servidores Web. Se puede diseñar una página sobre cualquier plataforma, correrla sobre cualquier servidor Web o servidor de aplicación habilitado para Web y accederla desde cualquier navegador.
- *Separación de Roles*  
JSP permite separar mas claramente los roles: los *desarrolladores Web* se encargan de crear el contenido estático y el contenido dinámico de la página buscando obtener una buena presentación, los *desarrolladores de software* escriben componentes que interactúan con objetos del lado del servidor. Cada uno de estos roles requiere diferentes tipos de habilidades y aunque puedan estar presentes en el mismo individuo generalmente no es así.
- *Reuso de componentes y etiquetas*  
JSP enfatiza el uso de componentes reusables tales como JavaBeans, Enterprise JavaBeans y etiquetas. Estos pueden ser usados en herramientas interactivas para desarrollo de componentes y en la creación de páginas, permitiendo ahorrar tiempo de desarrollo. Además proporcionan robustez a la plataforma y flexibilidad al lenguaje de programación Java y otros lenguajes de scripting.
- *Separación de contenido dinámico y estático*  
JSP permite la separación de contenidos estáticos y dinámicos, simplificando significativamente la creación de contenido. Esta separación es soportada por Beans específicamente diseñados para la interacción con objetos del lado del servidor y especialmente por el mecanismo de extensión de etiquetas<sup>3</sup>. El mecanismo consiste en crear etiquetas personalizadas que en últimas extienden las etiquetas existentes para páginas JSP.

---

<sup>1</sup> Extraído de <http://developer.java.sun.com/developer/technicalArticles/index.html>

<sup>2</sup> Extraído de <http://developer.java.sun.com/developer/technicalArticles/index.html>

<sup>3</sup> Sun Microsystems. Especificación 1.2 de JSP. 2000. Documento de internet jsp12.pdf

- *Contenido dinámico puede ser servido en variedad de formatos*  
 No es obligatorio que la información dinámica en una pagina JSP tenga que seguir un formato determinado. JSP puede servir a un rango de clientes bastante amplio, que van desde los navegadores convencionales usando HTML/DHTML, dispositivos inalámbricos como teléfonos móviles y PDAs (Personal Digital Assistance) usando WML, hasta otras aplicaciones B2B (Business to Business) usando XML<sup>1</sup>.
- *Capa de acceso Web para arquitecturas de aplicación empresarial de N-hilos*  
 JSP es parte integral de J2EE, lo que permite llevar tecnología Java a ambientes de computación empresarial. Se puede desarrollar aplicaciones servidoras multi-hilo usando un sitio Web que soporta tecnología JSP como Front End para componentes Enterprise JavaBeans, en un ambiente J2EE.

### 3.4.2 Comparación JSP con ASP

Aunque las características ofrecidas por JSP pueden parecer similares a las ofrecidas por Microsoft con ASP, realmente son tecnologías diferentes. La Tabla 2 muestra la comparación entre estas dos tecnologías.

Tabla 2. Comparación JSP con ASP

Característica	Java Server Pages	Active Server Pages
Servidor Web soportado	La mayoría de servidores Web populares incluyendo Apache, Netscape y Microsoft Internet information Server (IIS) pueden ser fácilmente habilitados para JSP.	Soporte para Microsoft IIS, Personal Web Server y Apache.
Plataformas soportadas	Independiente de la plataforma. Se puede ejecutar sobre todas las plataformas habilitadas para Java	Completamente soportado bajo Windows
Modelo de Componentes	Usa componentes JavaBeans, Enterprise JavaBeans y Etiquetas personalizadas	Usa el modelo de componentes COM basado en Win32
Scripting	Puede usar el lenguaje de programación Java o JavaScript	Soporta VBScript y JScript para scripting
Seguridad	Trabaja con el modelo de seguridad Java	Puede trabajar con la arquitectura de seguridad Windows NT
Acceso a bases de datos	Usa JDBC para acceso a bases de datos	Usa Active Data Objects para acceso a bases de datos
Etiquetas personalizadas	JSP es extensible con librerías de etiquetas personalizadas	No puede usar librerías de etiquetas personalizadas y no es extensible.

<sup>1</sup> Sun Microsystems. Java Server Pages Fundamentals. Documento de internet. Jspintro.pdf

### 3.4.3 JSP o Servlets

Es cierto que los servlets y las páginas JSP tienen muchas cosas en común y pueden ser usados para servir contenido Web dinámico. Esto puede causar alguna confusión al momento de decidir cuando optar por una u otra tecnología. Los Blueprints J2EE de Sun<sup>1</sup> ofrecen una guía sobre el uso de tecnologías JSP y Servlets.

De acuerdo a los Blueprints J2EE, los servlets deben ser usados estrictamente como una tecnología de extensión del servidor Web. Esto podría incluir la implementación de componentes controladores especializados que ofrezcan servicios como autenticación, validación en bases de datos entre otros. De esta forma, JSP únicamente trata con datos textuales, por lo que se debe continuar usando los servlets para la comunicación con applets Java y aplicaciones.

Por otra parte, JSP debe ser usado para desarrollar aplicaciones Web típicas que requieran de contenido dinámico. Además podría ser usado para manejar contenido repetitivo, en lugar de usar extensiones de servidor Web propietarias.

“Es importante entender que aunque la tecnología JSP es un sucesor robusto para los servlets básicos, las dos tecnologías tienen una relación de evolución y pueden ser usadas de manera cooperativa y complementaria”<sup>2</sup>.

### 3.4.4 Arquitectura JSP

La especificación JSP está definida como una extensión estándar de la API Servlet. Por tanto no es sorprendente que en su estructura, los servlets y las páginas JSP tengan mucho en común.

Típicamente, las páginas JSP están sujetas a una *fase de traducción* y a una *fase de ejecución*<sup>3</sup>. La primera fase se lleva a cabo solamente una vez, a menos que la página JSP cambie en cuyo caso se ejecuta nuevamente esta fase; el resultado obtenido es un archivo de clase que implementa la interface Servlet y corresponde a la página JSP. El objeto de implementación de la página JSP manejará solicitudes y creará respuestas. La Figura 31 ilustra la fase de traducción.

Esta fase es llevada a cabo por el motor JSP, que no es más que un Servlet especializado, cuando este recibe una solicitud para la página JSP por primera vez. La especificación JSP 1.1 permite que las páginas JSP sean precompiladas en archivos de clase, es decir Servlets Java.

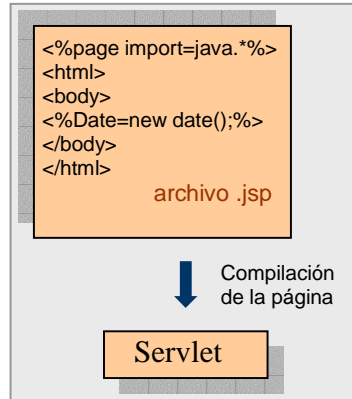
---

<sup>1</sup> <http://www.java.sun.com/j2ee/blueprints>

<sup>2</sup> MALKS, Dan. Professional JSP. 2000. Capítulo 12 “JSP Architecture”

<sup>3</sup> Sun Microsystems. Java Server Pages Fundamentals. Documento de internet. Jspintro.pdf

Figura 31. Fase de traducción de una página JSP



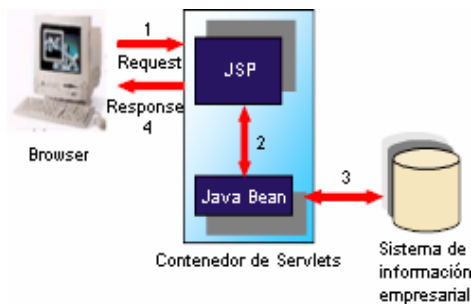
Durante la *fase de ejecución* el contenedor JSP, entidad que gestiona el ciclo de vida y da soporte en tiempo de ejecución a las páginas JSP, entrega eventos al objeto de implementación. El contenedor también será responsable de instanciar los objetos de solicitud y respuesta.

### Modelos de Acceso JSP

A continuación se muestran dos modelos que constituyen la base para la utilización de tecnología JSP.

#### Modelo 1: Página Céntrica<sup>1</sup>

Figura 32. Modelo Página Céntrica



La solicitud desde un navegador Web es enviada directamente a una página JSP, la cual es responsable de procesarla y enviar respuesta al cliente. La presentación y el contenido (datos) son separados, y el acceso a los datos es realizado usando Java Beans. Aunque es útil para aplicaciones simples no es deseable para aplicaciones complejas. El uso inadecuado de esta arquitectura usualmente repercute en una cantidad significativa de código Java embebido dentro de las páginas JSP. Aunque desde el punto de vista de los desarrolladores de software, no parece ser un problema, lo cierto es que para proyectos

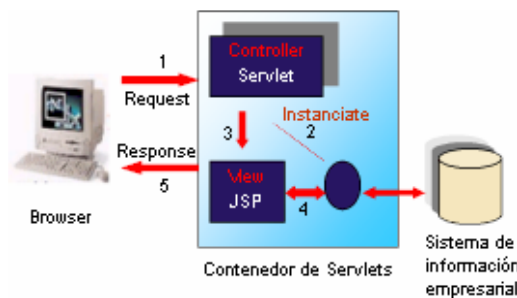
<sup>1</sup> MALKS, Dan. Professional JSP. 2000. Capítulo 12 "JSP Architecture"

grandes se generan inconvenientes cuando los diseñadores realizan mantenimiento a las páginas JSP.

Otra desventaja de esta arquitectura radica en que cada página JSP debe ser responsable de manejar el estado de la aplicación y verificar la autenticación y seguridad.

### Modelo 2: MVC (Model/View/Controller)<sup>1</sup>

Figura 33. Modelo MVC



Esta arquitectura es una implementación del lado del servidor, sigue el modelo de diseño Model/View/Controller<sup>2</sup>. El procesamiento es dividido entre los componentes de presentación y los componentes front. Los primeros son páginas JSP que generan respuesta HTML/XML que determina la interfaz de usuario a ser mostrada en el navegador Web. Los componentes front, conocidos como *controllers*, no manejan ningún tipo de presentación sino que procesan solicitudes HTTP. Los controllers son responsables de crear cualquier Bean u objeto usado por los componentes de presentación, como también decidir, dependiendo de la acción del usuario, a cual componente de presentación transmitir la solicitud. Los controllers pueden ser implementados como un servlet o una página JSP.

La ventaja de esta arquitectura radica en que no existe procesamiento dentro de los componentes de presentación. Estos son responsables simplemente de recuperar objetos o Beans que han sido previamente creados por el controller, y extraer el contenido dinámico para ser insertado dentro de la página Web. La separación de roles y responsabilidades tanto para los desarrolladores de software como para los diseñadores de páginas está garantizada.

Otra ventaja de este modelo radica en que los controllers presentan un punto único de entrada a la aplicación, facilitando así la gestión del estado de la aplicación y la seguridad.

### 3.5 Sistemas de Gestión de Bases de Datos (DBMS)

Los DBMS son el núcleo de la mayoría de sistemas cliente/servidor. Estos proveen acceso seguro a datos compartidos para las aplicaciones cliente y le permite a los

<sup>1</sup> MALKS, Dan. Professional JSP. 2000. Capitulo 12 "JSP Architecture"

<sup>2</sup> Sun Microsystems. Java Server Pages Fundamentals. Documento de internet. Jspintro.pdf

desarrolladores de sistemas cliente/ servidor dividir la carga de procesamiento entre el cliente y el servidor.

Los DBMS pueden realizar las siguientes funciones:

- Asumir alguna carga de procesamiento de aplicaciones a través del uso de procedimientos almacenados y triggers<sup>1</sup>.
- En sistemas que pueden fallar, regreso al estado anterior usando el mecanismo de recuperación por roll-back()<sup>2</sup>
- Proveer mecanismos de seguridad y control de acceso multiusuario para proteger los datos de los peligros del acceso concurrente.
- Proveer mecanismos para optimización de consultas a la base de datos.
- Proveer herramientas de monitoreo y administración de bases de datos. Las herramientas de administración incluyen cualquier software que puede restaurar una base de datos y desempeñar administración de usuarios, administración de seguridad y control del desempeño.

Aunque hay muchos modelos de bases de datos, los DBMSs relacionales aun rigen el modelo cliente/servidor. El éxito de las bases de datos relacionales se debe a la simplicidad del modelo relacional. El uso de tablas en dos dimensiones enlazadas por campos comunes es un concepto mas fácil de entender y trabajar que los que se manejan en sistemas de bases de datos jerárquicas.

Los DBMSs relacionales son la mejor forma de atacar los sistemas cliente/servidor. Estos exponen una aplicación a la menor cantidad de riesgo. Por ejemplo, la mayoría de herramientas de desarrollo cliente/servidor se comunican con la mayoría de DBMSs relacionales usando un middleware de bases de datos nativo o usando ODBC. Sin embargo muchas herramientas populares aun no permiten comunicarse con los principales DBMSs no relacionales.

### 3.5.1 Plataformas y procesadores

En muchos aspectos, la plataforma sobre la cual corre el DBMS determina el desempeño del servidor de base de datos. Los DBMSs también dependen de la capacidad del sistema operativo para proveer disco, caché y servicios de red eficientemente. Por lo tanto seleccionar una plataforma adecuada es tan importante como el mismo DBMS.

Para explotar las mejores características de Unix los fabricantes de DBMSs tales como Informix, Sybse y Oracle han dominado las capacidades de este sistema operativo para soportar numerosas conexiones cliente simultáneamente, procesos universales e hilos (procesos ligeramente pesados). También se apropiaron de la gestión de memoria y capacidades de gestión de tareas de los sistemas Unix. Todo esto al costo de tener que manejar la complejidad de estos sistemas operativos. Adicionalmente los servidores Unix, tales como IBM's RS/6000 and HP's HP-9000, tiene un costo bastante alto.

---

<sup>1</sup> <http://www.mysql.com/documentation.html>

<sup>2</sup> <http://www.dbmsmag.com/index.html>



Aunque Unix es aun un sistema operativo favorable para los DBMSs, la popularidad de los sistemas Windows y la capacidad para proveer características similares a las de Unix, ha permitido la difusión de estos con plataformas como Windows NT. Productos tales como Microsoft SQL Server pueden explotar las capacidades de Windows NT, incluyendo la capacidad para soportar muchas conexiones cliente en un hilo único y de esta forma incrementar la capacidad de los servidores para manejar una gran carga de conexión.

Para este proyecto se ha optado por utilizar un sistema de gestión de bases de datos relacionales (RDBMS) como MySQL corriendo sobre plataforma Windows. A continuación se describirán algunas características de MySQL.

### 3.5.2 MySQL

MySQL es un Sistema de Gestión de Bases de Datos Relacionales que permite almacenar datos en tablas separadas en lugar de ubicarlas en una tabla centralizada. Las tablas se enlazan a través de relaciones haciendo posible combinar en una solicitud, datos que se encuentran dispersos en tablas.

El servidor de bases de datos de MySQL es MySQL Server y se caracteriza por ser rápido, confiable y fácil de utilizar. Este servidor fue desarrollado para manejar grandes bases de datos mucho más rápido que las soluciones existentes y ha sido exitosamente usado por varios años, en ambientes de producción con niveles de demanda bastante altos. Para desarrollos que no requieren alta complejidad MySQL ofrece una amplia y útil configuración de funciones. Su conectividad, velocidad y seguridad hacen de MySQL Server un sistema de Gestión ampliamente usado para acceder a las BD en Internet.

El SW de bases de datos de MySQL es un sistema cliente/servidor que consiste de un servidor SQL multi-hilo que soporta diferentes librerías y programas cliente, herramientas administrativas y un amplio rango de interfaces de programación (APIs).

En cuanto a la escalabilidad se puede destacar que: "MySQL Server ha sido utilizado con bases de datos que contienen 50 millones de registros y se conoce de ambientes corporativos que utilizan MySQL Server con 60.000 tablas y aproximadamente 5.000.000.000 de filas"<sup>1</sup>.

Las primeras versiones de MySQL trabajaban bajo el paradigma de Operaciones Atómicas. Este paradigma nos asegura que ningún usuario puede interferir mientras una actualización específica está ejecutándose sobre una base de datos. Bajo este paradigma trabajan los tipos de tablas no transaccionales tales como MyISAM, las cuales son las tablas por defecto de MySQL. La ventaja de este tipo de tablas es que ofrecen una velocidad superior en comparación con las tablas transaccionales.

A partir de la versión 4, MySQL soporta transacciones usando el manejador de tablas transaccionales InnoDB<sup>2</sup>. Este provee a MySQL de un mecanismo transaccional seguro con rollback y capacidades de recuperación frente a fallos, además de proveer a MySQL de un estilo Oracle consistente. Para este proyecto se hizo uso de tablas transaccionales

---

<sup>1</sup> Extraído de <http://www.mysql.com/documentation.html>

<sup>2</sup> Extraído de [http://www.mysql.com/documentation/mysql/by\\_chapter/manual\\_toc.html](http://www.mysql.com/documentation/mysql/by_chapter/manual_toc.html)

las cuales por sus características permiten controlar la base de datos para evitar inconsistencias. En el paradigma transaccional, las aplicaciones están escritas de tal forma que dependen de lo que se llama rollback, es decir volver los registros al estado anterior en situaciones críticas. Las transacciones también aseguran que actualizaciones no terminadas o actividades erróneas, no sean ejecutadas en la Base de Datos. Al servidor se le da la oportunidad de hacer un rollback automático.

Las tablas transaccionales se caracterizan por mantener ante todo la integridad de la base de datos, por su durabilidad y consistencia. De esta forma la base de datos nunca es inconsistente. Los siguientes ejemplos ilustran el paradigma transaccional.

Imagine transferir \$ 100 desde una cuenta A a una cuenta B. Siempre, los \$100 estarán en una o en otra cuenta (al usar transacciones), pues en el caso de que ocurra un error actualizando una tabla, el rollback automático devuelve los \$100 a la cuenta A. Otro caso sucede cuando se quiere adicionar \$50 a los salarios de 1000 empleados. En este caso sería crítico que por algún error, solamente la mitad de los registros de empleados se hayan actualizado. Las transacciones garantizan que se efectúa la operación sobre todos los 1000 registros de empleados o no se hace la operación; en caso de que una parte de la transacción no ocurra se retorna al estado original los registros que alcanzaron a ser modificados. De esta forma una persona que accese en ese momento verá una base de datos consistente, nadie verá un empleado con \$50 más en su salario.

## 4. SISTEMA GEOS - ESPECIFICACIÓN DE REQUISITOS

### 4.1 Esencia del Sistema

Las prácticas para recolección de datos se basan fundamentalmente en la utilización de formatos escritos en papel los cuales requieren procesos posteriores de transcripción para efectos de organización y procesamiento de información. Esto genera inconsistencia en los datos recolectados debido a errores de transcripción, pérdida de datos por exposición de formatos a condiciones adversas, procesos exhaustivos de organización de información y transcripciones adicionales para introducir datos a un software estadístico. El sistema integrado de recolección y procesamiento de datos permitirá llevar a cabo la recolección de datos a través de dispositivos móviles que funcionan bajo plataforma Palm OS de tal forma que los datos recolectados puedan ser descargados a un sistema de información central.

El sistema ha sido desarrollado para el Instituto de Investigación e Información Geocientífica, Minero-Ambiental y Nuclear (INGEOMINAS)<sup>1</sup> que tiene entre sus funciones el monitoreo de volcanes en Colombia. La información manejada por esta entidad es de suma importancia por ser catalogada como información científica, por lo cual mecanismos de recolección de datos soportados en formatos escritos en papel que requieren procesos posteriores de organización y transcripción, pueden resultar críticos al momento de obtener resultados. Es por esto que el Sistema integrado de recolección y procesamiento de datos es de gran utilidad para INGEOMINAS al soportar procesos de recolección de datos en campo con PDAs y ofrecer la gestión y procesamiento de información de forma rápida y eficiente.

El sistema está enfocado de tal forma que exista un administrador y diferentes tipos de usuarios que pueden realizar funciones diversas. Para el administrador, el sistema se constituye en una herramienta de gestión y control tanto de usuarios como de información. Para los usuarios, el sistema ofrece una nueva estrategia de recolección de datos en la cual ellos no deben preocuparse por pesados procesos de transcripción y organización de información. Además la información puede ser procesada estadísticamente y si es el caso transferida desde el sistema a un paquete estadístico como Excel para ofrecer extensibilidad y mayores capacidades de procesamiento.

La información recolectada puede ser observada a través de la Web o en ambientes reducidos como por ejemplo una Intranet, en el caso de que la organización desee instalar el sistema en un servidor corporativo.

---

<sup>1</sup> <http://www.ingeominas.gov.co>

## 4.2 Descripción General del Sistema

El Sistema Integrado de Recolección y Procesamiento de Datos con Dispositivos Móviles bajo la Plataforma Palm consistirá de dos componentes principales y complementarios: un *componente Web* que permite la elaboración de formatos para recolección de datos, gestión de información recolectada e interfaz con el software estadístico Excel, y un *componente móvil* que permite descargar formatos desde el componente Web, recolectar información de manera electrónica y transferir información al componente Web; usando para ello dispositivos móviles (PDAs) bajo la plataforma Palm.

### Objetivos del Sistema

- Desarrollar una herramienta que permita generar formatos que se adapten al tipo de información que se desea recolectar.
- Permitir la recolección de datos a través de dispositivos móviles bajo la plataforma Palm.
- Garantizar que los datos recolectados a través de los dispositivos móviles se lleven de forma automática a un sistema de información central.
- Obtener resultados estadísticos a partir de información recolectada que se encuentra almacenada en el sistema.
- Soportar la transferencia de datos desde el sistema hacia un paquete estadístico como Excel.
- Soportar algún mecanismo seguro de autenticación para restringir el acceso de usuarios al sistema.
- Permitir que el sistema sea gestionado a través de un administrador.
- Restringir el acceso a la información, de acuerdo al tipo de usuario.
- Evitar que la información recolectada, catalogada como información científica, sea eliminada del sistema.

## 4.3 Glosario

**Analista:** Persona encargada del análisis de la información recolectada. Se soporta en la obtención de estadísticas para lo cual debe copiar los datos necesarios al software estadístico para que sean procesados. El analista genera informes en base a los resultados obtenidos.

**Área:** Componente organizacional que existe al interior de INGEOMINAS, en el que se realizan labores específicas de monitoreo, control y gestión. Existen para nuestro caso, 3 áreas independientes que son: geología, geoquímica y sismología. Cada área esta a cargo de uno o más individuos de la organización.

**Diseñador de formatos:** Persona encargada de la realización de formatos para recolección de datos.

**Formato:** Documento estándar escrito en papel para llevar a cabo la recolección de datos de un sitio. El formato contiene una serie de preguntas que varían de acuerdo a la información que se desea recolectar. Los formatos son propios de cada área.

**Objeto de Medida:** Se refiere al objeto del cual se obtiene información. Puede estar representado por plantas, rocas, lugares o personas.

**Recolector:** Persona que se encarga de la recolección de datos utilizando para ello formatos escritos en papel. El recolector es responsable de entregar organizada la información que haya recolectado.

**Sede Central:** Corresponde a un ente externo a la organización que está en capacidad de emitir formatos estándar para llevar a cabo recolección de datos.

**Sitio general de visita:** Corresponde a un sitio al cual se puede llevar a cabo una visita. Se tienen 3 categorías de sitios generales de visita: Volcán, Región y Falla. Estos tienen un nombre para identificar elementos de una categoría. Por ejemplo se tiene el Volcán Puracé o el Volcán Nevado del Huila.

**Sitio específico de visita:** Corresponde a un sitio que se encuentra asociado a un sitio general de visita. Se tienen 3 categorías de sitios específicos de visita: estación, campo fumarólico, fuente termal, estación sismológica y estación acelerográfica. Los sitios específicos de visita están relacionados también con un área. La Tabla 3 muestra la relación de los sitios específicos con un sitio general y un área.

Tabla 3. Relación sitio general-sitio específico-área

Sitios generales de visita	Sitios específicos de visita para cada Area		
	Geología	Geoquímica	Sismología
Volcán	Estación	Campo fumarólico Fuente termal	Estación sismológica
Región	Estación		Estación acelerográfica
Falla	Estación	Estación	

**Visita:** Corresponde a una salida de campo programada previamente para llevar a cabo recolección de datos alrededor de un sitio general de visita. Las visitas son programadas por cada una de las áreas. La periodicidad con la que se programa una visita depende específicamente de cada área.

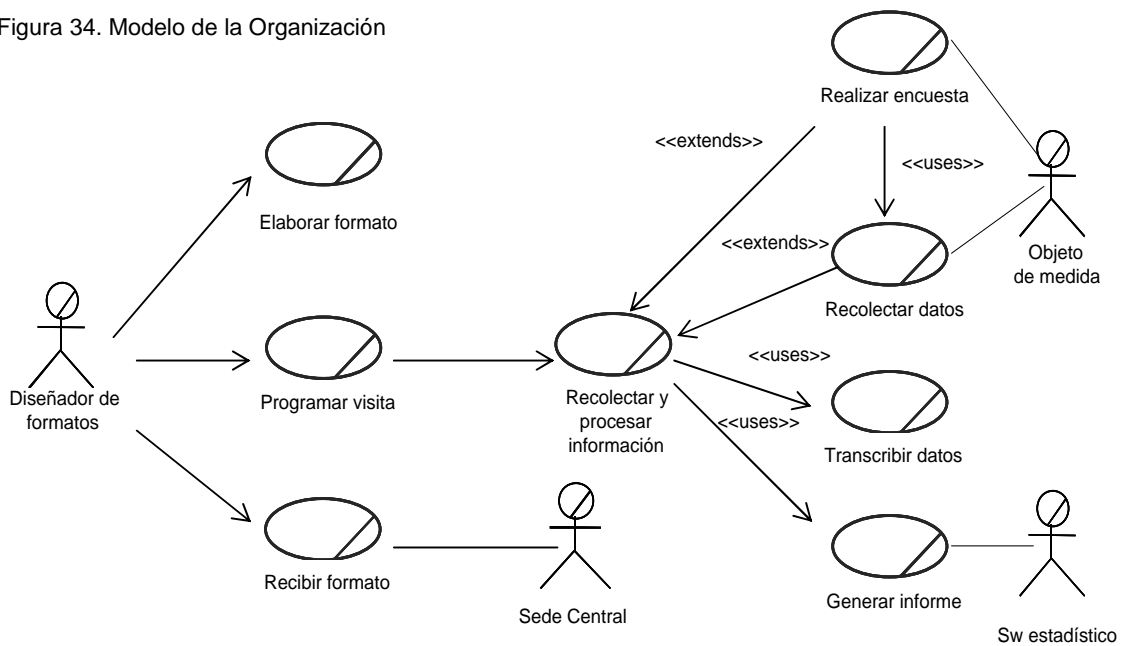
#### 4.4 Modelo de la Organización

Para cumplir sus tareas de observación, monitoreo y control de elementos instalados alrededor de la geografía, INGEOMINAS organiza sus actividades por medio de un conjunto de procesos de negocio. Cada proceso es llevado a cabo por individuos que recolectan información en campo para posteriormente organizarla y analizarla. Los procesos ocurren al interior de cada una de las áreas las cuales manejan independientemente la información. La finalidad del modelo de la organización es describir cada proceso del negocio, especificando sus datos, actividades (o tareas), roles (o agentes) y reglas de negocio.

**Modelo de Casos de Uso del Negocio:** Inicialmente se identificará los procesos de negocio, y los agentes o actores del negocio involucrados en su realización. Cada uno de estos actores del negocio desempeña cierto papel (*juega un rol*) cuando colabora con otros para llevar a cabo las actividades que conforman los casos de uso del negocio.

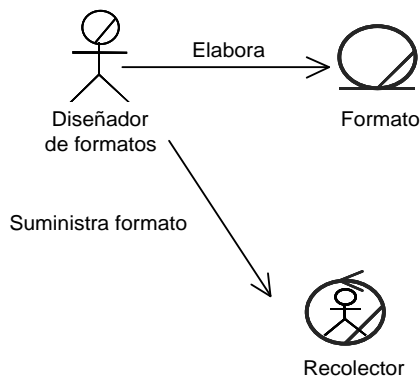
El diagrama de casos de uso del negocio, permitirá tener una visión general de los diferentes procesos de negocio de la organización los cuales aparecen como casos de uso. Los actores del negocio aparecen desempeñando funciones externas al sistema. En la Figura 34 se muestra el diagrama de casos de uso del negocio para INGEOMINAS.

Figura 34. Modelo de la Organización



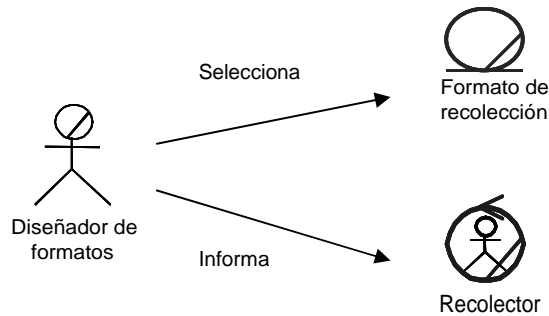
**Elaborar formato:** A través de este caso de uso, el Diseñador de formatos construye un formato para recolección de datos. El Diseñador introduce cada una de las preguntas que debe contener el formato de acuerdo a los parámetros que se desean medir en campo. Posteriormente este formato es suministrado al Recolector para llevar a cabo el proceso de recolección.

Figura 35. Caso de Uso Elaborar Formato



**Programar visita:** El diseñador de formatos selecciona los formatos que llevará para la visita. Se determina la fecha en la que se llevara a cabo y el sitio de visita. El diseñador informa a los recolectores para que lo acompañen en una determinada visita.

Figura 36. Caso de Uso Programar visita



**Recibir formato:** El diseñador de formatos tiene la posibilidad de recibir un formato de la Sede Central y lo transcribe para generar el formato de recolección a ser utilizado.

Figura 37. Caso de Uso Recibir formato

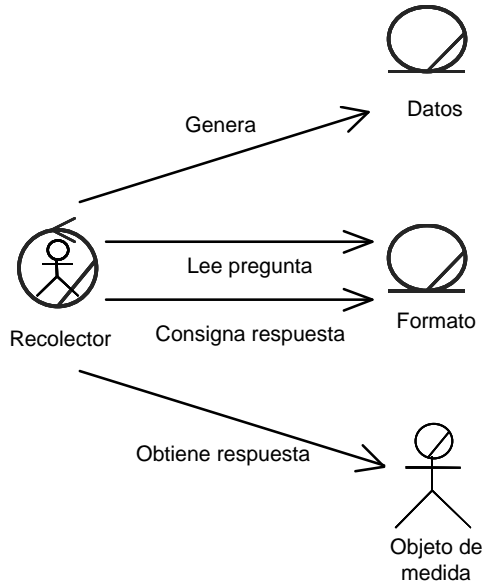


**Recolectar y Procesar Información:** Utiliza la funcionalidad descrita en los casos de uso Recolectar datos o Realizar encuesta, Transcribir datos y Generar informe que se describen a continuación.

**Recolectar datos:** En este caso de uso se describe el procedimiento de recolección de información en campo. El Recolector lee la pregunta que se encuentra en el formato de recolección, obtiene la respuesta del objeto de medida y la consigna en el formato, generando de esta manera el conjunto de datos (respuestas) correspondientes a un objeto de medida específico.

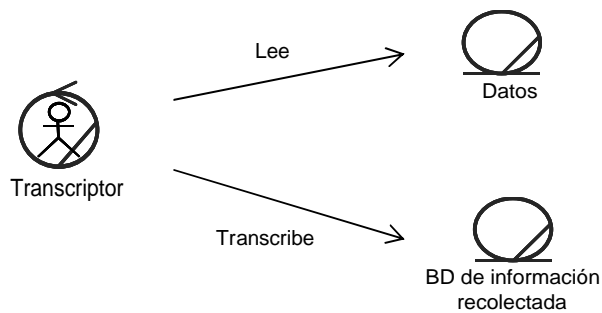
**Realizar encuesta:** Este caso de uso sigue la funcionalidad descrita en el caso de uso Recolectar datos, teniendo en cuenta que el objeto de medida corresponde específicamente a una persona.

Figura 38. Caso de uso Recolectar Datos



**Transcribir datos:** El transcriptor se encarga de leer los datos que han sido recolectados y realizar un proceso de copiado de los mismos pero ahora de forma organizada. Una vez transcriba la totalidad de los datos procede a almacenar la información organizada en la base de datos de información de recolectada.

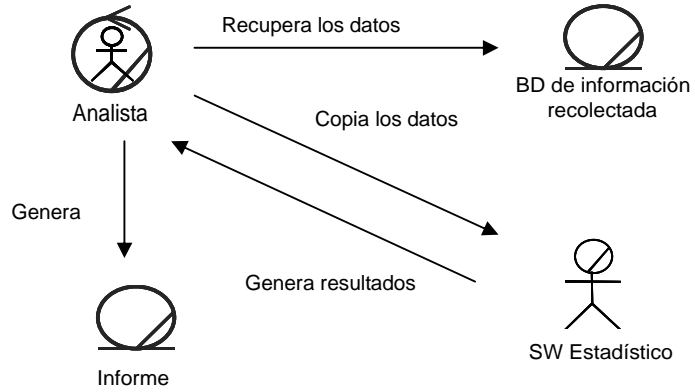
Figura 39. Caso de Uso Transcribir datos



**Generar informe:** El analista recupera los datos de la base de datos de información recolectada y lleva a cabo un proceso de organización con el fin de adaptarlos a un formato requerido por el software estadístico donde serán procesados. El analista introduce los datos al software estadístico, el cual genera los resultados correspondientes. El analista genera un informe con base en el análisis de los resultados estadísticos.



Figura 40. Caso de Uso Generar informe



A continuación se muestra el diagrama general del modelo de la organización y el diagrama general de objetos de la organización:

Figura 41. Diagrama General del Modelo de la Organización

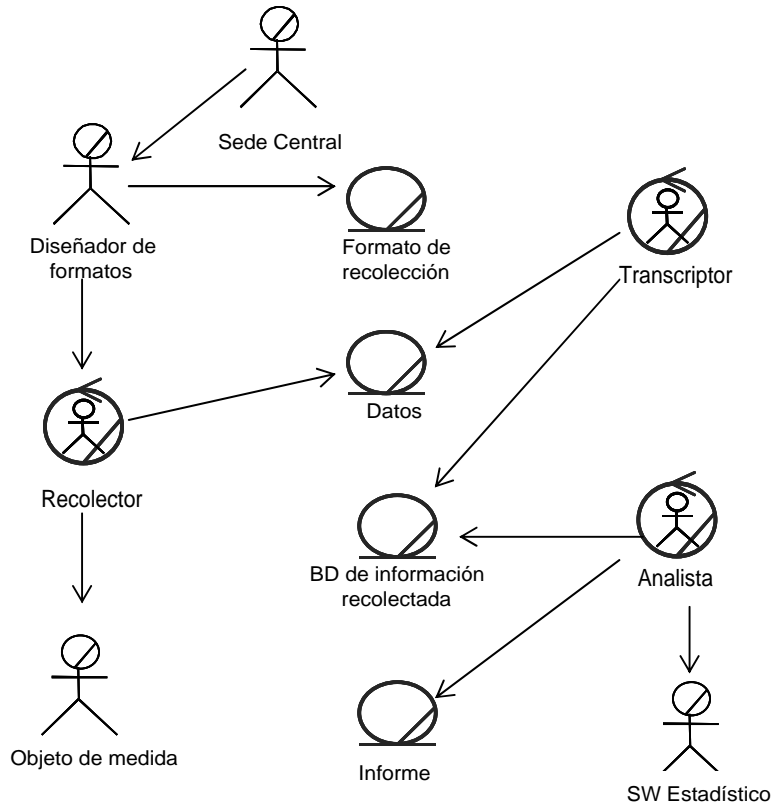
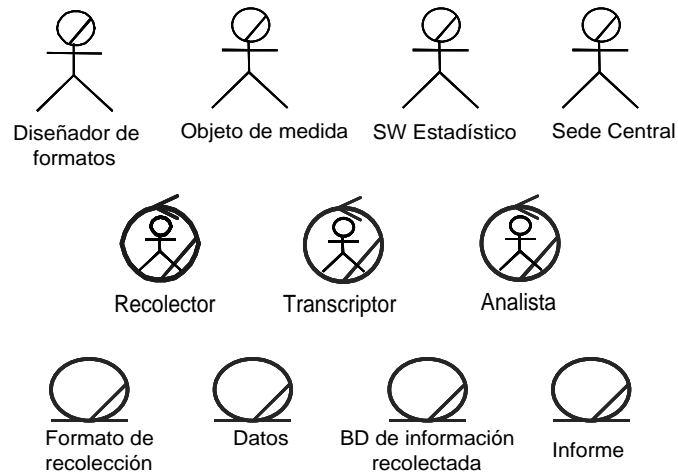


Figura 42. Diagrama General de Objetos de la Organización



#### 4.5 Arbol de funciones

##### 1. Usuario

- 1.1 Acceder al sistema
- 1.2 Cambiar password
- 1.3 Consultar información

##### 2. Diseñador de formatos

- 2.1 Elaborar formato
  - 2.1.1 Elaborar formato tipo Encuesta
  - 2.1.2 Elaborar formato Simple
- 2.2 Consultar formato
  - 2.2.1 Consultar formatos del Area de Geología
  - 2.2.2 Consultar formatos del Area de Geoquímica
  - 2.2.3 Consultar formatos del Area de Sismología
- 2.3 Modificar formato
- 2.4 Definir visita
  - 2.4.1 Definir visita en el Area de Geología
  - 2.4.2 Definir visita en el Area de Geoquímica
  - 2.4.3 Definir visita en el Area de Sismología
- 2.5 Consultar visitas pendientes
- 2.6 Terminar visita pendiente

##### 3. Recolector

- 3.1 Seleccionar visita a descargar
- 3.2 Sincronizar datos
  - 3.2.1 Enviar información al dispositivo móvil
  - 3.2.2 Obtener información del dispositivo móvil
- 3.3 Recolectar datos
  - 3.3.1 Iniciar nuevo proceso de recolección
  - 3.3.2 Interrumpir proceso de recolección

- 3.3.3** Reanudar proceso de recolección
- 3.3.4** Consultar información recolectada
- 3.3.5** Modificar información recolectada
- 3.3.6** Eliminar información recolectada

#### **4. Administrador**

##### **4.1** Gestión de usuarios

- 4.1.1** Agregar usuario
- 4.1.2** Consultar usuario
- 4.1.3** Modificar usuario
- 4.1.4** Eliminar usuario

##### **4.2** Gestión de sitios

- 4.2.1** Agregar sitio general de visita
- 4.2.2** Consultar sitio general de visita
- 4.2.3** Modificar sitio general de visita
- 4.2.4** Eliminar sitio general de visita
- 4.2.5** Agregar sitio específico de visita
- 4.2.6** Consultar sitio específico de visita
- 4.2.7** Modificar sitio específico de visita
- 4.2.8** Eliminar sitio específico de visita

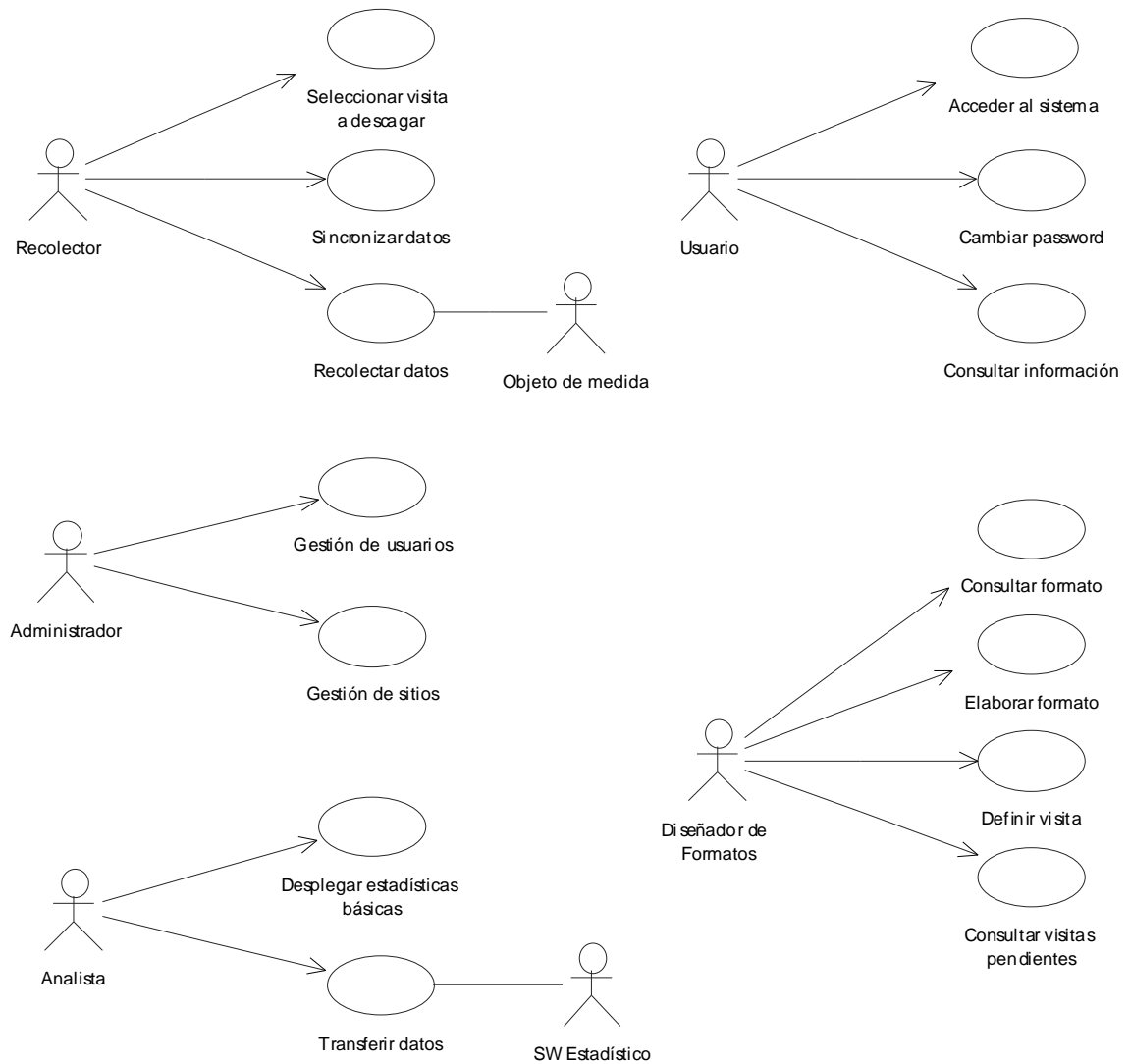
#### **5. Analista**

- 5.1** Desplegar estadísticas básicas
- 5.2** Transferir datos a software estadístico
- 5.3** Eliminar información

## 4.6 Casos de Uso de Alto Nivel

### 4.6.1 Diagrama de Casos de Uso de Alto Nivel

Figura 43. Diagrama de Casos de Uso de Alto Nivel



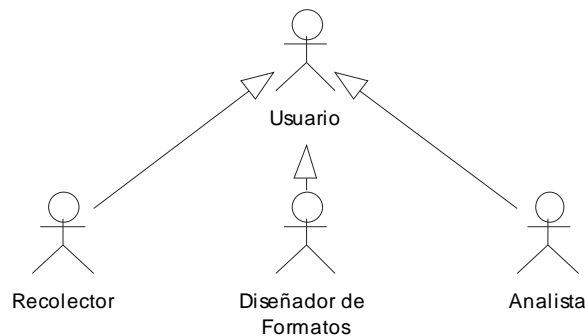
### 4.6.2 Descripción de los Casos de Uso de Alto Nivel

A continuación se describe el rol de cada uno de los usuarios que hacen parte del sistema:

### **Usuario**

En este grupo se encuentran el diseñador de formatos, el recolector y el analista es decir, aquellos usuarios que no desempeñan funciones de gestión en el sistema. Estos usuarios son adicionados al sistema por el administrador, quien les asigna un login y password de acceso. Estos usuarios tienen la capacidad de consultar información correspondiente a las visitas que se han realizado en campo para cualquiera de las áreas. A continuación se describen los roles específicos de cada uno de estos usuarios.

Figura 44. Usuario



### **Diseñador de Formatos**

Este usuario es el encargado de elaborar los formatos de recolección de datos y programar las visitas que deben llevarse a cabo en campo, lo cual implica la definición de la fecha de realización de la misma y la selección de los formatos. Este usuario solo puede llevar a cabo estas tareas en el área a la cual pertenece.

### **Recolector**

Este usuario se encarga de llevar a cabo la recolección de información en campo, utilizando los formatos establecidos para la visita por parte del Diseñador de Formatos. El Recolector confirma su participación en una visita particular al descargar los formatos correspondientes en el dispositivo móvil y es su responsabilidad la descarga de información desde el dispositivo a la base de datos central.

### **Analista**

La función principal del analista es el procesamiento de la información que se ha recolectado en campo. Para llevarlo a cabo, el analista hace uso de las funciones estadísticas de la aplicación o realiza la transferencia de la información a Excel, con el objeto de obtener conclusiones a partir de los resultados mostrados.

### **Administrador**

El Administrador se encarga de las funciones de gestión del sistema, lo cual implica tareas de gestión de usuarios como la adición, modificación o eliminación de los mismos, así como también tareas de gestión de los sitios de visita como la adición, modificación o eliminación de estos para cada una de las áreas.

A continuación se realiza una breve descripción de los casos de uso de alto nivel:

### **C.U. Elaborar formato**

**Actor:** Diseñador de formatos

**Tipo:** Primario

**Descripción:**

Este caso de uso permite la elaboración de formatos para recolección de datos. Se dispone de varias opciones que permiten adicionar diferentes tipos de preguntas, modificar o eliminar las preguntas que se han adicionado, guardar parcialmente el formato como lo hacen los procesadores de texto y adicionar datos generales del objeto de medida. Adicionalmente, se define el sitio para el cual se elabora el formato de acuerdo al área a la que pertenezca el Diseñador de formatos.

### **C.U. Consultar formato**

**Actor:** Diseñador de formatos

**Tipo:** Primario

**Descripción:**

Para usuarios que tengan el perfil de diseñador de formatos, el sistema permite consultar para cada área los formatos que hayan sido creados. Los formatos contienen un título, el área al cual pertenece, el sitio general de visita y el sitio específico para los cuales fue creado, el diseñador que lo elaboró, en algunos casos una descripción general y las diferentes preguntas.

### **C.U. Definir visita**

**Actor:** Diseñador de formatos

**Tipo:** Primario

**Descripción:**

El sistema funciona en base a la creación de visitas las cuales se programan para llevar a cabo la recolección de datos. Un diseñador de formatos puede programar visitas solo en el área a la cual pertenezca. Para definir una visita se requiere de un título, una fecha de realización, el sitio general que será visitado, los formatos y opcionalmente los sitios específicos que serán visitados. Un formato solo puede ser seleccionado para una visita siempre y cuando pertenezca a la misma área para la cual se está definiendo la visita.

### **C.U. Consultar visitas pendientes**

**Actor:** Diseñador de formatos

**Tipo:** Primario

**Descripción:**

El diseñador de formatos puede consultar las visitas para las cuales no se haya terminado la recolección de datos. Seleccionando una visita pendiente, el diseñador puede observar los formatos que contiene la visita y los recolectores que han descargado información. Desde este caso de uso el diseñador puede terminar la visita cuando considere que toda la información ya ha sido recolectada.

### **C.U. Seleccionar visita a descargar**

**Actor:** Recolector

**Tipo:** Primario

**Descripción:**

Este caso de uso permite al Recolector seleccionar la visita en la cual va a participar. Con este fin, el Recolector consulta la información correspondiente a la visita la cual incluye la fecha de realización y los formatos seleccionados. Una vez que el Recolector se encuentra seguro de la elección de la visita, lo notifica al sistema para descargar los formatos al dispositivo móvil.

### **C.U. Recolectar datos**

**Actor:** Recolector

**Tipo:** Primario

**Descripción:**

Este caso de uso agrupa todas las funciones involucradas en la recolección de datos en campo. A través del dispositivo móvil el Recolector selecciona un formato de la lista e inicia un nuevo proceso de recolección para el mismo o consulta la información que se ha recolectado. Adicionalmente, el Recolector puede modificar la respuesta a una pregunta específica o eliminar los datos correspondientes a un objeto de medida cuando lo considere conveniente. Si el formato es muy extenso, el Recolector tiene la posibilidad de interrumpir el proceso de recolección para el objeto de medida actual y reanudarlo más adelante.

### **C.U. Sincronizar datos**

**Actor:** Recolector

**Tipo:** Primario

**Descripción:**

Este caso de uso permite al Recolector descargar los formatos de una visita al dispositivo o enviar los datos recolectados en campo a la base de datos central. El Recolector puede enviar información hacia el servidor central cuando lo desee y solo notificará que el proceso de recolección ha finalizado cuando se encuentre seguro de haber descargado la totalidad de los datos. El dispositivo móvil rastrea todos los cambios ocurridos desde la última sincronización exitosa para enviarlos adecuadamente cuando el recolector lo decida. Adicionalmente, el sistema detecta posibles fraudes denegando el acceso o fallas asociadas al borrado de la memoria del dispositivo, caso en el cual descarga nuevamente los formatos para que el Recolector pueda continuar su labor.

### **C.U. Desplegar estadísticas básicas**

**Actor:** Analista

**Tipo:** Primario

**Descripción:**

Este caso de uso procesa la información recolectada para un determinado formato con el fin de obtener resultados estadísticos. El analista puede obtener estadísticas para datos recolectados a través de un formato llevado durante una visita. Las estadísticas incluyen generación de tabla resumen y gráfico de barras correspondiente. Además se pueden obtener estadísticas parciales de información utilizando los datos que hayan sido recolectados. El analista puede eliminar los datos recolectados para el formato durante la visita.

### **C.U. Transferir datos**

**Actor:** Analista

**Tipo:** Primario

**Descripción:**

Este caso de uso permite transferir información histórica que se encuentra almacenada en el sistema, a un software estadístico como Excel. Se busca obtener mayores capacidades de procesamiento de información.

### **C.U. Gestión de usuarios**

**Actor:** Administrador

**Tipo:** Primario

**Descripción:**

El administrador es quien ingresa al sistema los nuevos usuarios, quienes tienen alguno de los siguientes perfiles: diseñador de formatos, recolector, analista. Un usuario podría reunir todos los perfiles o combinación de ellos. Para cada nuevo usuario se deben almacenar datos personales tales como nombre, apellidos, login, password, teléfono, dirección, perfil de usuario, email y un área al cual pertenece. El administrador es el encargado además de consultar, modificar y eliminar un usuario.

### **C.U. Gestión de sitios**

**Actor:** Administrador

**Tipo:** Primario

**Descripción:**

El administrador es el encargado de adicionar sitios generales de visita para los cuales se debe almacenar un nombre y un tipo que puede ser volcán, región o falla. Para cada sitio general adicionado, se pueden adicionar sitios específicos de visita. Además el administrador puede modificar y eliminar tanto sitios generales como sitios específicos de visita.

### **C.U. Acceder al sistema**

**Actor:** Usuario

**Tipo:** Primario

**Descripción:**

Para que el usuario pueda acceder al sistema y hacer uso de cada una de sus funciones debe introducir un login y password. Esta información es validada de acuerdo con los datos que el administrador haya introducido en la base de datos durante el proceso de registro de usuario.

### **C.U. Cambiar password**

**Actor:** Usuario

**Tipo:** Primario

**Descripción:**

Para usuarios que hayan accedido al sistema, se brinda la posibilidad de cambiar el password de usuario.

### **C.U. Consultar información**

**Actor:** Usuario

**Tipo:** Primario

**Descripción:**

El usuario puede consultar información recolectada para una visita, seleccionando el área al cual pertenece la visita y el sitio general. El usuario elige el año para el cual desea consultar visitas y selecciona una de ellas para acceder a información de la misma. La información consultada se presenta en tablas resumen o como formatos asociados con sus respectivas respuestas.



## 5. SISTEMA GEOS - ANÁLISIS DEL SOFTWARE

### 5.1 Casos de Uso Extendidos

#### Definiciones

**BD:** Base de datos

**Datos Coherentes:** Son aquellos donde los caracteres que los componen obedecen a una lógica de validación específica, por ejemplo que contenga solo números o se represente en un formato específico como las fechas (mm/dd/aaaa).

**Formato Simple:** Corresponde a un tipo de formato que no excede de 5 preguntas. En la Tabla 4 se muestra la descripción de los tipos de pregunta que componen este formato.

Tabla 4. Tipos de pregunta que conforman un Formato Simple

Pregunta	Descripción	Ejemplo
Lista	Se dispone de una lista de opciones	Temperatura: <input type="text" value="25"/> ↓ 50
Lista y Texto	Se dispone de un campo para llenar y una lista de opciones	Dirección: __15__ <input type="text" value="NE"/> ↓ NW
Texto	Se dispone de un campo para llenar	Ph: _____ 4 _____

**Sitio General de Visita:** Hace referencia a una región geográfica extensa donde se realiza recolección periódica de datos. Los sitios generales se han clasificado en tres categorías: Volcán, Región y Falla.

**Sitio Específico de Visita:** Hace referencia a un punto geográfico específico dentro de un sitio general de visita.

**Proceso de Recolección:** Hace referencia a cada una de las ocasiones en las cuales se toman datos con un formato específico para un objeto de medida en particular.

**Sesión:** Hace referencia al estado de las peticiones realizadas por un usuario durante un periodo de tiempo.

**Formato Tipo Encuesta:** Corresponde a un tipo de formato que no excede de 30 preguntas, sin tener en cuenta los datos generales que se toman del objeto de medida tales como nombre, dirección, teléfono, etc. En la Tabla 5 se muestra la descripción de los tipos de pregunta que componen este formato.

Tabla 5. Tipos de pregunta que conforman un Formato Tipo Encuesta

Pregunta	Descripción	Ejemplo
Selección Múltiple Respuesta Unica	Se dispone de una lista de opciones, de las cuales se puede seleccionar solo una.	Sintió el sismo?: <input type="radio"/> Sí ↓ <input type="radio"/> No
Selección Múltiple Respuesta Múltiple	Se dispone de una lista de opciones, y es posible elegir mas de una	Lugares visitados: <input type="checkbox"/> París ↓ <input checked="" type="checkbox"/> Madrid <input checked="" type="checkbox"/> Bogotá
Combinada	Se dispone de una lista de opciones y una o más opciones pueden contener una subpregunta	Va a cine?: <input type="radio"/> Sí ↓ <input type="radio"/> No Por qué? _____
Contenido	Se dispone de un campo para llenar	Observaciones: _____

**Encuesta:** Formato Tipo Encuesta utilizado en el área de Sismología para recolectar datos a personas.

En la Tabla 6 se resumen los casos de uso principales del sistema, agrupados de acuerdo a su funcionalidad:

Tabla 6. Casos de Uso Principales

Función	Caso de uso	Usuario
Gestión Sesión	Acceder al sistema	Usuario – Administrador
	Cambiar password	Usuario – Administrador
	Salir del sistema	Usuario – Administrador
Gestión Usuarios	Agregar usuario	Administrador
Gestión Sitios	Agregar sitios de visita	Administrador
	Consultar sitios de visita	Administrador
	Modificar sitios de visita	Administrador
	Eliminar sitios de visita	Administrador
Gestión Formato	Elaborar formato	Diseñador de formatos
Gestión Visita	Definir visita	Diseñador de formatos
	Consultar visitas pendientes	Diseñador de formatos
	Consultar información	Usuario
Estadísticas	Desplegar estadísticas básicas	Analista
	Transferir información	Analista
Recolección de datos	Seleccionar visita a descargar	Recolector
	Recolectar información en campo	Recolector
	Consultar información en campo	Recolector
	Modificar información en campo	Recolector
	Eliminar información en campo	Recolector
Sincronización	Sincronizar datos	Recolector

## Gestión Sesión

### Información general

Caso de Uso:	<b>Acceder al sistema-usuario</b>
Actores:	Usuario (iniciador)
Propósito:	Permitir el acceso de un usuario al sistema a través de un login y password.
Resumen:	El usuario introduce su login y password para acceder al sistema. Los datos introducidos son verificados de acuerdo con los datos almacenados en la base de datos. El sistema recupera el perfil de usuario (diseñador de formatos, recolector, analista) y despliega un menú de usuario personalizado de acuerdo al perfil que haya recuperado.
Tipo:	Primario
Referencias Cruzadas:	Funciones: 1.1

### Precondiciones

- El usuario debe estar registrado en el sistema, acción que es realizada por el Administrador.
- El sistema debe contar con la siguiente información para poder iniciar una sesión: login de usuario, password, área a la que pertenece (Geología, Geoquímica, Sismología) y perfil de usuario.

### Flujo Principal

- Este caso de uso empieza cuando el usuario elige utilizar el sistema.
- El sistema presenta al usuario la interfaz Acceso Usuario, que le solicita un login y password.
- El usuario ingresa su login y password y pulsa Entrar.
- El sistema verifica el login y password (E1).
- El sistema recupera el perfil de usuario y de acuerdo a este escoge el contexto del menú de usuario que será desplegado. Las figuras posteriores ilustran los menús para cada uno de los perfiles de usuario.
- El sistema inicia una sesión de usuario.

### Flujos de Excepción

E1: El sistema despliega un mensaje informando que el login o password de usuario no es correcto.

Figura 45. Interfaz Acceso Usuario



Figura 46. Menús de Usuario





Menú Diseñador de Formatos

### Información general

Caso de Uso:	<b>Cambiar password</b>
Actores:	Usuario (iniciador)
Propósito:	Permitir que un usuario pueda cambiar su password.
Resumen:	Una vez que el usuario ha accedido al sistema puede cambiar su password.
Tipo	Secundario y abstracto
Referencias	Funciones 1.1 y 1.2
Cruzadas:	CU: Acceder al sistema-usuario

### Precondiciones

El usuario debe haber ejecutado el caso de uso Acceder al sistema-usuario

### Flujo Principal

- Este caso de uso inicia cuando el usuario selecciona la opción Cambiar password en el Menú de Usuario.
- El sistema presenta la interfaz Cambiar password en donde se solicita el password, el nuevo password y su confirmación.
- El usuario introduce los datos y elige Aceptar la operación.
- El sistema verifica el password de usuario (E1).
- El sistema cambia el password de usuario.
- El sistema regresa al Menú de Usuario.

Figura 47. Cambiar password

The screenshot shows a web form titled "Cambiar password". It contains three text input fields stacked vertically, each with a label to its left: "Password", "Nuevo Password", and "Confirmar Nuevo Password". Below these fields is a blue button with the text "Enviar" in white.

### Flujos de Excepción

E1: El sistema despliega un mensaje informando que el password no es correcto por lo tanto la operación no será efectuada.

### Información general

Caso de Uso:	<b>Salir del sistema</b>
Actores:	Usuario (iniciador)
Propósito:	Permitir que un usuario pueda cerrar la sesión que ha creado y salir del sistema.
Resumen:	El usuario puede optar por salir del sistema en el momento que lo desee.
Tipo	Secundario y abstracto
Referencias	Funciones 1.1 y 1.3
Cruzadas:	CU: Acceder al sistema-usuario

### Precondiciones

El usuario debe haber ejecutado el caso de uso Acceder al sistema-usuario

### Flujo Principal

- Este caso de uso empieza cuando el usuario selecciona la opción Salir en alguna de las interfaces del sistema que contienen esta opción.
- El sistema cierra la sesión iniciada por el usuario.
- El sistema regresa al usuario la interfaz *Acceso de Usuario*

### Información general

Caso de Uso:	<b>Acceder al sistema-administrador</b>
Actores:	Administrador (iniciador)
Propósito:	Permitir el acceso del administrador al modulo de gestión que conforma el sistema.

Resumen:	El administrador introduce su login y password para acceder al módulo de gestión. Los datos introducidos son verificados de acuerdo con los datos almacenados en la base de datos. El sistema despliega un menú de administrador.
Tipo:	Primario
Referencias Cruzadas:	Funciones: 1.1

### Precondiciones

- El sistema debe tener un login y password almacenados como valores por defecto, para que el administrador pueda iniciar el sistema.

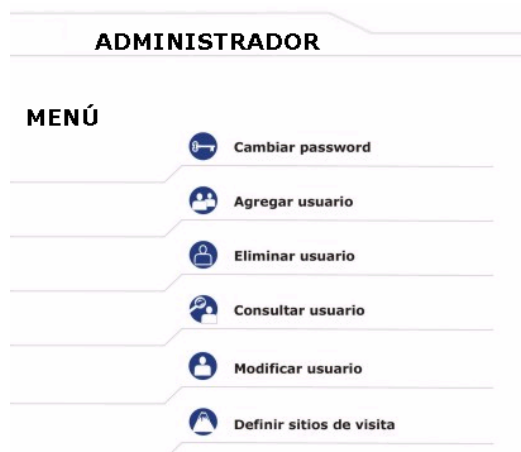
### Flujo Principal

- Este caso de uso empieza cuando el administrador elige utilizar el sistema.
- El sistema presenta al administrador la interfaz Acceso Administrador, la cual es similar a la interfaz Acceso Usuario, que le solicita un login y password.
- El administrador ingresa su login y password y pulsa Entrar.
- El sistema verifica el login y password (E1).
- El sistema despliega el Menú de Administrador.

### Flujos de Excepción

E1: El sistema despliega un mensaje informando que el login o password de administrador no es correcto.

Figura 48. Menú de Administrador



## Gestión usuarios

### Información general

Caso de Uso:	<b>Agregar usuario</b>
Actores:	Administrador (iniciador)
Propósito:	Permitir que un usuario sea adicionado al sistema para que pueda hacer de todas sus funciones.
Resumen:	Este caso de uso despliega un formulario donde se solicita toda la información necesaria para que un usuario sea agregado al sistema.
Tipo	Primario y abstracto
Referencias	Funciones: 4.1.1
Cruzadas:	CU: Acceder al sistema-administrador

### Precondiciones

El administrador debe haber ejecutado el caso de uso Acceder al sistema-administrador

### Flujo Principal

- El caso de uso empieza cuando el administrador selecciona la opción Agregar usuario en el Menú de Administrador.
- El sistema despliega la interfaz Agregar Usuario donde se solicita toda la información necesaria para crear un nuevo usuario.
- El administrador introduce la información de usuario y pulsa el botón Agregar.
- El sistema verifica que todos los campos estén llenos y que los datos (variables o números) suministrados por el administrador coincidan con las características exigidas por el campo (consistencia de los campos) (E1).
- Si los datos suministrados son consistentes y los campos requeridos son llenados completamente, el sistema verifica que la identificación y el login de usuario no estén registrados en la base de datos (E2).
- El sistema despliega un mensaje informando que el usuario ha sido adicionado al sistema y muestra los datos que fueron agregados para este usuario.

### Flujos de Excepción

E1: El sistema despliega un mensaje informando campos vacíos o inconsistentes. Para cada campo se realiza la validación mostrando mensajes de advertencia.

E2: El sistema despliega un mensaje informando identificación o login de usuario no permitido.



Figura 49. Interfaz Agregar Usuario

**Agregar usuario**

Nombre	<input type="text"/>
Apellidos	<input type="text"/>
Dirección	<input type="text"/>
Teléfono	<input type="text"/>
Identificación	<input type="text"/>
E-mail	<input type="text"/>
Login	<input type="text"/>
Password	<input type="text"/>
Confirmar Password	<input type="text"/>

Perfil del Usuario

Diseñador de Formatos  
 Recolector  
 Analista

Centro Operativo:  Area:

## Gestión Sitios

### Información general

Caso de Uso:	<b>Agregar sitios de visita</b>
Actores:	Administrador (iniciador)
Propósito:	Permitir que el administrador agregue sitios generales de visita.
Resumen:	Este caso de uso despliega una interfaz donde se permite agregar sitios generales de visita (volcanes, regiones, fallas). Para cada sitio general pueden ser agregados sitios específicos de visita si el administrador lo desea.
Tipo	Primario y abstracto
Referencias	Funciones: 4.2.1
Cruzadas:	CU: Acceder al sistema-administrador

### Precondiciones

El administrador debe haber ejecutado el caso de uso Acceder al sistema-administrador

### Flujo Principal

- Este caso de uso empieza cuando el administrador selecciona Definir sitios de visita en el Menú de Administrador.
- El sistema despliega la interfaz Gestión para sitios generales de visita, donde el administrador puede adicionar un volcán, una región o una falla. En la Figura 50 se ilustra la gestión para volcanes, de igual forma es para regiones y fallas.
- El administrador introduce el nombre del sitio general de visita y opcionalmente puede chequear la casilla Agregar sitios, para agregar sitios específicos de visita que

corresponden al sitio general, ver numeral 4.3 (Tabla 3) para consultar sitios específicos que se permite adicionar.

- El sistema verifica la consistencia del campo (E1).
- El sistema verifica el nombre del sitio general agregado (E2).
- El sistema agrega el sitio general de visita a la base de datos.

### Flujos de Excepción

E1: El sistema despliega un mensaje informando que el campo está vacío o inconsistente. La validación se realiza mostrando un mensaje de advertencia.

E2: El sistema despliega un mensaje informando que el sitio general que se intenta agregar se encuentra registrado en el sistema.

Figura 50. Interfaz Gestión para Sitios generales de visita



### Información general

Caso de Uso:	<b>Consultar sitios de visita</b>
Actores:	Administrador (iniciador)
Propósito:	Permitir que el administrador consulte sitios generales de visita.
Resumen:	Este caso de uso despliega ordenadamente los diferentes sitios generales de visita que han sido adicionados al sistema.
Tipo	Secundario y abstracto
Referencias	Funciones: 4.2.2
Cruzadas:	CU: Acceder al sistema-administrador

### Precondiciones

El administrador debe haber ejecutado el caso de uso Acceder al sistema-administrador

### Flujo Principal

- Este caso de uso empieza cuando el administrador selecciona Definir sitios de visita en el Menú de Administrador.

- El sistema despliega ordenadamente una lista de los diferentes volcanes, regiones y fallas que han sido agregadas al sistema. La interfaz Gestión para sitios generales de visita muestra la información.

### Información general

Caso de Uso:	<b>Modificar sitios de visita</b>
Actores:	Administrador (iniciador)
Propósito:	Permitir que el administrador modifique el nombre de un sitio general de visita.
Resumen:	Este caso de uso despliega una interfaz donde se puede agregar el nuevo nombre del sitio general de visita.
Tipo	Secundario y abstracto
Referencias	Funciones: 4.2.3
Cruzadas:	CU: Acceder al sistema-administrador

### Precondiciones

El administrador debe haber ejecutado el caso de uso Acceder al sistema-administrador

### Flujo Principal

- Este caso de uso empieza cuando el administrador selecciona Definir sitios de visita en el Menú de Administrador.
- El sistema despliega ordenadamente una lista de los diferentes volcanes, regiones y fallas que han sido agregadas al sistema. Cada nombre de volcán, región o falla tiene un enlace para modificar el nombre del sitio general de visita. La interfaz Gestión para sitios generales de visita muestra la información.
- El administrador pulsa sobre el enlace de un volcán, región o falla.
- El sistema despliega una interfaz para modificación de sitio general, donde se muestra el nombre del sitio general de visita y se solicita un nuevo nombre.
- El administrador introduce un nuevo nombre para el sitio general de visita y pulsa el botón Modificar.
- El sistema verifica la consistencia del campo (E1).
- El sistema verifica el nuevo nombre introducido (E2).
- El sistema realiza la modificación y despliega la interfaz Gestión para sitios generales de visita actualizada.

### Flujos de Excepción

E1: El sistema despliega un mensaje informando que el campo está vacío o inconsistente. La validación se realiza mostrando un mensaje de advertencia.

E2: El sistema despliega un mensaje informando que el nuevo nombre introducido ya se encuentra registrado en el sistema.

### Información general

Caso de Uso:	<b>Eliminar sitios de visita</b>
Actores:	Administrador (iniciador)
Propósito:	Permitir que el administrador elimine sitios generales de visita.
Resumen:	Este caso de uso elimina un sitio general de visita almacenado en el sistema.
Tipo	Primario y abstracto
Referencias	Funciones: 4.2.4
Cruzadas:	CU: Acceder al sistema-administrador

### Precondiciones

El administrador debe haber ejecutado el caso de uso Acceder al sistema-administrador

### Flujo Principal

- Este caso de uso empieza cuando el administrador selecciona Definir sitios de visita en el Menú de Administrador.
- El sistema despliega ordenadamente una lista de los diferentes volcanes, regiones y fallas que han sido agregadas al sistema junto con un enlace Eliminar. La interfaz Gestión para sitios generales de visita muestra la información.
- El administrador presiona el enlace Eliminar.
- El sistema solicita confirmación para eliminar el sitio general de visita.
- El administrador confirma la eliminación del sitio general de visita.
- El sistema despliega la interfaz Gestión para sitios generales de visita actualizada.

## Gestión Formato

### Información General

Caso de Uso:	<b>Elaborar Formato</b>
Actores:	Diseñador de Formatos (iniciador)
Propósito:	Permite la elaboración de formatos para recolección de datos en campo.
Resumen:	Este caso de uso proporciona una interfaz que permite seleccionar el tipo de pregunta que se desea adicionar de acuerdo al tipo de formato. El Diseñador de Formatos puede editar o modificar el contenido de las preguntas, incluyendo las opciones cuando sea el caso. Adicionalmente, es posible eliminar preguntas y guardar el formato periódicamente.
Tipo:	Primario
Referencias	Funciones: 2.1
Cruzadas:	

### Precondiciones

- El Usuario debe ejecutar el Caso de Uso Acceder al Sistema
- El Usuario debe estar registrado en el sistema

## Flujo Principal

- Este caso de uso empieza cuando el Diseñador de Formatos selecciona Elaborar Formato (Simple o Tipo Encuesta) del Menú Principal.
- El sistema despliega la interfaz correspondiente a la herramienta de construcción de formatos.
- El Diseñador de Formatos puede acceder a la sección de información general del formato o a la sección de preguntas.
- Si el Diseñador de Formatos accede a la sección de información general, subflujo S1: Introducir información general del formato.
- Si el Diseñador de Formatos accede a la sección de preguntas, subflujo S2: Introducir las preguntas del formato.

Figura 51. Sección de Información General del Formato

Información General preguntas

Título:

Fecha de Elaboración: 2003-10-17

Area: Sismologia

Diseñado por: Francisco Martinez

Lugar donde se va a realizar la visita:

Volcán  
 Región

Descripción:

Datos Generales:

Nro	Dato	Tipo
	Estacion Sismologica	
	Fecha	

## Subflujos

### S1: Introducir información general del formato

- El Diseñador de formatos introduce el Título y selecciona el Sitio general y el Sitio específico de visita para el formato.
- De manera opcional el Diseñador de Formatos puede adicionar una descripción y datos generales extras para el formato, aparte de la fecha de recolección (E1).
- Similarmente, el Diseñador de Formatos puede eliminar cualquiera de los datos generales que haya adicionado previamente.

Figura 52. Adición y Eliminación de Datos Generales al Formato

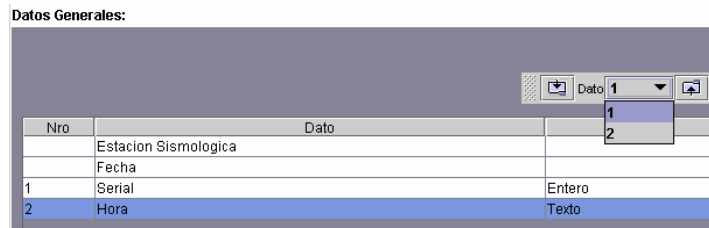
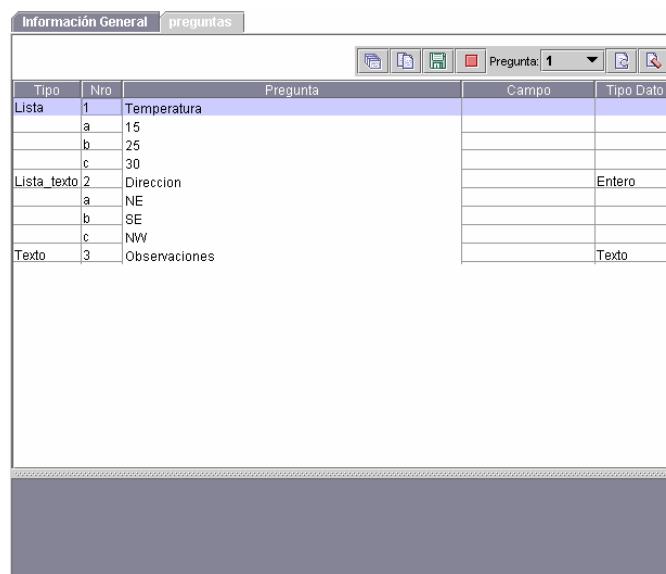


Figura 53. Sección de Preguntas del Formato



**S2: Introducir las preguntas del formato**

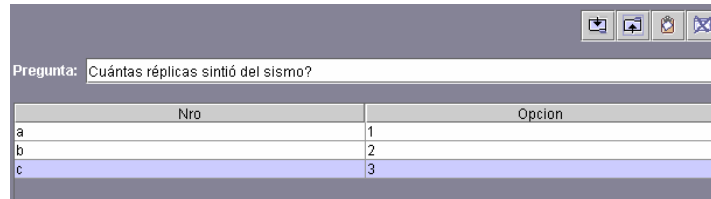
- El Diseñador de Formatos puede seleccionar cualquiera de las opciones disponible en la barra de herramientas.
- Si el Diseñador de formatos selecciona la opción Pregunta de selección múltiple, subflujo S3 Elaborar pregunta de selección múltiple.
- Si el Diseñador de formatos selecciona la opción Pregunta de contenido o texto, subflujo S4 Elaborar pregunta tipo contenido o texto.
- Si el Diseñador de formatos selecciona la opción Modificar pregunta, subflujo S5 Modificar pregunta.
- Si el Diseñador de formatos selecciona la opción Eliminar pregunta, subflujo S6 Eliminar pregunta.
- Si el Diseñador de formatos selecciona la opción Guardar, subflujo S7 Guardar Formato.
- Si el Diseñador de formatos selecciona la opción Terminar, subflujo S8 Terminar Formato.

**S3: Elaborar pregunta de selección múltiple**

- El sistema despliega la interfaz de elaboración de la pregunta, adicionando dos opciones por defecto. Previamente el usuario ha seleccionado si la respuesta es única o múltiple.

- El Diseñador de Formatos introduce el texto de la pregunta.
- El Diseñador de Formatos puede adicionar o eliminar opciones en la pregunta a través de las funciones disponibles en la barra de herramientas (E2).
- Finalmente el Diseñador de Formatos selecciona Aceptar para adicionar la pregunta al formato (E3) o Cancelar para suspender la operación.

Figura 54. Elaboración de una Pregunta de Selección Múltiple



Nro	Opcion
a	1
b	2
c	3

#### S4: Elaborar pregunta tipo contenido o texto

- El sistema despliega la interfaz de elaboración de la pregunta.
- El Diseñador de Formatos introduce el texto de la pregunta.
- El Diseñador de Formatos define el tipo de dato que debe contener la respuesta en la barra de herramientas(entero, decimal o texto).
- Finalmente el Diseñador de Formatos selecciona Aceptar para adicionar la pregunta al formato (E3) o Cancelar para suspender la operación.

Figura 55. Elaboración de una Pregunta tipo Contenido o Texto



#### S5: Modificar Pregunta

- El sistema despliega la información correspondiente a la pregunta seleccionada.
- El Diseñador de Formatos modifica cualquiera de los parámetros de la pregunta a través de la barra de herramientas como se menciona en los subflujos S3 y S4.
- Finalmente el Diseñador de Formatos selecciona Aceptar para guardar los cambios de la pregunta (E3) o Cancelar para suspender la operación.

#### S6: Eliminar Pregunta

- El sistema despliega un mensaje para confirmar la eliminación de la pregunta seleccionada.
- Si el Diseñador de Formatos confirma la operación, la pregunta es eliminada del formato.

#### S7: Guardar Formato

- El sistema busca las modificaciones realizadas desde la última vez que se guardó el formato (E4).
- El sistema envía las modificaciones a la base de datos central.

- El sistema despliega un mensaje confirmando que el formato se ha guardado satisfactoriamente (E5).

**S8: Terminar Formato**

- El sistema despliega una mensaje para confirmar la finalización del formato.
- Si el Diseñador de Formatos confirma la operación, el sistema despliega un nuevo mensaje para establecer si se deben guardar las últimas modificaciones realizadas.
- Si el Diseñador de Formatos responde afirmativamente, el sistema guarda el formato como se indica en el Subflujo S7.
- Finalmente el sistema regresa al Menú Principal.

**Flujos de Excepción**

E1: No se permite la adición de datos generales extras cuando el objeto de medida corresponde a Personas o el tipo del formato seleccionado es Simple.

E2: Si la pregunta es combinada, el sistema permite la definición de las subpreguntas (denominadas campo) para cada una de las opciones, además del tipo de respuesta que corresponda. Si se trata de una pregunta Lista y Texto, además de las opciones el sistema permite la definición del tipo de respuesta en la barra de herramientas.

E3: El sistema despliega un mensaje indicando que el texto de la pregunta no puede ser vacío.

E4: El sistema despliega un mensaje para indicar que el título del formato no puede ser vacío.

E5: El sistema despliega un mensaje que indica que no ha sido posible conectarse a la base de datos.

Figura 56. Elaboración de una Pregunta Combinada

Nro	Opcion	Campo	Tipo
a	Si	Cuál?	Texto
b	No		

Figura 57. Elaboración de una Pregunta Lista y Texto

Nro	Opcion
a	°F
b	°C
c	°K



## Gestión Visita

### Información general

Caso de Uso:	<b>Definir visita</b>
Actores:	Diseñador de formatos (iniciador)
Propósito:	Definir una visita que será llevada a cabo para recolección de datos.
Resumen:	Este caso de uso permite definir una visita seleccionando un título y un sitio general de visita donde se realizará. Se definen también los formatos que serán utilizados y opcionalmente los sitios específicos que serán visitados.
Tipo	Primario y abstracto
Referencias	Funciones: 2.4.1, 2.4.2, 2.4.3
Cruzadas:	CU: Acceder al sistema-usuario

### Precondiciones

- El diseñador de formatos debe haber ejecutado el caso de uso Acceder al sistema-usuario.
- El sistema debe contar con información sobre perfil de usuario, área y nombre completo del diseñador de formatos.

### Flujo Principal

- Este caso de uso empieza cuando el diseñador de formatos selecciona Definir visita en el Menú de Usuario.
- El sistema verifica el área a la que pertenece el diseñador de formatos y despliega la interfaz Definir visita, fijando el nombre de la persona que define la visita, que corresponde con el nombre de diseñador de formatos y el área a la que pertenece la visita, que corresponde con el área a la que pertenece el diseñador de formatos. De esta forma, un diseñador de formatos solo puede configurar visitas para el área a la que él pertenezca. Para cada área los sitios generales a seleccionar varían de acuerdo con la siguiente tabla.

Tabla 7. Relación área-sitio general-visita

Sitios generales de visita	Áreas en las que se puede realizar una Visita		
	Geología	Geoquímica	Sismología
Volcán	X	X	X
Región	X		X
Falla	X	X	

Figura 58. Interfaz Definir visita

**Definir visita**

Nombre de la Visita	
Area	Geoquimica
Configurada por	Diego andres gonzales
Fecha de Realización	año 2003 -mes 01 -día 01

Seleccione el sitio al que se realizará la visita

Volcán Galeras

Falla Norte

**Aceptar**

- El diseñador de formatos introduce la información, selecciona un sitio general de visita y pulsa Aceptar.
- El sistema recupera el área y determina si el sitio general seleccionado por el diseñador de formatos corresponde a un volcán, región o falla. El sistema selecciona los formatos teniendo en cuenta que el área recuperada coincida con el área del formato y además que el sitio general de visita para el cual fue creado el formato corresponda con el sitio general recuperado por el sistema (E1).
- El sistema despliega los formatos de forma organizada.
- El diseñador de formatos selecciona los formatos que serán utilizados durante la visita.
- El sistema despliega la interfaz Formatos seleccionados para Visita donde se muestran los formatos seleccionados para la visita.
- Si el diseñador de formatos hace clic en el enlace definir sitios, Subflujo S1: definir sitios específicos para un formato
- Si el diseñador de formatos selecciona un formato y pulsa el botón Borrar, Subflujo S2: Borrar formato seleccionado.
- Si el diseñador de formatos hace clic en el icono seleccionar otro formato, Subflujo S3: Seleccionar otro formato para la visita.
- Si el diseñador de formatos hace clic en el icono Eliminar visita creada, Subflujo S4: Eliminar visita creada.

Figura 59. Interfaz Formatos seleccionados para visita



## Subflujos

### S1: Definir sitios específicos para un formato

- El sistema verifica si existen sitios específicos para los cuales se pueda recoger datos y que puedan ser seleccionados para el formato (E2).
- El sistema despliega los sitios específicos de visita habilitados para el formato.
- El diseñador de formatos selecciona los sitios específicos y los agrega al formato.
- El sistema despliega la interfaz Formatos seleccionados para Visita actualizada.

### S2: Borrar formato seleccionado

- El sistema valida la visita (E3).
- El sistema elimina de la visita el formato que ha sido seleccionado.
- El sistema despliega la interfaz Formatos seleccionados para Visita actualizada.

### S3: Seleccionar otro formato para la Visita

- El sistema valida que existan formatos para adicionar a la visita (E4).
- El diseñador de formatos selecciona el formato que desea adicionar a la visita.
- El sistema adiciona el formato a la visita y despliega la interfaz Formatos seleccionados para Visita actualizada.

### S4: Borrar visita creada

- El sistema despliega un mensaje solicitando que la operación sea confirmada.
- El diseñador de formatos confirma la operación.
- El sistema elimina la visita creada y regresa al Menú de Usuario.

## Flujos de Excepción

- E1: El sistema despliega un mensaje informando que la visita no puede ser creada porque no existen formatos para un área y sitio general determinados.
- E2: El sistema despliega un mensaje informando que para el formato no existen sitios específicos de visita que puedan ser seleccionados.
- E3: El sistema valida teniendo en cuenta que una visita está definida correctamente cuando tiene al menos un formato. Si esta condición no se cumple el sistema despliega un mensaje informando que el formato no puede ser eliminado.
- E4: El sistema despliega un mensaje informando que no existen otros formatos que puedan ser adicionados a la visita.

## Información general

Caso de Uso:	<b>Consultar información</b>
Actores:	Usuario (iniciador)
Propósito:	Consultar información que ha sido recolectada durante una visita.
Resumen:	Este caso de uso permite consultar información de visitas para las cuales se ha terminado la recolección de datos. La información recolectada por medio de encuestas no es desplegada por este caso de uso.
Tipo	Primario y abstracto
Referencias	Funciones: 1.3
Cruzadas:	CU: Acceder al sistema-usuario

## Precondiciones

El diseñador de formatos debe haber ejecutado el caso de uso Acceder al sistema-usuario.

## Flujo Principal

- Este caso de uso empieza cuando el usuario selecciona Consultar información en el Menú de Usuario.
- El sistema solicita el área a la que pertenece la visita y el sitio general visitado.
- El usuario introduce información para el área y el sitio general de visita.
- El sistema valida que existan visitas para el área y sitio general seleccionado por el usuario (E1).
- El sistema despliega la interfaz Consulta de información para Sitios específicos, donde se pueden consultar las visitas en las cuales se recogió datos para algún sitio específico. Los sitios específicos que se despliegan en esta interfaz, dependen del área y sitio general seleccionado, ver numeral 4.3 Tabla 3.
- El usuario introduce el nombre del sitio general de visita, que contiene los sitios específicos para los cuales se consultarán visitas.
- El sistema despliega una lista con todas las visitas que fueron realizadas para recolectar datos en un sitio específico de visita.

Figura 60. Interfaz Consulta de información para sitios específicos

SITIO DE VISITA: VOLCAN

Area Sismologia

Consultar Visitas a Estaciones Sismologicas

Seleccione el volcan al que estan relacionadas las estaciones sismologicas

Volcan:  
Nevado del Huila

Enviar

Figura 61. Interfaz Visitas

**Resultados de la Búsqueda**

Visitas durante las cuales se tomaron datos para el Volcán Nevado del Huila

Visita	Fecha
<a href="#">Visita de sismologia- Enero</a>	2003-01-20
<a href="#">Visita de sismologia - Marzo</a>	2003-03-01
<a href="#">Visita de sismologia - Junio</a>	2003-06-21

- El usuario hace clic en la visita que desea consultar.
- El sistema despliega los formatos utilizados durante la visita para recoger datos en el sitio específico seleccionado. Los formatos pueden ser simples o tipo encuesta.
- Si el usuario selecciona un formato simple, Subflujo S1: Consultar información recolectada a través de un formato simple.
- Si el usuario selecciona un formato tipo encuesta y selecciona un sitio específico de visita para el que se recolectó datos, Subflujo S2: Consultar información recolectada a través de un formato tipo encuesta.

### Subflujos

#### S1: Consultar información recolectada a través de un formato simple

- El sistema despliega la interfaz Consultar información utilizando Formato Simple, donde se muestra las preguntas del formato y una tabla resumen con los diferentes datos recolectados para cada uno de los sitios específicos de visita.

**S2: Consultar información recolectada a través de un formato tipo encuesta**

- El sistema despliega la interfaz Consultar información utilizando Formato Tipo Encuesta, donde se muestran las preguntas del formato junto con la respectiva respuesta.
- El usuario puede seleccionar en la misma interfaz, otros sitios específicos de visita para los cuales se haya recolectado información utilizando formatos tipo encuesta.
- Si el perfil de usuario es analista, el sistema da la posibilidad de transferir datos al software estadístico Excel, esto activa el caso de uso Transferir información.

Figura 62. Consultar información utilizando Formato Simple

**Formato: Datos Estructurales Parte Alta del Nevado del Huila**

Preguntas del Formato

- 1 Direccion
- 2 Buzamiento
- 3 Tipo
- 4 OBSERVACIONES

**Datos Recolectados**

Estacion	Direccion	Buzamiento	Tipo	OBSERVACIONES	Fecha
BPNH113	NE	NE	Falla	Lavas brecha slavanda Abertura de 15 cm	2003-10-02
BPNH120	NW	SE	Diadaza	Posiblemente asociada a falla Fragmentos de brecha	2003-10-02
BPNH120	NE	NW	Diadaza	N	2003-10-02

Figura 63. Interfaz Consultar información utilizando Formato Tipo Encuesta

### Consultar información

**VISITA:**  
**VISITA SISMOLOGIA AL VOLCAN PURACE**

**TITULO DE FORMATO:**  
**FORAMTO PARA ESTACION SISMOLOGICA**

<b>Area</b>	<b>Sismologia</b>
<b>Sitio de Visita</b>	<b>Volcan Purace</b>
<b>Estacion Sismologica</b>	<b>Azufral</b>
<b>Fecha de mantenimiento</b>	<b>2003-10-19</b>

**Proceso de Recoleccion 1**

1. Condicion de la estacion sismologica  
 Rta/ a Buena
2. Voltaje de las baterias instaladas  
 Rta/ 12
3. Temperatura de la estacion  
 Rta/ 8
4. Voltaje de la fuente en el canal 1  
 Rta/ 5

Transferir datos a Excel

**Otras estaciones sismológicas visitadas**

Colon

## Flujo de Excepción

E1: El sistema despliega un mensaje informando que para el área y sitio general seleccionado por el usuario, no existen visitas para las cuales se haya terminado la recolección de datos.

## Estadísticas

### Información general

Caso de Uso:	<b>Desplegar estadísticas básicas</b>
Actores:	Analista (iniciador)
Propósito:	Desplegar estadísticas básicas para información recolectada desde personas.
Resumen:	Este caso de uso permite obtener resultados estadísticos para información recolectada desde personas. El escenario para este caso de uso es realizar una encuesta a un grupo de personas y obtener estadísticas a partir de los resultados obtenidos.
Tipo	Primario y abstracto
Referencias	Funciones: 5.1
Cruzadas:	CU: Acceder al sistema-usuario

### Precondiciones

- El analista debe haber ejecutado el caso de uso Acceder al sistema-usuario.
- Los formatos utilizados en este caso de uso son encuestas.
- Los datos recolectados para encuestas solo existen en el área de Sismología.
- Las visitas para recoger datos desde personas tienen como sitio general de visita una región.

### Flujo Principal

- Este caso de uso empieza cuando el usuario selecciona Desplegar estadísticas básicas en el Menú de Usuario.
- El sistema verifica que existen visitas para el área de Sismología donde se haya utilizado al menos una encuesta (E1).
- El sistema despliega una lista con las diferentes visitas para el área de Sismología, donde se haya utilizado al menos una encuesta.
- El analista selecciona una de las visitas.
- El sistema despliega las encuestas que han sido utilizadas durante esta visita.
- El analista selecciona una encuesta.
- El sistema despliega la interfaz Preguntas procesables estadísticamente, donde se muestra las preguntas que contiene la encuesta.
- Si el analista selecciona las preguntas que desea procesar y pulsa el botón Procesar, Subflujo S1: Procesar preguntas estadísticamente.
- Si el analista pulsa el botón Borrar para borrar datos recolectados para la encuesta durante esta visita, Subflujo S2: Borrar datos recolectados para encuesta durante una visita.

## Subflujos

### S1: Procesar preguntas estadísticamente

- El sistema recupera las preguntas seleccionadas y comienza a procesar cada una de ellas. Las preguntas que pueden ser procesadas son: selección múltiple respuesta única, selección múltiple respuesta múltiple y combinada. Para las preguntas combinadas el sistema da la posibilidad de observar lo que respondieron las personas al campo que acompaña una opción. Para las preguntas de contenido el sistema permite consultar las respuestas pero no se realiza ningún procesamiento estadístico.
- El sistema despliega, dependiendo del tipo de pregunta, la interfaz de procesamiento estadístico adecuada. Para resumir, se mostrará la interfaz Procesamiento estadístico pregunta combinada.

### S2: Borrar datos recolectados para encuesta durante una visita

- El sistema solicita confirmación de la operación
- El analista confirma la operación
- El sistema elimina los datos recolectados para la encuesta.

## Flujo de Excepción

E1: El sistema despliega un mensaje informando que no existen visitas tengan al menos una encuesta.

Figura 64. Interfaz Preguntas procesables estadísticamente

Departamento  
Municipio  
Direccion

Seleccione las preguntas que serán procesadas estadísticamente

Descripción:  
Se ha elaborado este cuestionario para complementar los datos registrados acerca de la Intensidad del Terremoto

Procesar todas las preguntas

1. Sintió usted el sismo?  
a. Si se sintió el terremoto en esta area Se asustó? \_\_\_\_  
b. No se sintió el terremoto en esta area Por qué? \_\_\_\_

2. Donde estaba usted cuando sucedió el sismo?  
a. En la casa(o en edificio) b. Fuera de la casa

3. Que estaba haciendo en ese momento?  
a. En movimiento b. Estático  
c. Subiendo a un transporte d. Durmiendo  
e. Otros

5. Cuantos pisos tiene la casa o edificio donde usted vive? \_\_\_\_

Procesar

Borrar datos recolectados para la encuesta durante esta visita  
Borrar



Figura 65. Interfaz Procesamiento estadístico pregunta combinada

**Pregunta 1** Sintió usted el sismo?

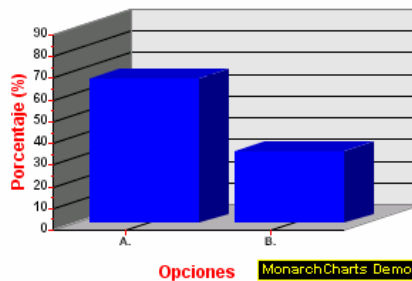
a. Se sintió el terremoto en esta area. Se asustó? \_\_\_\_

b. No se sintió el terremoto en esta area. Por qué? \_\_\_\_

**Estadísticas Básicas para la Pregunta 1**

Opcion	Porcentaje (%)	Nro. Encuestados
A.	66.666	2
B.	33.333	1

**Total de encuestados** 3



**Información general**

Caso de Uso:	<b>Transferir información</b>
Actores:	Analista (iniciador)
Propósito:	Transferir información histórica almacenada en el sistema al software estadístico Excel.
Resumen:	Este caso de uso permite transferir cierto tipo de información almacenada en la base de datos, al software estadístico Excel, con el fin de obtener mayores capacidades de procesamiento.
Tipo	Primario y abstracto
Referencias	Funciones: 5.2
Cruzadas:	CU: Acceder al sistema-usuario

**Precondiciones**

- El analista debe haber ejecutado el caso de uso Acceder al sistema-usuario.
- El analista debe haber ejecutado el caso de uso Consultar información.
- Debe existir información histórica para el sitio específico que se desea transferir información.

## Flujo Principal

- Este caso de uso se inicia cuando el analista hace clic en el icono Transferir datos a Excel de la interfaz Consultar información utilizando Formato Tipo Encuesta.
- El sistema verifica que para el sitio específico exista información histórica (E1).
- El sistema despliega la interfaz Transferir información donde se muestran las preguntas que pueden ser transferidas a Excel y se solicita el rango de tiempo en años, para realizar transferencia de datos. Las preguntas que pueden ser transferidas son las de tipo Contenido para las cuales se recogen datos enteros o decimales. Este tipo de pregunta permite realizar procesamiento a través del tiempo.
- El analista selecciona las preguntas que desea transferir y hace clic en el icono Transferir.
- El sistema le permite descargar una copia de la información al disco duro.

Figura 66. Interfaz Transferir información

**Preguntas que permiten observar comportamientos a través del tiempo**

Seleccione las preguntas que desea transferir a Excel


2. Voltaje de las baterías instaladas \_\_\_\_

3. Temperatura de la estación \_\_\_\_

4. Voltaje de la fuente en el canal 1 \_\_\_\_

Transferir desde - hasta

2003 - 2004

  
Transferir

## Flujo de Excepción

E1: El sistema despliega un mensaje informando que para el sitio específico de visita aun no existe información histórica.

## Recolección de Datos

### Información General

Caso de Uso:	<b>Seleccionar Visita a Descargar</b>
Actores:	Recolector (iniciador)
Propósito:	Permite al Recolector seleccionar la visita en la cual va a participar.
Resumen:	Este caso de uso da al Recolector la facultad de seleccionar la visita en la cual va a participar, a través de una interfaz que permite consultar los formatos seleccionados para la visita y ordenar la descarga de los mismos al dispositivo.

Tipo:	Primario
Referencias Cruzadas:	Funciones: 3.1

### Precondiciones

- El Recolector debe ejecutar el Caso de Uso Acceder al Sistema
- El Recolector debe estar registrado en el sistema

### Flujo Principal

- Este caso de uso empieza cuando el Recolector selecciona la opción Seleccionar Visita a Descargar del Menú Principal.
- El sistema despliega una tabla que muestra las visitas que se encuentran en curso para el área a la que pertenece el Recolector, especificando la fecha y el sitio en el que se van a realizar.

Figura 67. Lista de Visitas en Curso



- El Recolector selecciona una visita para consultar su información.
- El sistema despliega una tabla que muestra los formatos seleccionados para la visita, acompañados de su descripción y los sitios específicos seleccionados para cada uno.

Figura 68. Lista de Formatos seleccionados para la Visita

**Seleccionar visita a descargar**

**VISITA PENDIENTE:  
VISITA AL VOLCÁN GALERAS**

<b>Area</b>	Sismologia
<b>Sitio de Visita</b>	Volcan Galerías
<b>Fecha</b>	2003-11-01

La tabla siguiente muestra los formatos para recolección de datos que fueron seleccionados para esta visita

Formatos a descargar	Descripción	Sitios definidos
<a href="#">Formato para Estaciones Sismologicas - Galerías</a>		Estaciones Sismologicas Coconuco Nevados

- El Recolector puede seleccionar la opción “Configurar visita para descargar a la Palm”, Regresar a la interfaz de visitas en curso o regresar al Menú.
- Si el Recolector selecciona la opción “Configurar visita para descargar a la Palm”, el sistema despliega un mensaje indicando que la información de la visita será descargada al dispositivo la próxima vez que se realice una sincronización (E1).

### Flujos de Excepción

E1: El sistema despliega un mensaje de error cuando el Recolector ya ha seleccionado una visita para descarga previamente, ya que un Recolector solo puede participar en una visita.

### Información General

Caso de Uso:	<b>Recolectar Información en Campo</b>
Actores:	Recolector (iniciador), Objeto de medida
Propósito:	Permite la recolección de datos en campo a través del dispositivo móvil.
Resumen:	Este caso de uso permite al Recolector obtener información en campo para diferentes objetos de medida a través del dispositivo móvil. Se proporciona una interfaz específica para cada tipo de pregunta de acuerdo al tipo de formato. Adicionalmente, para los formatos tipo Encuesta el Recolector puede interrumpir el proceso en una pregunta determinada cuando así lo requiera para reanudarlo posteriormente.
Tipo:	Primario
Referencias Cruzadas:	Funciones: 3.3.1, 3.3.2

### Precondiciones

- El Recolector debe ejecutar el Caso de Uso Descargar Visita y Sincronizar Datos
- El Recolector debe estar registrado en el sistema

### Flujo Principal

- Este caso de uso empieza cuando el Recolector selecciona uno de los formatos que aparece en la lista de la interfaz principal en la PDA y pulsa el botón Nuevo.
- Si el formato es tipo Encuesta, Subflujo S1: Recolectar información para un formato Tipo Encuesta.
- Si el formato es Simple, Subflujo S2: Recolectar información para un formato Simple.

Figura 69. Interfaz Principal en la PDA

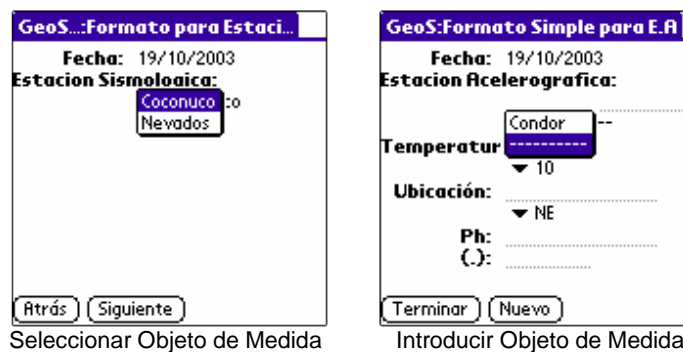


### Subflujos

#### S1: Recolectar Información para un formato Tipo Encuesta

- Inicialmente el sistema despliega una interfaz que permite introducir el objeto de medida o seleccionarlo de una lista si estos se han predefinido desde el servidor. En el primer caso, si se han definido objetos de medida en recolección anteriormente, es posible seleccionar alguno de ellos de la lista que aparece en la parte inferior si se desea adicionar un nuevo proceso para ese objeto de medida. Adicionalmente la interfaz permite introducir las respuestas para los datos generales que se hayan definido para el formato (E1).

Figura 70. Interfaz Datos Generales



- El Recolector pulsa el botón Siguiente para pasar a la primera pregunta del formato o Atrás para cancelar la operación y regresar a la Interfaz Principal (E2).
- Cuando el Recolector pasa a la sección de preguntas, a medida que se avanza con el botón Siguiente el sistema despliega una interfaz específica para cada tipo de pregunta y guarda la respuesta que el Recolector haya introducido para la pregunta inmediatamente anterior (E2).

Figura 71. Interfaz Pregunta Selección Múltiple

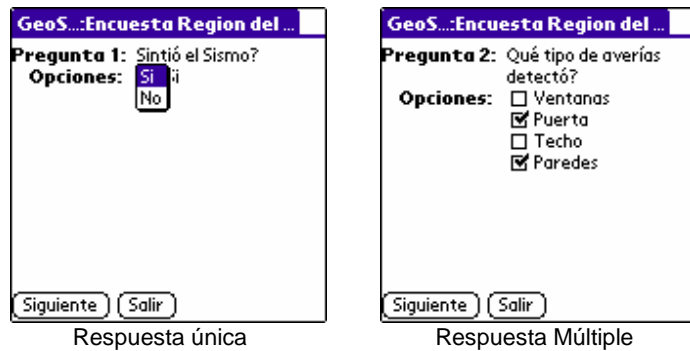


Figura 72. Interfaz Pregunta Combinada y Contenido

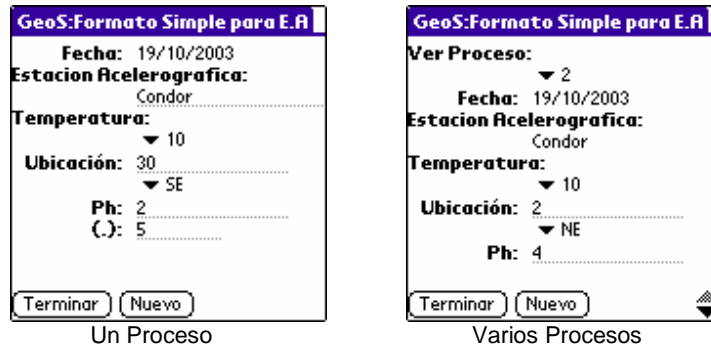


- Si el Recolector pulsa el botón Terminar en cualquiera de las preguntas, el sistema despliega un mensaje para confirmar la interrupción del proceso de recolección. Si el Recolector lo confirma, se regresa a la interfaz principal y el proceso queda pendiente.
- Cuando se contesta la última pregunta, el sistema despliega un mensaje indicando que el formato ha finalizado.

### S2: Recolectar Información para un formato Simple

- El sistema despliega una interfaz para introducir las respuestas del formato.
- La primera pregunta corresponde al objeto de medida y sigue las mismas reglas mencionadas anteriormente para el formato Tipo Encuesta.
- Si el Recolector pulsa el botón Nuevo se inicia un nuevo proceso de recolección.
- El Recolector puede navegar entre las respuestas de los diferentes procesos a través de la lista que aparece en la parte superior y modificar las respuestas cuando lo considere conveniente (Caso de Uso Modificar Información en Campo).

Figura 73. Interfaz para Recolección de Datos Formato Simple



- Cuando el Recolector presiona el botón Terminar, el sistema guarda las respuestas del formato y regresa a la interfaz principal.

### Información General

Caso de Uso:	<b>Consultar Información en Campo</b>
Actores:	Recolector (iniciador)
Propósito:	Permite la consulta de la información recolectada en campo.
Resumen:	A través de este caso de uso el Recolector puede consultar la información que se ha obtenido en campo. Se dispone de una interfaz para consultar los diferentes procesos de recolección de acuerdo al tipo de formato.
Tipo:	Primario
Referencias Cruzadas:	Funciones: 3.3.3, 3.3.4

### Precondiciones

- El Recolector debe ejecutar el Caso de Uso Descargar Visita y Sincronizar Datos
- El Recolector debe estar registrado en el sistema
- El Recolector debe ejecutar el Caso de Uso Recolectar Información en Campo.

### Flujo Principal

- Este caso de uso empieza cuando el Recolector selecciona uno de los formatos que aparece en la lista de la interfaz principal en la PDA y pulsa el botón Consultar.
- El sistema despliega una lista de los objetos de medida disponibles, colocando un asterisco \* al lado de los objetos que tengan un proceso de recolección pendiente. Si el numero de objetos es mayor que 5, se dispone de botones de navegación Atrás y Siguiente para avanzar entre pantallas de hasta 5 objetos cada una.

Figura 74. Interfaz Objetos de Medida

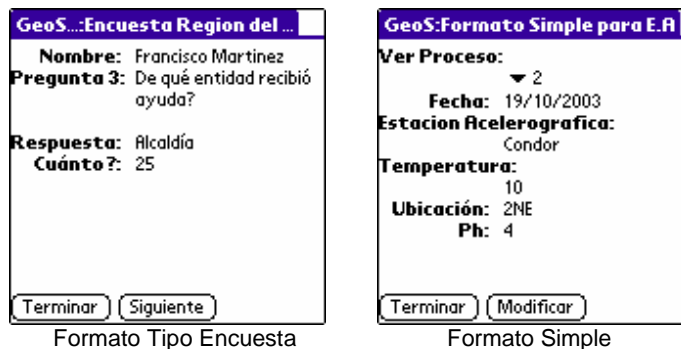


- El Recolector puede seleccionar un objeto cuyos procesos estén terminados o un objeto que tenga un proceso pendiente.
- Si el Recolector selecciona un objeto cuyos procesos están terminados, Subflujo S1: Consultar información asociada a un objeto de medida (E3).
- Si el Recolector selecciona un objeto que tiene un proceso pendiente, Subflujo S2: Reanudar proceso de recolección.

S1: Consultar información asociada a un objeto de medida

- Si el Formato es Simple, el sistema despliega la totalidad de las respuestas obtenidas en campo para el primer proceso de recolección y proporciona una lista para navegar entre los diferentes procesos existentes para este objeto, si es el caso.
- Si el Formato es Tipo Encuesta, el sistema despliega la primera pregunta y se permite avanzar hacia delante a través del botón Siguiente. En cada una de las preguntas existe una lista que permite navegar entre los diferentes procesos de recolección si es el caso, lo cual permite confrontar las respuestas de una misma pregunta en los diferentes procesos de recolección existentes.

Figura 75. Interfaz de Consulta



S2: Reanudar proceso de recolección

- El sistema despliega un mensaje para confirmar si el Recolector desea reanudar el proceso de recolección.
- Si el Recolector lo confirma, el sistema despliega la interfaz correspondiente a la pregunta donde fue interrumpido el proceso de recolección en cuestión.
- El Recolector sigue la funcionalidad descrita en el Caso de Uso Recolectar Información en Campo.



## Información General

Caso de Uso:	<b>Modificar Información en Campo</b>
Actores:	Recolector (iniciador)
Propósito:	Permite la modificación de la información recolectada en campo.
Resumen:	Este caso de uso permite al Recolector modificar la información obtenida en campo. Se proporciona una interfaz para modificar una o varias preguntas de acuerdo al tipo de formato.
Tipo:	Primario
Referencias Cruzadas:	Funciones: 3.3.5

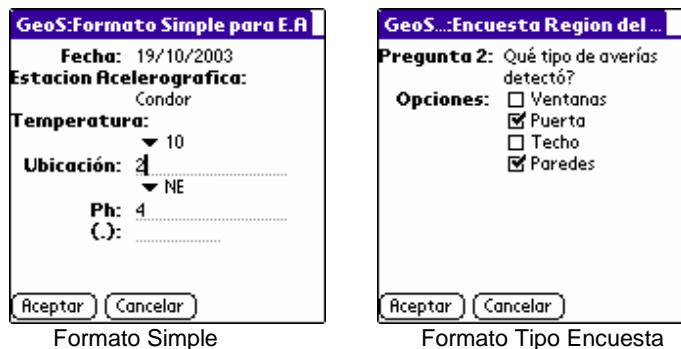
## Precondiciones

- El Recolector debe ejecutar el Caso de Uso Descargar Visita y Sincronizar Datos
- El Recolector debe estar registrado en el sistema
- El Recolector debe ejecutar el Caso de Uso Recolectar Información en Campo.

## Flujo Principal

- Este caso de uso empieza cuando el Recolector selecciona la opción Modificar en la Interfaz de consulta, a la cual se puede llegar de dos maneras: Cuando el Recolector navega entre los diferentes procesos de recolección en el Caso de Uso Recolectar Información en Campo o cuando selecciona un objeto de medida para consultar su información en el Caso de Uso Consultar Información en Campo.
- El Sistema despliega la interfaz correspondiente a la(s) pregunta(s) correspondiente(s), dependiendo del tipo de formato.

Figura 76. Interfaz de Modificación



- El Recolector edita el contenido que desea.
- El Recolector puede seleccionar Aceptar o Cancelar.
- Si el Recolector selecciona Aceptar, el sistema regresa a la interfaz de consulta con los cambios realizados.
- Si el Recolector selecciona Cancelar, el sistema regresa a la interfaz de consulta sin realizar ninguna modificación.

## Información General

Caso de Uso:	<b>Eliminar Información en Campo</b>
Actores:	Recolector (iniciador)
Propósito:	Permite la eliminación de información recolectada en campo.
Resumen:	Este caso de uso permite al Recolector eliminar información obtenida en campo. La interfaz permite eliminar la totalidad de la información asociada a un objeto de medida específico.
Tipo:	Primario
Referencias Cruzadas:	Funciones: 3.3.6

### Precondiciones

- El Recolector debe ejecutar el Caso de Uso Descargar Visita y Sincronizar Datos
- El Recolector debe estar registrado en el sistema
- El Recolector debe ejecutar el Caso de Uso Recolectar Información en Campo.

### Flujo Principal

- Este caso de uso empieza cuando el Recolector elige un objeto de medida y selecciona la opción Eliminar del Menú alterno del dispositivo disponible en la interfaz de Objetos de medida (Caso de uso Consultar Información en Campo).
- El sistema despliega un mensaje para confirmar la operación.
- Si el Recolector la confirma, el sistema elimina toda la información asociada al objeto de medida específico y regresa a la interfaz anterior con los cambios realizados.
- Si el Recolector no confirma la operación, el sistema regresa a la interfaz anterior sin ninguna modificación.

### Flujos de Excepción

E1: Cuando el objeto de medida corresponde a personas, no se presentan datos generales extras en el formato, consecuentemente con el Caso de Uso Elaborar Formato, solo se solicita el Nombre del objeto.

E2: El sistema despliega un mensaje que señala que la respuesta introducida no puede ser vacía.

E3: Si el objeto de medida seleccionado ha sido predefinido desde el servidor y no existen procesos de recolección asociados al mismo, el sistema despliega un mensaje indicando que no existe información para consultar.

## Sincronización

### Información General

Caso de Uso:	<b>Sincronizar Datos</b>
Actores:	Recolector (iniciador)
Propósito:	Permite al Recolector sincronizar los datos del dispositivo con la base de datos central.
Resumen:	Este caso de uso proporciona funciones de sincronización de datos entre el dispositivo móvil y el sistema central. El conducto de sincronización proporciona una interfaz al Recolector para introducir su login y password antes de activar el proceso. Cuando el Recolector lo activa, el conducto realiza las tareas de sincronización respectivas.
Tipo:	Primario
Referencias Cruzadas:	Funciones: 3.2

### Precondiciones

- El Recolector debe ejecutar el Caso de Uso Seleccionar Visita a Descargar
- El Recolector debe estar registrado en el sistema

### Flujo Principal

- Este caso de uso empieza cuando el Recolector selecciona el conducto de la lista proporcionada por HotSync Manager y pulsa el botón Cambiar.
- El sistema despliega el Formulario de Acceso.
- 

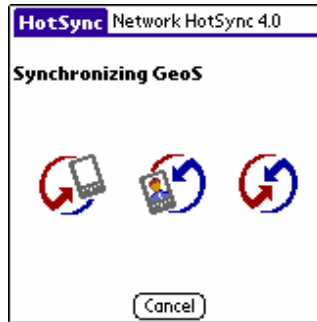
Figura 77. Formulario de Acceso para Sincronización



- El Recolector introduce el Login y Password. Además puede configurar la dirección del servidor e indicar si desea finalizar la recolección de datos para la visita en curso.
- El Recolector puede seleccionar Aceptar o Cancelar.
- Si el Recolector selecciona Aceptar el sistema guarda la información introducida para ser enviada al servidor.
- Si el Recolector selecciona Cancelar, el sistema no almacena el password de autenticación.

- El Recolector activa el proceso de sincronización.
- A través de la interfaz de HotSync Manager, el sistema muestra el progreso de la sincronización.

Figura 78. Progreso de Sincronización



- El sistema envía la información de autenticación al servidor.
- El servidor envía el estado de la autenticación y el tipo de sincronización que debe realizarse (E1).
- El sistema procede a realizar el intercambio de información entre la PDA y el servidor.
- Cuando el intercambio finaliza, el sistema lo notifica a HotSync Manager para actualizar la interfaz de estado.
- El Recolector puede consultar el Log de sincronización disponible a través de la opción Ver Log de HotSync Manager, para visualizar información detallada del proceso, lo cual puede ser útil para identificación de errores en la conexión.

### Flujos de Excepción

E1: Se imprime un mensaje de error en el Log de sincronización que hace referencia a cualquiera de las siguientes eventualidades:

- La conexión con el servidor no se puede establecer
- Ocurrió un error en el servidor
- No se ha configurado un login y/o password.
- La visita en la que participa el Recolector ha sido cancelada.
- Ha ocurrido un error interno en el dispositivo o en el PC donde se ejecuta el HotSync Manager.

## 5.2 Diagrama de clases del sistema

Figura 79. Diagrama de clases del sistema

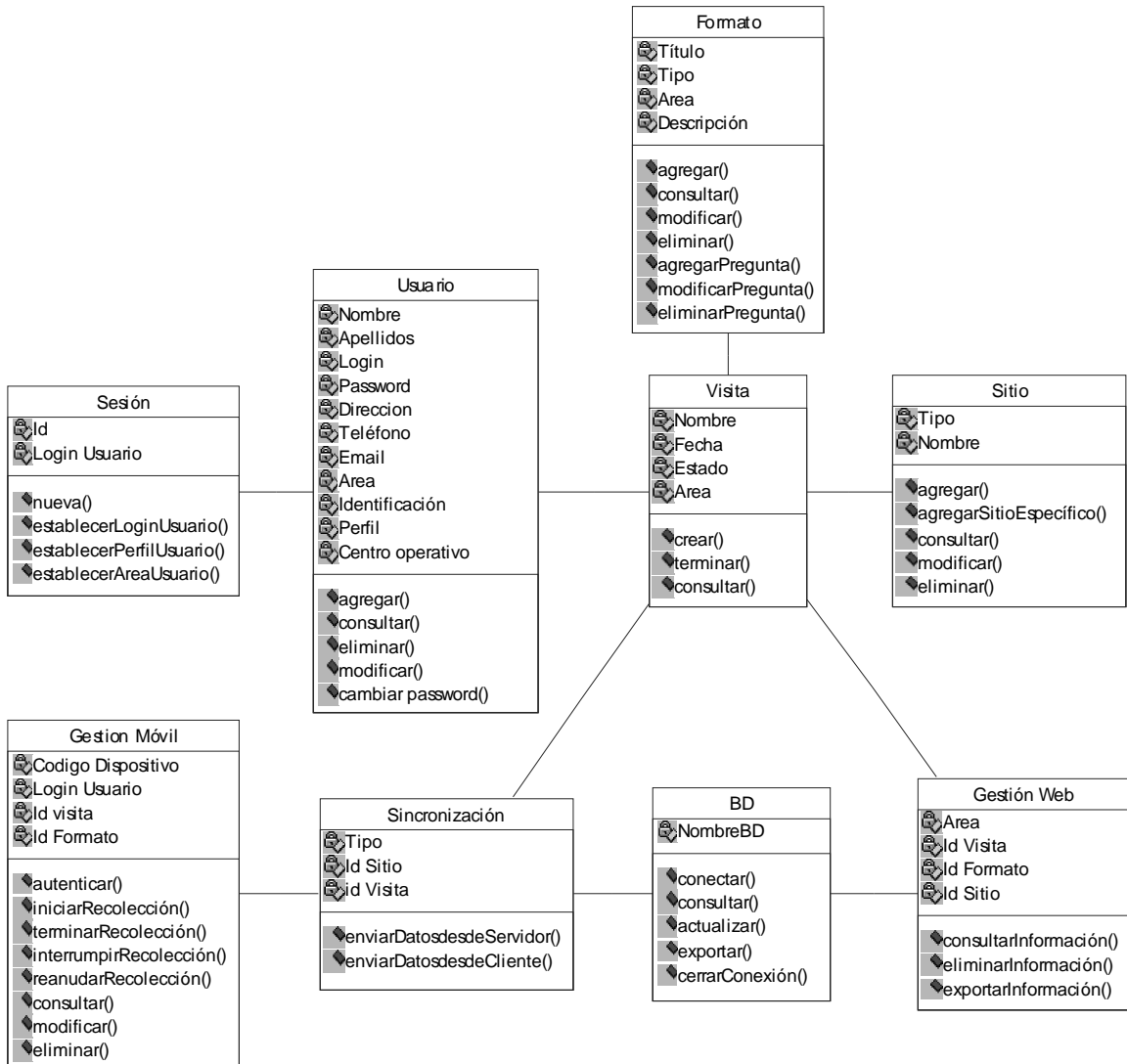
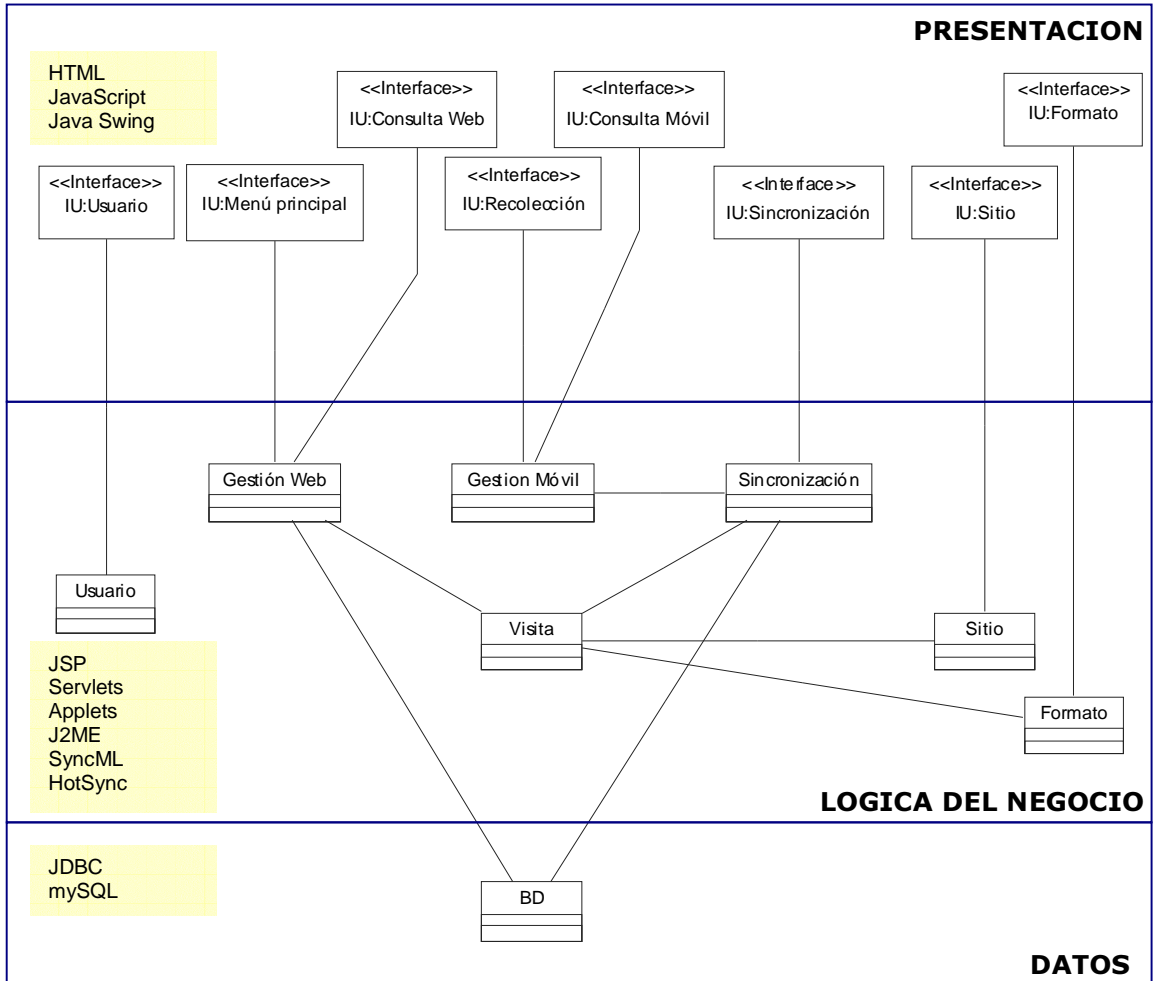


Figura 80. Diagrama de clases Arquitectura en 3 niveles



### 5.3 División en Clases Específicas

#### Grupo de Casos de Uso Gestión sesión

Figura 81. Clases del C.U. Iniciar sesión

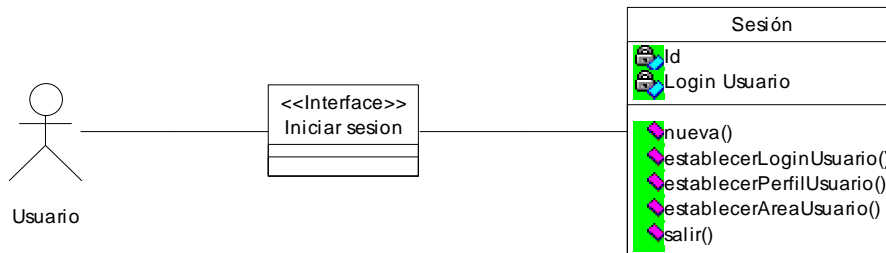
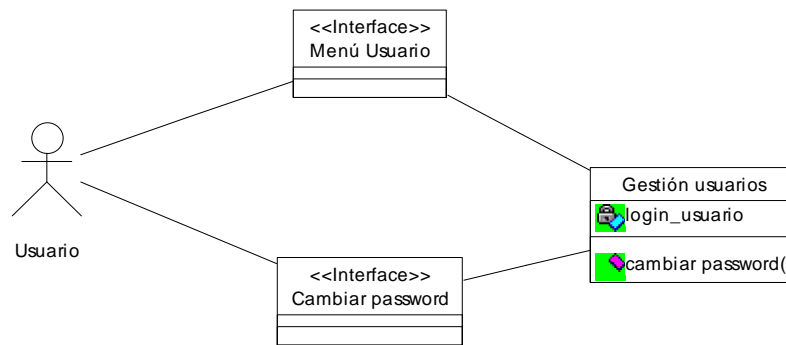
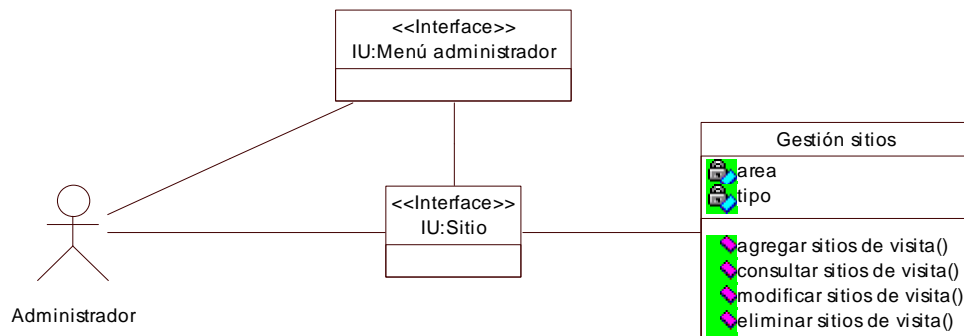


Figura 82. Clases del C.U. Cambiar password



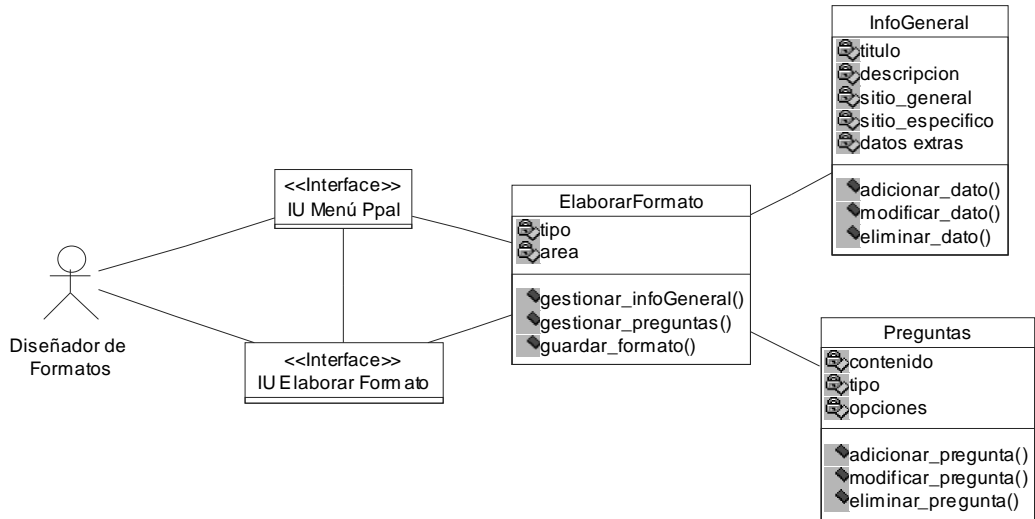
#### Grupo de Casos de Uso Gestión sitios

Figura 83. Clases del C.U. Agregar sitios de visita



Grupo de casos de uso Elaborar Formato

Figura 84. Clases del caso de uso Elaborar Formato



Grupo de casos de uso Recolección de Datos

Figura 85. Clases del caso de uso Recolectar Información en Campo

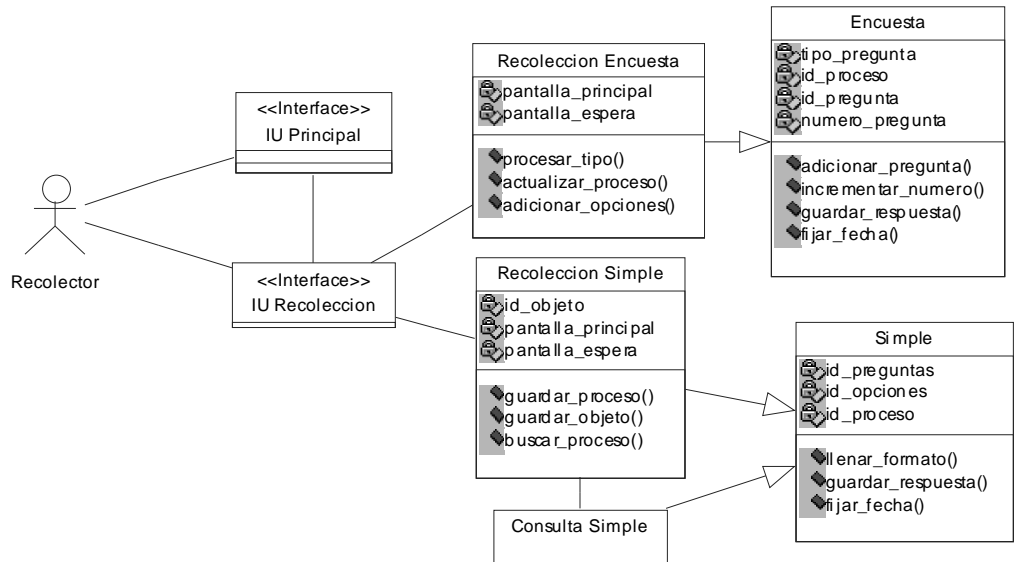
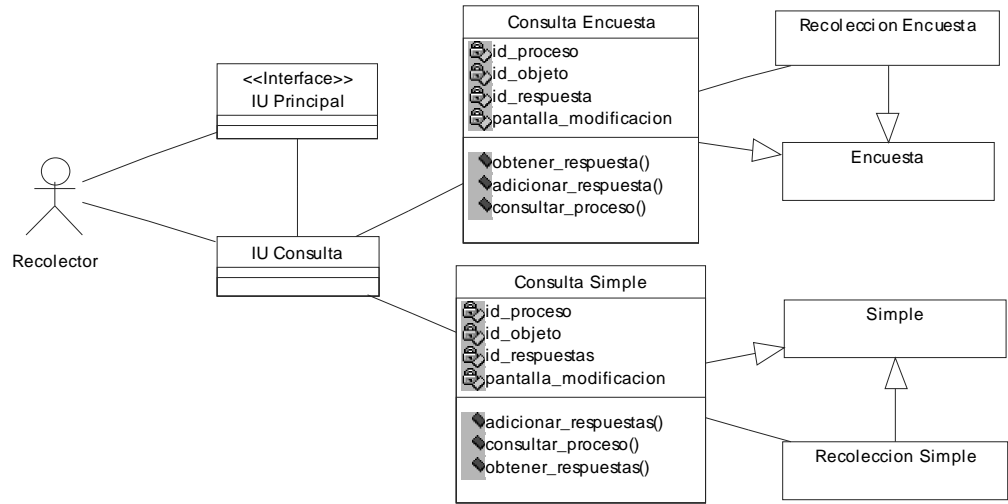




Figura 86. Clases del caso de uso Consultar Información en Campo



Grupo de Casos de Uso Estadísticas

Figura 87. Clase del C.U. Desplegar estadísticas básicas

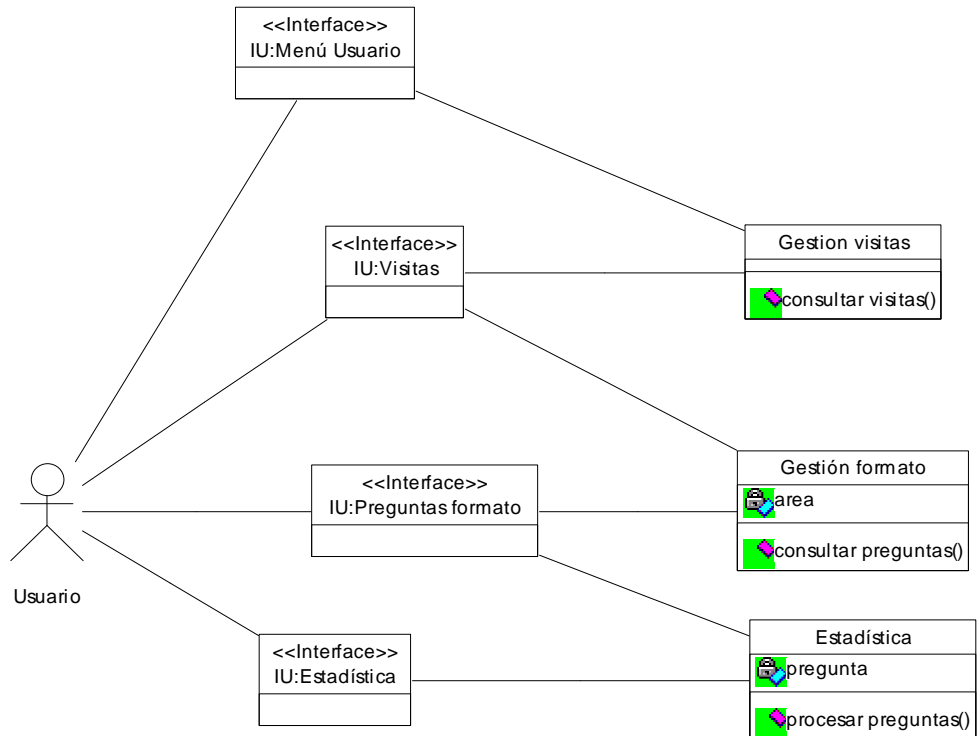
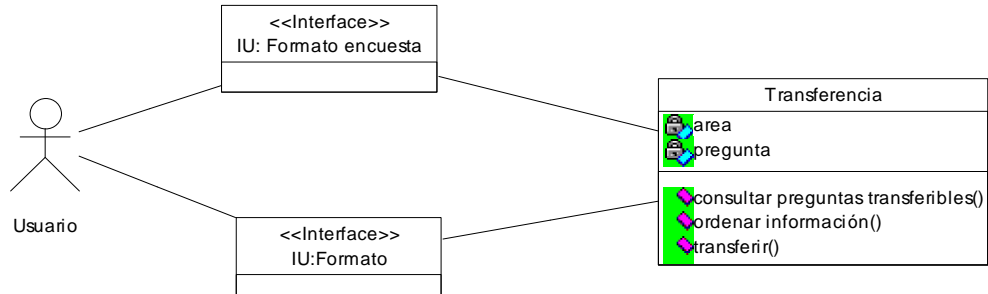
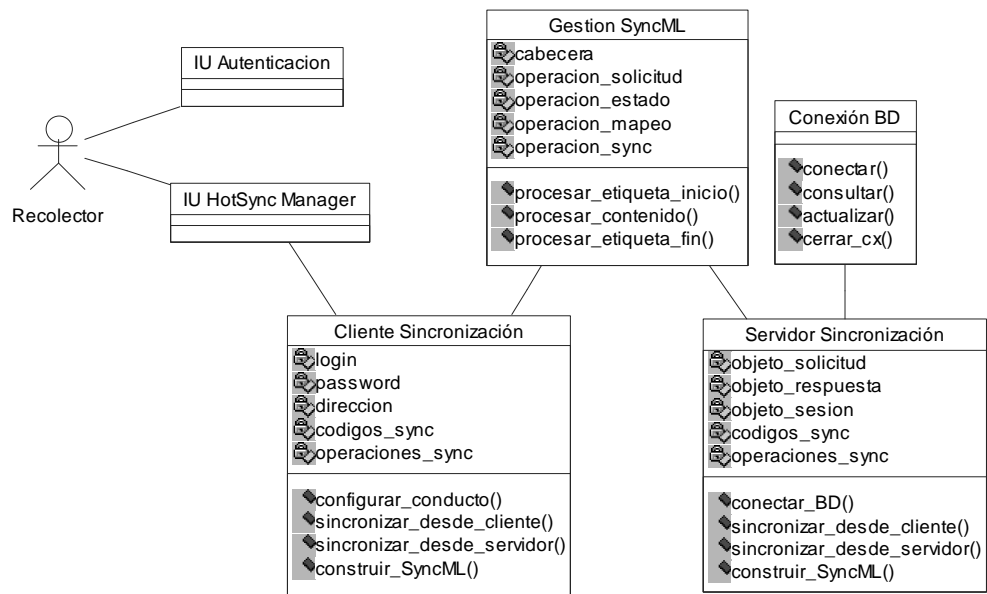


Figura 88. Clase del CU Transferir información

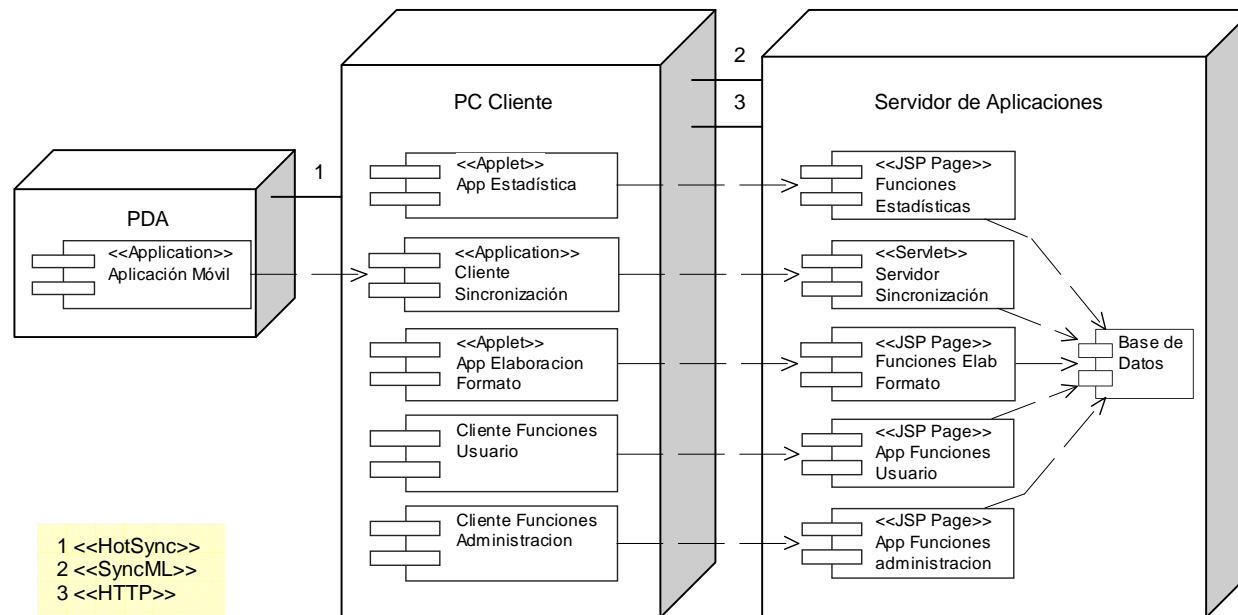


### Grupo de casos de uso Sincronizar Datos

Figura 89. Clases del caso de uso Sincronizar Datos



### 5.4 Diagrama de Implantación



## CONCLUSIONES Y RECOMENDACIONES

- La computación móvil es una de las tecnologías más importantes en el mundo moderno gracias a los múltiples beneficios derivados de la portabilidad de aplicaciones. Sin embargo, el verdadero valor agregado de una aplicación móvil se encuentra en su capacidad para interactuar con los PC tradicionales, sobre todo en ambientes empresariales.
- Debido a las limitaciones hardware que aún poseen los dispositivos móviles modernos, es necesario realizar un estudio cuidadoso de la funcionalidad de la aplicación con el objeto de definir acertadamente las tareas que va a desarrollar el dispositivo. En este sentido, una integración adecuada con máquinas de mayor capacidad de procesamiento como un PC tradicional, juega un papel importante pues es posible trasladar las funciones que requieren mayores capacidades hardware como una sincronización, a este tipo de máquinas. Esto se traduce en aplicaciones móviles más livianas y en un mejoramiento significativo de la experiencia del usuario final.
- Las tecnologías de sincronización juegan un papel fundamental en la integración de los dispositivos móviles al ámbito empresarial. Uno de los resultados más importantes de este trabajo es una implementación parcial del protocolo SyncML, el cual se perfila como un estándar de sincronización en el mercado. A través del estudio del mismo y las pruebas realizadas, es posible concluir que se trata de un protocolo flexible y altamente integrable con las tecnologías web convencionales, lo cual lo hace adecuado para este tipo de ambientes. Adicionalmente, este protocolo hace más flexible la evolución de la aplicación hacia el mundo inalámbrico donde los requerimientos de ancho de banda son más exigentes, gracias a su compatibilidad con tecnologías como WBXML, que en la práctica, demostró una reducción del tráfico entre el 50 y 60%.
- El Sistema Integrado de Recolección y Procesamiento de Datos “GeoS”, es el resultado de un proceso de desarrollo basado en la metodología UP, lo cual garantiza un estudio efectivo de los requerimientos del cliente y un software de calidad ajustado a los mismos. A través del proceso de desarrollo y luego de un estudio técnico detallado, se hizo la selección de la plataforma Java como la más adecuada, principalmente por la facilidad que ofrece para trabajo en red y la posibilidad de construir sistemas más robustos. De esta manera se combinaron tecnologías como J2ME, JSP, Servlets y Applets con tecnologías de soporte como SyncML y HotSync para lograr el mejor resultado posible.
- Las aplicaciones web pueden ser construidas a partir de arquitecturas en 2 niveles siguiendo el modelo cliente/servidor. La aplicación web desarrollada se soporta sobre una Arquitectura web en 3 niveles en donde el nivel de presentación hace uso de herramientas de Macromedia basadas en JavaScript para crear presentaciones llamativas para el cliente, el nivel de lógica de negocio aprovecha las potencialidades de JSP y Servlets y el nivel de datos utiliza las características transaccionales que han sido adicionadas al RDBMS MySQL para mantener ante todo la integridad de los datos.

## BIBLIOGRAFÍA

- QUSAY, Mahmoud. Learning Wireless Java. O' Reilly. 2001.
- FENG, Yu y ZHU, Jun. Developing Wireless Applications Using Java. SAMS. 2001.
- Piroumian, Vartan. Wireless Java 2 ME Platform Programming. Prentice Hall. 2002.
- HUNTER, Jason. Java Servlet Programming. Segunda Edición. 2001:EU.
- MALKS, Dan. Professional JSP. Wrox Press. 2000
- HANNA, Phil. JSP, Manual de Referencia. Mc-Graw Hill. Osbourne Multimedia. 2000.
- RENDON, Alvaro. El lenguaje unificado de modelado UML. Universidad del Cauca. Abril 2000.
- RENDON, Alvaro. Apuntes sobre el proceso unificado para el desarrollo de programas. Universidad del Cauca. Abril 2000.

## ANEXOS

### ANEXO 1 SISTEMA GEOS - DISEÑO DEL SOFTWARE

Diagrama de casos de uso extendidos

Realización de los casos de uso

- Gestión Sesión
- Gestión Usuarios
- Gestión Sitios
- Gestión Visita
- Estadísticas
- Gestión Formato
- Recolección de datos
- Sincronización

Modelo Entidad Relación base de datos central

Modelo Entidad Relación base de datos PDA

### ANEXO 2 PAQUETE DE GRAFICOS MONARCH CHARTS

Introducción

Tipo de Gráfico utilizado para el proyecto GeoS

Arquitectura

- Elementos de un gráfico
- Creando un gráfico básico

### ANEXO 3 MANUAL DE INSTALACION

Manual de instalación \_ parte móvil

- Instalación de HotSync Manager
- Instalación de Palm Desktop
- Creación de cuentas de usuario
- Instalación de programas en la PDA
- Instalación del conducto de sincronización

Manual de instalación \_ parte web

- Instalación del servidor web Apache
- Instalación de Java 2 SDK
- Instalación de Tomcat
- Integración de Apache y Tomcat
- Instalación de MySQL

## CONTENIDO CD ROM

- Información teórica relacionada con el desarrollo del proyecto
- Archivos ejecutables de la aplicación GeoS
- Monografía del proyecto en .pdf
- Anexos del proyecto en .pdf
- Herramientas utilizadas para el desarrollo del proyecto