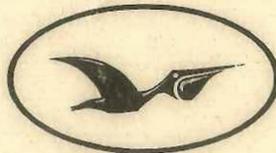


**Centro de Investigación Científica y de
Educación Superior de Ensenada**



CICESE.

**DIVISIÓN DE FÍSICA APLICADA
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN**

**Apoyo a la operación desconectada en sistemas de workflow en dispositivos
móviles**

TESIS

**que para cubrir parcialmente los requisitos necesarios para obtener el grado de
MAESTRO EN CIENCIAS**

**Presenta:
PEDRO ARROYO SANDOVAL**

Ensenada, Baja California, Agosto del 2003.

TESIS DEFENDIDA POR
PEDRO ARROYO SANDOVAL
Y APROBADA POR EL SIGUIENTE COMITÉ

Dra. Ana Isabel Martínez García

Director del Comité

Dr. Jesús Favela Vara

Miembro del Comité

Dr. José Antonio García Macías

Miembro del Comité

Dr. Guillermo Licea Sandoval

Miembro del Comité

Dr. José Rosario Gallardo López

Miembro del Comité

Dr. Jesús Favela Vara

*Jefe del Departamento de
Ciencias de la Computación*

Dr. Luis Alberto Delgado Argote

Director de Estudios de Posgrado

Centro de Investigación Científica y de
Educación Superior de Ensenada



DIVISIÓN DE FÍSICA APLICADA
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

Apoyo a la operación desconectada en sistemas de workflow en dispositivos
móviles

TESIS

que para cubrir parcialmente los requisitos necesarios para obtener el grado de
MAESTRO EN CIENCIAS

Presenta:
PEDRO ARROYO SANDOVAL

Ensenada, Baja California, Agosto del 2003.

RESUMEN de la tesis de **PEDRO ARROYO SANDOVAL**, presentada como requisito parcial para la obtención del grado de MAESTRO EN CIENCIAS en CIENCIAS DE LA COMPUTACIÓN. Ensenada, Baja California. Agosto del 2003.

Título.

Resumen aprobado por:


Dra. Ana Isabel Martínez García
Director de Tesis

Los Sistemas Administradores del Flujo de Trabajo o *Workflow* (WFMS por sus siglas en Inglés) son una tecnología de rápida evolución la cual provee apoyo a la ejecución automática de las tareas en los procesos. Por esta razón, esta tecnología esta siendo cada vez más utilizada por las organizaciones. Por otro lado, estas organizaciones se están desplazando hacia ambientes en los que la movilidad es requerida. En años recientes, hemos visto el uso de tecnologías de cómputo más allá de los ambientes tradicionales de escritorio hacia nuevas tecnologías como los Asistentes Digitales Personales (PDAs), sin embargo, pocas opciones existen para apoyar el *workflow* y la cooperación entre usuarios en plataformas de cómputo no tradicionales como PDAs. Considerando la creciente necesidad de movilidad por parte de las organizaciones y el creciente uso que éstas hacen de dispositivos móviles, es necesario proveer soporte para la integración de estos dispositivos a los ambientes de trabajo. Estas clases de dispositivos observan características especiales como conexiones temporales a redes de comunicación lo cual nos lleva a investigar la operación desconectada como en enfoque para apoyar la movilidad de usuarios en áreas sin soporte a red. La operación desconectada se refiere a la capacidad de un cliente para continuar con sus operaciones cuando y donde conexiones de red no están presentes y por lo tanto, el acceso a un repositorio central de datos no es posible. Con el desarrollo de mecanismos que soporten la operación desconectada en sistemas de *workflow*, los usuarios en las organizaciones pueden trabajar independientemente de una unidad de cómputo central.

En este trabajo se presenta el desarrollo de extensiones al WFMS *SysCoor*, cuya arquitectura provee soporte a la generación semiautomática de sistemas de *workflow* desconectado a través del uso de modelos de procesos organizacionales descritos en el Lenguaje eXtensible de Marcado (XML). En este trabajo se establecen los requerimientos para el apoyo de la operación desconectada en sistemas de *workflow* y se discute la arquitectura del sistema basada en el uso de XML para la especificación de sistemas de *workflow*. La especificación del sistema de *workflow* es generada en XML para permitir la interoperabilidad con otras máquinas de *workflow*. La arquitectura de *SysCoor* consiste de cinco componentes: un generador de sistemas de *workflow*, un repositorio de datos para almacenar modelos y definiciones de sistemas de *workflow*, una máquina de coordinación, un módulo para la operación desconectada y la arquitectura ligera para clientes PDAs. El

desarrollo de los mecanismos para el apoyo a la operación desconectada se presenta a través de una arquitectura ligera para dispositivos PDAs la cual ha sido desarrollada utilizando el lenguaje de programación SuperWaba, sin embargo, la arquitectura descrita puede ser implementada utilizando otros lenguajes de programación para dispositivos PDAs.

Palabras clave: Sistemas administradores de workflow, workflow desconectado, dispositivos móviles, modelo de procesos en XML.

Abstract of the thesis of **PEDRO ARROYO SANDOVAL**, presented as partial requirement, to obtain the master title of **MASTER IN SCIENCIAS in Computer Science**. Ensenada, Baja California, México. August of 2003.

Title.

Abstract approved by:



Dra. Ana Isabel Martínez García
Thesis Director

Workflow Management Systems (WFMS) are fast evolving technology, which provides support to the guided performance of a process' tasks. Thus, this technology is increasingly being used by organizations. On the other hand, organizations are moving forward to environments where mobility is required. In recent years we have seen the spread of computer technologies beyond the traditional desktop environment towards the use of new technologies like Personal Digital Assistants (PDAs). However, few options exist to support workflow and cooperation among mobile users using non-traditional computing platforms such as PDAs. Considering the growing need for mobility and the more frequent use that organizations make of mobile devices, it is necessary to provide support for the integration of them to the work environments. These kinds of devices observe special characteristics such as, intermittent network connectivity. This characteristic leads us to research disconnected operation as an approach to support user's mobility in areas where there are no communication networks, looking for options that enable process agents to continue with their work, and still keep coordinated interactions with other agents and tasks, even if a network connection is not present.

Disconnected operation is the ability of a client to continue his/her work operations when and where network connections are not present and therefore access to the central main data repository is not possible. With the development of mechanisms to support the disconnected operations, users within an organization can work independently of the main computer facility

In this work, we present the development of the extensions for the SysCoor WFMS architecture, which provides support for the semi-automatic generation of disconnected workflow systems through the use of models described in the eXtensible Markup Language (XML). In this work we establish the requirements for supporting disconnected workflow and discuss the architecture of the system based on the use of XML for workflow systems definition. The workflow systems' specification is generated in XML to enable interoperability with other WFMS engines.

The architecture of *SysCoor* consists of five components: a workflow system generator, a data repository to store models and workflow systems' definitions, a coordination engine, an operation disconnection module and a lightweight client architecture intended for PDA devices. The development of the disconnection mechanism is presented in terms of a lightweight architecture for PDA devices developed using the SuperWaba programming language, however, the architecture described can be implemented using other programming tools

Keywords: Workflow Management System, disconnected workflow, mobile devices, process models in XML.

Dedicatoria

*A mis padres, Alicia y Pedro,
gracias a quienes estoy aquí.*

Agradecimientos

Quiero agradecer

Especialmente a mi asesora la Dra. Ana Isabel Martínez García, por toda su ayuda, consejo y apoyo para llevar a cabo este trabajo

A los miembros del comité de tesis, el Dr. Jesús Favela Vara, Dr. Antonio García Macías, Dr. José Rosario Gallardo López y al Dr. Guillermo Licea Sandoval por sus aportaciones en favor de la mejora de este trabajo.

Al Consejo Nacional de Ciencia y Tecnología.

CONTENIDO

Página

CAPÍTULO I. INTRODUCCIÓN	1
I.1 Antecedentes.....	1
I.2 Descripción del Problema.....	5
I.3 Objetivo General	7
I.4 Objetivos Específicos	7
I.5 Contenido de la tesis.....	7

CAPÍTULO II. SISTEMAS DE WORKFLOW Y WORKFLOW DESCONECTADO	10
II.1 Introducción	10
II.2 Modelos de Procesos.....	10
II.3 Workflow	11
II.4 Sistemas Administradores de Workflow (WFMS)	12
II.4.1 Aportaciones de los Sistemas de Workflow en las Organizaciones	14
II.4.2 Diferentes tecnologías en WFMSs.....	15
II.5 Coalición para la Administración del Workflow (WfMC)	18
II.5.1 Meta-modelo de Workflow y el modelado de sistemas de workflow.....	18
II.6 Limitaciones en los Sistemas Administradores de Workflow (WFMS).....	20
II.6.1 Interoperabilidad	21
II.6.2 Apoyo a la movilidad y a la operación desconectada	22
II.7 Operación desconectada en sistemas de workflow	23
II.7.1 Trabajo previo para apoyar la operación desconectada en sistemas de workflow	23
II.8 <i>SysCoor</i>	28
II.8.1 Arquitectura básica de <i>SysCoor</i>	29
II.9 Resumen.....	31

CAPÍTULO III. REQUERIMIENTOS PARA EL APOYO A LA OPERACIÓN DESCONECTADA EN SISTEMAS DE WORKFLOW ..	33
III.1 Introducción.....	33
III.2 Casos de Estudio	33
III.2.1 Escenarios en el Caso de Estudio en el IMSS	34
III.2.2 Escenario del Caso de Estudio en Chateau Camou	39
III.3 Requerimientos obtenidos de los casos de estudio	41
III.4 Requerimientos documentados en la literatura.....	42
III.5 Requerimientos establecidos por <i>SysCoor</i>	43
III.6 Requerimientos genéricos.....	43
III.7 Resumen.....	44

CAPÍTULO IV. ARQUITECTURA DE SYSCOOR Y EXTENSIONES PARA EL APOYO AL WORKFLOW DESCONECTADO	45
IV.1 Introducción	45
IV.2 La Arquitectura de <i>SysCoor</i>	45
IV.3 Funcionalidad de <i>SysCoor</i>	48
IV.4 Extensiones a la arquitectura para apoyar el trabajo desconectado en sistemas de workflow	50
IV.4.1 Mecanismos en el Cliente	54
IV.4.2 Mecanismos en el Servidor	55
IV.5 Análisis y diseño de las extensiones para apoyar el trabajo desconectado en <i>SysCoor</i>	56
IV.5.1 Casos de Uso	56
IV.5.2 Diagramas de Secuencia	62
IV.5.3 Diagramas de Clases	66
IV.6 Resumen	68
CAPÍTULO V. IMPLEMENTACIÓN DE LAS EXTENSIONES PARA EL APOYO AL WORKFLOW DESCONECTADO EN SYSCOOR	69
V.1 Introducción	69
V.2 La arquitectura extendida de <i>SysCoor</i>	69
V.3 Mecanismos en el servidor	71
V.3.1 Transformador de código y modelos base WF y WFD	71
V.3.2 Modelo base de workflow (modelo base WF)	72
V.3.3 Modelo base GUI para la definición de Interfaces Gráficas del Usuario	80
V.3.4 Definición de datos por medio de XML	804
V.3.5 El modelo base de workflow desconectado (modelo base WFD)	84
V.3.6 Implementación del transformador de código	85
V.3.7 Modificaciones al AGP	88
V.3.8 Mecanismo de determinación de desconexión	89
V.3.9 Mecanismo de transferencia de documentos XML	91
V.3.10 Mecanismo de sincronización	94
V.4 Arquitectura cliente	95
V.4.1 Coordinador Local	95
V.4.2 Modelo base para la reconexión	96
V.4.3 Generador de interfaz gráfica	98
V.5 Funcionalidad del sistema	102
V.6 Relación de las actividades con la funcionalidad del sistema	104
V.6.1 Definición y validación de las interfaces gráficas de usuario (GUIs)	107
V.6.3 Generar el sistema de workflow (roles base) y el modelo base WFD	109
V.7 Resumen	118

CAPÍTULO VI. CONCLUSIONES, APORTACIONES Y TRABAJO

FUTURO	120
VI.1 Conclusiones.....	120
VI.2 Aportaciones.....	122
VI.3 Trabajo futuro.....	123
VI.4 Logros adicionales.....	125

REFERENCIAS	127
--------------------------	------------

APÉNDICE A. CASO DE ESTUDIO DEL “PROCESO DE VENTAS” DE LA COMPAÑÍA VITIVINÍCOLA CHATEAU CAMOU.....131

A.1 Introducción.....	131
A.2 Antecedentes.....	132
A.3 Objetivos.....	132
A.4 Captura.....	133
A.5.1 Descripción textual del proceso.....	133
A.5.2 Gráfica rica de la estructura del proceso.....	135
A.6 Aspectos sociales y técnicos.....	137
A.6.1 Sociales.....	137
A.6.2 Técnicos.....	137
A.7 Modelado de Procesos.....	138
A.7.1 Catálogo de usuarios.....	138
A.7.3 Modelo del sistema.....	139
A.7.4 Modelo de los objetivos.....	140
A.7.5 Diagramas de rol actividad.....	140

APÉNDICE B. MODELOS DE PROCESOS EN XML.....150

B.1 Introducción.....	150
B.2 Elementos del DTD del <i>modelo base de transición de estados (modelo base DTE)</i>	150
B.3 Elementos del DTD del <i>modelo base de información (modelo base EI)</i>	151
B.4 Elementos del DTD del <i>modelo base de workflow (modelo base WF)</i>	152
B.5 Elementos del DTD del <i>modelo base de workflow desconectado (modelo base WFD)</i>	156



Lista de figuras

Figura	Descripción	Página
1	Esquema general de un WFMS	13
2	Elementos necesarios para generar un WFMS	29
3	Arquitectura básica de <i>SysCoor</i>	30
4	Arquitectura <i>SysCoor</i> propuesta de García Carrillo y Ramírez Fernández	46
5	Esquema general de la funcionalidad de los sistemas generados por <i>SysCoor</i>	49
6	Arquitectura de <i>SysCoor</i> con extensiones propuestas	55
7	Caso de uso “ <i>generar modelo de workflow desconectado</i> ”	57
8	Caso de uso “ <i>Evaluar la factibilidad de la desconexión de un rol</i> ”	58
9	Caso de uso “ <i>Preparar el rol para ser transferido</i> ”	58
10	Caso de uso “ <i>Notificar roles, actividades y EI preparadas para transferir</i> ”	59
11	Caso de uso “ <i>Procesar transacciones recibidas</i> ”	60
12	Diagrama de secuencia del caso de uso “ <i>Evaluar la factibilidad de la desconexión de un rol</i> ”	62
13	Diagrama de secuencia del caso de uso “ <i>Preparar el rol para ser transferido</i> ”	63
14	Diagrama de secuencia del caso de uso “ <i>Notificar actividades y Entidades de Información (IE) preparadas para transferir</i> ”	64
15	Diagrama de secuencia de los casos de uso “ <i>Procesar transacciones recibidas</i> ”	65
16	Diagrama de clases de las extensiones propuestas para la arquitectura servidor de <i>SysCoor</i>	67
17	Diagrama de clases para la arquitectura cliente para dispositivos PDA	67
18	La arquitectura de <i>SysCoor</i> incluyendo las extensiones para el apoyo a la operación desconectada en sistemas de workflow como se presentó en el capítulo anterior	70
19	Ejemplo de una parte de un <i>modelo base WF</i> que incluye la definición de dos estados del proceso “ <i>Realizando venta</i> ” del caso de estudio realizado en la compañía Chateau Camou	76
20	Un estado del proceso “ <i>Realizando venta</i> ” el cual muestra una actividad que interactúa con otra del mismo proceso y recibe información del segundo	77
21	El estado “ <i>Finalizar</i> ” del proceso “ <i>Realizando venta</i> ” el cual muestra una actividad que envía una entidad de información a otra actividad en otro proceso	78
22	Definición de la actividad <i>a2</i> en la que se implementa la plantilla 4 para servir como pantalla con el usuario que ejecuta la actividad	78
23	DTD para la creación de GUIs con el modelo base GUI	82

Lista de figuras (Continuación)

Figura	Descripción	Página
24	Ejemplo de un <i>modelo base GUI XML</i> que define una pantalla en XML	83
25	Parte de un <i>modelo base WF</i> el cual define la referencia a una plantilla (GUI) (a) y el mismo fragmento del modelo una vez que se agrego la definición de la pantalla dando como resultado <i>modelo base WFD</i> (b)	86
26	Pantalla principal de <i>SysCoor</i> en la que se muestran las principales secciones que la forman	88
27	Flujo de actividades del Servlet <i>determinación</i> en el que se muestran las consultas y llamadas a Servlets que ejecuta	90
28	Algunas de las operaciones que lleva cabo el método <i>analiza()</i> en e orden en el que ocurren	92
29	Ejemplo de un modelo base para la reconexión que muestra la definición de una actividad con una entidad de información para ser reconectadas a la máquina central de workflow	97
30	Las fases a través de las cuales se genera en <i>SysCoor</i> un sistema de workflow con apoyo a la operación desconectada, desde la definición del proceso, su transformación a un sistema de workflow y la transferencia a un dispositivo PDA	100
31	Escenarios de la fase de reconexión a la máquina central (<i>AGP</i>)	101
32	Definición de una pantalla para la captura de un pedido en el proceso de “Realizando Venta”.	107
33	El detallado del <i>modelo base WF</i> utilizando el editor que implementa el <i>generador de sistemas</i>	109
34	Generación de los <i>roles base</i> del <i>modelo base WFD</i> utilizando el <i>modelo base WF</i>	110
35	Los sistemas de workflow generados	111
36	Pantalla principal de <i>SysCoor</i> (<i>AGP</i>)	112
37	Pantalla principal de <i>SysCoor</i> (<i>AGP</i>) indicando al usuario que la desconexión no es posible en ese momento	113
38	Pantalla principal de <i>SysCoor</i> (<i>AGP</i>) indicando al usuario la definición del sistema está lista para ser transferida	114

Lista de figuras (Continuación)

Figura	Descripción	Página
39	Escenarios en el dispositivo PDA en el orden en el que ocurren. Primero se (a) consultan los roles que se encuentran en el dispositivo, para cargar nuevos roles (b) se toma la opción “cargar roles”, el dispositivo se conecta (c) y carga los roles que han sido preparados (d) para terminar con la ejecución de alguno de estos roles (e)	115
40	Pantalla principal de <i>SysCoor (AGP)</i> indicando al usuario que la actividad “ <i>revisar pedido</i> ” no puede llevarse a cabo, ya que depende de otra que aun no termina	116
41	Escenarios en el dispositivo PDA en el orden en el que ocurren cuando se requiere reconexión. Primero se toma la opción “reconectar” (a), el dispositivo muestra los roles que se encuentran cargados (b) de los cuales el usuario debe elegir aquellos que se reconectarán, el coordinador local notifica la reconexión (c), por último, se muestran los roles que aún están en el dispositivo (d)	117
42	Interfaz gráfica principal de <i>SysCoor (AGP)</i> indicando al usuario la reconexión de un rol	118
43	Gráfica rica de la estructura del <i>proceso de ventas</i>	136
44	Modelo del sistema del proceso de ventas	139
45	Modelo de los objetivos del proceso de ventas	140
46	Diagramas RAD globales de las actividades del <i>vendedor</i> y del <i>encargado de facturación</i>	141
47	Diagrama RAD del subproceso <i>realizando la venta</i>	142
48	Diagramas RAD que forman el subproceso <i>llenando formato de pedido</i>	143
49	Diagramas RAD del subproceso procesar pedido	144
50	Diagramas RAD del subproceso de recibir y entregar mercancía entre <i>vendedor</i> y el <i>encargado de facturación</i>	145
51	Diagramas RAD del subproceso de recibir y entregar mercancía entre <i>vendedor</i> y <i>cliente</i>	145
52	El <i>vendedor</i> preparando la relación diaria	146
53	Diagrama RAD del subproceso <i>Entregando Facturas para Cobro</i>	146
54	Diagramas RAD del subproceso de cobrar facturas	147
55	El <i>vendedor</i> realizando reporte de pagos	147
56	Diagramas RAD del subproceso <i>recibiendo pagos</i>	148
57	Diagramas RAD del <i>vendedor</i> realizando el reporte de visitas	149
58	El <i>vendedor</i> entregando facturas autorizadas	149
59	<i>Documento base de transición de estados</i>	151
60	<i>Documento base de información</i>	152
61	<i>Documento base del workflow</i>	154
62	<i>Documento base del workflow desconectado</i>	156

Lista de tablas

Tabla	Descripción	Página
I	Cuadro comparativo de las diferentes propuestas para el apoyo al trabajo desconectado en sistemas de workflow	26
II	Categorías y su descripción	27
III	Descripción de algunos de los elementos y sus atributos del DTD del <i>modelo base WF</i> utilizado para la definición de un sistema de workflow	73
IV	Ejemplo de <i>GUIs</i> en <i>SysCoor</i> para el Web y para <i>SysCoor</i> en PDAs	80
V	Descripción del elemento <i><elementogui></i> y sus atributos	82
VI	Elementos GUI básicos y su representación en XML	83
VII	Estados de las entidades de información y su descripción	90
VIII	Estados de las actividades y su descripción	90
IX	Fases de funcionalidad de <i>SysCoor</i> y los usuarios involucrados	102
X	Relación actividad-funcionalidad	104

Capítulo I. Introducción

I.1 Antecedentes

Actualmente las organizaciones enfrentan los efectos de los cambios tecnológicos y de una mayor dinámica en la realización de sus actividades [Silver, 1999]. Estos aspectos establecen la necesidad de llevar a cabo estudios que le permitan a estas organizaciones contar con un mejor entendimiento de la forma en la que se están llevando a cabo las actividades dentro de la organización. A través de estos estudios de procesos, es posible exponer el comportamiento de los procesos a un análisis racional y a la investigación para comprobar su consistencia [Warboys *et al.*, 1999], así como buscar opciones para mejorar la forma en que el trabajo se está llevando a cabo.

Con la finalidad de lograr un entendimiento común de los procesos existentes en una organización por parte de sus miembros, así como la documentación uniforme de éstos, se requiere de técnicas que apoyen las tareas de definición y análisis de los procesos. Entre estas técnicas, la *Ingeniería de Procesos (IP)* es de las más utilizadas. La *IP* es definida como la colección de técnicas para el análisis, diseño y evaluación de los procesos organizacionales, basadas en el uso del modelado de procesos [Kawalek, 1998].

Un modelo es una representación abstracta de la realidad. El propósito de un modelo es reducir la complejidad en el entendimiento de éste, eliminando los detalles que no influyen en el comportamiento de un fenómeno [Curtis *et al.*, 1992]. Un proceso es definido como “*un conjunto de pasos parcialmente ordenados con la finalidad de alcanzar un objetivo*” [Humphrey y Feiler, 1992]. Algunos de los principales elementos que componen un proceso son: *agentes* que son humanos o máquinas encargados de la ejecución de pasos en el proceso, *roles* que son unidades funcionales de responsabilidad asignadas a agentes, y *artefactos* comprendidos por productos modificados o creados durante la ejecución del proceso.

Existen varias técnicas de modelado para la representación de procesos, entre las cuales los Diagramas de Integración Definida (IDEF0) y los Diagramas de Rol Actividad (RAD) están entre los más importantes [Miers, 1996]. Los diagramas IDEF0 representan los elementos del proceso con cajas y flechas. Las cajas representan las actividades o funciones del sistema, las flechas representan la información o productos necesarios para llevar a cabo las actividades [Miers, 1996]. Los diagramas Rol Actividad (RADs) modelan el proceso desde el punto de vista de roles e interacciones entre roles. Los roles llevan a cabo actividades y toman decisiones acerca de qué hacer y cuándo hacerlo de acuerdo a reglas de la organización. Los roles pueden llevar a cabo actividades en paralelo y condicionadas de acuerdo a las reglas mencionadas [Miers, 1996; Ould, 1995]. Estos diagramas capturan las tareas, objetivos y agentes involucrados en el proceso, en roles, actividades, decisiones e interacciones [Rojas y Martínez, 1998].

Otros diagramas como el de transición de estados STD [Booch *et al.*, 1999] son también utilizados durante el modelado del proceso para capturar los aspectos dinámicos del mismo.

A través de la realización de estudios de procesos usando el modelado de procesos, las organizaciones pueden lograr un entendimiento de sus procesos fundamentales y buscar opciones para hacerlos más eficientes a través del apoyo automatizado a éstos [Curtis *et al.*, 1992].

El modelo de un proceso es una descripción abstracta de un proceso actual o de uno propuesto que representa elementos seleccionados que son considerados importantes para el propósito del modelo y cuya realización o “*enactment*” por un humano o una máquina se puede lograr [Curtis *et al.*, 1992]. Por *enactment*, entendemos la ejecución simultánea y sincronizada de un proceso llevada a cabo por un humano a través de un modelo ejecutable del mismo, utilizando el soporte que brinda la computadora al hombre [Fernström y Lennart, 1991].

Sin embargo, los modelos de procesos son solo la representación de la secuencia de eventos que ocurren dentro de una organización; el *enactment* de un proceso es llevado a cabo por tecnologías como los sistemas de automatización del flujo de trabajo (*sistemas de workflow*).

Actualmente, la tecnología existente ofrece los medios necesarios para apoyar el *enactment* de los procesos, facilitando además el trabajo cooperativo entre los miembros del grupo a través de soluciones innovadoras. Para lograr este objetivo se ha elaborado software que toma la definición del proceso para generar sistemas de workflow hacia el *enactment* de los procesos, este tipo de software es conocido con el nombre de sistemas administradores del workflow (*Workflow Management System - WFMS*)

Un Sistema Administrador del Workflow (*WFMS*) es un sistema que automatiza el flujo de información y control en un proceso a través del manejo y administración de la secuencia de actividades dentro del mismo, así como la invocación de recursos humanos y/o tecnológicos [WfMC, 1999]. De esta forma, los agentes del proceso (las personas involucradas) se concentran sólo en la ejecución de sus actividades, permitiendo al *WFMS* el manejo del flujo y control del trabajo.

Tradicionalmente los *WFMSs* han sido diseñados bajo consideraciones de diseño y arquitectura que proponen la implementación de esquemas cliente-servidor. En este enfoque la información sobre el proceso y sus estados es almacenada en una estructura centralizada compuesta de bases de datos y redes de comunicación [Alonso *et al.*, 1995], lo cual requiere que los usuarios que participan, mantengan una conexión constante a una red de comunicación en la que el *WFMS* reside. Esta centralización permite sincronizar y controlar la realización del proceso de una forma sencilla con la desventaja de establecer limitaciones para la movilidad.

Nuevos enfoques como las arquitecturas de workflow implementadas para trabajar en el Web (sistemas basados en Web) han incrementado la disponibilidad de sistemas de

workflow para usuarios distribuidos [Bolcer, 2000], aumentando de esta forma las posibilidades de que los usuarios se encuentren geográficamente dispersos a lo largo de varios lugares, y aun así, mantener la ejecución de tareas coordinadas por estos sistemas. Sin embargo, este enfoque aun requiere de la existencia de una conexión permanente a una red de comunicación para poder utilizar el sistema de workflow.

En contraste, las organizaciones empiezan a trabajar en ambientes con recursos variables, heterogéneos y donde la movilidad es requerida o bien las conexiones a redes de comunicación son temporales. Hay necesidad de desarrollar sistemas para el apoyo al trabajo en grupo en esos ambientes [Litiu y Prakash, 2000]. La idea de movilidad no debe ser limitación para mantener los flujos normales de trabajo y la cooperación en las organizaciones, tampoco para romper con los conceptos de pertenencia a un grupo de trabajo. De aquí que la idea de movilidad ofrezca extensiones interesantes a los sistemas de workflow existentes con la finalidad de ofrecer un soporte más flexible hacia estas nuevas formas de trabajo.

En años recientes se ha visto la expansión de las tecnologías de computación más allá del enfoque tradicional de computadoras de escritorio, hacia nuevas tecnologías como asistentes personales digitales PDAs (*personal digital assistants*) y teléfonos celulares inteligentes en donde las personas pueden almacenar y recuperar información personal relevante [Greenberg *et al.*, 1999]. Estos dispositivos ofrecen la idea de movilidad pero ésta no queda restringida a su uso, ya que se expande a computadoras tipo laptop e inclusive a computadoras de escritorio que se conectan ocasionalmente a una red [Alonso *et al.*, 1995].

De esta forma los usuarios de sistemas de workflow incrementalmente se han vuelto móviles, conectándose desde varios puntos utilizando una variedad de dispositivos. Sin embargo, existen pocas opciones para apoyar los sistemas de workflow en usuarios móviles que utilizan dispositivos como PDAs [Bolcer, 2000]. Es importante remarcar que la movilidad no debe ser una limitación para mantener los flujos regulares de trabajo ni una barrera para la colaboración en la organización, lo que da lugar a buscar opciones para

permitir a los usuarios de estos dispositivos continuar en contacto con los procesos en los que intervienen aún cuando no se encuentran conectados a la red en la que normalmente desarrollan su trabajo. La idea principal es la de permitir a los agentes continuar con la realización de sus roles y seguir relacionándose con el resto de los agentes, roles y artefactos de los procesos en los que intervienen, de forma coordinada aún cuando no están bajo una misma red cerrada. Esto da lugar a la idea de roles que presentan movilidad y a sistemas de workflow desconectados.

Considerando el uso cada vez mayor que se hace en las organizaciones de dispositivos móviles como PDAs y teléfonos celulares inteligentes su integración al esquema de trabajo debe ser apoyada.

Por otro lado, estos dispositivos tienen características especiales como presentar conexiones temporales a redes de comunicación, limitaciones en el almacenamiento, y algunos, limitaciones de despliegue gráfico.

La característica de estos dispositivos de realizar conexiones temporales a redes de comunicación nos lleva a buscar opciones que permitan a los agentes de un proceso continuar con sus actividades coordinadas aun en ambientes donde las redes de comunicación no se encuentran presentes.

I.2 Descripción del Problema

En el período 2000-2001 se formó el *Grupo de Ingeniería de Procesos (GIP)* en el departamento de Ciencias de la Computación del CICESE. Dentro de las actividades de este grupo, García Carrillo desarrolló un Sistema Administrador del Flujo de Trabajo (WFMS) llamado *SysCoor* cuya arquitectura permite la generación semi-automática de sistemas de workflow que pueden ser ejecutados en navegadores Web [García Carrillo, 2001]. El diseño de esta arquitectura se basó en el desarrollo de sistemas de workflow definidos a través del concepto de rol y la coordinación entre ellos para la automatización

de procesos organizacionales. Un rol dentro de un proceso es una unidad funcional de responsabilidad asignada a un agente [Humphrey y Feiler, 1992]. Un rol representa un conjunto definido de actividades agrupadas con el fin de lograr un objetivo.

Más tarde, Ramírez Fernández extendió la arquitectura propuesta por García Carrillo con la finalidad de ampliar la funcionalidad de *SysCoor* [Ramírez Fernández, 2002].

La arquitectura propuesta por García Carrillo y Ramírez Fernández implementó la definición de un modelo base para la representación de sistemas de workflow utilizando el Lenguaje de Marcado eXtensible (XML – eXtensible Markup Language). Al definirlo en XML se permitió contar con un modelo independiente de la plataforma y reutilizable en otras máquinas de workflow capaces de interpretar esta definición. Las reglas bajo las que se construye este modelo base fueron definidas a través de una Definición de Tipo de Documento (DTD, por sus siglas en inglés). Un DTD permite establecer los elementos y las reglas sintácticas y léxicas para estructurar y analizar un documento XML.

Los sistemas de workflow generados en base a la arquitectura propuesta por García Carrillo y Ramírez Fernández [García Carrillo, 2001; Ramírez Fernández, 2002], estaban orientados al Web, permitiendo a usuarios distribuidos participar en la ejecución de sistemas de workflow.

Considerando la necesidad de apoyar este tipo de tecnología en usuarios móviles que requieren desconexiones de una red de comunicación, en este trabajo buscamos establecer los mecanismos que permitan apoyar el trabajo cuando este requiere movilidad, e integrar estos mecanismos a la herramienta *SysCoor* para permitir a los usuarios de dispositivos PDA continuar en contacto con los procesos organizacionales en los que intervienen, aún cuando no se encuentran conectados a la red en la que normalmente desarrollan su trabajo. La idea principal es la de permitir a los agentes continuar con la

realización de sus roles y seguir relacionándose con el resto de los agentes, roles y artefactos de los procesos, de forma coordinada aún en desconexión.

I.3 Objetivo General

Definir los mecanismos necesarios para apoyar la operación desconectada en sistemas de workflow, implementándolos en la herramienta *SysCoor* para proveer apoyo a procesos que se realizan de forma desconectada, por medio de sistemas de workflow capaces de trabajar en dispositivos PDA.

I.4 Objetivos Específicos

- 1) Definir un modelo base en XML para la representación de un sistema de workflow para ser ejecutado en un dispositivo PDA en desconexión.
- 2) Analizar la arquitectura de la herramienta *SysCoor* para la generación de sistemas de workflow con el objeto de proponer extensiones a dicha arquitectura para apoyar el desarrollo de sistemas de workflow que soporten movilidad.
- 3) Desarrollar las extensiones propuestas a la arquitectura dando lugar a un software para la generación de sistemas de workflow que soporten movilidad, extendiendo de esta manera el apoyo que *SysCoor* puede dar a la etapa de soporte.

I.5 Contenido de la tesis

Esta tesis esta constituida por seis capítulos y dos apéndices, los cuales se describen brevemente a continuación.

En el capítulo II se presenta una introducción al área en la que se desarrolla este trabajo: los sistemas de workflow. El objetivo principal es establecer la definición, características y necesidades presentes en estos sistemas, se discuten los términos básicos sobre el concepto de workflow, la tecnología que se ha desarrollado para proveer apoyo automatizado al workflow (sistemas de workflow) y se plantean algunas de las limitaciones de esta tecnología, se aborda el problema de la operación móvil y desconectada, tema principal de esta tesis, y se presentan los trabajos más importantes que se han llevado a cabo con el fin de apoyar este tipo de operación.

En el capítulo III se describe el análisis de los requerimientos para apoyar el trabajo móvil y desconectado en WFMSs a través de la descripción de escenarios obtenidos a partir de la realización de dos casos de estudio, del análisis de los requerimientos establecidos por trabajos anteriores, y por último, los requerimientos que la arquitectura de *SysCoor* (base tecnológica de nuestra propuesta) establece.

En base a los requerimientos establecidos en el capítulo III, en el capítulo IV se describe la propuesta para extender la funcionalidad actual de *SysCoor* hacia la generación de sistemas de workflow que soporten el trabajo móvil y desconectado en sistemas de workflow. Se presenta parte del análisis y diseño de las extensiones propuestas para la arquitectura por medio del uso del Lenguaje de Modelado Unificado (*Unified Modeling Language - UML*).

En el capítulo V se describe el desarrollo llevado a cabo con la finalidad de extender la funcionalidad de *SysCoor*. Aquí se establecen puntualmente los cambios que fueron realizados a algunos de los elementos existentes y las extensiones que fueron agregadas a la herramienta *SysCoor*. La implementación realizada se presenta en términos de la descripción de la tecnología utilizada y su funcionalidad. Adicionalmente, este capítulo muestra a través de un escenario de uso la finalidad que las extensiones desarrolladas para el apoyo al trabajo móvil desconectado cumplen. La implementación realizada fue

evaluada en términos de funcionalidad, por medio del desarrollo de un sistema de workflow desconectado para un proceso de uno de los casos de estudio.

Finalmente el capítulo VI presenta las conclusiones y aportaciones de este trabajo, así también, se señalan algunas de las áreas que representan retos relacionados con esta tesis.

El apéndice A contiene la descripción de uno de los casos de estudio realizados. El apéndice B contiene los DTDs que son utilizados para la generación de los sistemas de workflow desconectado.

Capítulo II. Sistemas de Workflow y Workflow Desconectado

II.1 Introducción

El presente capítulo ofrece una introducción al área en la que se desarrolla este trabajo: los sistemas de workflow. El objetivo principal es establecer los conceptos básicos relacionados con el área, definiciones, características y necesidades presentes en estos sistemas. Con este fin, se discuten los términos básicos sobre el concepto de workflow, la tecnología que se ha desarrollado para proveer apoyo automatizado al workflow (sistemas de workflow) y se plantean algunas de las limitaciones de esta tecnología. Por otro lado, se describe el meta-modelo que ha sido definido como base general para la representación del workflow. Este aspecto es fundamental ya que si se requiere definir un sistema de workflow necesitamos de un marco establecido para su representación.

Se presentan algunas de las limitaciones existentes en los sistemas de workflow actuales, en particular se aborda la necesidad de contar con apoyo a la operación móvil y desconectada. En este aspecto, se presentan los trabajos más importantes que se han llevado a cabo con el fin de apoyar este tipo de operación. Lo anterior se establece con el objetivo de analizar el trabajo que se ha realizado y los retos que se presentan en esta área. Por último, se presenta una breve introducción a la arquitectura de *SysCoor*, una herramienta que da apoyo al workflow y que representa la base sobre la que se prueba la propuesta tecnológica de este trabajo.

II.2 Modelos de Procesos

Como se estableció en el capítulo anterior, un modelo es una representación abstracta de la realidad y su propósito es el de reducir la complejidad en el entendimiento de una situación determinada [Curtis *et al.*, 1992]. En el contexto de las organizaciones, los

modelos de procesos permiten capturar el comportamiento de un proceso establecido. De acuerdo con Warboys, cuando se captura un proceso dentro de una organización a través de un modelo, es posible exponer su comportamiento a un análisis racional y a la investigación para comprobar su consistencia [Warboys *et al.*, 1999]. Más aún, este autor afirma que la representación de los procesos de una organización a través de modelos, provee de un contenedor de conocimiento para la organización, lo que permite facilitar el aprendizaje dentro de la misma.

Warboys hace también una clasificación de los diferentes tipos de modelos de procesos: *modelos pasivos* y *modelos activos* [Warboys *et al.*, 1999]. Con el término de *modelos pasivos* los autores se refieren a aquellas representaciones estáticas de un proceso. El modelo, una vez que es creado, es independiente del objeto modelado. Este último podrá cambiar o no. Si las cosas cambian, el modelo podrá ser revisado y modificado, pero no existe la liga directa entre estos dos eventos. Ejemplos de estos modelos son los diagramas de flujo.

Los *modelos activos* por otro lado, son construidos en un medio que permite mantener la relación entre el objeto modelado y el modelo. Estos modelos permiten monitorear y mantener el control del proceso que ha sido modelado y conocer el estado en el que éste se encuentra. La representación cambia como cambian los estados del proceso. Estos modelos activos son principalmente encontrados en los sistemas de workflow [Fischer, 1995] y en general en cualquier tipo de sistema cuya premisa sea mantener el control sobre un proceso. En la siguiente sección, se establecen los conceptos necesarios para entender la naturaleza de los sistemas de workflow.

II.3 Workflow

El término *flujo de trabajo* o workflow, se refiere al flujo de datos y de control entre actividades de un proceso [Barbara *et al.*, 1996]. Por otro lado, un sistema de workflow involucra la automatización de procedimientos en los cuales documentos,

información y tareas son pasados entre los participantes del proceso de acuerdo a reglas bien definidas, para lograr, o contribuir al cumplimiento de un propósito [WfMC, 1995].

Por lo general, se asocia el término de workflow con el de *Reingeniería de Procesos* [Silver, 1995], el cual se refiere al análisis, modelado, definición e implementación operacional de los procesos en una organización. De aquí que el modelado de un proceso sea la parte esencial para la definición del workflow. Por otro lado, la representación de un proceso a través de modelos es solo una descripción de la secuencia de eventos que suceden en las tareas de una organización. La ejecución real de esos procesos es generalmente efectuada por un Sistema Administrador del Workflow (*Workflow Management System - WFMS*), el cual toma la definición del proceso y lo implementa de forma activa. En este punto es importante distinguir entre el concepto de *WFMS* y el de sistema de workflow. Los *WFMSs* (también conocidos como máquinas de workflow), definen, administran y generan el sistema de workflow el cual es la representación del modelo de forma activa.

II.4 Sistemas Administradores de Workflow (WFMS)

Un sistema administrador del workflow (*Workflow Management System - WFMS*) es un sistema que automatiza el flujo de información y control en un proceso a través del manejo y administración de la secuencia de actividades dentro del mismo, así como la invocación de recursos humanos y/o tecnológicos [WfMC, 1999]. De esta forma, los agentes del proceso (las persona involucradas) se concentran solo en la ejecución de sus actividades, permitiendo al *WFMS* el manejo del flujo y control del trabajo.

Los *WFMSs* han atraído la atención de las organizaciones por su capacidad para mejorar la eficiencia de las mismas a través de la automatización total o parcial de los procesos. Técnicamente, estos sistemas apoyan el modelado, la ejecución coordinada y el monitoreo de las actividades que se llevan a cabo dentro de la organización que implementa el sistema. En general los *WFMSs* son herramientas colaborativas que permiten a un administrador definir la estructura de un proceso, sus reglas, excepciones y recursos con los

cuales dicho proceso será ejecutado. Los usuarios o agentes del proceso cooperan en la ejecución del mismo delegando al WMFS la coordinación de sus actividades, de forma que estos agentes solo se concentran en la ejecución de la tarea. La figura 1 muestra el esquema general de un WFMS. En esta figura se muestra como la entrada de esta clase de sistemas es el modelo de un proceso, y cómo es que una máquina de coordinación se encarga de distribuir el trabajo entre los agentes del proceso.

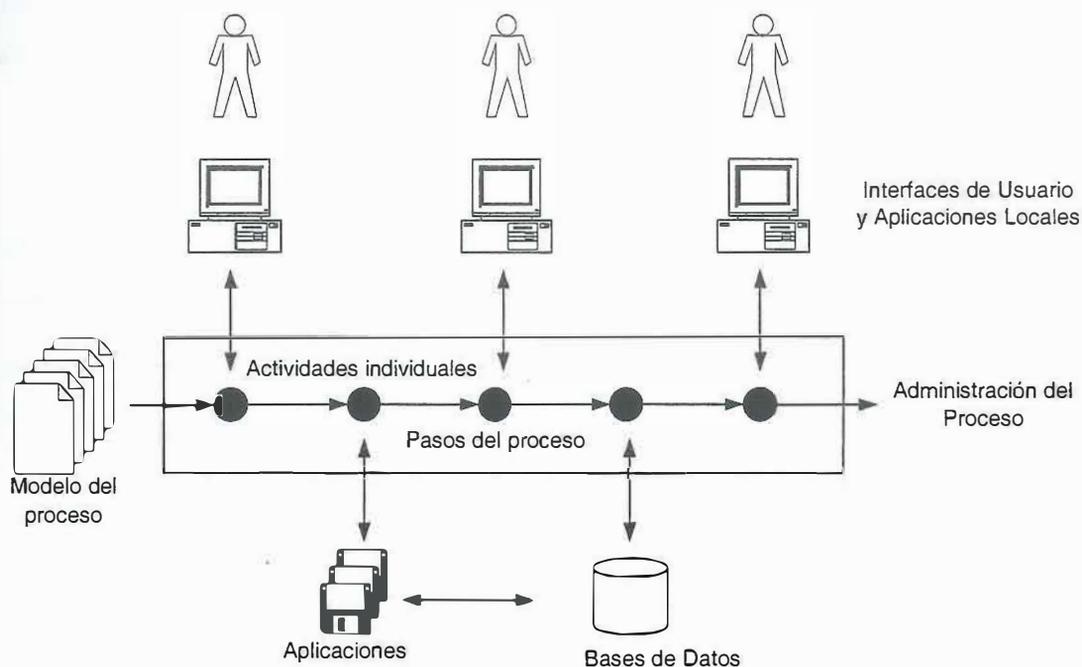


Figura 1. Esquema general de un WFMS

Para la definición de un sistema de workflow, es necesario definir los parámetros del proceso que se desea automatizar, como por ejemplo, los actividades, el orden y las condiciones bajo los cuales estas actividades son llevadas a cabo, incluyendo aspectos como el flujo de datos entre éstas, los responsables de las tareas y las aplicaciones (bases de datos, editores de texto, hojas de cálculo) que se usan en cada actividad [Alonso *et al.*, 1997].

Una vez que el proceso ha sido definido, la ejecución del mismo es realizada por el sistema: el WFMS entrega las tareas que cada agente del proceso debe ejecutar, recolecta

los resultados de la ejecución, determina los siguientes pasos, controla las actividades de cada usuario y detecta el momento en el que el proceso ha sido terminado [Alonso *et al.*, 1995]. Es importante señalar que un WFMS puede no automatizar completamente las tareas, a partir del hecho de que la intervención humana puede ser necesaria para solucionar y determinar qué hacer en caso de eventos no previsible. Sin embargo, el uso de estos sistemas simplifica la coordinación entre los usuarios del proceso, lo cual representa una de las principales aportaciones que los WFMSs han dado a las organizaciones. Con el fin de valorar estas aportaciones, a continuación se presentan algunas de las más relevantes.

II.4.1 Aportaciones de los Sistemas de Workflow en las Organizaciones

Como se mencionó con anterioridad, la implementación de WFMSs dentro de las organizaciones ha permitido observar una serie de beneficios, entre los cuales se encuentran [Silver, 1999]:

- a) *Reducción en los tiempos.* Uno de los mayores beneficios que brindan los WFMSs a las organizaciones es la reducción en el tiempo que requiere para completar la ejecución de un proceso. Con la eliminación del tiempo que involucra el mover material de trabajo (documentos, formas, etc.) entre los agentes involucrados en él, y mediante la automatización de ciertas tareas, el tiempo de ejecución del proceso se reduce.
- b) *Administración del trabajo.* Los WFMSs permiten filtrar, establecer prioridades y organizar el trabajo de los individuos por medio del establecimiento de reglas que definen la secuencia del trabajo y los recursos para cada una de las tareas, llevando el trabajo adecuado al recurso adecuado.
- c) *Control del Proceso.* Por medio del establecimiento de las reglas del proceso y su introducción a un sistema de cómputo, los WFMSs permiten automatizar el

cumplimiento de las estas. Reglas como *¿qué material necesita un procesamiento específico?* o *¿qué documentos necesitan ser enviados junto con determinado formato?* Son definidas en un formato entendible por la computadora de forma que ésta asegura su cumplimiento.

- d) *Apoyan la cooperación y el compartir conocimiento.* Estos sistemas establecen un repositorio de conocimiento que describe la forma en que se llevan a cabo las tareas en una organización. Lo anterior permite el establecimiento de un conocimiento común para todos los miembros de la organización.

Estas aportaciones han sido el resultado de la implementación de estos sistemas en las organizaciones. Sin embargo, las diferencias en las organizaciones han planteado la necesidad de desarrollar diversas tecnologías que apoyen diferentes tipos de procesos. A continuación se presentan las diferentes opciones tecnológicas que han sido desarrolladas. Esta clasificación nos permite definir la clase de procesos que buscamos apoyar a través de la propuesta que se presenta en este trabajo. Con el fin de establecer el tipo de sistemas de workflow a los que nos enfocamos, en la siguiente sección se presentan las diferentes clasificaciones que han sido propuestas.

II.4.2 Diferentes tecnologías en WFMSs

Diferentes autores han tratado de establecer categorías para los sistemas de workflow. Sin embargo, dichas categorías aun no quedan bien definidas [Plesums, 2002]. Una taxonomía ampliamente aceptada se basa en el tipo de sistemas de workflow que son generados, de esta forma se distingue entre sistemas de workflow administrativos, ad-hoc, colaborativos y de producción [Alonso *et al.*, 1997]. Los parámetros de esta clasificación son: la similitud entre los procesos involucrados y el valor de los mismos para la organización. Las características de los tipos de workflow que forman la taxonomía mencionada se presentan a continuación.

a) Workflow Administrativo

En general, los sistemas de workflow administrativos se refieren a sistemas en los que los procesos, los pasos y secuencia se encuentran bien definidos y las reglas involucradas en la ejecución son conocidas por todos los agentes que participan. Ejemplos de este tipo de workflow son: el registro de cursos en una universidad, el registro de un vehículo y en general, la mayoría de los procesos en los cuales existe un conjunto de formas en papel que deben ser llenadas y llevadas a través de una serie de pasos.

b) Workflow Ad hoc

Los sistemas de workflow ad hoc son similares a los administrativos, excepto que son creados para tratar con excepciones o situaciones únicas. Igualmente, nos referimos a sistemas de workflow en los cuales los procesos no están completamente definidos y en los cuales existen los mecanismos para tratar con este tipo de procesos.

c) Workflow Colaborativo

Esta tercera clase de sistemas de workflow es principalmente caracterizada por el número de participantes involucrados y por las interacciones entre ellos. En contraste con otros tipos de sistemas de workflow basados en la premisa de un proceso continuo y hacia adelante, el workflow colaborativo puede involucrar iteraciones sobre el mismo paso hasta que algún acuerdo sea alcanzado, e inclusive, puede involucrar el retorno a fases previas. Un ejemplo de este tipo de workflow es el proceso de escritura de un artículo por varios autores. Este tipo de sistemas de workflow generalmente son dinámicos en el sentido de que son definidos conforme van progresando.

d) Workflow de Producción

Los sistemas de workflow de producción son caracterizados por la implementación de procesos directamente relacionados con el funcionamiento de una organización. Solicitudes de crédito y préstamos, y los reclamos de fianzas son ejemplos típicos de estos sistemas. Usualmente cuando se habla de sistemas de workflow de producción, los principales puntos a considerar son los ambientes en los que se utilizan, su tamaño,

complejidad, el número de personas involucradas y la naturaleza de las tareas. En particular, esta clase de sistemas son ejecutados en sistemas heterogéneos, frecuentemente aplicaciones legadas (heredadas).

Otra clasificación a la que generalmente se hace mención en la literatura es con respecto a la tecnología en la que está basado el WFMS [Alonso *et al.*, 1997]. De acuerdo con esta clasificación se tienen WFMSs basados en correo electrónico, basados en documentos y basados en procesos. La primer categoría de sistemas trabajan sobre una plataforma de correo electrónico y son por lo general utilizados en sistemas colaborativos y ad hoc. Por otra parte, los sistemas basados en documentos trabajan bajo la premisa del enrutamiento de documentos, sin embargo, la capacidad para interactuar con aplicaciones externas es limitada. Muchos sistemas de workflow administrativos, principalmente aquellos basados en formatos establecidos (en papel) pueden ser implementados utilizando esta tecnología.

Finalmente, los sistemas de workflow basados en procesos se relacionan directamente con el workflow de producción. Por lo general esta clase de sistemas implementan sus propios mecanismos de comunicación, son desarrollados sobre bases de datos y proveen de un amplio rango de interfaces que permiten la interacción con aplicaciones nuevas y legadas. Es en esta clasificación en la que se encuentran los sistemas de workflow que son abordados en esta tesis, por lo que en el resto de este trabajo, nos concentramos solo en WFMSs que generan workflows de producción o bien, basados en procesos.

Hasta aquí hemos establecido la definición y clasificación de los sistemas administradores de workflow. Hemos visto como se ha diversificado el desarrollo de esta tecnología atendiendo a procesos de diferente naturaleza, lo que ha implicado también, una diversidad en las formas como se definen estos sistemas. Para establecer una base común en la definición y desarrollo de este tipo de tecnología, se formó la Coalición para la Administración del Workflow (*Workflow Management Coalition - WfMC*), la cual trabaja

para definir bases comunes en el desarrollo de tecnología de workflow, aspecto cuya importancia reside en áreas como el apoyo a la interoperabilidad entre diferentes herramientas.

II.5 Coalición para la Administración del Workflow (WfMC)

La Coalición para la Administración del Workflow (Workflow Management Coalition – WfMC), es una organización no comercial formada a partir de la necesidad de establecer estándares para el desarrollo de tecnologías de workflow. Entre las actividades que la WfMC lleva a cabo están: el identificar áreas funcionales, desarrollo de especificaciones y de estándares [WfMC, 1999].

Una de las primeras aportaciones de la WfMC es el establecimiento de un meta-modelo para la definición de sistemas de workflow [WfMC, 1995]. Por medio de este modelo, se establecen los aspectos básicos que deben contemplarse en la definición de un sistema de workflow. El cumplir con este modelo, asegura que la definición del sistema de workflow pueda ser en un momento dado transferida a otra máquina de workflow que observe este modelo y que provea los mecanismos para su interpretación, apoyando de esta forma la interoperabilidad y reutilización de modelos de workflow.

En el contexto de esta tesis, es importante revisar este meta-modelo ya que estaremos trabajando con la definición de sistemas de workflow, de aquí que es necesario conocer y seguir un modelo establecido para su definición de forma que el modelo resultante, sea completo, considere los aspectos generalmente propuestos y permite la reusabilidad. La descripción de este meta-modelo se presenta a continuación.

II.5.1 Meta-modelo de Workflow y el modelado de sistemas de workflow

La base de cualquier workflow son los procesos dentro de una organización. El modelo de referencia del workflow define a un proceso organizacional como un

procedimiento en el cual, documentos, información o tareas son pasadas entre los participantes de acuerdo a un conjunto de reglas definidas [WfMC, 1995].

Con la finalidad de lograr una representación del modelo a un formato que pueda ser ejecutado por una computadora, se define a un modelo de workflow como un grafo acíclico dirigido, en los cuales los nodos representan pasos de ejecución y las aristas representan el flujo del control y de datos entre las diferentes actividades [WfMC, 1995]. A continuación se presentan los elementos del meta-modelo propuesto por la WfMC.

a) Procesos. la descripción de la secuencia de pasos que deben llevarse a cabo para lograr un objetivo. Por regla, debe tener un nombre, condiciones de inicio y fin así como datos para seguridad y control. Un proceso consiste de actividades y datos relevantes.

b) Actividades, o pasos dentro de un proceso, las cuales deben tener un nombre, pre y post condiciones así como condiciones de ejecución.

c) Flujo de control, especificado a través de *conectores de control* entre actividades. Representan el orden en el que las actividades son ejecutadas.

d) Contenedores de entrada, representados a través de variables y estructuras de datos que son utilizadas como la entrada de datos por los procesos.

e) Flujo de datos, representados a través de variables y estructuras de datos que son utilizadas para almacenar las salidas por los procesos.

f) Condiciones, las cuales especifican las circunstancias bajo las cuales ciertos eventos ocurren. Existen tres tipos básicos de condiciones. *Condiciones de transición*, las cuales están asociadas con *conectores de control*, y especifican si una condición se evalúa a falso o verdadero. Un conector de control que evalúa a falso no activará la ejecución de una actividad. Las *condiciones de inicio* especifican cuando una actividad debe comenzar: por

ejemplo, ya sea cuando todos los conectores de control evalúan a verdadero. Las *condiciones de salida* especifican cuando una actividad debe ser terminada.

g) Datos relevantes, representan las entidades de información que son utilizadas durante la ejecución del workflow. Básicamente se refiere a los datos que son creados y utilizados en cada una de las actividades del proceso en ejecución. Estos datos se hacen disponibles a las actividades y otras aplicaciones, y son utilizados para pasar información persistente o resultados intermedios entre las actividades ya sea para la evaluación de condiciones o como resultados finales de algún proceso.

Este modelo representa solo una abstracción de lo que debe ser considerado para realizar el modelado, pero en ningún momento establece detalles de implementación por lo que este aspecto queda a libertad del modelador. En este sentido, se han propuesto varias formas para realizar el modelado de un sistema de workflow. Entre estas técnicas se encuentran las *redes de petri*, las *redes de petri con color* [van der Aalst, 1998] y los *diagramas Rol-Actividad (RAD)* [Flores Rios, 2002; García *et al.*, 2002].

Este meta-modelo ha resuelto en una parte el problema de la interoperabilidad, estableciendo un modelo común para la definición de sistemas de workflow. Existen por otro lado diferentes limitaciones en sistemas administradores de workflow. Estas limitaciones se exponen a continuación.

II.6 Limitaciones en los Sistemas Administradores de Workflow (WFMS)

A pesar del desarrollo que se ha llevado a cabo en cuanto a tecnología de workflow, existen aun limitaciones en esta clase de sistemas. Estas limitaciones se originan principalmente por los antecedentes de estos sistemas. Los orígenes de los sistemas de workflow se encuentran en áreas como la *automatización de la oficina*, el procesamiento de imágenes y documentos y el trabajo cooperativo asistido por computadora (*CSCW*). En

estos ambientes, los principales problemas por resolver eran el compartir recursos y el apoyar al trabajo cooperativo, por lo que aspectos como la interoperabilidad, la escalabilidad y la flexibilidad no eran considerados [Alonso *et al.*, 1997].

En general, la literatura establece como retos y deficiencias que deben ser superadas en las máquinas de workflow, aspectos como la interoperabilidad, el apoyo a la movilidad y a la operación desconectada [Alonso *et al.*, 1997], la integración de agentes autónomos de software, la flexibilidad y adaptabilidad de workflow, entre otros. En este sentido, la motivación y aportación de esta tesis se enfoca en el apoyo a la movilidad y la operación desconectada en sistemas de workflow. Así mismo, el paradigma que utilizamos en la implementación del WFMS y en la definición de los sistemas generados por éste, aporta también un enfoque de solución para el problema de la interoperabilidad. Para clarificar ambos retos, a continuación se presenta la problemática en ambas áreas y la forma en que han sido apoyadas.

II.6.1 Interoperabilidad

Los WFMSs existentes en la actualidad son casi en su totalidad incompatibles entre sí [WfMC, 1995]. Recientemente, se ha estado llevando a cabo trabajo para apoyar este aspecto en las máquinas de workflow. En este sentido la WfMC, ha realizado varios esfuerzos, el primero fue el meta-modelo descrito anteriormente, después un modelo de referencia del workflow en el que se establecen los componentes estándar que debe considerar una máquina de workflow [WfMC, 1995]. Por otra parte, la misma WfMC ha empezado a establecer formatos de intercambio mediante el uso del Lenguaje de Mercado eXtensible (XML) [WfMC, 2002] con la finalidad de establecer una serie de reglas sintácticas y léxicas para la definición de procesos de forma que se puedan transportar entre productos o máquinas diferentes que observen este formato. De esta forma el uso de XML ofrece alternativas viables para el logro de la interoperabilidad entre diferentes máquinas [WfMC, 2002; Yung, 2002]. En este mismo sentido, y con el fin de apoyar este aspecto, la máquina de workflow (WFMS) propuesta en este trabajo tiene la capacidad de definir

sistemas de workflow utilizando el lenguaje XML y de interpretar esta definición para convertirla en un sistema de workflow funcional.

II.6.2 Apoyo a la movilidad y a la operación desconectada

El siguiente reto abordado en esta tesis es el del apoyo al cómputo móvil y desconectado. Este tipo de tecnología ha estado tomando un auge bastante importante. Nixon apuntó que el número de usuarios de PDAs para el año 2004 será de entre 30 y 40 millones [Nixon, 1998]. Sin embargo, de acuerdo a varios autores [Bolcer, 2000; Seungil *et al.*, 2001; Yung, 2002] la investigación para el desarrollo de sistemas de workflow en esta área no ha sido explorada del todo.

Es importante distinguir entre dos tipos de desconexiones en sistemas de workflow, las voluntarias y las involuntarias. Las desconexiones involuntarias ocurren cuando una falla inesperada en el sistema de comunicación impide al usuario continuar con su trabajo, mientras que las desconexiones voluntarias son aquellas en las que el usuario toma explícitamente sus tareas y las desconecta de un sistema de comunicación. En esta tesis limitamos la operación de sistemas de workflow a ambientes de desconexiones voluntarias.

Tradicionalmente los sistemas de workflow han sido diseñados bajo consideraciones en las que se presupone la existencia constante de una red de comunicación en la que reside la definición del sistema, los datos y los mecanismos para mantener la coordinación existente en estos ambientes, en contraste con el uso cada vez mayor dentro de las organizaciones de dispositivos móviles; estos dispositivos presentan características especiales como conexiones temporales a redes de comunicación, razón por la cual, en esta tesis planteamos la *operación desconectada* como una opción que permite a los agentes que participan en un sistema de workflow continuar con su trabajo y mantener interacciones coordinadas en casos donde la conexión a una red de datos no está disponible.

II.7 Operación desconectada en sistemas de workflow

El término *operación desconectada* se refiere a la habilidad de un cliente en un sistema de cómputo para continuar con sus tareas aún cuando no se cuenta con una red en donde éste generalmente ejecuta su trabajo y por lo tanto, el acceso a un contenedor central de datos no es posible [Seungil *et al.*, 2001]. Con mecanismos que apoyen la operación desconectada, los usuarios pueden trabajar independientemente de una computadora servidor [Alonso *et al.*, 1997]. Podemos observar como la combinación de los conceptos de workflow y operación desconectada ofrecen un amplio rango de aplicaciones en los escenarios actuales de workflow. Mediante la combinación de ambos conceptos, encontramos el concepto de workflow desconectado como la *disponibilidad de servicios de un WFMS en estado desconectado*. El concepto de workflow desconectado es la base de la propuesta tecnológica que se presenta en esta tesis. Por este motivo, en la siguiente sección se hace una revisión de los trabajos que han considerado el problema de la operación desconectada. La finalidad de esta revisión es poner en contexto el trabajo presentado en esta tesis con respecto a esfuerzos anteriores y mostrar la contribución que buscamos con nuestro trabajo.

II.7.1 Trabajo previo para apoyar la operación desconectada en sistemas de workflow

Se han realizado diferentes esfuerzos para apoyar un desempeño confiable en ambientes desconectados, la mayoría de estos trabajos se han realizado en ambientes de sistemas de archivos y han demostrado que el apoyo a la operación desconectada es efectivo al nivel de operaciones de archivo [Kistler, 1993]. Sin embargo, a un nivel de aplicación, como en sistemas de workflow, no se ha realizado mucho trabajo con profundidad [Seungil *et al.*, 2001].

A continuación se presentan cinco de los trabajos más importante que se han realizado para apoyar el trabajo desconectado en sistemas de workflow.

- 1) *FlowMark*.- Desarrollado en IBM, el proyecto *FlowMark* [Alonso *et al.*, 1995] representa el primer trabajo de investigación en el área de workflow desconectado. En este trabajo, los autores proponen una serie de mecanismos que permiten a la arquitectura apoyar a clientes desconectados. Una de las principales contribuciones de este trabajo, es el señalar el impacto de clientes desconectados en sistemas WFMS. La propuesta de *FlowMark*, es el implementar mecanismos relacionados con el manejo de datos, la notación de actividades bloqueadas, y la introducción de mecanismos para establecer compromisos de los clientes para efectuar las actividades que se desconectan.
- 2) *Domino Everyplace Enterprise*.- La arquitectura del *Domino Everyplace Enterprise* [Pepin *et al.*, 2002], provee de una plataforma para el desarrollo de aplicaciones que se pueden utilizar en dispositivos móviles, las cuales se integran a la plataforma de *Lotus Domino*. El aspecto interesante de esta propuesta es el uso del lenguaje XML para la definición de las aplicaciones, aspecto que apoya la compatibilidad con otras máquinas de workflow. Sin embargo, esta propuesta no permite evaluar factibilidad de la desconexión de las aplicaciones.
- 3) *MAGI*.- *Micro Apache Generic Interface*.- *Magi* [Bolcer, 2000] implementa una arquitectura en términos de un servidor HTTP (micro-Apache). Esta implementación hace uso de primitivas de servidor para la administración, sincronización y almacenamiento de información en línea. Sin embargo, *Magi* fue diseñada específicamente para el manejo de workflow basado en el paso de documentos y debido a su estructura representa una implementación robusta.
- 4) Propuesta de Yun Yang.- Por otro lado el Dr. Yun Yang en la Universidad de Tecnología de Swinburne en su trabajo *Disconnected Workflow for Web-based teamwork support* [Yung, 2002] define algunos requerimientos y estrategias de diseño para el workflow pero sólo hace mención de laptops como dispositivos móviles. La premisa principal de su aportación es el proveer al usuario de

interfaces gráficas de usuario (GUI) idénticas aún cuando este se encuentra en desconexión. Los datos de los estados de los procesos son mantenidos en una base de datos, en la cual también se almacenan los cambios realizados a las entidades de información modificadas durante el período de desconexión.

- 5) ICU/COWS.- La arquitectura ICU/COWS [Seungil *et al.*, 2001] representa un proyecto de investigación en el cual los autores analizan el modelo de una tarea, identificando el tipo y el ámbito de la misma, con la finalidad de evaluar una posible desconexión. Por el tipo de la tarea, los autores se refieren al conjunto de actividades involucradas en el proceso, y mediante el término ámbito, se refieren a los servicios sobre los cuales se ejecutan las tareas dentro del sistema. Basados en ese análisis, los autores identifican tres consideraciones importantes para el apoyo a la operación desconectada en sistemas de workflow. Estas consideraciones son: la clasificación de la tarea, el manejo de las aplicaciones y los datos relacionadas con el workflow y la emulación de la secuencia y transiciones entre tareas cuando estas se encuentran en estado desconectado.

En la tabla 1, se presenta el resumen de las características que se consideran más importantes para el apoyo del trabajo desconectado en sistemas de workflow como se acaban presentar.

Tabla I. Cuadro comparativo de las diferentes propuestas para el apoyo al trabajo desconectado en sistemas de workflow

Sistema	FlowMark	Domino Everyplace Enterprise	Magi	Yun Yang	Seungil et al (ICU/COWS)
Descripción	Producto comercial de IBM	Producto comercial de Lotus	Originalmente un proyecto "open source". Ahora un producto comercial	Desarrollo académico	Desarrollo académico
Arquitectura	Distribuida	Distribuida	Distribuida	Distribuida	Distribuida
Tipo de Workflow	basado en procesos	basado en procesos	basado en el paso de documentos	basado en procesos	basado en procesos
Lenguaje para la definición del proceso	API propietaria	API propietaria	No se contempla	API definida	WPDL (propuesto por la WfMC)
Soporte a XML para apoyar la interoperabilidad	No	Sí	Sí	No	No
Apoyo a la movilidad y la operación desconectada	Sí	Sí	Sí	Sí	Sí
Mecanismo para apoyar la desconexión	Proveer al cliente con la información necesaria	Proveer al cliente con la información necesaria	Proveer al cliente con la información necesaria	Proveer al cliente con la información necesaria	Proveer al cliente con la información necesaria
Dispositivos que Apoya	Notebooks	PDA	PDA	Notebooks	PDA
Evaluación de la desconexión	No	No	No se contempla	No	Sí. Análisis del tipo de tarea y de la información relacionada
Uso de Agentes	No	Sí	Sí	No	No
Interacción a través del Web	No	Sí	Sí	No	No
Interacción con otras aplicaciones	No se contempla	No se contempla	Sí	No	No

Las categorías que se presentan en la tabla 1 representan las características más importantes que se buscan para el apoyo a la operación desconectada en sistemas de workflow. La explicación de cada una de estas categorías se presenta en la tabla 2.

Tabla II. Categorías y su descripción

Categoría	Descripción
Descripción	Alguna característica importante del sistema.
Arquitectura	Tipo de arquitectura: centralizada, distribuida, etc.
Tipo de Workflow	Clasificación del workflow de acuerdo a la taxonomía presentada en el capítulo.
Lenguaje para la definición del proceso	Lenguaje por medio del cual los sistemas de workflow son generados.
Soporte a XML	Indica si el WFMS soporta el uso de XML para la definición de sistemas.
Apoyo a la desconexión	Indica si el WFMS puede manejar la operación desconectada de tareas.
Mecanismos para apoyar la desconexión	Mecanismos que se implementan para manejar la operación desconectada.
Dispositivos que apoya	Clase de dispositivos móviles en los que los sistemas pueden ejecutarse.
Evaluación de la desconexión	Indica si el WFMS puede o no evaluar si es factible desconectar un rol tomando en cuenta que éste se relaciona con otros y puede depender de una entrada previa.
Interacción a través del Web	Indica si los sistemas de workflow pueden ejecutarse en Internet a través de navegadores Web.
Interacción con otras aplicaciones	Indica si los sistemas de workflow generados pueden acceder a otras aplicaciones como procesadores de palabras, hojas de cálculo, sistemas de información, etc.

Dos áreas son las que más interesan en el contexto de esta tesis. Estas categorías son el apoyo al lenguaje XML como medio para proveer interoperabilidad con otras herramientas, y la forma en que se apoya la desconexión, lo que incluye los mecanismos y la evaluación de la factibilidad de la desconexión de tareas. Por medio de la propuesta tecnológica que presentamos en los siguientes capítulos, estos aspectos son implementados en la herramienta *SysCoor* lo que coloca a este WFMS como una herramienta completa entre las propuestas analizadas.

II.8 SysCoor

SysCoor es un WFMS con una arquitectura centralizada cuyo propósito es facilitar la generación de sistemas de workflow a través del uso de modelos de procesos capturados por medio de técnicas diagramáticas.

SysCoor implementa los mecanismos necesarios para que los sistemas de workflow generados trabajen en Internet y sean accedidos por medio de un navegador Web. Esta herramienta ha sido desarrollada por el grupo de Ingeniería de Procesos del CICESE.

En *SysCoor*, la definición de un sistema de workflow se inicia con la definición del proceso que se quiere automatizar. Para realizar esto, se captura el proceso utilizando los diagramas Rol-Actividad (RAD)¹, diagramas de transición de estado (DTE) con los cuales se obtiene el comportamiento del proceso a través del tiempo [Greenwood *et al.*, 1998] y por medio de la definición de las entidades de información (datos y documentos) utilizadas en cada tarea que compone el proceso.

Estos tres modelos son representados formalmente por medio del lenguaje XML y validados usando una Definición de Tipo de Documento (DTD) para cada modelo, el cual define las reglas sintácticas y léxicas del documento XML. Estos documentos son conocidos como los *modelos base RAD, DTE y EI*. Estos tres modelos son analizados de forma sintáctica y léxica y transformados en un solo modelo conocido como el *modelo base del sistema de workflow (modelo base WF)* el cual define un sistema de workflow a través del lenguaje XML. La figura 2 describe el flujo en el cual un sistema de workflow es generado en *SysCoor*.

¹ Los diagramas Rol-Actividad (RAD) son una de las técnicas de modelado de procesos más utilizadas [Miers,1996]. Estos diagramas capturan las tareas, los objetivos y las personas (agentes) involucradas en el proceso en roles, actividades, decisiones e interacciones[Rojas y Martínez, 1998].

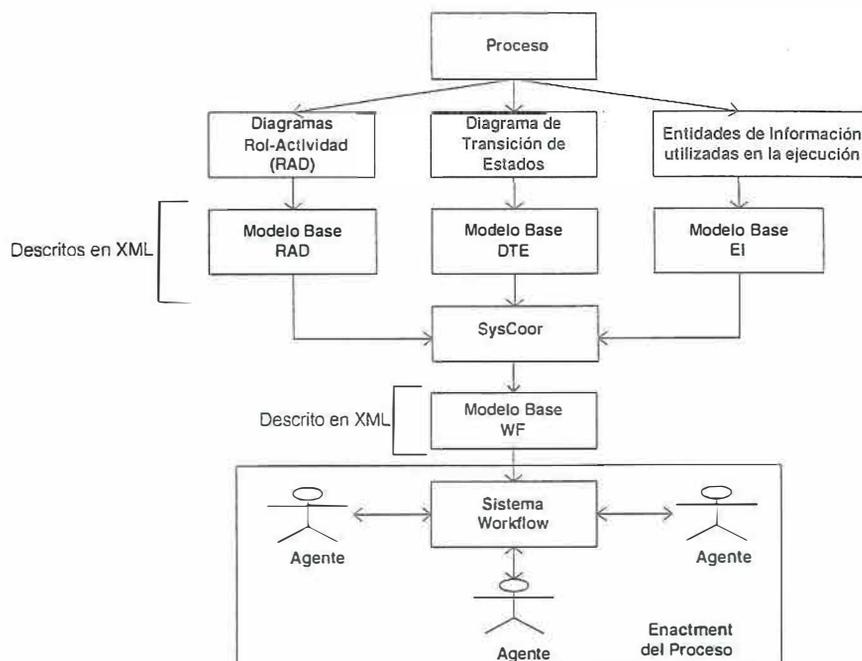


Figura 2. Elementos necesarios para generar un WFMS

II.8.1 Arquitectura básica de *SysCoor*

La arquitectura básica de *SysCoor* se presenta en la figura 3. La arquitectura original consiste de cuatro componentes principales: un *generador de sistemas de workflow*, un *contenedor de datos y modelos*, una *máquina de coordinación* y el componente conocido como *rol base*.

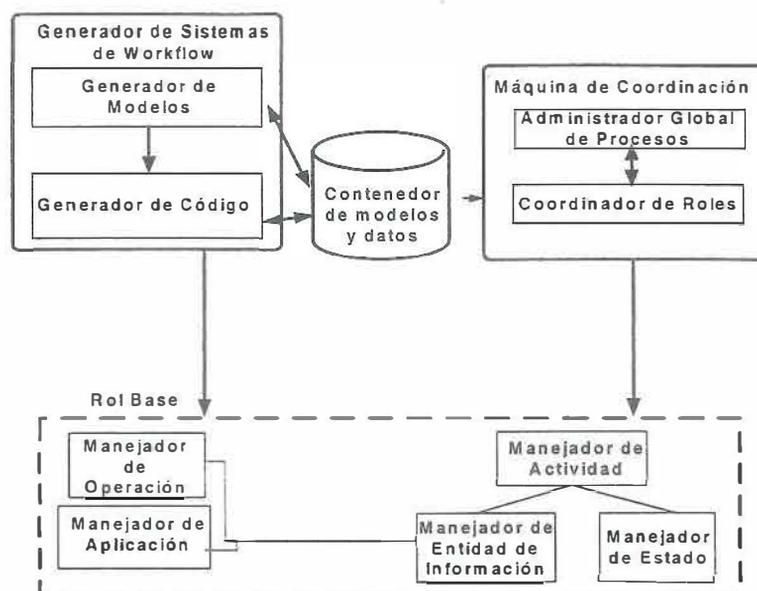


Figura 3. Arquitectura básica de SysCoor

El *generador de sistemas de workflow* se compone de dos submódulos: el *generador de modelos* y el *generador de código*. El *generador de modelos* es responsable de leer y procesar los *modelos base RAD, STD y EI*, y crear el *modelo base WF*. Por su parte, el *generador de código* crea la estructura del *rol base* en el que se definen las características de un rol dentro del proceso, como las actividades internas, los estados por los cuales atraviesa, las entidades de información que utiliza, las operaciones y las aplicaciones externas que son invocadas durante la ejecución de éste [Arroyo *et al.*, 2003]. En el *contenedor de modelos y datos* se encuentra la información de los sistemas de workflow que han sido generados, los *modelos base* y las entidades de información (datos relevantes). Finalmente, la *máquina de coordinación*, la cual es responsable de mantener la colaboración, la comunicación, la coordinación y el intercambio de información entre los procesos.

La arquitectura básica de SysCoor fue desarrollada utilizando la tecnología de *Servlets* y *Páginas de Servidor de Java* (JSP, *Java Server Pages*). Los *Servlets* son programas escritos en el lenguaje de programación Java que se ejecutan en un servidor

[Hall, 2000] , mientras que los JSP son una combinación de código HTML y código Java que en conjunto funcionan como un *Servlet*. Estos mecanismo se comunican entre sí para recuperar y desplegar la información almacenada en *objetos serializados* los cuales representan los roles base. Esta implementación permite al usuario de un sistema de workflow, ejecutar sus tareas a través de un navegador Web.

II.9 Resumen

En el presente capítulo se introdujo al concepto de workflow y las tecnologías de WFMS estableciendo de esta forma un entendimiento básico del área en el que se desarrolla esta tesis.

Se presentaron varias de las limitaciones señaladas por la literatura en el área de workflow, siendo éstas, el apoyo a la interoperabilidad entre herramientas y el apoyo a la operación desconectada, útil en ambientes donde la movilidad es requerida. Siendo este último aspecto el tema principal de esta tesis. Por otro lado, se describió brevemente la herramienta *SysCoor* y su arquitectura. Esta herramienta, es la base de la propuesta tecnológica que se expone en esta tesis para abordar el problema de la operación desconectada en sistemas de workflow.

En los siguientes capítulos se presenta cómo la arquitectura de *SysCoor* ha sido extendida de forma que apoya la operación desconectada en sistemas de workflow. A través de la implementación de una serie de mecanismos y de una arquitectura cliente, la funcionalidad de *SysCoor* ha sido extendida para no sólo generar sistemas de workflow que trabajan en el Web, si no que también pueden ser utilizados en dispositivos PDA.

El desarrollo llevado a cabo para apoyar la operación desconectada en sistemas de workflow, consistió en implementar extensiones a la arquitectura existente para el servidor, de forma que el apoyo a este tipo de operación fuese posible en una arquitectura cuyo escenario principal eran los sistemas de workflow basados en Web. Además se desarrolló

una arquitectura ligera para un cliente de forma que pueda ser implementada en dispositivos PDA.

En el siguiente capítulo se describen dos casos de estudio que permitieron observar escenarios en los que los agentes de los procesos cambian de un ambiente en el que cuentan con toda la información para llevar a cabo su trabajo, a uno en el que los recursos son limitados. Estos escenarios no permitieron establecer requerimientos que complementan los obtenidos de la literatura, para el desarrollo de las extensiones a la arquitectura de *SysCoor* de forma que se provea soporte a la operación desconectada en sistemas de workflow. Así mismo, se retoman los trabajos que se han realizado en el área para obtener requerimientos adicionales.

Capítulo III. Requerimientos para el Apoyo a la Operación Desconectada en Sistemas de Workflow

III.1 Introducción

En el presente capítulo se describe el análisis de los requerimientos para apoyar el trabajo móvil y desconectado en WFMSs. Estos requerimientos han sido obtenidos de tres fuentes: la realización de dos casos de estudio, el análisis de los requerimientos establecidos por trabajos anteriores con el fin de apoyar la operación desconectada en aplicaciones de workflow, y por último, los requerimientos que establece la arquitectura de *SysCoor*, base tecnológica de nuestra propuesta.

El desarrollo de los casos de estudio nos ha permitido observar escenarios reales de trabajo, donde las personas se mueven de un ambiente en el que cuentan con todos los elementos para llevar a cabo su trabajo, a uno en donde los recursos pueden no estar a la mano. A continuación se presentan algunos de los escenarios analizados en los casos de estudio y los requerimientos que fueron obtenidos a partir de éstos.

III.2 Casos de Estudio

Con la finalidad de obtener requerimientos a partir de escenarios reales de aplicación, se realizaron dos casos de estudio, los cuales se presentan a detalle en el apéndice A de este documento. Estos casos de estudio se llevaron a cabo en la compañía vitivinícola “Chateau Camou” en el área de ventas y en la Clínica Hospital General de Zona IV No. 8 del Instituto Mexicano del Seguro Social (IMSS) en las áreas de medicina interna y urgencias. La metodología utilizada para realizar estos casos de estudio se basó en la aplicación de entrevistas semi-estructuradas con duración de entre 30 a 50 minutos

con miembros del personal de ambas organizaciones, observación de actividades y análisis de documentos.

En ambos casos de estudio, se buscaron escenarios que presentaran características naturales de los sistemas de workflow como la necesidad de coordinación entre las actividades y los participantes, y además, que involucraran actividades cuya realización requiriese de movilidad física por parte de los agentes participantes. Más detalladamente las características buscadas fueron:

- a) Que el agente o agentes del proceso presentaran movilidad física.
- b) Que existiera la necesidad de coordinación con otros agentes.
- c) Que en las actividades del agente existiera intercambio de documentos y de información.
- d) Que durante el tiempo en el que el agente se encuentra separado de su área física de trabajo, éste modificara información.
- e) Que el regreso al lugar de trabajo requiriera reincorporar la información modificada durante la ausencia del área de trabajo.

En ambos casos de estudio se encontraron escenarios que muestran estas características y a partir de ellos, se obtuvieron un primer conjunto de requerimientos, los cuales ayudan en la definición de mecanismos que pueden apoyar a los agentes a llevar a cabo sus actividades. Estos escenarios se presentan a continuación, iniciando con aquellos analizados en el caso de estudio realizado en el IMSS y continuando con el escenario encontrado en la compañía Chateau Camou.

III.2.1 Escenarios en el Caso de Estudio en el IMSS

Como resultado del caso de estudio realizado en la clínica 8 del IMSS se definieron tres escenarios los cuales se presentan a continuación.

Escenario 1. Programa de Atención Domiciliaria a Enfermos Crónicos.

El programa de Atención Domiciliaria a Enfermos Crónicos (ADEC), provee servicio médico a domicilio a enfermos considerados como crónicos. Un médico, una enfermera y una trabajadora social visitan en su domicilio al paciente inscrito dentro del programa ADEC (agentes del proceso). El médico responsable prepara y actualiza el calendario de las visitas y los datos de localización del paciente previo a la realización de la visita. Da a conocer este calendario a los miembros del grupo para que preparen la información necesaria para apoyar las tareas correspondientes que cada uno lleva a cabo durante las visitas. Por ejemplo, la enfermera de ADEC revisa el calendario y obtiene las hojas médicas (expediente médico) de los pacientes con la finalidad de conocer sus padecimientos y determinar el material necesario para la visita. La trabajadora social prepara los formatos necesarios, como lo son el formato para realizar el diagnóstico social y las evaluaciones a los familiares.

Durante la visita cada uno de los agentes genera varios documentos:

- a) *Médico*: Genera solicitudes de laboratorio, estudios de gabinete (rayos X, etc), estudios subrogados (aquellos por realizarse en laboratorios fuera del IMSS), traslados, etc. También realiza las anotaciones médicas que sirven a la enfermera para planear el material de la próxima visita, generar recetas, etc. Otra de sus actividades es decidir la alta del paciente si así se considera necesario.
- b) *Enfermera*: Registra los signos vitales y lleva el inventario del material utilizado con cada paciente en cada visita.
- c) *Trabajadora Social*: Realiza el proceso conocido como *diagnóstico social*, el cual involucra diversas tareas como la evaluación del medio físico en el que se encuentra el paciente.

Una vez que el grupo ADEC regresa al hospital, cada uno de los miembros procede a integrar la información generada durante la visita:

- a) *Médico*: Entregar a la jefa de piso las solicitudes de laboratorio, estudios de gabinete, estudios subrogados, traslados, etc., también solicitar el medicamento necesario para la próxima visita al paciente. Otro de las actividades es dar trámite a las altas de aquellos pacientes que así lo requieran.
- b) *Enfermera*: Integrar el registro de los signos vitales al expediente del paciente, realizar las solicitudes de material al almacén basándose en el inventario y necesidades del material para las visitas del próximo día.
- c) *Trabajadora Social*: Llenar el *reporte de trabajo social*.

Este escenario cuenta con las siguientes características:

1. Coordinación entre médico, enfermera y trabajo social previo a la visita.
2. Alejamiento por un período prolongado de tiempo del área de trabajo.
3. Trabajo por hacer cuando se regresa al área de trabajo a partir de la información obtenida durante el alejamiento.
4. Necesidad de comunicar el estado en el que se encuentra el trabajo cuando éste se está llevando a cabo.
5. Múltiples roles ejecutados por el mismo agente durante un mismo lapso de tiempo, por ejemplo, el médico que lleva a cabo funciones como valoración del paciente y al mismo tiempo genera solicitudes de diversos tipos.
6. Anotaciones breves durante el período de movilidad.

Escenario 2. Tareas del asistente médico.

La asistente médico del hospital se encarga de recorrer todos los servicios del hospital, cama por cama, para actualizar el listado de los pacientes, saber qué pacientes egresaron, quiénes ingresaron y quiénes están en el estado conocido como *pre-alta*. Una vez que termina este recorrido, elabora la lista de los pacientes hospitalizados que incluye el nombre del paciente, número de afiliación y servicio en el que se encuentra. El asistente genera copias de esta lista y las distribuye a las áreas de admisión hospitalaria, urgencias,

vigilancia y en cada uno de los servicios del hospital de forma que haya conocimiento de los pacientes que se encuentran en cada una de las áreas del hospital.

Durante este recorrido, el asistente médico tiene que revisar todos los expedientes con la finalidad de verificar si el paciente es un derechohabiente vigente por medio de la forma 4-30-51, la cual es el registro de ingreso del paciente a cualquier servicio. El proceso para verificar la vigencia de un derechohabiente es el siguiente: si la forma 4-30-51 no se encuentra sellada y firmada, el asistente médico toma la forma y la lleva al archivo, ahí revisa la información del paciente, si se comprueba la vigencia, la forma regresa al expediente; si no es posible confirmar, se localiza a algún familiar para que presente documentos que comprueben la vigencia del paciente; si no se comprueba la vigencia del derechohabiente, es necesario avisar a trabajo social para que en coordinación con la asistente médico se elabora una carta de obligación de pago por el servicio que el paciente está recibiendo y en caso contrario, se determina el envío del paciente a un hospital general.

Las características encontradas en este escenario son:

1. Coordinación entre el asistente médico y trabajo social.
2. Información previa necesaria para la separación del área de trabajo, en este caso el listado de los pacientes.
3. Movilidad del asistente médico a través de todos los servicios del hospital.
4. Alejamiento por un período de tiempo corto del área de trabajo.
5. Trabajo por hacer cuando se regresa al área de trabajo a partir de la información obtenida durante la movilidad.
6. Anotaciones breves durante el período de movilidad.

Escenario 3. Asistente médico en apoyo a otras actividades.

Durante el recorrido que la asistente médico realiza por los diferentes servicios del hospital, surgen algunas actividades en las que la asistente debe apoyar a otros agentes, estas actividades son:

- a) El médico le comunica la alta de un paciente. En algunas ocasiones, esta alta requiere del trámite de una cita con algún especialista, posterior a la salida del paciente del hospital. Esta cita es tramitada por la asistente médico. La alta de un paciente requiere también que el asistente prepare otros documentos para el trámite.
- b) El médico directamente, o a través de una jefa de piso, solicita a la asistente médico el traslado en ambulancia de un paciente, para lo que la asistente médico elabora la papeleta de ambulancia y la entrega al servicio de transporte para que este servicio recoja al paciente.

En ambos casos (*a* y *b*) el asistente médico recoge los datos de estas solicitudes durante su recorrido por el hospital para después llegar a su escritorio y darles trámite.

Las características encontradas en este caso son:

1. Coordinación entre el asistente médico, los médicos, jefas de piso y personal de transporte.
2. Movilidad del asistente médico en los diferentes servicios del hospital.
3. Alejamiento por un período de tiempo corto del área de trabajo.
4. Anotaciones breves durante el periodo de movilidad.
5. Trabajo por hacer cuando se regresa al área de trabajo a partir de la información obtenida durante la movilidad.
6. Múltiples roles ejecutados por un mismo agente durante un mismo lapso de tiempo.

Los escenarios encontrados en el caso de estudio realizado en la clínica 8 del IMSS nos han permitido observar varias características del trabajo cuando este requiere de la movilidad de los agentes. Para lograr una caracterización completa de este forma de desempeñar actividades, se necesitaría un número considerable de casos de estudio en diferentes ambientes que pudieran ayudar a refinar, contrastar y establecer un marco general de características del trabajo en ambientes móviles. Ante las limitaciones de tiempo

en el que se desarrolló esta tesis, y con la finalidad de validar y buscar nuevas características, se llevo a cabo solo un segundo caso de estudio más, esta vez en la compañía Chateau Camou. Al igual que en el caso de estudio anterior, la metodología consistió en la aplicación de entrevistas semi-estructuradas con duración de entre 30 a 50 minutos, observación de actividades y análisis de documentos.

III.2.2 Escenario del Caso de Estudio en Chateau Camou

El caso de estudio realizado en Chateau Camou, se centró en el análisis del proceso de ventas, el cual es el que mayor movilidad presenta. A continuación se describe el escenario a través del cual se desarrolla el proceso. El agente principal en este caso es el vendedor quien se relaciona con dos agentes más: el cliente y el asistente administrativo.

Escenario. Vendedor realizando venta.

En la mayoría de las ocasiones, los vendedores necesitan visitar a los clientes de la compañía en el lugar donde éstos se localizan, en algunas ocasiones fuera de la ciudad. Estas visitas requieren que el vendedor se ausente del lugar de trabajo por períodos que van desde un día hasta varios días. Durante estas visitas, un vendedor necesita recolectar información respecto al inventario que el cliente tiene del producto, llenar pedidos, entregar facturas, entre otras actividades. Para tomar un pedido, el vendedor necesita tener información detallada acerca de las compras previas que el cliente ha realizado y su estado del crédito, lo anterior para tomar una decisión respecto a la venta de tal forma que sea benéfica para ambas partes.

Un tipo de visita que es muy importante para el vendedor son las llamadas visitas de promoción a clientes potenciales de la compañía, las cuales requieren que el vendedor realice una presentación de los viñedos y el área de producción para que el cliente tenga una idea completa de la compañía. Para este tipo de visitas el vendedor prepara una serie de material gráfico para hacer una presentación más completa.

Normalmente el vendedor prepara su calendario de actividades por las mañanas, lo que incluye calendarizar visitas y preparar todo el material que requiere para éstas. Si durante la visita a un cliente surge la necesidad de aclarar detalles sobre el crédito del cliente o bien sobre el inventario existente en el almacén de Chateau Camou, el vendedor necesita llamar a la oficina y solicitar la ayuda del asistente administrativo, quien consulta el sistema de información para responder estas preguntas. De esta forma, el vendedor siempre depende de la disponibilidad del asistente para tomar la llamada.

Al final del día, el vendedor puede o no regresar a la oficina, pero ya cuenta con la información que se ha recolectado durante las visitas. Esta información es necesaria para calendarizar las visitas del siguiente día, actualizar la información de las compras y crédito del cliente e iniciar el proceso de entrega de producto. Toda la información recolectada por el vendedor se mantiene en papel, de forma que cuando el vendedor regresa a la oficina debe capturar los pedidos recolectados en el sistema de información de donde se originan las órdenes para entregar el producto.

Como resultado del caso de estudio realizado, se validaron y fortalecieron las características obtenidas del caso de estudio realizado en la clínica 8 del IMSS. Las características encontradas en el escenario de Chateau Camou se presentan a continuación:

1. Movilidad del vendedor a diferentes localidades en la ciudad y fuera de ésta.
2. Alejamiento por períodos de tiempo variable, los cuales pueden ir desde un día hasta varios días.
3. Anotaciones breves durante el período de movilidad.
4. Coordinación entre el vendedor y el asistente administrativo.
5. Trabajo por hacer cuando se regresa al área de trabajo a partir de la información obtenida durante la movilidad.
6. Necesidad de comunicar el estado en el que se encuentra el trabajo cuando éste se está llevando a cabo.

7. Múltiples roles ejecutados por el mismo agente durante un mismo lapso de tiempo; el vendedor que puede estar tomando pedidos, realizando visitas promocionales, y visitas de cobranza durante el mismo período de desconexión.
8. Información previa necesaria para la separación del área de trabajo, en este caso, información de compras previas del cliente y el estado de su crédito.

III.3 Requerimientos obtenidos de los casos de estudio

A partir de las características encontradas en los escenarios de nuestros casos de estudio, podemos definir algunos requerimientos iniciales para el desarrollo de una herramienta que provea soporte al trabajo, cuando éste requiere movilidad y desconexión. Como se menciona en el primer capítulo, nuestro propósito es apoyar el trabajo móvil en desconexión a través del uso de asistentes digitales personales (PDAs).

Algunos de los requerimientos los obtuvimos a través de agrupar las características encontradas en los casos de estudio. Estos requerimientos son:

- a) La coordinación entre los miembros del proceso se debe mantener a pesar de la desconexión.
- b) Un agente en movilidad debe poder ejecutar de 1 hasta n roles durante una desconexión.
- c) La ejecución de un rol requiere de una entrada de información que deberá estar disponible al momento de la desconexión.
- d) Debe existir el manejo de la transición entre el trabajo desconectado y la reconexión a un área de trabajo de forma que la información sea integrada al flujo de trabajo.

Como se mencionó, estos requerimientos nos permiten pensar en las funciones que deben cumplir los mecanismos que se propongan para apoyar este tipo de operación en WFMSs. Sin embargo, debemos considerar también los requerimientos que han sido establecidos por trabajos anteriores ya que nos permiten enriquecer los obtenidos de los

casos de estudio. Así también, a partir del hecho de que desarrollamos nuestras extensiones sobre el WFMS *SysCoor*, debemos considerar los requerimientos establecidos por este último. En las siguientes secciones se presentan estos dos grupos de requerimientos.

III.4 Requerimientos documentados en la literatura

Como parte de la revisión de la literatura existente en el área del workflow desconectado, la cual se presentó en el capítulo anterior, hemos identificado los requerimientos que han sido establecidos e implementados en los trabajos mencionados en ese capítulo. Los requerimientos más importantes que han sido señalados son:

- a) Para permitir que los clientes trabajen de forma desconectada, éstos deben tener su propio medio de almacenamiento y acceso a la información necesaria para llevar a cabo sus tareas sin consultar una estructura centralizada [Alonso, 1995].
- b) Los clientes deben ser autónomos, es decir, un servidor central debe asegurar que no existan conflictos entre diferentes clientes que se encuentran en desconexión relacionados a una misma tarea [Alonso, 1995].
- c) Cuando se provee información a un cliente para ser utilizada durante la desconexión, no se debe utilizar información que no esté actualizada [Yung, 2002].
- d) Cuando un cliente se reconecta a un WFMS, después de un período de desconexión, debe existir el manejo de la información de salida del cliente, de forma que se asegure la consistencia de los datos [Yung, 2002].
- e) Las tareas deben clasificarse para decidir si una tarea puede ser ejecutada en modo desconectado. Esta clasificación debe considerar el tipo de la tarea y la información relevante utilizada por la misma [Seungil *et al.*, 2001].
- f) Debe existir el control del estado de las tareas en desconexión, las aplicaciones relacionadas con las tareas y los datos utilizados durante la desconexión [Seungil *et al.*, 2001].

III.5 Requerimientos establecidos por *SysCoor*

Por otra parte, los requerimientos que establece la arquitectura de *SysCoor* fueron determinados por medio de una revisión detallada de la herramienta.

- a) El uso del lenguaje XML debe mantenerse como medio para apoyar la interoperabilidad.
- b) El control centralizado del estado de los procesos se debe realizar por medio de un administrador global de procesos.
- c) Se debe permitir a los agentes del proceso visualizar el estado de las tareas involucradas.

De esta forma, hemos obtenido requerimientos a partir de tres fuentes: requerimientos obtenidos directamente de los escenarios, requerimientos que la literatura existente ha propuesto y los requerimientos que la arquitectura *SysCoor* establece. De igual forma, este conjunto de requerimientos nos permiten establecer un marco más general de requerimientos para el apoyo a la operación desconectada. Este marco se presenta a continuación.

III.6 Requerimientos genéricos

En resumen, podemos agrupar los requerimientos obtenidos de los casos de estudio, del trabajo previo y los existentes en *SysCoor* en cuatro áreas generales que se deben considerarse para el apoyo a la operación desconectada en sistemas de workflow, estas cuatro áreas son:

1. *Evaluación de la factibilidad en la desconexión de un rol.*
2. *Provisión de información al cliente previo a la desconexión.*
3. *Sincronización de la información generada o modificada durante la desconexión cuando ésta finaliza.*

4. *Control de la coordinación, principal característica de los sistemas de workflow mientras los clientes se encuentran desconectados.*

Estos requerimientos y la forma en como se pueden implementar en *SysCoor* se describen en el siguiente capítulo a través de extensiones que buscan apoyar el trabajo móvil y la operación desconectada en sistemas de workflow. Consideramos que estas extensiones cubren los requerimientos encontrados en nuestra revisión de la literatura y en la revisión de los casos de estudio, e igualmente, aprovechan la funcionalidad de los componentes en la arquitectura existente.

III.7 Resumen

En el presente capítulo se han descrito escenarios obtenidos a partir de la realización de dos casos de estudio. Estos casos de estudio han sido de suma importancia ya que a partir del análisis de la forma en como las personas llevan a cabo sus actividades, hemos obtenido características y requerimientos para apoyar el trabajo, cuando éste requiere movilidad de los agentes dueños de los procesos. Así mismo, la revisión de los trabajos existentes en el área de workflow desconectado nos ha permitido analizar los mecanismos que han sido utilizados para apoyar este tipo de tecnología.

En el siguiente capítulo, se describe ampliamente la arquitectura básica de *SysCoor* y se presentan las extensiones propuestas en términos de mecanismos funcionales cuyo propósito es extender la arquitectura de forma que apoye la operación desconectada en los sistemas de workflow que esta misma arquitectura ya genera.

Capítulo IV. Arquitectura de *SysCoor* y Extensiones para el apoyo al Workflow Desconectado

IV.1 Introducción

En base a las cuatro áreas de requerimientos establecidos en el capítulo anterior, el presente capítulo describe la propuesta para extender la funcionalidad actual de *SysCoor* hacia la generación de sistemas de workflow que soporten el trabajo móvil y desconectado. Se presenta parte del análisis y diseño de las extensiones propuestas para la arquitectura por medio del uso del Lenguaje de Modelado Unificado (*Unified Modeling Language - UML*).

Con la finalidad de poder plantear la forma en como se propone cumplir con las cuatro áreas de requerimientos establecidas en el capítulo anterior, es necesario explicar los componentes básicos que forman la arquitectura de *SysCoor*, ya que depende de éstos la forma en como se cubren los requerimientos establecidos previamente.

IV.2 La Arquitectura de *SysCoor*

García Carrillo desarrolló un Sistema Administrador del Flujo de Trabajo (WFMS) llamado *SysCoor* cuya arquitectura permite la generación semi-automática de sistemas de workflow que pueden ser ejecutados en navegadores Web [García Carrillo, 2001]. El diseño de esta arquitectura se basó en el desarrollo de sistemas de workflow definidos a través del concepto de rol² y la coordinación entre ellos para la automatización de procesos organizacionales. Más tarde, Ramírez Fernández extendió la arquitectura propuesta por García Carrillo con la finalidad de ampliar la funcionalidad de *SysCoor* [Ramírez Fernández, 2002].

² Un rol dentro de un proceso es una unidad funcional de responsabilidad asignada a un agente [Humphrey y Feiler, 1992]. Un rol representa un conjunto definido de actividades agrupadas con el fin de lograr un objetivo.

La arquitectura definida por García Carrillo y Ramírez Fernández [García Carrillo, 2001; Ramírez Fernández, 2002] se muestra en la figura 4. Esta arquitectura se forma de cuatro componentes: un *generador de sistemas de workflow*, un *contenedor de datos y modelos*, una *máquina de coordinación* y el componente conocido como *rol base*.

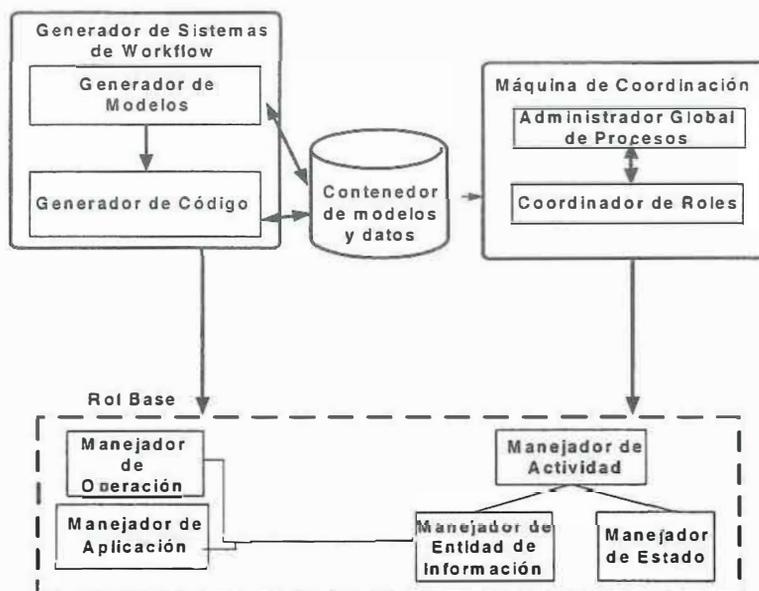


Figura 4. Arquitectura SysCooor propuesta de García Carrillo y Ramírez Fernández

El *rol base* representa los aspectos de un rol dentro del proceso, como las actividades internas, los estados por los cuales atraviesa, las entidades de información que utiliza, las operaciones y las aplicaciones externas que son invocadas durante la ejecución de éste [Arroyo *et al.*, 2003].

El *generador de sistemas de workflow* se compone de dos submódulos: el *generador de modelos* y el *generador de código*. El *generador de modelos*, como se explicó en el capítulo II, es responsable de leer y procesar los *modelos base RAD, STD y EI*, y crear el *modelo base WF* en XML. Por su parte, el generador de código crea la estructura del *rol base*.

En el *contenedor de modelos y datos* se encuentra la información de los sistemas de workflow que han sido generados, los *modelos base* y las entidades de información (datos relevantes).

En la *máquina de coordinación* se encuentran el *administrador global de procesos (AGP)* y el *coordinador de roles*, los que en conjunto forman el mecanismo que coordina las actividades que realizan los roles involucrados en un proceso, estableciendo la comunicación entre ellos mediante el envío de mensajes de acuerdo a las interacciones definidas entre los mismos [García Carrillo, 2001]. La *máquina de coordinación* es la base del WFMS ya que es responsable de realizar y mantener la colaboración, comunicación, coordinación e intercambio de información entre los procesos.

Como se ha mencionado anteriormente, *SysCoor* permite la generación semi-automática de sistemas de workflow a partir de modelos de procesos descritos en el lenguaje XML. Al señalar que los sistemas se generan de forma semi-automática, se hace necesario describir cuales son las tareas que un *modelador* debe realizar de forma manual. Básicamente, cada vez que se genera un *modelo base WF*, el *modelador* debe de agregar una serie de detalles al modelo que incluyen la definición de la interfaz gráfica de usuario (GUI) para aquellas actividades que así lo requieran y el orden en el que deben aparecer las pantallas. Una pantalla se refiere al conjunto de elementos (cuadros de texto, botones, etiquetas, etc) de despliegue gráfico. También debe especificar el flujo en el envío de información entre roles, es decir, quién emite y quién recibe entidades de información durante la ejecución del rol.

Cabe mencionar que los cuatro modelos base (*RAD*, *DTE*, *EI* y *WF*), son validados por el *generador de modelos*, para asegurarse de que se encuentren bien escritos y cumplan con las reglas de sintaxis correspondientes que se describen en las definiciones de tipo de documentos (DTDs) los cuales se presentan a detalle en el apéndice B. Ya que se cuenta con el *modelo base WF* validado, el *generador de código* es el encargado de generar los

roles base del proceso, que serán instanciados durante la ejecución del sistema de workflow.

Como se ilustra en la figura 4, para cada *rol base* se definen *manejadores de actividades, de estados, de entidades, de operaciones y de aplicaciones*. En el *manejador de actividades* se encuentra la definición de todas las actividades de un rol particular, de manera similar en el *manejador de estados* y el *manejador de entidades*, se almacena la información del estado del rol y de las entidades de información utilizadas (solo la referencia a éstas).

Antes de continuar con la descripción de las extensiones que se proponen para apoyar la operación desconectada en *SysCoor*, es preciso detallar la forma en que los sistemas de workflow generados por la herramienta proveen funcionalidad y permiten la automatización de un proceso.

IV.3 Funcionalidad de SysCoor

En el capítulo II se explicó como a través de modelos descritos en el lenguaje XML (*RAD, DTE y EI*) conocidos también como modelos base, se logra la definición de un sistema de workflow conocido como el *modelo base WF*. A continuación se describirá como un modelo pasivo del sistema de workflow definido en XML se vuelve un modelo activo (sistema de workflow ejecutable).

Una vez que el modelo base de workflow (en XML) ha sido generado, éste sirve como entrada al generador de código, el cual se encarga de leer este modelo y generar el *rol base* a través de un *objeto serializado*. Este *objeto serializado* contiene toda la información necesaria para llevar cabo el control interno del rol así como las interacciones con otros roles. Un *objeto serializado* permite guardar en un archivo, un objeto definido en el lenguaje de programación Java [Deitel y Deitel, 1997].

Cuando un usuario del proceso se conecta al servidor, donde residen las definiciones de los sistemas de workflow, a través de un navegador Web y elige uno de esos sistemas para ejecutar, una serie de páginas definidas en el lenguaje de marcado HTML entran en actividad, estas páginas representan los escenarios típicos de un workflow como son: esperar una respuesta, enviar un documento, etc.

Estas páginas HTML llaman a los *objetos serializados* que representan los *roles base* y obtienen su contenido para desplegar información o bien obtener la próxima actividad por ejecutar. La coordinación entre los roles participantes y los estados de los *roles base* son controlados por la máquina de coordinación. En la figura 5 se muestra un diagrama general sobre los elementos involucrados durante la transformación de un *modelo base WF* a un sistema de workflow ejecutable y su respectiva ejecución como acabamos de explicar.

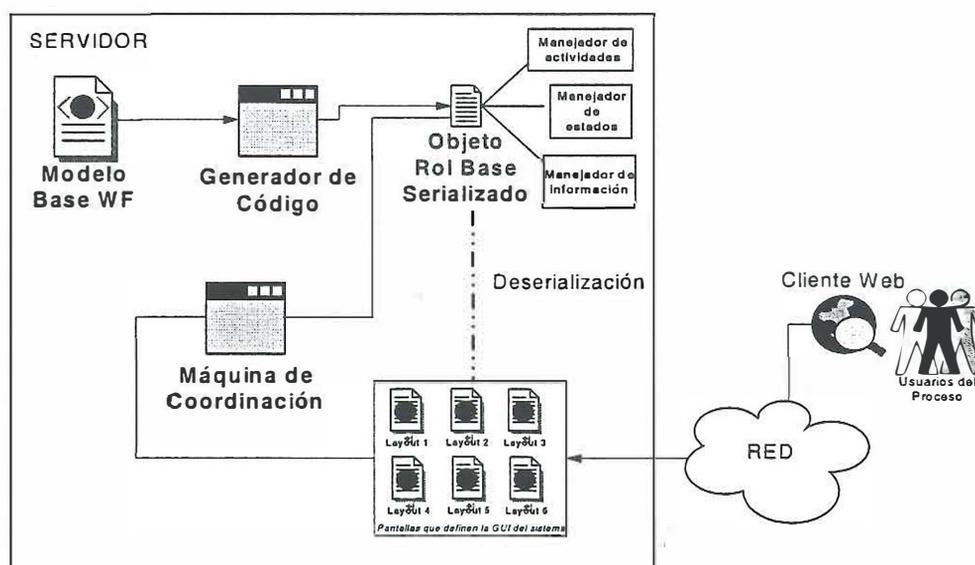


Figura 5. Esquema general de la funcionalidad de los sistemas generados por SysCoor

La explicación de los componentes que forman la arquitectura de SysCoor nos permite ahora presentar la forma en que las cuatro áreas de requerimientos planteadas en el capítulo anterior buscan ser cubiertas. Con el entendimiento del funcionamiento de los

componentes de la arquitectura básica de *SysCoor* podemos ahora explicar como buscamos apoyar la operación de desconectada aprovechando la arquitectura existente, y definiendo extensiones para cumplir con aquellas áreas que la arquitectura básica no logra cubrir.

Recordando, las cuatro áreas generales que se deben considerarse para el apoyo a la operación desconectada en sistemas de workflow son:

1. *Evaluación de la factibilidad en la desconexión de un rol.*
2. *Control de la coordinación (principal característica de los sistemas de workflow) mientras los clientes se encuentran desconectados.*
3. *Provisión de información al cliente previo a la desconexión.*
4. *Sincronización de la información generada o modificada durante la desconexión cuando ésta finaliza.*

En la siguiente sección, explicaremos cada una de estas áreas y las extensiones propuestas para cubrirlas.

IV.4 Extensiones a la arquitectura para apoyar el trabajo desconectado en sistemas de workflow

Con la finalidad de extender la arquitectura del WFMS *SysCoor*, hemos propuesto varias extensiones para implementar soporte para usuarios móviles que requieren desconexión en sistemas de workflow. Consideramos que estas extensiones cubren los requerimientos encontrados en nuestra revisión de la literatura y en la revisión de los casos de estudio, e igualmente, aprovechan la funcionalidad de los componentes en la arquitectura existente.

Requerimiento: Evaluación de la factibilidad de la desconexión de un rol.

En sistemas de workflow no es posible apoyar la desconexión de tareas de cualquier tipo ni en cualquier momento. Con respecto al tipo de la tarea, se distinguen entre tareas manuales y tareas automáticas. Las tareas automáticas son aquellas que dependen fuertemente para su ejecución de otros mecanismos externos al WFMS que las controla, estos mecanismos externos pueden ser sistemas de información u otras aplicaciones externas al WFMS. En este caso, el transferir la tarea involucra el transferir la o las aplicaciones externas utilizadas al dispositivo, lo cual resulta difícil por varios factores como el tamaño de la aplicación externa o por las características del ambiente en el que ésta trabaja. Por esta razón, se recomienda evaluar el tipo de la tarea antes de la desconexión y limitarla a tareas manuales [Seungil *et al.*, 2001].

El no poder apoyar una tarea en cualquier momento se debe a que las actividades de workflow requieren del uso de información que debe estar disponible al momento en el que se desea llevar a cabo la tarea. Esta información puede no encontrarse disponible al momento en el que la desconexión es necesaria.

Para cubrir este requerimiento definimos un mecanismo llamado *mecanismo de determinación de desconexión*, el cual se encarga de evaluar las entidades de información que requiere un rol para llevar a cabo su función y verificar el estado de éstas. Asimismo, verifica si existen relaciones del rol con otros sistemas de información los cuales no puedan ser transferidos al dispositivo. De esta forma, se evalúa si la desconexión de una tarea es posible en base a la disponibilidad de la información necesaria para ejecutarla y las relaciones del rol con otros roles o con aplicaciones externas.

Requerimiento: Control de la coordinación mientras los clientes se encuentran desconectados.

Uno de los principales requerimientos de los WFMSs es el mantener las interacciones entre los usuarios que participan en el workflow de forma coordinada. Este requerimiento es cubierto en la arquitectura de *SysCoor* a través del *administrador global de procesos (AGP)* que se encarga de coordinar las actividades que realizan los roles involucrados en un proceso, además de establecer la comunicación entre ellos [García *et al.*, 2002]. Sin embargo este componente supone que existe una conexión permanente de comunicación a través de la cual los estados de los roles y las modificaciones a éstos pueden ser realizadas. Si consideramos que durante la desconexión se debe mantener la coordinación entre roles que residen en el PDA, necesitamos un mecanismo en el dispositivo que lleve a cabo tareas similares a las del *AGP*. En este sentido definimos un mecanismo llamado *coordinador local*. Este coordinador local controla la comunicación entre los roles que se encuentran en el PDA, así como entre las actividades dentro de un rol, llevando a cabo el manejo de los estados de tareas y roles y de las entidades de información utilizadas.

Requerimiento: Provisión de información al cliente previo a la desconexión.

Previo a una desconexión, se debe proveer al cliente con la información suficiente para continuar su trabajo sin necesidad de consultar el contenedor central [Yung, 2002]. Esta información incluye no solo datos, sino también las definiciones de las tareas que se desconectan. Este requerimiento implica dos puntos: la definición de la información que se desconectará (actividades y datos) y el mecanismo encargado de realizar la transferencia.

Además, es importante seguir apoyando la interoperabilidad a través del uso de XML. De aquí que hemos propuesto un modelo definido en XML llamado el *modelo base de workflow desconectado (modelo base WFD)* para definir los datos y las actividades necesarios para trabajar de forma desconectada. Este modelo define un sistema de

workflow con información suficiente respecto a las actividades y datos, de forma que pueda ejecutarse en un dispositivo PDA sin necesidad de consultar el contenedor central cuando se está ejecutando la tarea. Este modelo será analizado a detalle en el siguiente capítulo.

Para realizar la transferencia de la información desde la máquina central de workflow (AGP) al dispositivo PDA y viceversa, se propone un *mecanismo de transferencia de documentos XML* el cual es responsable de transferir el *modelo base WFD* (descrito en el siguiente capítulo) que define al sistema de workflow desconectado y los datos utilizados.

Requerimiento: Sincronización de la información generada o modificada durante la desconexión cuando ésta finaliza.

En la etapa de reconexión, los estados de los procesos centrales son actualizados y los resultados de las operaciones efectuadas durante la desconexión son reportados [Alonso *et al.*, 1995]. En este sentido se plantean tres requerimientos: la definición de los datos y los cambios que son reportados al final de la desconexión, el mecanismo que transfiera esta definición del PDA a la máquina central y un mecanismo que realice las funciones de manejar candados en los estados de roles y entidades de información, de forma que durante la ejecución global del sistema de workflow exista conocimiento cuando un rol se encuentran en desconexión o bien, cuando se reconecta, así mismo, que integre los resultados de las operaciones que se llevaron a cabo durante la desconexión. Para implementar este último requerimiento proponemos un mecanismo conocido como *mecanismo de sincronización*.

Para la definición de los datos y los cambios que necesitan ser reportados a la máquina central de workflow, proponemos un documento XML que permite la definición de los datos y las actividades que regresan después de una desconexión. A este documento

lo llamamos el *modelo base de reconexión*. Este documento, al igual que los otros modelos en XML propuestos, se detalla en el capítulo siguiente.

Para transferir esta definición del PDA a la máquina central se utiliza el *mecanismo de transferencia de documentos XML*.

La definición del sistema de workflow desconectado a través del uso del lenguaje XML (*modelo base WFD*), establece la necesidad de proponer otros mecanismos para convertir esa definición en un sistema de workflow funcional, esta necesidad da lugar a los mecanismos que forman la arquitectura cliente.

IV.4.1 Mecanismos en el Cliente

El *modelo base WFD* se forma de tres partes, la definición del proceso, de los datos y de las pantallas del usuario. Para manejar la definición del proceso, de los datos e implementar la funcionalidad descrita en la definición del sistema, definimos un *rol base* en el dispositivo. Este rol base reúne la información de las actividades del rol (secuencia, entidades de información utilizadas y operaciones) de la misma forma que el *rol base* en el servidor. Definimos también un *generador de interfaz gráfica del usuario* residente en el dispositivo el cual interpreta la definición de la pantallas del usuario y la transforma en elementos de interfaz gráfica.

Dado que los dispositivos PDA tienen recursos limitados, se propone una arquitectura cliente ligera con mecanismos mínimos. Esta arquitectura ligera se compone de cuatro componentes principales: el *coordinador local*, el *generador de interfaz gráfica del usuario*, el *rol base* y un *contenedor de datos*.

Podemos ahora agrupar los mecanismos definidos a partir de las cuatro áreas de requerimientos analizadas al inicio de esta sección e incorporarlos en la arquitectura básica del servidor de *SysCoor* descrita anteriormente.

IV.4.2 Mecanismos en el Servidor

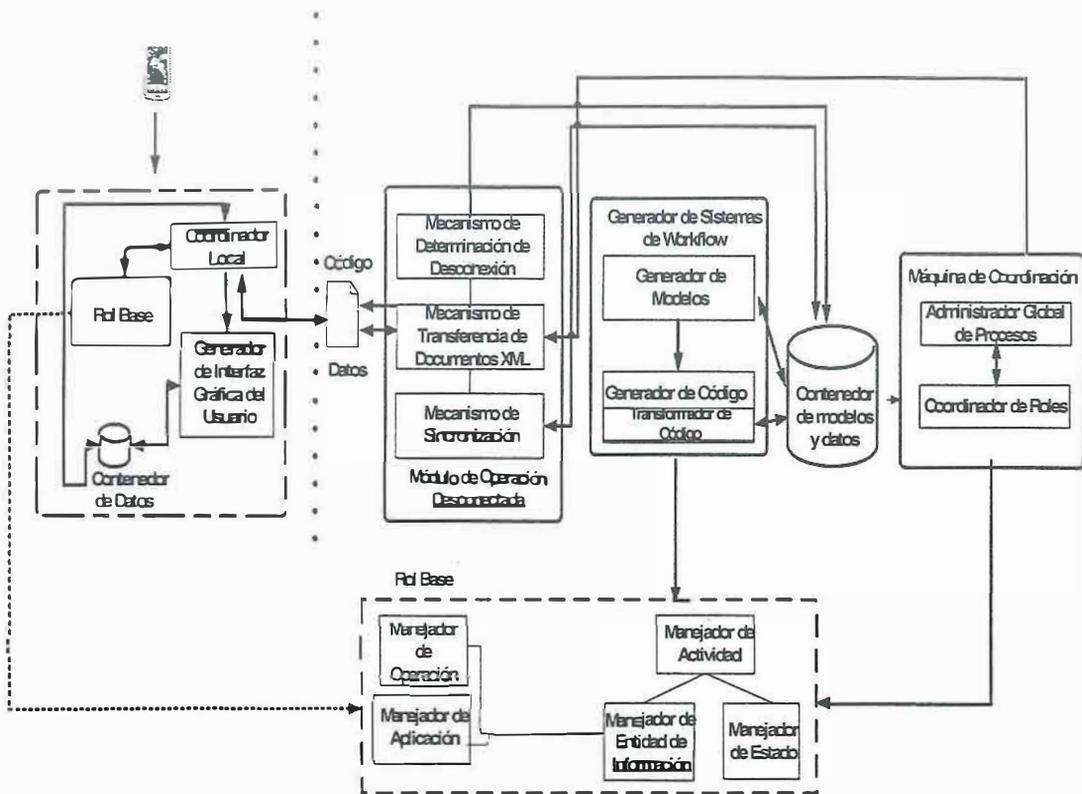


Figura 6. Arquitectura de *SysCoor* con extensiones propuestas

De acuerdo al análisis de las cuatro áreas de requerimientos y a la forma en como se les propone dar soporte, seguiremos haciendo uso de la máquina de coordinación (AGP) existente en *SysCoor* para mantener la coordinación entre los agentes que participan en el proceso, ya sea de modo conectado o desconectado. También seguiremos haciendo uso del *generador de sistemas de workflow*. Sin embargo, motivados por las diferencias entre GUIs de computadoras de escritorio (plataforma original de *SysCoor*) y dispositivos PDAs,

implementamos un *transformador de código* en el *generador de sistemas de workflow*, el cual manejará la creación del *modelo base WFD*. Adicionalmente, introducimos un nuevo módulo llamado *Módulo de Desconexión*. Este módulo consiste de tres mecanismos: *mecanismos de determinación de desconexión*, *mecanismo de transferencia de documentos XML* y el *mecanismo de sincronización*. En la figura 6 presentamos la arquitectura extendida de *SysCoor* de acuerdo a los mecanismos establecidos.

En la siguiente sección se presenta parte del análisis y diseño llevado a cabo para implementar las extensiones propuestas a la arquitectura de *SysCoor*.

IV.5 Análisis y diseño de las extensiones para apoyar el trabajo desconectado en *SysCoor*

En esta sección presentamos el análisis y el diseño que se ha llevado a cabo con la finalidad de extender la funcionalidad de *SysCoor*. Este proceso de desarrollo se llevó a cabo a través del Lenguaje de Modelado Unificado (UML por sus siglas en Inglés). UML es un conjunto de especificaciones utilizadas para modelar los componentes de un desarrollo de software [Booch *et al.*, 1999]. Con el propósito de ilustrar el análisis y diseño de las extensiones, se ilustra la propuesta a través de casos de uso, diagramas de secuencia y el diagrama de clases de los elementos que construyen las extensiones.

IV.5.1 Casos de Uso

El propósito de los casos de uso es mostrar la interacción que tienen los actores con un sistema o las metas que éstos desean alcanzar. Un actor puede ser tanto una persona como un sistema [Booch *et al.*, 1999]. A continuación se presentan algunos de los casos de uso más representativos de las extensiones a la arquitectura de *SysCoor*.

a) Transformador de Código

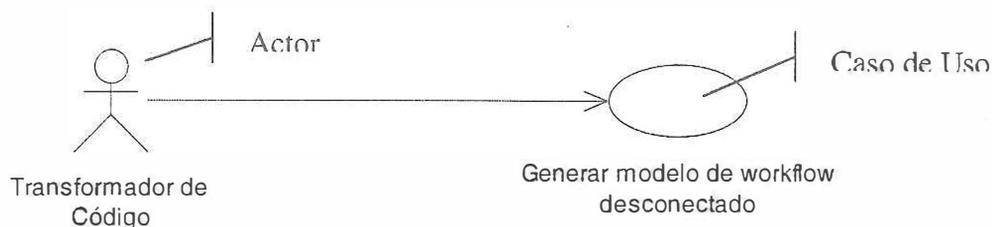


Figura 7. Caso de uso “*generar modelo de workflow desconectado*”

Caso de Uso: Generar el modelo base de workflow desconectado (*modelo base WFD*).

Actor: Transformador de Código.

Propósito: A partir del *modelo base WF* que ha sido generado de los modelos *base RAD, STD y EI*, se genera una nueva representación del sistema de workflow para ser ejecutado en dispositivos PDA.

Descripción: Una vez que se ha elaborado el *modelo base WF*, el *transformador de código*, implementado dentro del *generador de código*, toma la definición del *modelo base WF* y genera un nuevo modelo, el *modelo base workflow desconectado (modelo base WFD)*; el cual proporciona la definición de un sistema de workflow desconectado; reutilizando la información ya definida en el *modelo base WF* que proporciona una representación dinámica para el sistema de workflow. El *transformador de código* sin embargo, agrega la información referente a las pantallas definidas de acuerdo a las actividades de cada uno de los roles. Por otro lado, crea una serie de estructuras de datos a partir de la definición del *modelo base WF* referente a los roles, sus actividades, relaciones con otros roles y con entidades de información. Esta información será utilizada en fases posteriores.

El *modelo base WFD* es almacenado en el *contenedor de modelos y datos* para ser accesado más tarde por otros componentes.

b) *Mecanismo de Determinación de Desconexión*



Figura 8. Caso de uso “*Evaluar la factibilidad de la desconexión de un rol*”

Caso de Uso: Evaluar la factibilidad de la desconexión de un rol.

Actor: Mecanismo de Determinación de Desconexión.

Propósito: Examinar las entidades de información utilizadas por las tareas de un rol y los estados de estas entidades para determinar la factibilidad en la desconexión de ese rol.

Descripción: La información de interacciones así como la de entidades de información y sus estados es obtenida por el *transformador de código* cuando éste genera el *modelo base WFD* y se encuentra almacenada en estructuras de datos. De esta forma el *mecanismo de determinación de desconexión* verifica en el contenedor, los estados de las entidades de información utilizadas, si éstas se encuentran disponibles, este mecanismo notifica al *mecanismo de transferencia de documentos XML* para preparar el *modelo base WFD* para ser transferido. En caso que las entidades de información utilizadas por el rol no puedan ser utilizadas se notifica al usuario.

c) *Mecanismo de Transferencia de Documentos XML*

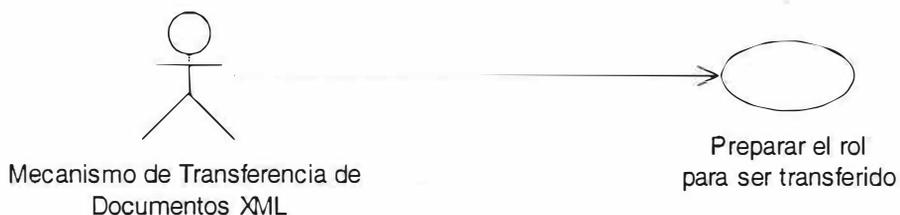


Figura 9. Caso de uso “*Preparar el rol para ser transferido*”

Caso de Uso: Preparar el rol para ser transferido

Actor: Mecanismo de Transferencia de Documentos XML.

Propósito: Recuperar del contenedor de modelos el *modelo base WFD* para preparar el o los roles para ser transferido al dispositivo, esta operación involucra también, el agregar los datos que son utilizados por el rol y que necesitan ser transferidos.

Descripción: Una vez que ha sido determinado que la desconexión es posible, el *mecanismo de transferencia de documentos XML* inicia la fase conocida como “*recolección de datos*”. En esta fase, este mecanismo prepara el *modelo base WFD* para ser transferido. Con este fin, el mecanismo recupera el *modelo base WFD*, obtiene la definición completa solo del rol o roles que se desconectan, recupera del contenedor de modelos y datos la definición de los datos que utilizan los roles y los agrega a la definición de los roles. De esta forma, deja preparado el *modelo base WFD* con la definición completa para que el sistema trabaje de forma independiente del servidor central. En seguida, espera por un cliente PDA para establecer la comunicación con el dispositivo y enviar la definición de los roles por medio del *modelo base WFD*.

d) *Mecanismo de Transferencia de Documentos XML y Mecanismo de Sincronización*

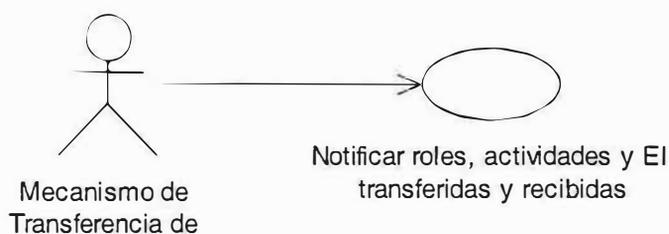


Figura 10. Caso de uso “*Notificar roles, actividades y EI preparadas para transferir*”

Caso de Uso: Notificar roles, actividades y entidades de información (IE) preparadas para transferir.

- Actor: Mecanismo de Transferencia de Documentos XML.
- Propósito: Notificar al *mecanismo de sincronización* sobre los roles, las actividades y los datos que han sido preparados para ser transferidos al dispositivo.
- Descripción: El *mecanismo de transferencia de documentos XML* comunica al *mecanismo de sincronización* sobre los roles y datos que están listos para ser transferidos. De esta forma, el *mecanismo de sincronización* actualiza los estados de las entidades de información en el contenedor, así como el estado del rol o roles.

e) *Mecanismo de Sincronización*

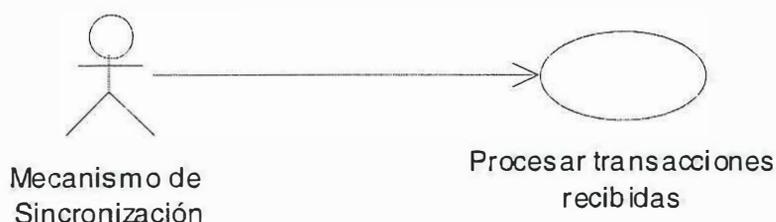


Figura 11. Caso de uso “*Procesar transacciones recibidas*”

- Caso de Uso: Procesar transacciones recibidas.
- Agente: Mecanismo de Sincronización.
- Propósito: Recibir las transacciones llevadas a cabo sobre los datos durante la desconexión y actualizar la información en el contenedor de datos.
- Descripción: El *mecanismo de transferencia de documentos XML* espera por un cliente PDA para establecer la comunicación con el dispositivo y recibir la definición de los roles que se reconectan junto con los datos que fueron modificados en cada actividad (esta definición se forma en base a las reglas establecidas en el DTD del *modelo base de reconexión*). El mecanismo de transferencia analiza estos datos, los separa y los comunica al *mecanismo de sincronización*.

El *mecanismo de sincronización* actualiza los datos modificados durante la desconexión y actualiza la copia que existe de estos datos en el contenedor central. De la misma forma, utiliza la información de los roles que se reconectan para actualizar el estado de estos, de forma que el *AGP* pueda notificar al usuario conectado sobre los roles que han terminado un período de desconexión.

Una vez descrito el comportamiento esperado de los componentes del sistema por medio de las metas que éstos desean alcanzar (casos de uso), ahora presentamos el comportamiento dinámico de estos componentes a través de *diagramas de secuencia*. Los *diagramas de secuencia* nos permiten modelar las interacciones entre las instancias de los objetos en un contexto de colaboración y son principalmente usados para modelar las interacciones a alto nivel de los objetos de un sistema [Booch *et al.*, 1999].

IV.5.2 Diagramas de Secuencia

a) *Evaluar la factibilidad de la desconexión de un rol*

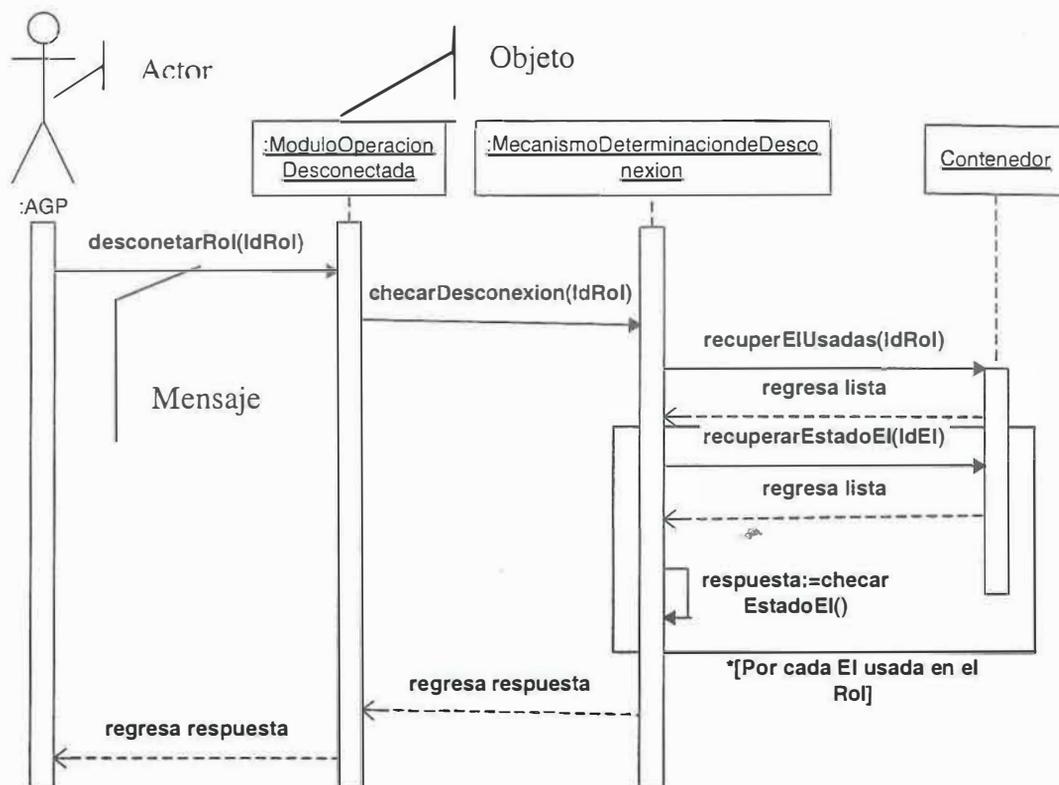


Figura 12. Diagrama de secuencia del caso de uso “*Evaluar la factibilidad de la desconexión de un rol*”

Las actividades de recuperar y analizar las interacciones de un rol y las entidades de información usadas por el mismo, son ejecutadas por el *mecanismo de determinación de desconexión* el cual es activado por petición del *módulo de operación desconectada*. En la figura 12 se presenta el diagrama de secuencia que muestra la secuencia en que se lleva a cabo la evaluación de la factibilidad de la desconexión de un rol. Como se muestra en la figura, el *AGP*, hace una petición al *módulo de operación desconectada* para desconectar un rol, este módulo, pide al *mecanismo de determinación de desconexión* verificar si la desconexión se puede llevar a cabo. Para realizar esta tarea, este mecanismo recupera del *contenedor de modelos y datos* la información de todas las entidades de información que

son utilizadas por el rol y el estado de las mismas. El *mecanismo de determinación de desconexión*, regresa al *módulo de operación desconectada* una respuesta, si la respuesta es negativa, se envía un mensaje al *AGP* explicando la razón por la que la desconexión no es posible. Si la respuesta es positiva, la fase de recolección de datos inicia.

b) *Preparar el rol para ser transferido*

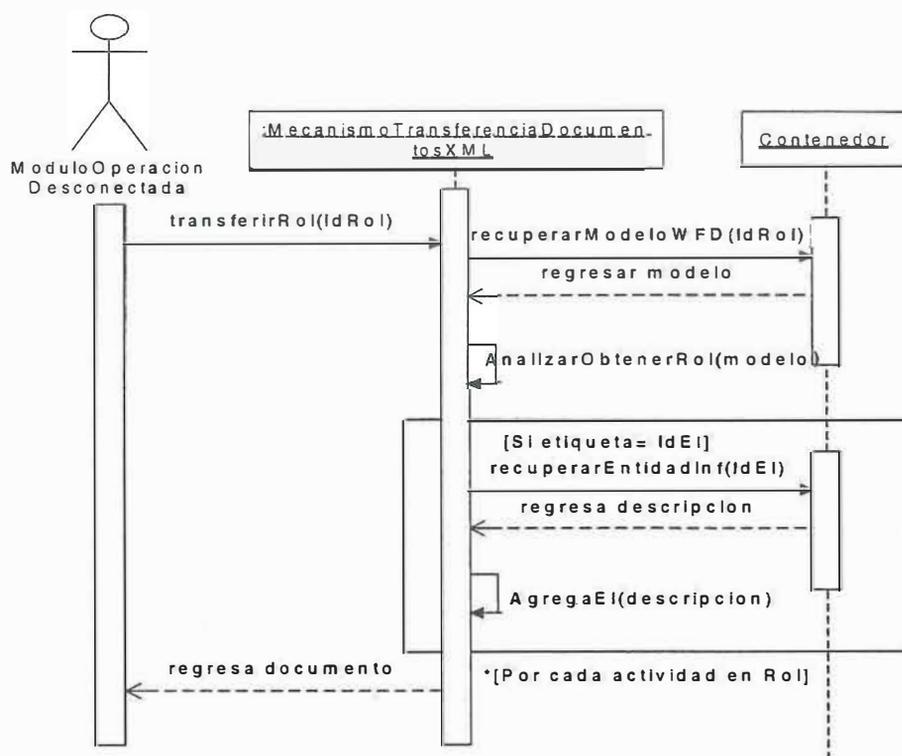


Figura 13. Diagrama de secuencia del caso de uso “Preparar el rol para ser transferido”

El *módulo de operación desconectada* envía al *mecanismo de transferencia de documentos XML* la solicitud de desconexión de uno o más roles. El mecanismo recupera del *contenedor de modelos y datos* el *modelo base WFD*, y mediante un análisis sintáctico y léxico de este documento, obtiene solo la definición del rol o los roles que se desconectan. Durante este mismo análisis, localiza en la definición de los roles la referencia a las entidades de información que son utilizadas durante la ejecución del rol, y con esta referencia, busca y recupera del *contenedor de modelos y datos* el contenido de las

entidades de información, los agrega a la definición de los roles que se desconectarán y los deja listos para que sean transferidos al dispositivo PDA.

c) *Notificar roles, actividades y EI preparadas para transferir*

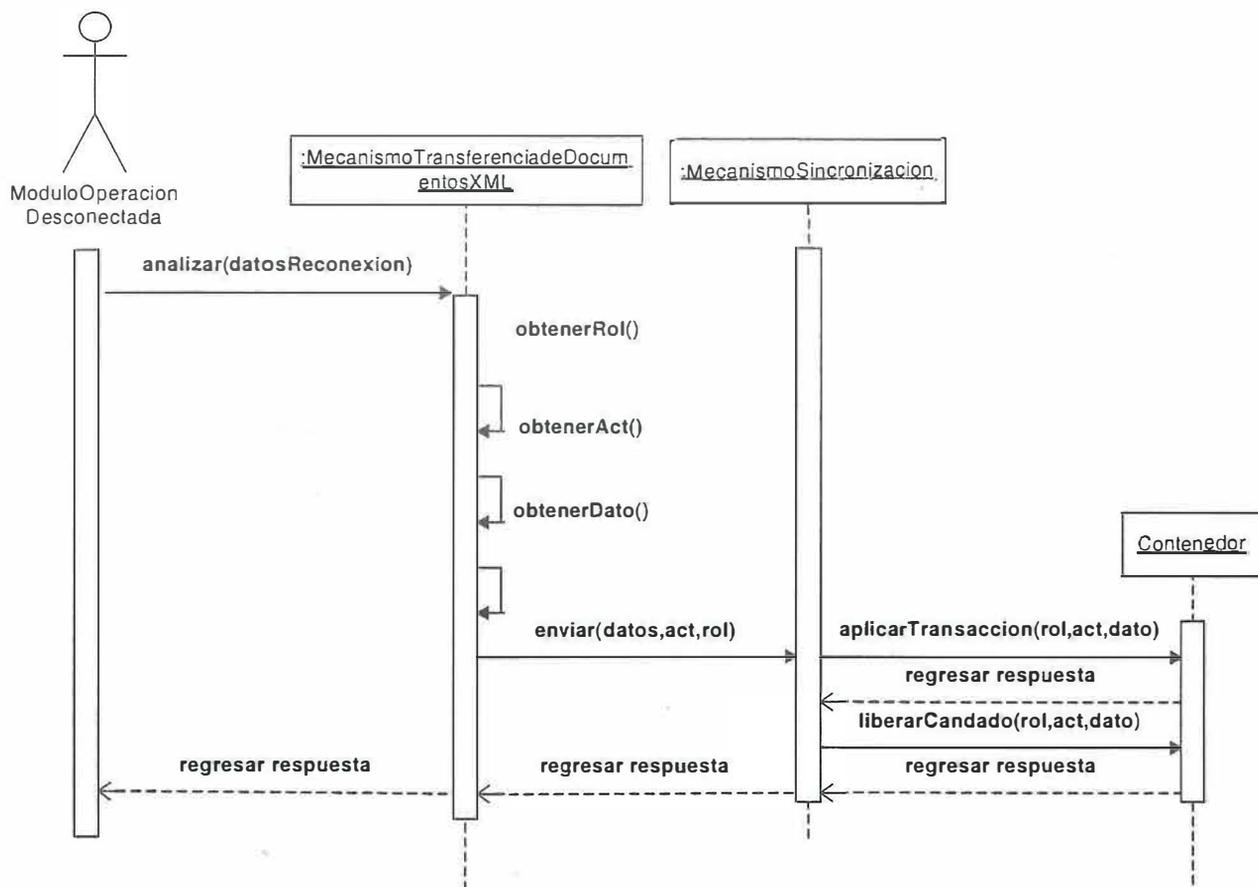


Figura 14. Diagrama de secuencia del caso de uso “Notificar actividades y Entidades de Información (IE) preparadas para transferir”

Una vez que el *mecanismo de transferencia de documentos XML* ha preparado la definición de los roles para ser transferidos, este comunica al *mecanismo de sincronización* sobre los roles, sus actividades y los datos que fueron transferidos al dispositivo PDA. En la figura 14 se muestra la secuencia en la que ocurren estas actividades. Una vez que ha notificado esta información, el *mecanismo de sincronización* accesa el *contenedor de modelos y datos* en donde se encuentran las estructuras de datos que controlan los estados de los elementos del proceso, para colocar candados en las entidades de información y las

actividades que fueron transferidas, de forma que el resto de los roles tengan conocimiento de que estas entidades de información y actividades no se encuentran disponibles.

d) *Procesar transacciones recibidas*

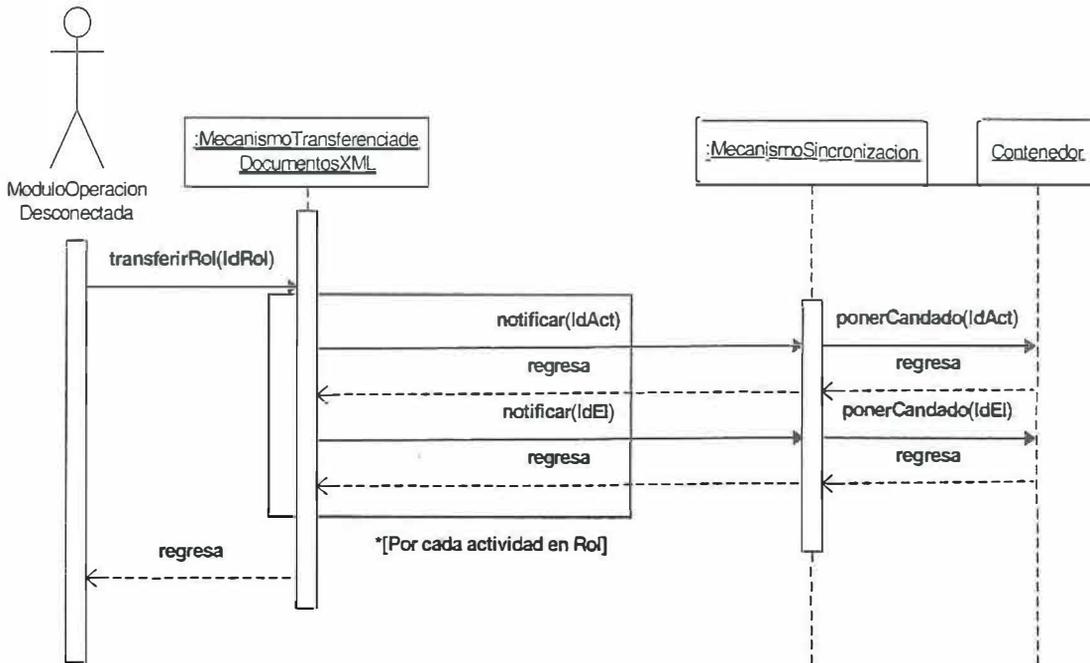


Figura 15. Diagrama de secuencia de los casos de uso “*Procesar transacciones recibidas*”

Cuando un cliente regresa después de un período de desconexión, el *módulo de sincronización* debe integrar los cambios llevados a cabo sobre los datos al contenedor central. Con este fin, el *mecanismo de transferencia de documentos XML* recibe la definición de los roles que se reconectan junto con los datos que fueron modificados en cada actividad. El *mecanismo de transferencia* analiza estos datos, los separa y los comunica al *mecanismo de sincronización*. El *mecanismo de sincronización* actualiza los datos modificados durante la desconexión y actualiza la copia que existe de estos datos en el *contenedor central*. De la misma forma, utiliza la información de los roles que se reconectan para actualizar el estado de estos, de forma que el *AGP* pueda notificar al

usuario conectado sobre los roles que han terminado un periodo de desconexión. La secuencia de estas actividades se muestra en la figura 15.

Para finalizar esta sección de análisis y diseño, se presenta el diagrama de clases el cual es una representación gráfica de la vista estática que muestra una colección de elementos como clases, tipos, y su contenido y relaciones [Rumbaugh *et al.*, 2000].

IV.5.3 Diagramas de Clases

En la figura 16 se presenta el diagrama de las clases más representativas que construyen las extensiones a la arquitectura servidor del sistema de WFMS *SysCoor*. Se muestra la manera en que se relacionan las clases del *AGP*, el *módulo de operación desconectada*, el *transformador de código* y el *contenedor de modelos y dato*.

Como se observa, el *transformador de código* requiere de dos componentes que son: un analizador sintáctico y léxico para la definición de las pantallas de usuario (*DTDModeloGUI*) y un analizador más para la definición de datos (*DTDDocumentoDatos*). El *módulo de operación desconectada* se compone a su vez de tres clases que representan cada uno de los mecanismos que forman este módulo (*de determinación*, *de transferencia* y *de sincronización*).

Las clases mencionadas anteriormente, hacen uso de una clase conocida como *contenedormodelosydatos*, la cual incluye los mecanismos para la recuperación y manipulación información en el *contenedor central de modelos y datos*.

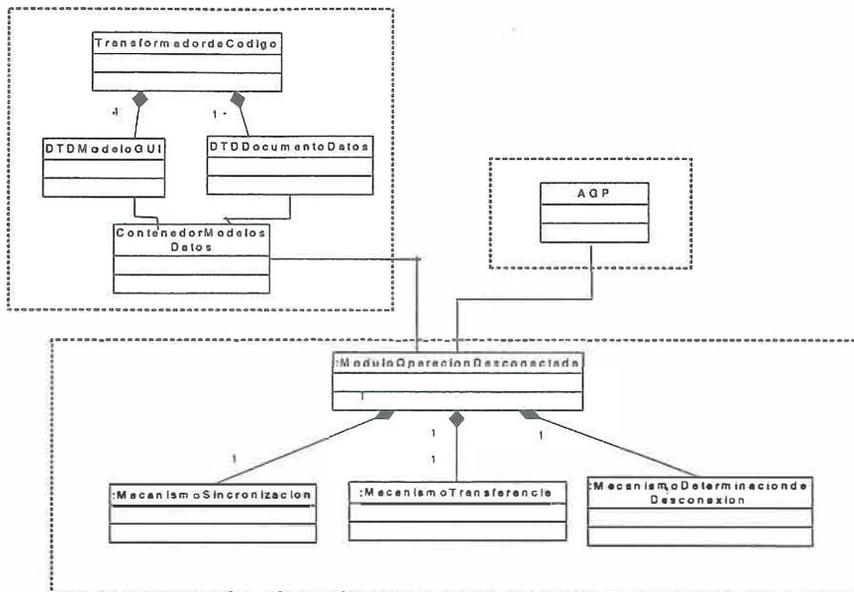


Figura 16. Diagrama de clases de las extensiones propuestas para la arquitectura servidor de SysCoor

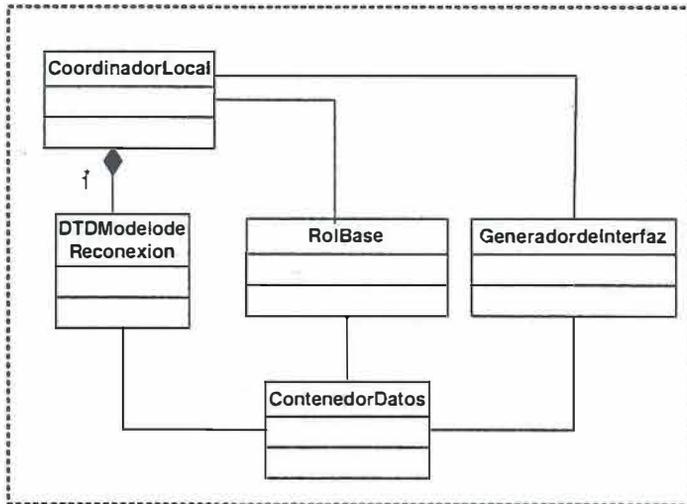


Figura 17. Diagrama de clases para la arquitectura cliente para dispositivos PDA

Por otro lado, la arquitectura cliente se forma de cinco clases las cuales se muestran en la figura 17. Se estableció una clase por cada uno de los mecanismos propuestos en la arquitectura del cliente. Adicionalmente, se incluye un analizador sintáctico y léxico para

formar el documento XML que define las actividades y los datos que son transferidos del dispositivo a la máquina central cuando termina la desconexión.

En el próximo capítulo, se describe la implementación que se llevó a cabo de las extensiones propuestas para apoyar la operación desconectada en el WFMS *SysCoor* a través de la generación de sistemas de workflow capaces de trabajar en modo desconectado.

IV.6 Resumen

En el presente capítulo se presentó la propuesta para extender la funcionalidad de *SysCoor* hacia la generación de sistemas de workflow que soporten el trabajo móvil y desconectado en workflow. A través del cumplimiento de las cuatro áreas de requerimientos establecidas en el capítulo III, se describieron las características que deben observar los mecanismos que se proponen.

La descripción de la funcionalidad esperada de los mecanismo propuestos, se presentó a través de una parte del análisis y diseño de las extensiones para la arquitectura por medio del uso del Lenguaje de Modelado Unificado (UML).

En el siguiente capítulo se presenta la implementación realizada de las extensiones propuestas para el apoyo a clientes móviles desconectados en sistemas de workflow.

Capítulo V. Implementación de las Extensiones para el apoyo al Workflow Desconectado en SysCoor

V.1 Introducción

El presente capítulo presenta el desarrollo llevado a cabo para extender la funcionalidad de *SysCoor* con el fin de brindar apoyo a sistemas de workflow con soporte al trabajo móvil desconectado en dispositivos PDA. Se establecen puntualmente los cambios que fueron realizados a algunos de los elementos existentes y las extensiones que fueron agregadas para apoyar este tipo de operación en la herramienta SysCoor.

Para lograr el apoyo a la operación desconectada, se parte de la definición que SysCoor crea en XML, para un sistema de workflow para el Web, conocida como el *modelo base WF*. A partir de este modelo se genera uno nuevo que permite la creación de un sistema de workflow capaz de trabajar cuando no se cuenta con acceso a una red y por lo tanto, al servidor central de workflow.

La implementación realizada se presenta en términos de la descripción de la tecnología utilizada y su funcionalidad.

V.2 La arquitectura extendida de SysCoor

La arquitectura de *SysCoor*, incluyendo las extensiones para el apoyo a la operación desconectada en sistemas de workflow, se forma de cinco componentes principales (figura 18): el *generador de sistemas de workflow*, el *contenedor de modelos y datos*, la *máquina de coordinación*, el *módulo de operación desconectada* y la *arquitectura cliente* para dispositivos PDA.

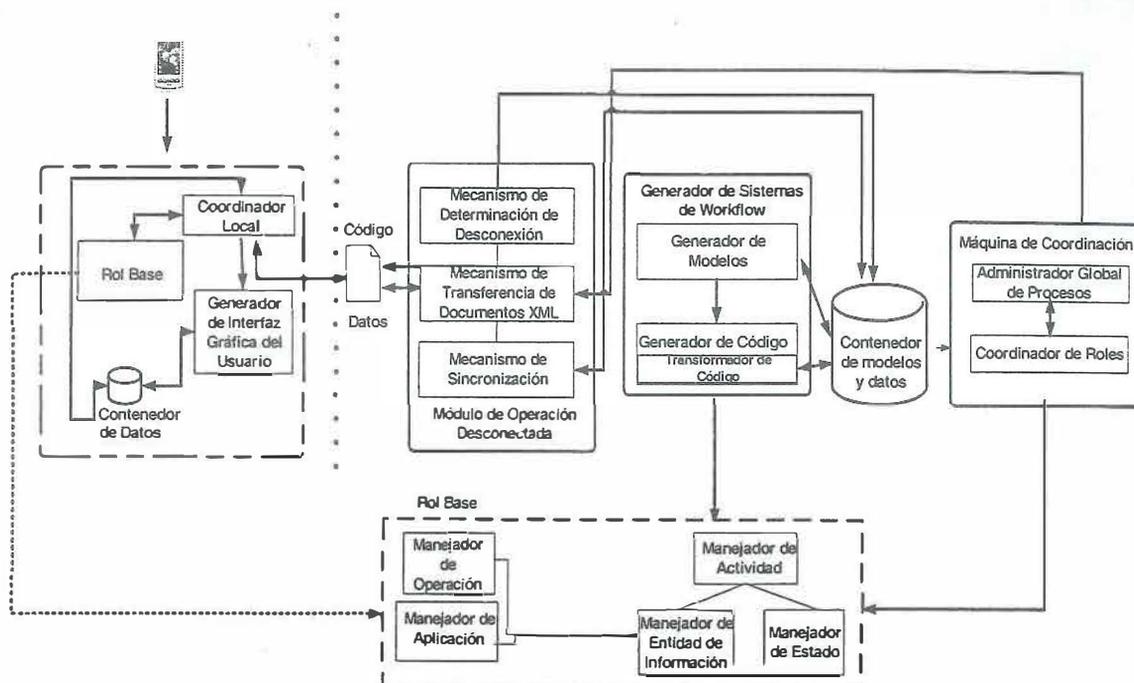


Figura 18. La arquitectura de SysCoor incluyendo las extensiones para el apoyo a la operación desconectada en sistemas de workflow como se presentó en el capítulo anterior

La implementación de las extensiones a la arquitectura consistió de modificaciones a los elementos existentes y desarrollo de nuevos componentes. Mas específicamente las extensiones realizadas fueron:

1. Mecanismos en el Servidor

- a) Modificación del *generador de código* para introducir un *transformador de código* que permite la generación del modelo base de workflow desconectado (modelo base WFD).
- b) Definición del *modelo base WFD* a partir de los elementos del modelo base WF y de un nuevo modelo para la definición de la interfaz gráfica de usuario en el PDA (modelo base GUI) y de la definición de datos a través de XML.

- c) Modificación de la pantallas principal de la máquina de workflow, para permitir al usuario desconectar un rol, ser notificado de mensajes generados durante las fases de desconexión y reconexión de roles, y observar el estado de los roles y las actividades.
- d) Desarrollo de un módulo de operación desconectada, el cual integra los mecanismos necesarios para evaluar la factibilidad de la desconexión de un rol, transferir y recibir datos hacia y desde un dispositivo PDA y mantener la información de estados y entidades de información, así como los cambios en el contenido de las últimas durante la operación desconectada.

2. Mecanismos en el Cliente

- e) Desarrollo de una arquitectura cliente ligera para dispositivos PDA, la cual permite interpretar la definición en XML de un sistema de workflow y darle funcionalidad.

En las siguientes secciones se presenta la implementación realizada de cada uno de estos puntos.

V.3 Mecanismos en el servidor

SysCoor fue desarrollada en el lenguaje de programación Java 2. Una de sus características principales es el ser independiente de plataformas lo cual quiere decir que si hacemos un programa en Java podrá funcionar en cualquier sistema operativo [Deitel y Deitel, 1996]. En este sentido, se decidió realizar todo el trabajo de desarrollo en las extensiones y modificaciones para el servidor utilizando este mismo lenguaje, con el fin de mantener uniformidad en el desarrollo y al mismo tiempo aprovechar las facilidades de este lenguaje para el desarrollo en Internet.

V.3.1 Transformador de código y modelos base WF y WFD

El *transformador de código*, implementado dentro del *generador de código*, se encarga de tomar la definición del *modelo base WF* y genera un nuevo modelo, el *modelo*

base workflow desconectado (modelo base WFD) el cual es la definición de un sistema de workflow desconectado. Antes de explicar la forma en la que este mecanismo fue implementado, es necesario establecer las características del *modelo base WFD*.

V.3.2 *Modelo base de workflow (modelo base WF)*

Como se ha mencionado en los capítulos previos, la arquitectura de *SysCoor* utiliza modelos de procesos descritos en el lenguaje XML para definir un sistema de workflow funcional. De acuerdo con el meta-modelo propuesto por la WfMC y descrito en el capítulo II [WfMC, 1995], los elementos necesarios para la definición de un sistema de workflow son: los procesos involucrados, las actividades dentro de los procesos, el flujo de control, los contenedores de entrada, el flujo de datos, las condiciones y los datos relevantes utilizados durante la ejecución de los procesos.

Para cumplir con los requerimientos propuestos por el meta-modelo, en la arquitectura de *SysCoor* se ha realizado la representación de cada uno de estos aspectos a través de la definición de diferentes modelos base. A través del *modelo base RAD* [Flores Ríos, 2001] y la representación a través de estados de las actividades del *modelo base RAD* por medio del *modelo base de transición de estados (modelo base STD)* [Ramírez Fernández, 2002] se realiza la representación de los procesos, las actividades que los componen, el flujo de control, las condiciones asociadas y el flujo de datos. En este mismo sentido se realiza la descripción de los datos relevantes por medio del *modelo base de entidades de información (modelo base EI)* [Ramírez Fernández, 2002] el cual describe la localización en un contenedor, de los datos utilizados durante la ejecución del workflow.

Por medio de la unión de los aspectos descritos en cada uno de los modelos mencionados, Ramírez Fernández y García Carrillo llegaron a una representación completa de un sistema de workflow en XML, a través de un solo modelo conocido como el *modelo base de workflow* o *modelo base WF*. En este mismo sentido, definieron un DTD el cual establece las reglas sintácticas y léxicas que se deben cumplir para que esta definición sea

correcta y pueda ser transformada por la arquitectura de *SysCoor*, en un sistema de workflow funcional. Este DTD se presenta en el apéndice B. Por otro lado, Ramírez Fernández implementó una serie de páginas HTML conocidas como *plantillas* las cuales representan escenarios típicos de workflow para servir como pantallas de usuario (GUI) durante la ejecución del sistema de workflow.

En la tabla III se presenta la relación de algunas de los elementos y sus atributos definidos en el DTD, las cuales son fundamentales para la definición del sistema.

Tabla III. Descripción de algunos de los elementos y sus atributos del DTD del *modelo base WF* utilizado para la definición de un sistema de workflow

<i>Elemento</i>	<i>Utilización</i>
<roles>.....</roles>	Esta etiqueta es utilizada para iniciar con la definición del sistema completo de workflow
<rol_coordinacion>....</rol_coordinacion>	Permite agrupar por roles la definición del sistema en los roles que lo forman
<nombre>....</nombre>	Utilizada para establecer el nombre del rol
<agente>....</agente>	Define el nombre del agente por cada rol.
	Atributos
	<i>quien</i> Permite asignar un identificador único en toda la definición del sistema para el agente
<estado>....</estado>	Utilizada para agrupar una serie de actividades que representan un estado en la ejecución del workflow
	Atributos
	<i>id</i> Identificador único del estado dentro de la definición de un rol
	<i>nom</i> Nombre del estado

Continuación de la Tabla III

<i>Elemento</i>	<i>Utilización</i>	
<act>...</act>	Se utiliza para definir una actividad	
	Atributos	
	<i>Id</i>	Identificador único de la actividad en toda la definición del sistema
	<i>nom</i>	Nombre de la actividad
	<i>plantilla</i>	Referencia a la página HTML que sirve como pantalla de usuario para la ejecución de la actividad
	<i>E_clave_info</i>	Para definir el nombre de una entidad de información si ésta se requiere como entrada para iniciar el estado
	<i>S_clave_info</i>	Para definir el nombre de una entidad de información si ésta se emite como salida del estado
	<i>Tipo</i>	Tipo de la entidad de información que se recibe o emite. Los tipos son "s" para cadenas de caracteres e "i" para enteros
<i>cual</i>	Identificador (id) del rol que recibe o emite la entidad de información especificada con <i>E_clave_info</i> o <i>S_clave_info</i>	
<interaccion>....</interaccion>	Permite definir el rol con el que se interactúa	
	Atributos	
	<i>cual</i>	Identificador de la actividad con la cual existe interacción
	<i>tipo</i>	R= Receptor E= Emisor
<transición>...</transición>	Definir la transición de un estado a otro	
	Atributos	
	<i>id_sig_edo</i>	Definir el estado siguiente

Este conjunto de etiquetas y elementos permite definir las actividades, las transiciones de estados, las entidades de información y las plantillas (páginas HTML) utilizadas por cada actividad para permitir la ejecución del proceso.

En la figura 19 se presenta un ejemplo que muestra parte de la definición de un *modelo base WF*. En la imagen se describen algunos de los elementos básicos y sus atributos.

validar que el documento cumple con las reglas léxicas y sintácticas establecidas

```
<?xml version='1.0' encoding="ISO-8859-1"?>
<!DOCTYPE estados SYSTEM "workflow.dtd">
<roles>
  <rol_coordinacion>
    <nombre>Realizando Venta</nombre>
    <agente quien="a">Vendedor</agente>
    <estado id="1" nom="Recibir solicitud">
      <act id="a1" nom="Recibir solicitud" E_clave_info="visitas" Tipo="s" cual="b1">
        <evento act_id="a1" nom_act="Recibir solicitud">
          <interaccion cual="b1" tipo="R">b</interaccion>
          <transicion id_sig_edo="2"></transicion>
        </evento>
      </act>
    </estado>
    .
    .
    <estado id="9" nom="Tomar inventario directo de la bodega del cliente" >
      <act id="a2" nom="Tomar inventario directo de la bodega del cliente" plantilla="4">
        <evento act_id="a2" nom_act="Tomar inventario directo de la bodega del cliente">
          <transicion id_sig_edo="10"></transicion>
        </evento>
      </act>
      <act>
        .
        .
      </act>
    </estado>
    .
    .
  </rol_coordinacion>
</roles>
```

Nombre que identifica al rol que se está definiendo

Agente que ejecuta el rol. La etiqueta *quien* permite asignar un identificador único al agente a lo largo de toda la definición del sistema

Los estados agrupan un conjunto de actividades. Para un estado se define una GUI o plantilla utilizada, la cual es usada por todas las actividades que forman el estado

Una vez terminada la actividad, esta etiqueta indica el estado siguiente

Identificador único del estado dentro de la definición de un rol

Nombre del estado

La definición del sistema de workflow se compone de una serie de estados, los cuales a su vez encapsulan las actividades que forman el proceso

La definición de una actividad es la unidad funcional del sistema de workflow. Su definición incluye un identificador, las interacciones y el evento que causa la transición a otro estado

Identificador de la actividad. Este es único dentro de la definición del sistema completo

Nombre o descripción de la actividad

Figura 19. Ejemplo de una parte de un *modelo base WF* que incluye la definición de dos estados del proceso "Realizando venta" del caso de estudio realizado en la compañía Chateau Camou

Para ilustrar la forma en que se definen las interacciones entre roles de un proceso, así como la descripción del intercambio de entidades de información, la figura 20 muestra la definición del estado “*Recibir solicitud*”, el cual se requiere para iniciar la actividad *a1*, de una entidad de información, en este caso *visitas* la cual es recibida de la actividad *b1*.

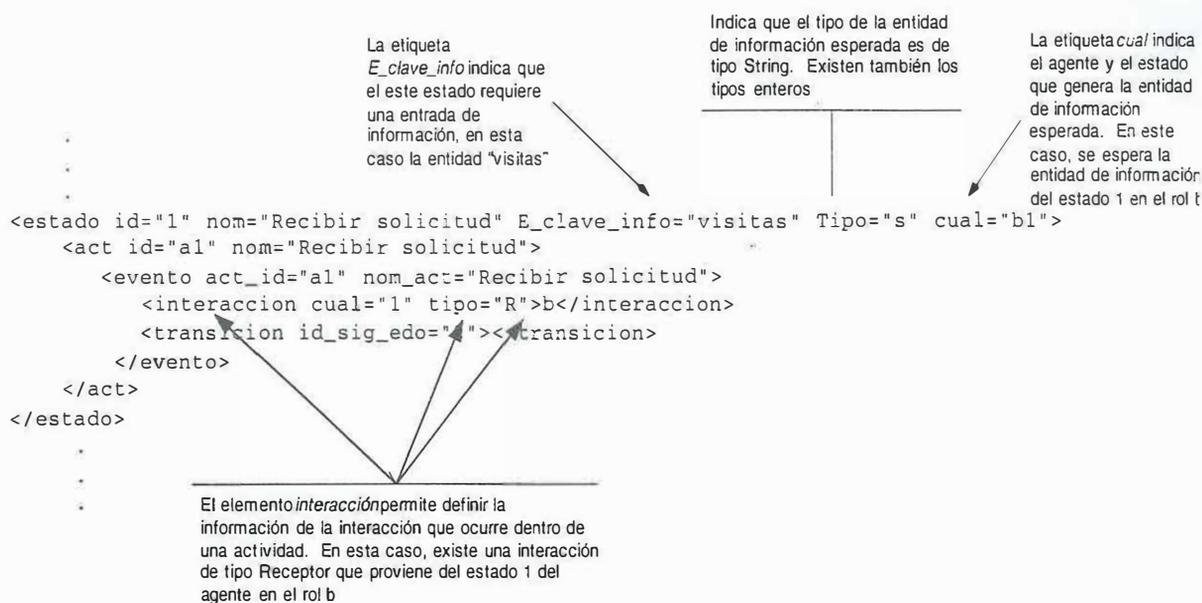


Figura 20. Un estado del proceso “*Realizando venta*” el cual muestra una actividad que interactúa con otra del mismo proceso y recibe información del segundo

De igual forma para ilustrar la forma en como se define un envío de entidades de información, la figura 21 muestra la definición del estado “*Finalizar*”, en el que la actividad *a7*, envía una entidad de información, definida como *pedido*, a la actividad *b3*.

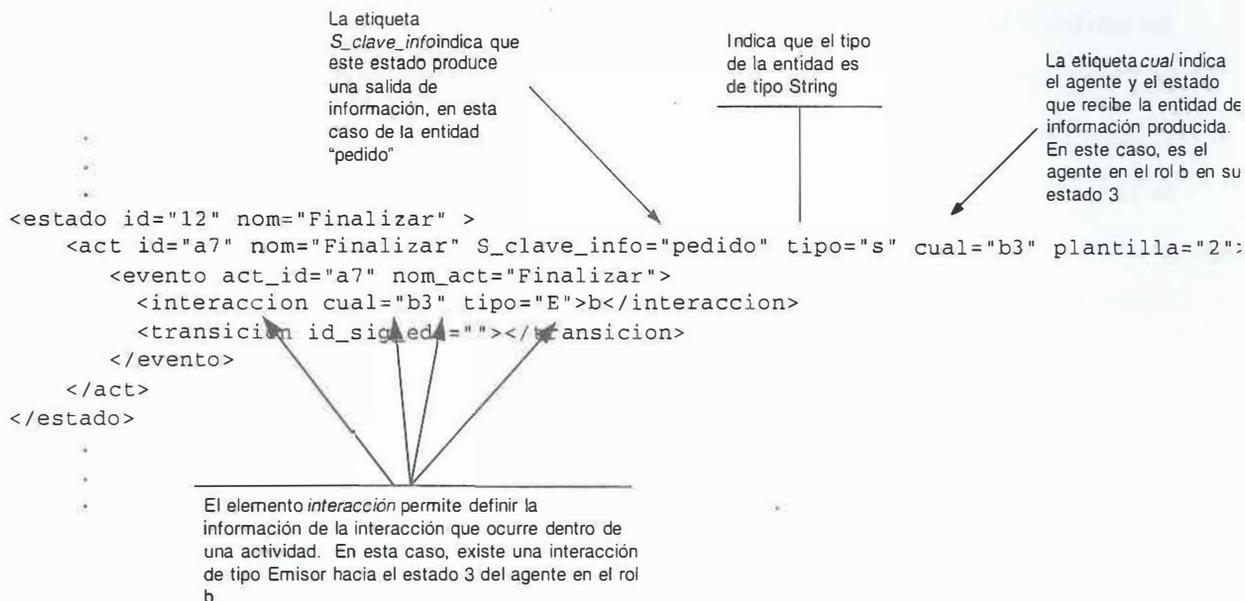


Figura 21. El estado "Finalizar" del proceso "Realizando venta" el cual muestra una actividad que envía una entidad de información a otra actividad en otro proceso

Por último, la figura 22 muestra como se define el uso de una plantilla o pantalla de usuario (GUI) para una actividad.

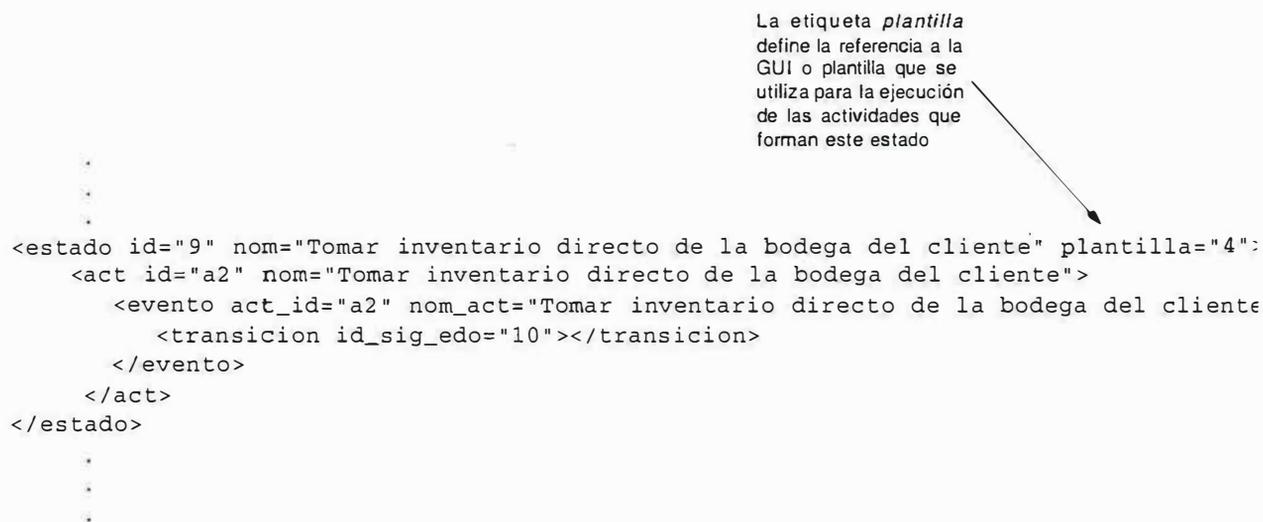


Figura 22. Definición de la actividad *a2* en la que se implementa la plantilla 4 para servir como pantalla con el usuario que ejecuta la actividad

El trabajo realizado por García Carrillo y Ramírez Fernández permite definir un sistema de workflow y transformar esta definición a un sistema ejecutable para ser accesado a través de un navegador Web. La definición realizada por medio del *modelo base WF* es una definición completa que cumple con los requisitos establecidos por el meta-modelo de la WfMC. Sin embargo, esta definición fue establecida bajo la consideración de que los datos y la interfaz gráfica de usuario se encuentran siempre disponibles en un contenedor central con el que existe una conexión constante.

Para permitir que un cliente trabaje desconectado de una máquina central de workflow, necesitamos proveer al cliente que se desconecta con la lógica del sistema, con los datos utilizados y con las pantallas, permitiendo de esta forma que el sistema se ejecute sin necesidad de consultar el contenedor central de datos.

Con la finalidad de completar la definición para un sistema de workflow que trabaja desconectado de una máquina central de workflow, en este trabajo se introduce la definición en XML de los datos y de GUIs que son utilizados por el workflow transferido. De igual forma, las diferencias entre GUIs de computadoras de escritorio (plataforma original de *SysCoor*) y dispositivos PDAs hizo necesario redefinir las plantillas (páginas HTML) utilizadas, a nuevas GUIs que puedan ser mostradas en esta clase de dispositivos. Con la definición de estos dos aspectos, y reutilizando la información creada en el *modelo base WF* llegamos a la representación de un sistema de workflow que puede ser ejecutado en un dispositivo PDA sin necesidad de consultar la máquina central de datos. Este modelo fue definido como el *modelo base de workflow desconectado (modelo base WF)*.

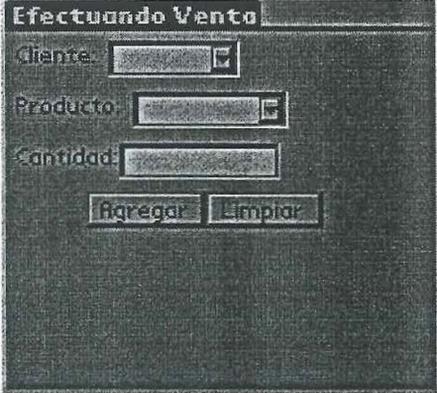
Para solucionar el problema de las interfaces gráficas introducimos el *modelo base GUI* el cual permite definir GUIs con elementos de interfaz gráfica que pueden ser mostrados en dispositivos PDA, y con el fin establecer la definición de los datos planteamos un conjunto de reglas sintácticas y léxicas a través del DTD del *modelo base WF*, las cuales permiten definir el contenido de los datos utilizados durante la ejecución del workflow. A continuación se describe la implementación del *modelo base GUI*.

V.3.3 Modelo base GUI para la definición de Interfaces Gráficas del Usuario

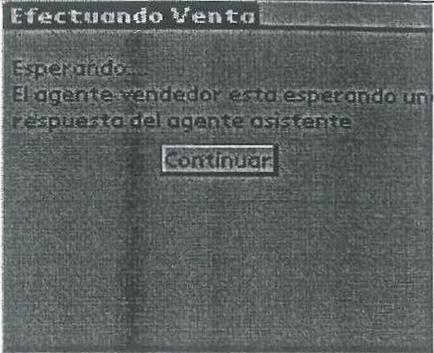
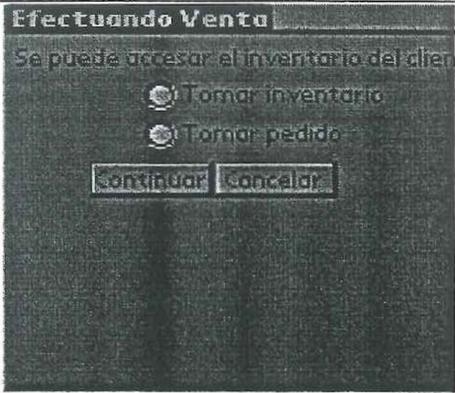
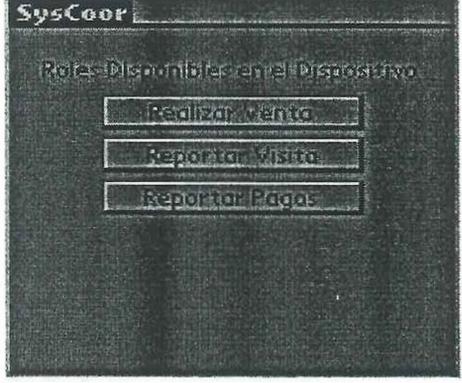
Para proveer a los usuarios de pantallas para interactuar con los sistemas de workflow generados en *SysCoor*, la arquitectura hace uso de una serie de pantallas de usuario (interfaces GUI) descritas por medio del lenguaje de marcado HTML (Hypertext Markup Language) y mostradas como páginas Web, las cuales representan los escenarios típicos de workflow. Estos escenarios son: capturar datos, esperar una respuesta, elegir entre varias actividades así como editar y enviar un documento.

Sin embargo, debido a las limitaciones de despliegue gráfico que aún muchos dispositivos PDA presentan, en este trabajo hemos redefinido las páginas HTML de *SysCoor* en interfaces GUI tradicionales con elementos de interfaz básicos a cualquier lenguaje de programación para PDAs. En la tabla IV se presentan las GUIs que se utilizan en los sistemas de workflow para Web en *SysCoor* y la forma en que pueden ser implementadas para la ejecución de workflow desconectado en dispositivos PDAs.

Tabla IV. Ejemplo de GUIs en *SysCoor* para el Web y para *SysCoor* en PDAs

Ejemplo de GUIs en <i>SysCoor</i>	GUIs en <i>SysCoor</i> para PDAs
 <p>The screenshot shows a web browser window with a title bar. The main content area displays the following text and form elements:</p> <ul style="list-style-type: none"> SubProceso UMF 8 Cristina en el rol del asistente médico En el estado ordenamiento Escuche la orden de formas para un cambio de jefe de sustancia Form fields: <ul style="list-style-type: none"> Nombre de la forma: <input type="text"/> Cantidad: <input type="text"/> Buttons: <input type="button" value="Enviar"/> <input type="button" value="Limpiar"/> 	 <p>The screenshot shows a PDA screen with a dark background and white text. The title is 'Efectuando Venta'. The form elements are:</p> <ul style="list-style-type: none"> Cliente: <input type="text"/> Producto: <input type="text"/> Cantidad: <input type="text"/> Buttons: <input type="button" value="Agregar"/> <input type="button" value="Limpiar"/>

Continuación Tabla IV

Ejemplo de plantilla	Plantilla en SysCooor para PDAs
	
	
	

Estas GUIs han sido definidas a través de un conjunto de etiquetas XML y validadas usando el DTD que permite crear el *modelo base GUI*. En la figura 23 se presenta una parte de este DTD en la que se muestra el elemento principal que permite definir un elemento de una GUI, el resto del documento se presenta en el Apéndice B.

En la tabla VI se presentan estos elementos básicos con su respectiva representación en XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Definiendo los elementos de una interfaz grafica -->
<!ELEMENT elementogui EMPTY>
<!ATTLIST elementogui
    id_gui CDATA #REQUIRED
    tipo (Label|Radio|InputBox|ItemButton|ComboBox) #REQUIRED
    texto CDATA #IMPLIED
    x CDATA #REQUIRED
    y CDATA #REQUIRED
    valor CDATA #IMPLIED>
<!ELEMENT act (elementogui+)>
.
.
.
```

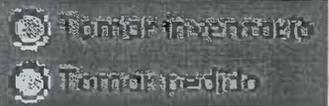
Figura 23. DTD para la creación de GUIs con el modelo base GUI

La descripción del elemento *elementogui* y la de sus atributos se presenta en la tabla V.

Tabla V. Descripción del elemento *<elementogui>* y sus atributos

<i>Elemento</i>	<i>Utilización</i>
<i><elementogui>....</i> <i></elementogui ></i>	Define un elemento de interfaz gráfica de usuario
	Atributos
<i>id_gui</i>	Permite asignar un identificador único en la definición del rol
<i>tipo</i>	Permite definir un tipo de elemento GUI. Los tipos válidos se muestran en la tabla VI
<i>texto</i>	Texto que acompaña al tipo de elemento que se define
<i>x</i>	Coordenada del elemento en X. El rango va desde 1 hasta 60
<i>y</i>	Coordenada del elemento en Y. El rango va desde 1 hasta 45
<i>valor</i>	Si el elemento muestra un valor inicial (dato), esta etiqueta permite definirlo

Tabla VI. Elementos GUI básicos y su representación en XML.

Nombre	Elemento GUI	Representación en XML
Button		<elementogui tipo="ItemButton"/>
Label		<elementogui tipo="Label"/>
Combo		<elementogui tipo="ComboBox"/>
InputBox		<elementogui tipo="InputBox"/>
Radio		<elementogui tipo="Radio"/>

Para ilustrar la forma en la cual las pantallas son definidas, en la figura 24 se presenta una GUI generada a partir de la definición en XML.

```
<act id_act="1">
```

```
<elementogui id_gui = "a12" tipo="ComboBox"
  texto="Cliente" x="2" y="5"
  valor = "Cliente 1,Cliente 2,Cliente 3,Cliente 4,
  Cliente,Cliente 6"/>
```

```
<elementogui id_gui = "a13" tipo="ComboBox"
  texto="Producto" x="2" y="15" valor= "Producto
  1,Producto 2,Producto 3,Producto 4,Producto 5,Producto 6"/>
```

```
<elementogui id_gui = "a14" tipo="InputBox"
  texto="Cantidad" x="2" y="25" valor="" />
```

```
<elementogui id_gui = "a15" tipo="ItemButton"
  texto="Agregar" x="10" y="35" />
```

```
<elementogui id_gui = "a16" tipo="ItemButton"
  texto="Limpiar" x="20" y="35" />
```

```
</act>
```

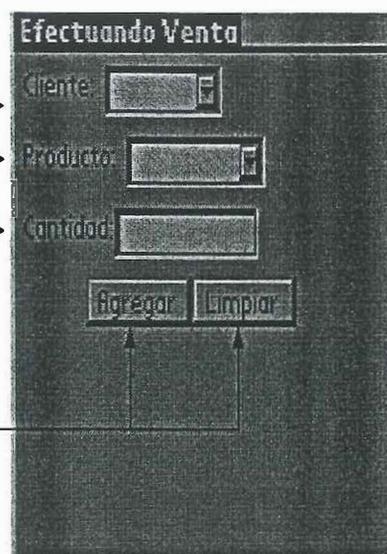


Figura 24. Ejemplo de un *modelo base GUI XML* que define una pantalla en XML

Por medio del uso del *modelo base GUI*, un modelador de sistemas de workflow puede definir diferentes pantallas para las actividades de acuerdo a necesidades específicas de estas. La definición de estas pantallas se puede realizar utilizando cualquier editor de texto, y será validada por medio de su DTD cuando sea utilizada para completar la definición del sistema de workflow desconectado. Este definición se realiza previa a la generación del *modelo base WFD*. Hemos propuesto una solución al problema de las GUIs, ahora presentamos la

V.3.4 Definición de datos por medio de XML

Para establecer la definición de los datos relevantes del workflow, el *modelo base WF* define la estructura de los datos y/o la referencia a la localidad donde éstos se encuentran pero no su contenido (figura 20), considerando que los datos se encuentran almacenados en un contenedor central y supone que el usuario tendrá una conexión constante a este contenedor del cual podrá recuperar estos datos.

Para proveer a un cliente con la definición de los datos relevantes a sus tareas, hemos propuesto la definición de éstos a través del lenguaje XML. Por medio de la definición del DTD del *modelo base WFD* se pueden describir los datos que existen en un contenedor de datos. Este DTD es utilizado por el *mecanismo de transferencia de documentos XML*, el cual agrega la definición de datos a los roles que se desconectan. La definición de datos es retomada cuando se describe la implementación del *mecanismo de transferencia de documentos XML*, por lo que dejamos para esa sección la descripción de las etiquetas que permiten definir los datos.

V.3.5 El modelo base de workflow desconectado (*modelo base WFD*)

Con información obtenida a partir de los *modelos bases WF* y *GUI*, y la representación de datos por medio de XML podemos obtener una definición completa y

suficiente de un sistema de workflow para ser ejecutado en un dispositivo PDA. Este modelo es conocido como el *modelo base de workflow desconectado (modelo base WFD)*. El modelo cuenta con la información necesaria sobre datos, pantallas de usuario, condiciones y descripción de actividades. Este modelo será analizado después de describir la implementación del *transformador de código* ya que es este último quien lo genera.

V.3.6 Implementación del transformador de código

El transformador de código crea a partir del *modelo base WF* el *modelo base WFD*, además, crea dos estructuras de datos con la definición de los roles y sus interacciones, y de los roles y las entidades de información utilizadas.

El *transformador de código* fue implementado dentro del *generador de código*, debido a que el primero necesita de la definición del *modelo base WF* y es en el generador en donde se realiza la definición de éste. El *generador de código* está implementado por medio de un *JApplet* usando el ambiente gráfico *Swing* de Java 2. Un *JApplet* es un programa escrito en java el cual es incluido en una página HTML y su código es interpretado en un visualizador de Web.

Para poder analizar el *modelo base WF* en el servidor se utilizó el conjunto de clases del API (*Application Program Interface*) SAX versión 1.0 para XML de Java.. Este API contiene una serie de clases que permiten analizar y validar cualquier modelo descrito en XML de acuerdo a las reglas definidas en un DTD correspondiente e inclusive sin éste.

El *transformador de código* se agregó como una clase dentro del *generador de código* con el nombre *transformadorCodigo*. En esta se hizo necesario implementar los métodos del analizador SAX para poder leer y analizar el *modelo base WF*. Durante este análisis se buscan las etiquetas necesarias para generar el *modelo base WFD*. Las actividades que se realizan durante este análisis son:

- Cuando se encuentra la etiqueta "*plantilla*", se toma el valor de esta etiqueta y se busca en el contenedor de modelos y datos (directorio tomcat4\webapps\coordinación\serializados) el archivo con el mismo nombre del valor de la etiqueta agregando la extensión ".xml". Por ejemplo, si el valor de la plantilla en el *modelo base WF* es gui1, se busca el archivo gui1.xml.
- Se lee el contenido del archivo y se envía a una segunda clase llamada *GUIDocHandler* la cual valida la GUI utilizando el *modelo base GUI*
- Una vez validada la GUI se agrega su definición a la cadena que se ha ido formando con el *modelo base WF*.

La figura 25 muestra un fragmento de un *modelo base WF* (a) y el fragmento resultante del *modelo base WFD* (b).

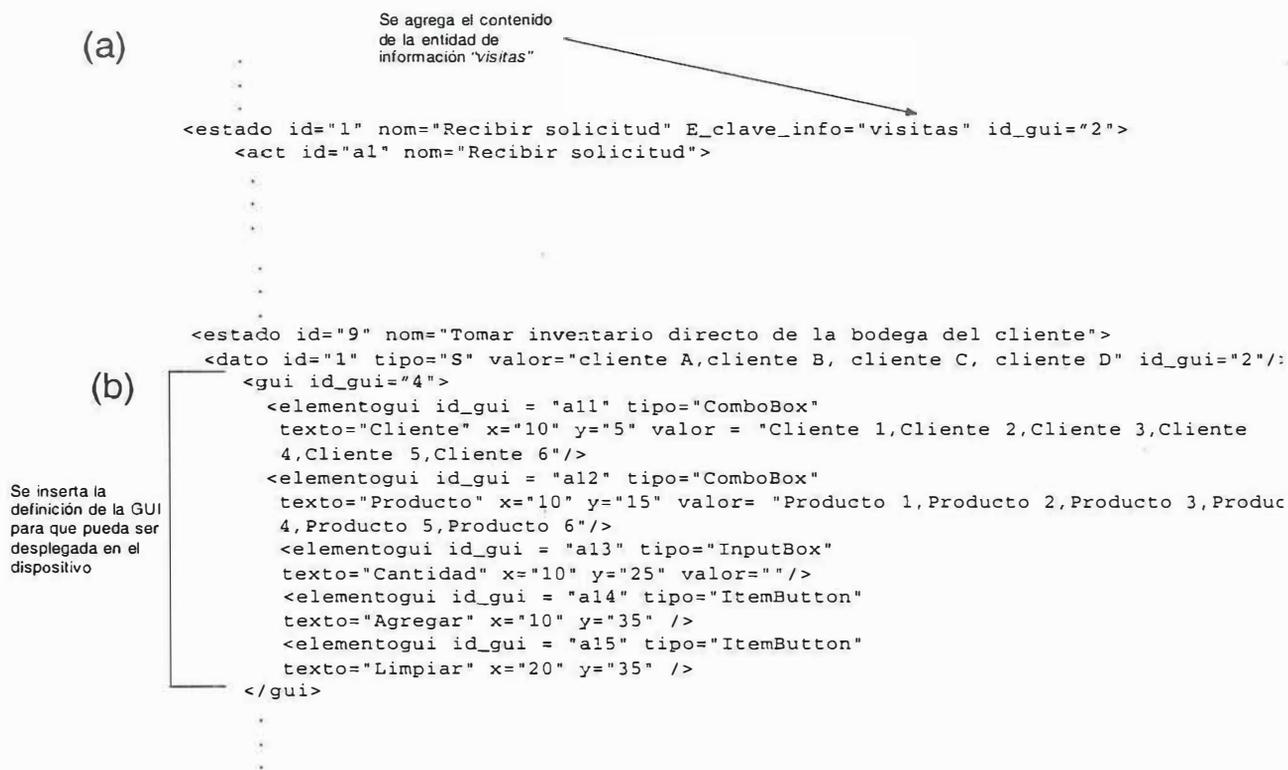


Figura 25. Parte de un *modelo base WF* el cual define la referencia a una plantilla (GUI) (a) y el mismo fragmento del modelo una vez que se agregó la definición de la pantalla dando como resultado *modelo base WFD* (b)

Al mismo tiempo que se va formando el *modelo base WFD*, se van almacenando en un vector (estructura de datos) los valores de los elementos *quien*, *id*, *E_clave_info*, *S_clave_info* dentro de la etiqueta <act>, *cual* y *tipo* en la etiqueta <interaccion>, ya que estos elementos permiten conocer el rol, interacciones y entidades de información. Una vez que el proceso de análisis termina, el *modelo base WFD* está listo para ser almacenado en el contenedor de datos y modelos y los datos en el vector se pueden pasar a las tablas *rolEI*, la cual almacena las entidades de información utilizadas, y a la tabla *rolAct*, la que almacena las actividades de cada rol y la relación con otros roles.

Para realizar las funciones de guardar el modelo y los datos se implementa un Servlet, ya que por cuestiones de seguridad a través del *Japplet* no es posible modificar los recursos del servidor mientras que los *Servlets* sí permiten esta funcionalidad. Este Servlet se conoce como *transformador*.

Un objeto de la clase *transformadorCodigo* hace una solicitud HTTP al Servlet *transformador*. En esta llamada se le envía al Servlet una cadena de texto, la cual es la definición completa del *modelo base WFD*. El Servlet *Transformador* recibe la cadena con el contenido del archivo y a través del método *grabarArchivo()* se crea el archivo en el contenedor de datos y modelos.

Este último objeto hace una segunda petición al Servlet en la que envía la información que se almacenó en el vector durante el análisis sintáctico y léxico. El Servlet *transformador* recibe la descripción de los datos para almacenar en la base de datos; por medio del método *guardarInformacion()* abre la conexión a la base de datos, y ejecuta una serie de comandos en el lenguaje SQL (queries) para grabar los datos recibidos en las tablas *rolEI* y *rolAct*.

V.3.7 Modificaciones al AGP

La pantalla principal de la máquina de workflow (*SysCoor*) se maneja a través del AGP. El AGP se encuentra implementado en el lenguaje Java 2 y se utilizan Páginas de Servidor de Java (JSP, *Java Server Pages*) las cuales son una combinación de código HTML y código Java que en conjunto funcionan como un *Servlet*.

Se realizaron tres modificaciones principales al AGP. La primera fue la construcción de un área de mensajes en la pantalla principal del usuario, para desplegar información que se produce durante el proceso de desconexión y reconexión de un rol. La segunda modificación fue agregar un mecanismo, en este caso un botón, para permitir desconectar un rol. Este mecanismo hace la llamada al Servlet *ModuloOperacionDesconectada* el cual inicia las llamadas a los Servlets que forman este módulo. La figura 26 muestra la interfaz que el AGP presenta con los cambios realizados.

The screenshot shows a web browser window displaying the main interface of the SysCoor system. The page title is "Grupo de Ingeniería de Procesos CICESE". The interface features a table with the following data:

Rol	Agente	Estado Actual	Conectado	
Realizando Venta	Vendedor	1 Recibe solicitud	Si	Desconecta
Solicitando Mercancia	Cliente	1 Enviar solicitud	No	

Callouts in the image provide the following descriptions:

- Top-left callout:** Roles participantes en el workflow mostrando el agente participante, el estado en el que se encuentra el proceso y si el usuario se encuentra conectado al servidor central o no.
- Bottom-left callout:** Área de mensajes por medio de la cual el AGP comunica al usuario sobre diferentes eventos.
- Bottom-center callout:** Botón para desconectar el rol. Al presionar el botón el AGP inicia todo el proceso para desconectar el rol.
- Bottom-right callout:** Roles participando en el workflow. A través de estas ligas el agente accesa a la actividad actual en la que se encuentra el proceso.

Figura 26. Pantalla principal de *SysCoor* en la que se muestran las principales secciones que la forman

El tercer cambio consistió en introducir la clase *instanciaRol()* implementada por el AGP. Esta clase permite generar un identificador único por cada vez que el usuario inicia

un rol, es decir, cada que inicia un instancia de algún rol. Cuando un usuario elige un rol para ejecutar, la clase *instanciaRol()* crea un identificador único para ese usuario y ese rol. Tomando la definición de actividades y entidades de información que forman el rol (tablas *rolEI* y *rolAct*), se crea en la tabla *estadosAct* un nuevo registro que permite mantener el estado de las actividades de esa instancia del rol, de igual forma, en la tabla *estadosEI* se agrega un registro para llevar el estado de las entidades de información utilizadas en esa instancia específica del rol. Esto permite mantener una mayor persistencia de los estados tanto de actividades como roles.

Lo anterior fue implementado solo para probar el prototipo de la parte de desconexión de los roles, es decir en el caso de los usuarios que ejecutan los sistemas de workflow para Web, el control de estados de actividades y entidades de información sigue manteniéndose a través de archivos serializados. Para probar nuestro prototipo se modificó el *AGP* de forma que los estados de ambos elementos que son mostrados en la interfaz de despliegue (figura 26) son consultados de la base de datos y no de los archivos serializados.

Los cambios en el generador de *código* y el *AGP* forman parte de las modificaciones a los elementos ya existentes en la arquitectura para apoyar la operación desconectada. A continuación se presenta la implementación de los mecanismos que fueron definidos como extensiones a la arquitectura.

V.3.8 Mecanismo de determinación de desconexión

Este mecanismo fue implementado por medio del Servlet *determinacion*. Este Servlet recibe como entrada el parámetro *instancia*, el cual es el identificador de la instancia del proceso que se está ejecutando. Utilizando este parámetro, el Servlet lleva a cabo las consultas a las tablas *estadoAct* y *estadoEI* definida en la base de datos. La tabla *estadoEI* almacena el estado en el que se encuentran las entidades de información que utiliza la instancia del rol que se requiere desconectar. En la tabla *estadoAct* revisa que las actividades del rol que se requiere desconectar estén disponibles. En las tablas VII y VIII

respectivamente, se muestran los estados que pueden presentar las entidades de información y las actividades en las tablas anteriores.

Tabla VII. Estados de las entidades de información y su descripción

Estado	Descripción
0	La entidad de información no puede ser utilizada.
1	La entidad de información puede ser utilizada por el rol que se desconecta para realizar operaciones de lectura y escritura sin ningún problema

Tabla VIII. Estados de las actividades y su descripción

Estado	Descripción
0	La actividad ha sido iniciada pero aun no ha terminado su ejecución.
1	La actividad está disponible

Para llevar a cabo la evaluación de la desconexión de un rol, el Servlet *determinación* lleva a cabo consultas a las tablas antes mencionadas y dependiendo del resultado obtenido toma una decisión.

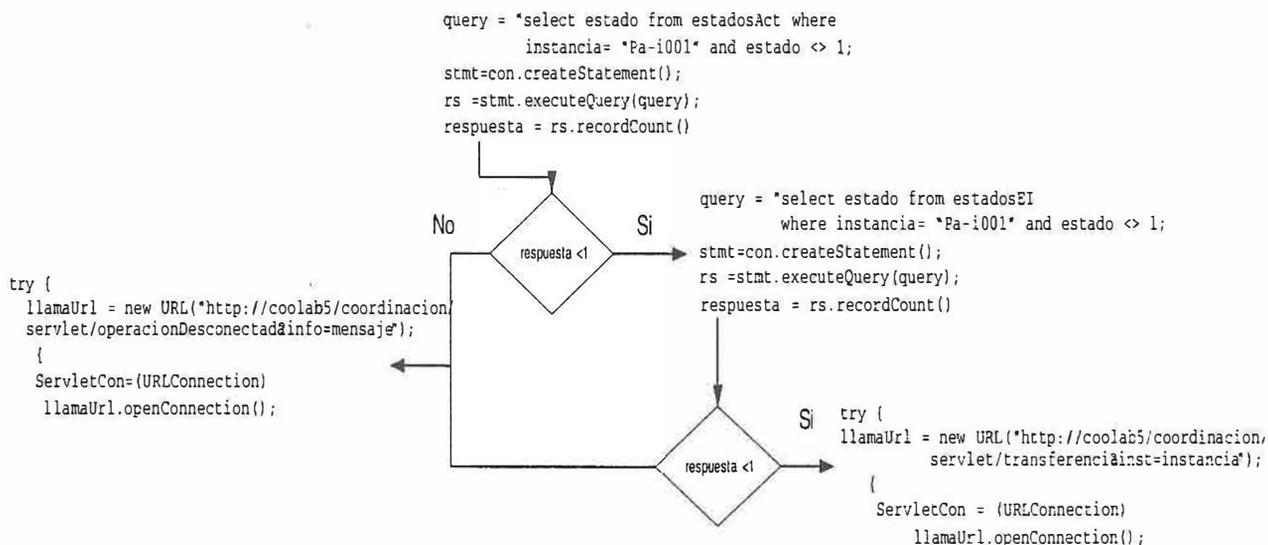


Figura 27. Flujo de actividades del Servlet *determinación* en el que se muestran las consultas y llamadas a Servlets que ejecuta

La figura 27 muestra las consultas que se llevan a cabo para conocer el estado tanto de actividades como de entidades de información. Las consultas solicitan información sobre las actividades y /o entidades de información usadas por una instancia específica del rol que *no* estén disponible. Si alguna de estas consultas regresa por lo menos un registro (*rs.recordCount*) indica que por lo menos una actividad y/o entidad no está disponible. En ese caso se envía el mensaje al Servlet *operacionDesconectada* indicando que la desconexión no es posible en ese momento. En caso contrario, se inicia el mecanismo de *transferencia* por medio de la llamada al Servlet *transferencia* pasándole el parámetro de la instancia del rol que se desconecta.

V.3.9 Mecanismo de transferencia de documentos XML

Una vez que ha sido determinado que la desconexión es posible, el *mecanismo de transferencia de documentos XML* inicial la fase conocida como “*recolección de datos*”. En esta fase, este mecanismo prepara el *modelo base WFD* para ser transferido. Con este fin, el mecanismo recupera el *modelo base WFD*, obtiene la definición completa solo del rol o roles que se desconectan, recupera del contenedor de modelos y datos la definición de los datos que utilizan los roles, basándose en las reglas descritas en el modelo base de *definición de datos* y los agrega a la definición de los roles. De esta forma, deja preparado el *modelo base WFD* con la definición completa para que el sistema trabaje de forma independiente de un servidor central. En seguida, espera por un cliente PDA para establecer la comunicación con el dispositivo y enviar la definición de los roles (*modelo base WFD*).

Este mecanismo fue implementado por medio del Servlet *transferencia*. Se compone de cinco métodos principales: *prepararRol()*, *guardarArchivo()*, *notificar()*, *enviarDatosHTTP()* y *recibirDatosHTTP()*.

El método *prepararRol()* se utiliza cuando un rol solicita desconexión. Este método utiliza la clase *WFDDocHandler* la cual implementa el analizar sintáctico y léxico para el *modelo base WFD*. El método *analiza()* en la clase *WFDDocHandler* recibe el identificador del rol o roles que se necesitan desconectar y el identificador de la instancia del rol. En la figura 28 se muestra parte de las operaciones que lleva a cabo este método.

(1)

El analizador léxico y sintáctico en el mecanismo de transferencia de documentos XML localiza la referencia a una entidad de información

```
<estado id="1" nom="Recibir solicitud" E_clave_info="visitas">
  <act id="a1" nom="Recibir solicitud">
```

(2)

```
query = "select c.id_ei, d.contenido from entidades as c,entidades_informacion as d
        where c.id_ei = a.id_ei and a.instancia_proceso = " + instancia;
stmt=con.createStatement();
rs =stmt.executeQuery(query);
while(rs.next()) {
  datos=rs.getString(2);
}
```

El metodo analiza() ejecuta una consulta a las tablas en la base de datos para recuperar el contenido de la entidad de información, utilizando el nombre de la entidad y el identificador único del sistema que se está ejecutando

id_ei	nombre	tipo
1	001	pedidos
2	002	visitas
3	003	clientes
4	004	productos
5	005	estado_cuenta

El nombre de la entidad de información. Esta es el nombre por medio del cual la entidad es conocida por la definición del sistema

Identificador del rol

id_ei	contenido	id_rol	instanc
001	"producto1,2,producto2,2,"producto3,1"	a	Pa-i001
001	"producto5,1,producto2,1,"producto3,1"	a	Pa-i002
001	"producto1,2,producto2,2,"producto3,1"	a	Pa-i003
002	"Cliente 1,Cliente 2,Cliente 3,Cliente4,Cliente 5,Cliente 6"	a	Pa-i001
002	"Cliente 8"	a	Pa-i002

Identificador único de instancia del rol que se está ejecutando

Identificador de la entidad de información

Contenido de la entidad de información

(3)

```
datos = "cliente 1,cliente 2, cliente 3, cliente 4, Cliente 5"
```

El contenido es regresado en la variable *datos* cuyo contenido se agrega a la definición de la actividad

(4)

Se agrega el contenido de la entidad de información "visitas"

```
<estado id="1" nom="Recibir solicitud" E_clave_info="visitas">
  <dato id="1" tipo="S" valor="cliente 1,cliente 2, cliente 3,
  cliente 4, Cliente 5" id_gui="2"/>
  <act id="a1" nom="Recibir solicitud">
```

Figura 28. Algunas de las operaciones que lleva a cabo el método *analiza()* en el orden en el que ocurren

Este método prepara dos cadenas de texto, una con la definición del rol que se desconectará incluyendo los datos usados por éste, y otra más con los identificadores (id) de las actividades y las entidades de información que se desconectarán. Una vez que termina de analizar el documento, regresa ambas cadenas de texto al método *prepararRol()* en el Servlet.

Con el método *guardarArchivo()* el Servlet *transferencia* guarda el contenido de la primer cadena de texto en un archivo en el contenedor de datos, el cual es accesado más tarde para ser enviado al dispositivo. La segunda cadena es utilizada por el método *notificar()* el cual llama al Servlet del mecanismo de sincronización.

Para la transferencia del *modelo base WFD* al dispositivo se implementó el envío de una cadena de texto que contiene la definición del modelo, a través de una conexión HTTP establecida por el método *enviarDatosHTTP()*. En este caso, el *coordinador local* en el dispositivo PDA realiza una conexión HTTP llamando al Servlet *transferencia* el cual responde con la cadena de texto que contiene el *modelo base WFD*.

De acuerdo a la documentación del lenguaje *SuperWaba* en el que fue desarrollada la arquitectura del cliente, es posible abrir la conexión HTTP por medio de un puerto serial, sin embargo, este aspecto no fue probado por no contar con la interfaz (cable serial) para realizar las pruebas. De aquí que la transferencia de la definición del *modelo base WFD* fue solo probada utilizando el emulador de un dispositivo PDA con el sistema operativo PalmOS.

El método *recibirDatos()* es utilizado cuando un dispositivo solicita la reconexión. En este caso, nuevamente el *coordinador local* en el dispositivo llama al Servlet *transferencia* enviando la cadena de texto que incluye las actividades y las entidades de información que se reconectan (*definidas con el modelo base de reconexión*). El método *recibirDatos()* recibe esta cadena y la envía por medio del método *notificar()* al Servlet *sincronizacion*.

V.3.10 Mecanismo de sincronización

El *mecanismo de transferencia de documentos XML* comunica al *mecanismo de sincronización* sobre los roles y datos que están listos para ser transferidos. De esta forma, el *mecanismo de sincronización* actualiza los estados de las entidades de información en el contenedor, así como el estado del rol o roles.

Este mecanismo es implementado por medio del Servlet *sincronizacion*. Este Servlet recibe del Servlet *transferencia* dos parámetros: la instancia y una cadena con el listado de identificadores de actividades y entidades de información que se desconectan o bien que se reciben del dispositivo. Las actividades que realiza son:

- a) Primero separa la cadena de actividades y entidades de información.
- b) Por medio de la instancia del rol, abre la base de datos y actualiza el estado actual tanto de datos como de actividades en las tablas *estadosEI* y *estadosAct*.
- c) Accesa también a la tabla *entidades_información* para aplicar el cambio realizado durante la desconexión. Este cambio puede ser que la entidad se haya agregado, actualizado o borrado. Esto queda definido en el *modelo base de reconexión* que se explica más adelante.

Para colocar la implementación de toda la arquitectura, se instaló el servidor de Web Tomcat 4. Se eligió este servidor por el soporte que tiene para las tecnologías utilizadas (Japplets, Servlets, paginas JSP y el SMDB MySql).

En la siguiente sección se presenta la implementación de la arquitectura cliente para dispositivos PDA la cual ha sido probada en el emulador del ambiente del sistema operativo PalmOS versión 3.5.

V.4 Arquitectura cliente

La arquitectura cliente para los dispositivos PDA se forma de cuatro componentes: un *coordinador local*, el *generador de interfaz gráfica*, un *rol base* y un *contenedor local de datos*.

Para la implementación de la arquitectura cliente, en su totalidad se utilizó el lenguaje de programación *SuperWaba*. Este lenguaje permite el desarrollo de aplicaciones para dispositivos PDA que utilizan los sistemas operativos PalmOS y WindowsCE los cuales son los más comúnmente usados en esta clase de dispositivos. Este lenguaje utiliza un subconjunto de las clases del lenguaje Java.

El contenedor de datos utilizado en el dispositivo PDA se forma a partir de registros (al menos en aquellos con el sistema operativo PalmOS) los cuales son arreglos de bytes de longitud variable que son accedidos secuencialmente y a los que nos referimos como estructuras de datos.

A continuación se presenta la implementación de los componentes de la arquitectura cliente, sus métodos más importantes y la funcionalidad de éstos.

V.4.1 Coordinador Local

El coordinador local se implementó como una clase con el nombre *coordinador*. Esta clase reside en el dispositivo PDA y sus métodos más importantes son:

obtenerDatosHTTP().- Este método realiza la llamada al Servlet *transferencia* por medio de una conexión HTTP. Este Servlet responde con la cadena de texto con la definición del *modelo base WFD*.

WFDDocHandler().- Implementa las clases *XMLListener* y *XMLParser* propias del lenguaje *SuperWaba*, estas clases permiten analizar la cadena de texto que se recibe con el

método *obtenerDatosHTTP()* que contiene el *modelo base WFD*, lo separa y crea las estructuras de datos con la información de la siguiente forma:

- a) Crea la estructura de datos *EDGui* en donde almacenan de forma separada el contenido de los diferentes atributos del elemento *<gui>* (ver tabla V). Esta estructura es utilizada por el *generador de GUIs* para recuperar la definición de una pantalla.
- b) Crea la estructura de datos *EDEntidadInfo* en la que se almacena el contenido de cada una de las entidades de información utilizadas. Busca el elemento *<dato>* y toma el contenido del atributo *valor* para almacenarlo. Agrega el campo *estado* para mantener el control de los estados de las entidades.
- c) Crea la estructura de datos *EDAct* en la que se almacena el identificador y el nombre de cada una de las actividades de los roles que fueron transferidas al dispositivo. Agrega el campo *estado* para mantener el control de los estados de las actividades.

reconecta().- Este método se encarga de preparar la cadena de texto con la que se definen las actividades y las entidades de información que se reconectan. Esta cadena se envía al servidor por medio de una conexión HTTP al Servlet *transferencia*. Para explicar como se forma esta cadena, es necesario introducir un nuevo modelo base que establece las reglas para la definición de esta información

V.4.2 Modelo base para la reconexión.

Adicionalmente a los modelos descritos anteriormente, ha sido necesario definir un modelo para representar los datos que son modificados durante un período de desconexión de un dispositivo, a este modelo le llamamos el *modelo base para la reconexión*. Este modelo permite definir a través del lenguaje XML, la información necesaria para que cada vez que el o los roles que se encuentran en desconexión se reconecten a la máquina central de workflow (*AGP*), ésta puede integrar los datos y las actividades realizadas durante la

desconexión al flujo de trabajo. El DTD establecido para el *modelo base de reconexión* se detalla en el apéndice B.

El modelo base define las actividades que son transferidas del dispositivo a la máquina central. Cada actividad que se reconecta contiene los datos que fueron modificados o generados durante la desconexión. La definición de las actividades se crea en el método *reconecta()* de la clase *coordinador*. Este método recupera de las estructuras de datos *EDEntidadInfo* y *EDAct* los elementos del rol que se reconecta, utilizando adicionalmente la clase *DescDocHandler* que implementa las clases *XMLListener* y *XMLParser* para formar correctamente la cadena de texto. En la figura 29 se muestra un ejemplo de un *modelo base para la reconexión* con la definición de algunos de sus elementos.

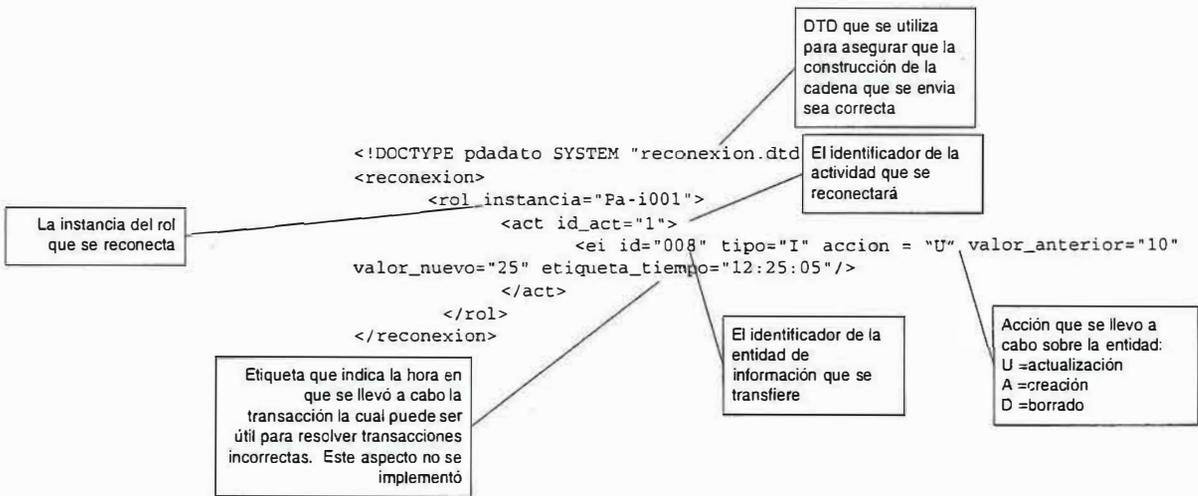


Figura 29. Ejemplo de un modelo base para la reconexión que muestra la definición de una actividad con una entidad de información para ser reconectadas a la máquina central de workflow

V.4.3 Generador de interfaz gráfica

La funcionalidad principal de este componente es la de interpretar la definición de una GUI en XML y transformarla en una GUI con elementos de interfaz gráfica visible en la pantalla de un dispositivo PDA.

El *generador de interfaz gráfica* se implementó por medio de la clase *generadorGUI* la cual reside en el dispositivo. El método más importante es:

generaGUI().- Este método obtiene la definición de una GUI de la estructura de datos *EDGui*. Hace uso de cinco clases más: *Item*, *itemButton*, *itemRadio*, *itemCheck*, *itemEdit* las cuales permiten traducir la definición de los atributos de un elemento de interfaz gráfica en un elemento de interfaz gráfica en el dispositivo como los que se muestran en la tabla VI.

V.4.4 Rol base

La finalidad del *rol base* en el PDA es mantener la definición de las actividades y las entidades de información utilizadas por el rol en el dispositivo. Este mecanismo se implementó como una clase en el dispositivo que se utiliza para actualizar los estados tanto de actividades como de las entidades de información en las tablas *EDEntidadInfo* y *EDAct*.

La arquitectura cliente ha sido probada solo en el emulador del ambiente del sistema operativo PalmOS versión 3.5.

Para finalizar esta sección, en la figura 30 se ilustra el flujo general de actividades que ocurren en SysCoo para la generación de un sistema de workflow desconectado, desde el inicio con la definición de los *modelos base RAD, EI y DTE* hasta la transferencia al PDA.

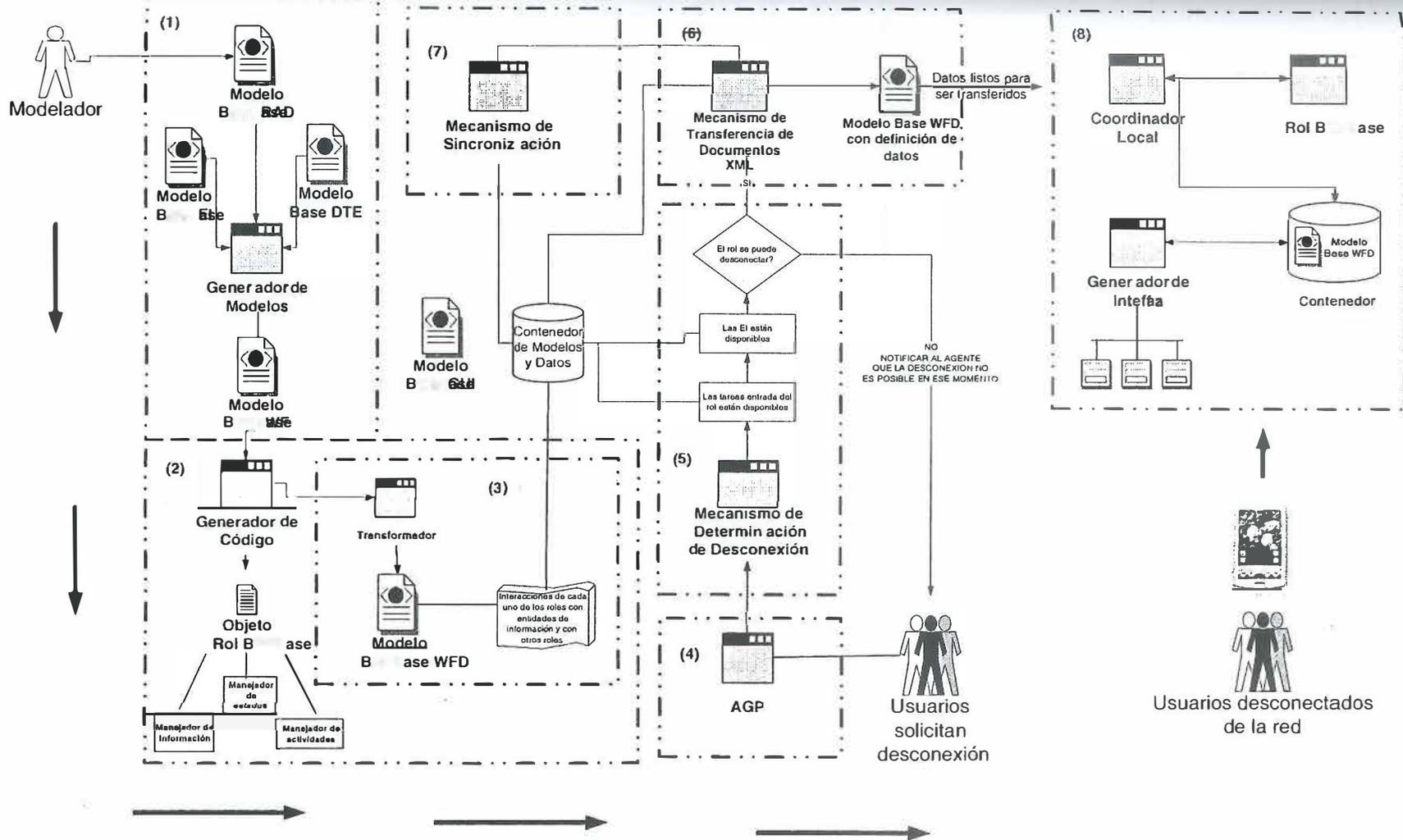


Figura 30. Las fases a través de las cuales se genera en SysCooor un sistema de workflow con apoyo a la operación desconectada, desde la definición del proceso, su transformación a un sistema de workflow y la transferencia a un dispositivo PDA

En la figura 31 se ilustra el flujo general de actividades que ocurren cuando un dispositivo reconecta uno o más roles a la máquina central de workflow.

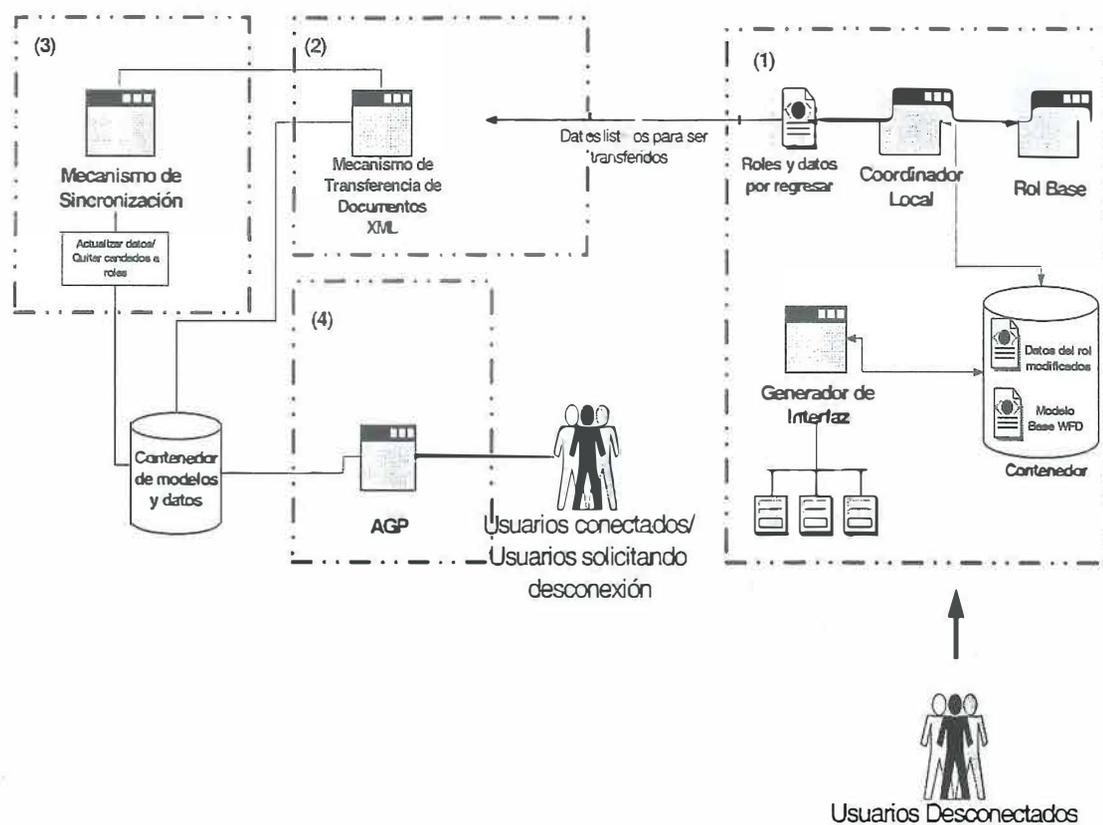


Figura 31. Escenarios de la fase de reconexión a la máquina central (AGP)

En la siguiente sección se describen los detalles de funcionalidad de las extensiones realizadas. Esta descripción se lleva a cabo utilizando el proceso *“Realizando Venta”* del caso de estudio realizado en la compañía vitivinícola *“Chateau Camou”*. Este escenario nos permite observar las fases en la generación del sistema de workflow desconectado, su transferencia al dispositivo, la funcionalidad durante la desconexión y la fase de reconexión a la máquina central de workflow.

V.5 Funcionalidad del sistema

Para observar la funcionalidad del sistema, hemos identificados tres fases de funcionalidad junto con los usuarios que participan en cada una de estas fases. Esta clasificación la presentamos en la tabla IX.

Tabla IX. Fases de funcionalidad de SysCoor y los usuarios involucrados

Fase	Usuarios
Generación del sistema de workflow desconectado	<i>Modelador</i> de las pantallas de usuario
	<i>Usuario</i> que carga al dispositivo la definición del sistema
Ejecución del workflow desconectado en el dispositivo PDA	<i>Usuario</i> que utiliza el sistema de workflow en el dispositivo PDA
	<i>Usuario</i> que utiliza el sistema de workflow en el navegador Web cuando existen roles desconectados
Reconexión a la máquina central de workflow (AGP)	<i>Usuario</i> que termina un período de desconexión
	<i>Usuario</i> que utiliza el sistema de workflow en el navegador Web cuando un rol termina un período de desconexión

En términos generales, el *modelador* es el responsable del diseño y creación de la información necesaria para la generación de los sistemas de workflow desconectado. El *usuario* es la persona de la organización que ejecuta los sistemas para realizar el *enactment* de procesos.

A continuación se enlistan las actividades a realizar para generar un sistema de workflow para ser utilizado en un dispositivo PDA. Estas actividades son definidas en base a la secuencia de escenarios que se debe seguir a fin de completar la definición de un sistema de workflow desconectado, realizar la ejecución del sistema en un dispositivo y terminar el período de desconexión integrando los resultados a la máquina central.

a) Generación del sistema de workflow desconectado

- ✓ El *modelador* se encarga de definir las interfaces gráficas de usuario (GUI) a través de un editor de texto. Las actividades se realizan son:
 - Crear el modelo para definir una GUI siguiendo las reglas establecidas en el DTD del *modelo base GUI*.
- ✓ El *transformador de código* del *generador de sistemas de workflow* se encarga de crear el *modelo base workflow desconectado (modelo base WFD)*. Las actividades involucradas son:
 - Leer el *modelo base WF*.
 - Crear el *modelo base WFD*.
- ✓ El *AGP* es la pantalla principal de la máquina de coordinación a través de la cual el usuario ejecuta los sistemas de workflow. Las actividades que realiza son:
 - Mostrar los sistemas de workflow generados.
 - Mostrar los roles participantes en cada sistema.
 - Mostrar los estados de los roles participantes en cada sistema.
 - Permitir la desconexión de los roles
- ✓ El *módulo de operación desconectada* lleva a cabo la desconexión y reconexión de los roles. Las actividades involucradas son:
 - Evaluar la factibilidad de la desconexión del rol.
 - Transferir la definición del sistema de workflow (*modelo base WFD*).
 - Recibir los roles que se reconectan.

b) Ejecución del workflow desconectado en el dispositivo PDA y reconexión a la máquina central de workflow (AGP)

- ✓ El *usuario* (miembro de la organización) accesa a la herramienta *SysCoor* para ejecutar su rol en algún sistema de workflow generado. Las actividades involucradas son:
- Seleccionar el sistema de workflow donde se encuentra el rol que desempeñará.
 - Ejecutar el sistema de workflow en el que necesita participar ya sea de forma conectada o desconectada.
 - Desconectar el rol.
 - Reconectar el rol.

En base a estas actividades definidas, en la siguiente sección se lista la funcionalidad esperada en cada una de las tareas establecidas.

V.6 Relación de las actividades con la funcionalidad del sistema

La Tabla X muestra la relación que existe entre la funcionalidad de la arquitectura de la herramienta *SysCoor* definida en la sección anterior, con respecto a cada una de las actividades descritas que se deben realizar.

Tabla X. Relación actividad-funcionalidad

Actividad	Funcionalidad
Crear el modelo de la GUI	- Definir el modelo de la GUI siguiendo las reglas del <i>modelo base GUI</i> - Guardar el modelo creado en el contenedor de modelos y datos.
Leer el <i>modelo base WF</i>	- Acceder a la ubicación del modelo en el servidor de Web y leerlo.
Crear el <i>modelo base WFD</i>	- Obtener la definición de las actividades en el <i>modelo base WF</i> . - Obtener la referencia a las GUIs utilizadas por cada actividad definida en el <i>modelo base WF</i> . - Recuperar la definición de GUIs utilizadas por cada actividad definida en el <i>modelo base WF</i> . - Validar la definición de la GUI usando el <i>modelo base GUI</i> - Agregar la descripción de las GUIs utilizadas por cada actividad que las implementen. - Crear una tabla en el contenedor de modelos y datos con la definición de las actividades definidas en el <i>modelo base WF</i> . - Crear una tabla en el contenedor de modelos y datos con la definición de las entidades de información definidas en el <i>modelo base WF</i> . - Guardar en un archivo el <i>modelo base WFD</i> .
Mostrar los <i>sistemas de workflow</i> generados	- Acceder a la ubicación de los archivos de procesos en el servidor de Web y leerlos (<i>roles base</i>). - Desplegar los archivos encontrados en pantalla.

<i>Continuación Tabla X</i>	
Actividad	Funcionalidad
Mostrar los roles participantes en cada sistema	<ul style="list-style-type: none"> - De acuerdo al archivo del proceso seleccionado, deserializarlo para buscar el nombre del rol y agente correspondiente. - Desplegar los roles y agente correspondiente en pantalla. - Mostrar el mecanismo para desconectar el rol
Permitir la desconexión de los roles	<ul style="list-style-type: none"> - Por cada rol definido en el proceso, mostrar un botón en la pantalla que activará el proceso de desconexión.
Mostrar los estados de los roles participantes en cada sistema	<ul style="list-style-type: none"> - De acuerdo al archivo del proceso seleccionado, buscar en las tablas del <i>contenedor de modelos y datos</i>, los estados de las actividades definidas en el proceso. - Desplegar los datos antes mencionados en pantalla.
Evaluar la desconexión de un rol	<ul style="list-style-type: none"> - Recuperar el <i>modelo base WFD</i>. - Consultar las tablas con la definición de las actividades que se requieren desconectar. - Evaluar la disponibilidad de las entidades de información. - Enviar mensajes al <i>AGP</i> sobre los resultados de la evaluación. - Activar al <i>mecanismo de transferencia de documentos XML</i>.
Transferir la definición del sistema de workflow (<i>modelo base WFD</i>)	<ul style="list-style-type: none"> - Leer el <i>modelo base WFD</i>. - Analizar el <i>modelo base WFD</i> y obtener la definición del rol que se desconecta. - Acceder al <i>contenedor de modelos y datos</i> para recuperar el contenido de las entidades de información que serán transferidas. - Agregar el contenido de las entidades de información a la definición del rol. - Analizar la validez del modelo. - Acceder al <i>contenedor de modelos y datos</i> para cambiar el estado de las entidades de información y de los roles que son transferidos. -Mostrar en el área de mensajes del <i>AGP</i> la notificación de que el rol ha sido transferido. -Actualizar en la pantalla principal del <i>AGP</i> los estados de los roles para reflejar la desconexión.
Recibir los roles que se reconectan	<ul style="list-style-type: none"> - Leer y analizar la definición de los datos y actividades que se desconectan o reconectan. - Acceder al <i>contenedor de modelos y datos</i> para cambiar el estado de las entidades de información y de los roles que son reconectados. -Mostrar en el área de mensajes del <i>AGP</i> la notificación de que el rol ha sido reconectado. -Actualizar en la pantalla principal del <i>AGP</i> los estados de los roles para reflejar la reconexión.
Seleccionar el sistema de workflow donde se encuentra el rol a ejecutar.	<ul style="list-style-type: none"> - Obtener el nombre del archivo del proceso seleccionado.
Seleccionar el rol en el que participa	<ul style="list-style-type: none"> - Obtener el nombre del rol seleccionado.
Desconectar el rol	<ul style="list-style-type: none"> - Mostrar el botón "<i>desconectar</i>" en los roles que se muestren en el <i>AGP</i>, solo de aquellos que contengan actividades. - Mostrar en el área de mensajes del <i>AGP</i> si se pudo realizar la desconexión del rol, si es así, mostrar el directorio donde fue generado el archivo para cargar al dispositivo. Si no fue posible desconectarlo, dar la información para que el usuario proceda.
Ejecutar el rol	<ul style="list-style-type: none"> - Mostrar las interfaces del rol y entidades de acuerdo al del proceso. - Guardar cada cambio realizado por el usuario.

<i>Continuación Tabla X</i>	
Actividad	Funcionalidad
Reconectar el rol	<ul style="list-style-type: none"> - Mostrar los roles que se encuentran cargados en el dispositivo para que el usuario seleccione el que necesita reconectar. - Mostrar en el dispositivo mensajes sobre el resultado de la reconexión. - Mostrar los roles que aun se encuentran cargados en el dispositivo. - En el <i>AGP</i>, mostrar en el área de mensajes, notificación de los roles que han sido reconectados.

Para describir las actividades que se realizan para generar un sistema de workflow desconectado, utilizamos uno de los modelos del caso de estudio realizado en la compañía Chateau Camou (caso descrito en el Apéndice A). Este modelo es el del proceso “*Realizando Venta*”, cuyo escenario se describe en el capítulo III. El objetivo de este proceso es llevar a cabo una venta satisfactoria mediante la realización de visitas a los clientes de la compañía.

Los pasos a seguir para generar el sistema de workflow son los siguientes:

1. Realizar la definición de pantallas de usuario (GUIs) a través del lenguaje XML y validar por medio del *modelo base GUI* utilizando el *generador de sistemas de workflow* cuando se genera el *modelo base WFD*.
2. Abrir el *modelo base WF* para realizar el detallado de este modelo, es decir, especificar las entidades de información manipuladas y las interfaces utilizadas por cada rol.
3. Seleccionar la opción del *generador de sistemas de workflow* de crear el sistema de workflow el cual crea los *roles base* y el *modelo base WFD*.
4. Desconectar un rol utilizando la herramienta.
5. Ejecutar un rol utilizando la herramienta *SysCoor* en el Web.
6. Ejecutar un rol utilizando la herramienta *SysCoor* en el PDA.
7. Ejecutar un rol utilizando la herramienta *SysCoor* en el Web, el cual dependa de la salida de otro que se encuentra en desconexión.
8. Solicitar la reconexión de un rol utilizando la herramienta *SysCoor* en el dispositivo.
9. Registrar los resultados y observaciones realizadas durante la generación del *modelo base WFD* y durante las fases desconexión, ejecución y reconexión del sistema de workflow desconectado generado en la herramienta *SysCoor*.

A continuación se describirán cada uno de estos pasos, mostrando las pantallas utilizadas en cada uno de ellos.

V.6.1 Definición y validación de las pantallas de usuario (GUIs).

Para la definición de las pantallas de usuario (GUIs) se utilizó un editor de texto genérico, realizando la especificación del mismo de acuerdo a la estructura que debe contener definida en su DTD. La figura 32 muestra parte del modelo realizado para definir una GUI la cual sirve para capturar un pedido del cliente. Se pueden realizar diferentes GUIs utilizando diferentes elementos de interfaces (tabla VIII en el capítulo V) de acuerdo a las necesidades específicas de la actividad. Como se observa en la figura 32, en la primer línea se indica el nombre del DTD que valida a este modelo.

```

<!DOCTYPE interfaz SYSTEM "GUI.dtd">
<interfaz>
<rol id_rol="a">
  <act id_act="1">
    <elementogui id_gui = "a11" tipo="ComboBox"
      texto="Cliente" x="10" y="5" valor = "Cliente 1,Client
2,Cliente 3,Cliente 4,Cliente 5,Cliente 6"/>
    <elementogui id_gui = "a12" tipo="ComboBox"
      texto="Producto" x="10" y="15" valor= "Producto
1,Producto 2,Producto 3,Producto 4,Producto 5,Producto 6"/>
    <elementogui id_gui = "a13" tipo="InputBox"
      texto="Cantidad" x="10" y="25" valor="" />
    <elementogui id_gui = "a14" tipo="ItemButton"
      texto="Agregar" x="10" y="35" />
    <elementogui id_gui = "a15" tipo="ItemButton"
      texto="Limpiar" x="20" y="35" />
  </act>
</rol>
</interfaz>

```

Figura 32. Definición de una pantalla para la captura de un pedido en el proceso de "Realizando Venta".

Este modelo se valida cuando se crea el *modelo base WFD*, por lo cual hay que almacenarlo en el directorio `C:\Tomcat4\webapps\coordinación\serializados`, que es el directorio específico en el servidor de Web en el que se encuentran los modelos y el *contenedor de datos*.

V.6.2 Apertura del modelo base WF en el generador de sistemas de coordinación para especificar las entidades de información manipuladas y las pantallas utilizadas por cada rol.

Una vez que se ha almacenado la definición de las GUI, se ejecutó el *generador de sistemas de workflow* desde un navegador de Web (<http://coolab5/coordinacion/Pagina.html>), se abrió el *modelo base WF* del proceso “*Realizando Venta*”. Este modelo había sido generado previamente en el *generador de sistemas de workflow*.

Una vez abierto el *modelo base WF*, se agregaron los detalles de las entidades de información utilizadas y de las GUI que se usaran en cada una de las actividades. Respecto a las entidades de información solo se agrega la referencia a estas. En la figura 33 se observa que para las actividades “*Obtener requerimientos de mercancía del cliente*” y “*Llenar formato de pedido*”, utilizaremos la GUI (conocida también como plantilla) que acabamos de definir. Se guardaron los cambios efectuados en el modelo, el cual reside en el directorio `C:\Tomcat4\webapps\coordinación\serializados`.

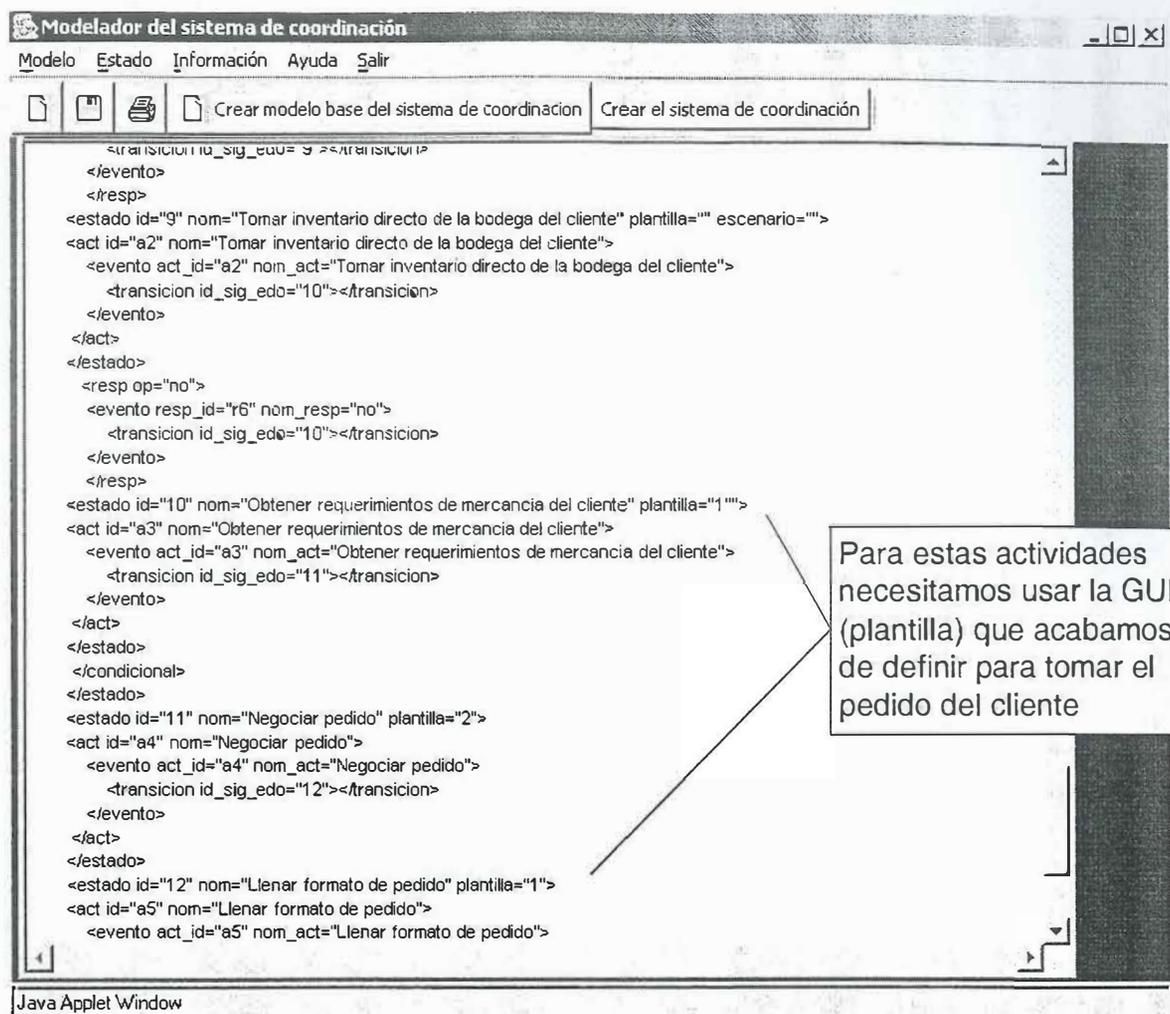


Figura 33. El detallado del *modelo base WF* utilizando el editor que implementa el *generador de sistemas*

V.6.3 Generar el sistema de workflow (roles base) y el modelo base WFD

Una vez que terminamos de detallar el *modelo base WFD*, tomamos la opción “*crear el sistema de coordinación*”. Esta opción crea los roles base para el sistema de workflow (figura 34 a). Además se forma el *modelo base WFD* (figura 34 b). La información sobre el procesamiento que se realiza para generar este modelo se despliega en la consola de JAVA (figura 34).

Generación de roles base

```

Java Console
Estas guardando esto: ventasEdo.xml
[Ljava.lang.String;@e29820
el generado DTE en manejadorContenidoRAD ventasEdo.xml
ventasInfo.xml
Estas guardando esto:
[Ljava.lang.String;@c4d04d
arch es: http://coolab5.coordinacion/serializados/ventas.xml
lo que envio para desplegar en el area nueva: [Ljava.lang.String;@47917a
valores que se le envian a la ventana de estado: [Ljava.lang.String;@47917a
101

Entrada: ventas.sis

EN SERIALIZAR : serializando ventas.sis
ACT:1
ACT:2
ACT:3
ACT:4

Archivo generad: ventas.dat
1 rol base

Sistema de workflow creado
  
```

Clear Copy Close

(a)

Generación del modelo base WFD

```

Java Console
Estas leyendo : ventas.sis
Estas guardando esto:
[Ljava.lang.String;@c4d04d
arch es: http://coolab5.coordinacion/serializados/ventas.sis

Estas leyendo : ventas.xml
Estas guardando esto:
act: 1 usa gui:
      usa dato:
act: 2 usa gui:
      usa dato:
act: 3 usa gui:
      usa dato:
act: 4 usa gui:
      usa dato:
act: 5 usa gui:
      usa dato:

Estas leyendo : gui01.xml
Estas guardando esto:
[Ljava.lang.String;@c4d04d
Archivo generad: ventas-wfd.xml
Sistema de workflow creado
  
```

Clear Copy Close

(b)

Figura 34. Generación de los roles base del modelo base WFD utilizando el modelo base WF

V.6.4 Accesar la herramienta SysCoor con el fin de ejecutar la desconexión de un rol.

Una vez que generamos los *roles base* y el *modelo base WFD* entramos a la interfaz del *AGP* utilizando el URL (*Uniform Resource Locator*) `http://coolab5/coordinación/Principal.jsp`, el cual es el JSP que se encarga de buscar en el directorio `C:\Tomcat4\webapps\coordinación\serializados`, los sistemas de workflow generados. Como se ve en la Figura 35, se encuentran cuatro sistemas generados, de los cuales seleccionamos “*ventas.dat*”.

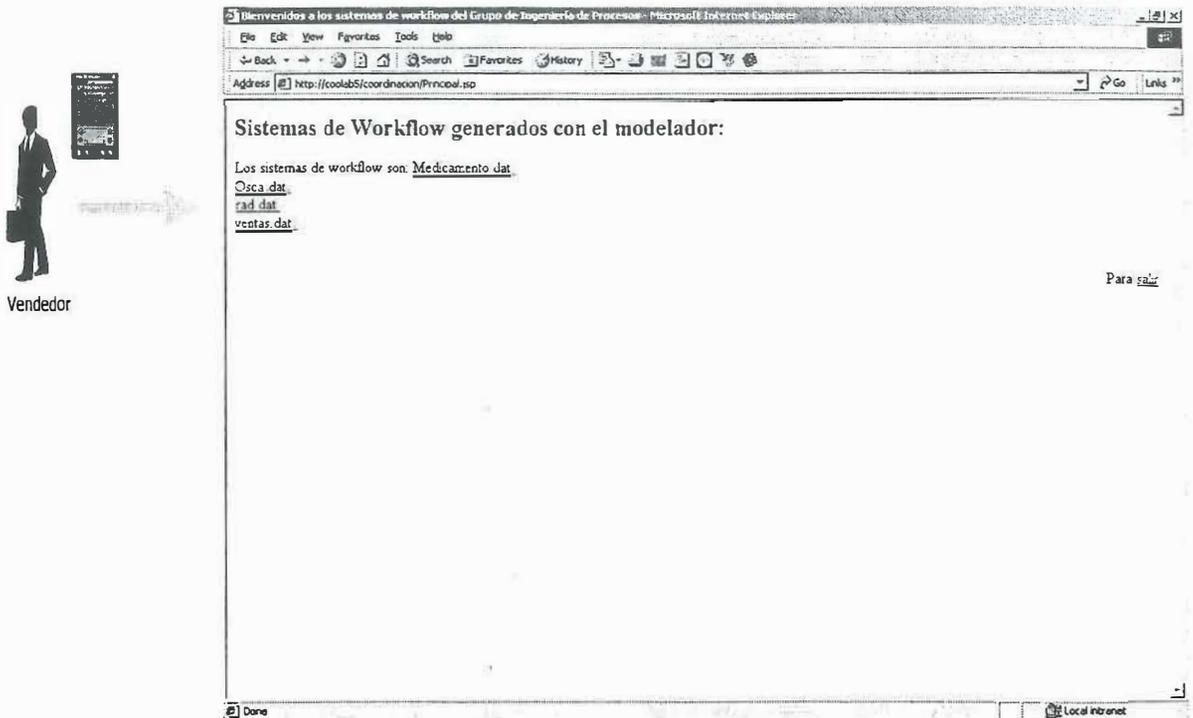


Figura 35. Los sistemas de workflow generados

Al seleccionar el sistema “*ventas.dar*”, se presenta la interfaz que se aprecia en la figura 36, en la parte derecha se presentan los roles participantes en el proceso y en la izquierda el estado actual de cada uno de ellos y el agente que los ejecuta. Esta es la pantalla principal del AGP.

Vendedor

Roles participantes en el workflow mostrando el agente participante, el estado en el que se encuentra el proceso y si el usuario se encuentra conectado al servidor central o no

Rol	Agente	Estado Actual	Conectado	
Realizando Venta	Vendedor	1 Recibir solicitud	Si	Desconectar
Solicitando Mercancia	Cliente	1 Enviar solicitud	No	

Mensajes:

Area de mensajes por medio de la cual el AGP comunica al usuario sobre diferentes eventos

Botón para desconectar el rol. Al presionar el botón el AGP inicia todo el proceso para desconectar el rol

Roles participando en el workflow. A través de estas ligas el agente accesa a la actividad actual en la que se encuentra el proceso

Inicio del proceso

Figura 36. Pantalla principal de SysCoor (AGP)

El siguiente paso es tratar de desconectar el rol “*Realizando Venta*” y esperar el resultado del proceso de *evaluación de desconexión de un rol*. Este resultado debe mostrarse en el área de mensajes de la pantalla de SysCoor. El resultado lo podemos observar en la figura 37.

Vendedor

Grupo de Ingeniería de Procesos
CICESE

Rol	Agente	Estado Actual	Conectado	
Realizando Venta	Vendedor	Recibir solicitud	Si	Desconectar
Solicitando Mercancia	Cliente	Emitir solicitud	No	

Mensajes:

Mensaje: El rol Realizando Venta no puede desconectarse
 Motiv: El rol Solicitando Mercancia no iniciado
 Solución: Inicie rol Solicitando Mercancia

Boton para desconectar el rol

Area de mensajes en la que se indica, en este caso, que el rol no se puede desconectar. También incluye un mensaje indicando una posible causa

Para salir del proceso

Figura 37. Pantalla principal de SysCoor (AGP) indicando al usuario que la desconexión no es posible en ese momento

De acuerdo a los mensajes desplegados en el área de mensajes, el *mecanismo de determinación de desconexión* evaluó la desconexión y determinó que ésta no se puede llevar a cabo porque el rol “Realizando venta” requiere de la actividad previa “Solicitando mercancía” haya sido realizada por el agente “Cliente” el cual, de acuerdo a la tabla de estados (parte superior de la ventana de mensajes), no se encuentra conectado y no ha iniciado la actividad.

Una vez que iniciamos la actividad “Emitir solicitud” en el rol del cliente, intentamos otra vez la desconexión del rol “Realizando venta”. El resultado de la evaluación de la desconexión se ilustra en la figura 38.

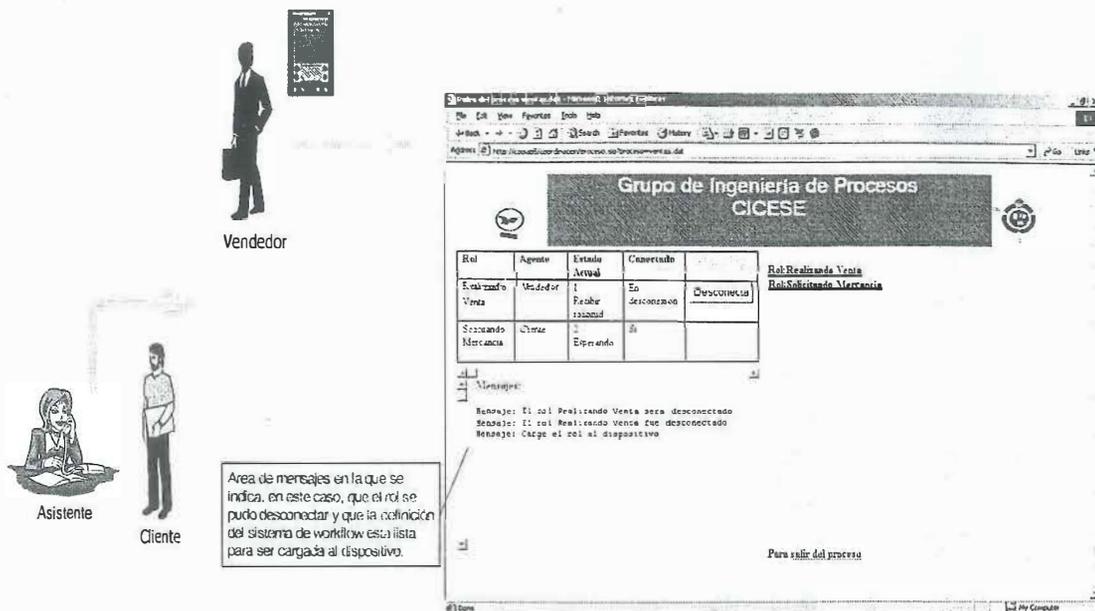


Figura 38. Pantalla principal de SysCoor (AGP) indicando al usuario la definición del sistema está lista para ser transferida

V.6.5 Accesar la herramienta SysCoor con el dispositivo con el fin de ejecutar un rol

Cuando el *modelo base WFD* ha sido preparado para ser transferido, por medio de la herramienta SysCoor que reside en el dispositivo este modelo puede ser cargado. La figura 39 muestra la secuencia de las actividades que se realizan para que toda vez que la definición del sistema se encuentra en el dispositivo, pueda ser accesada por el agente desconectado. En este caso, cargamos tres roles al dispositivo, de los cuales el que probamos fue el de “Realizar venta” para verificar que la forma para capturar el pedido que definimos fuere desplegada correctamente, y que la secuencia en la que sucede la ejecución del rol se cumpliera.

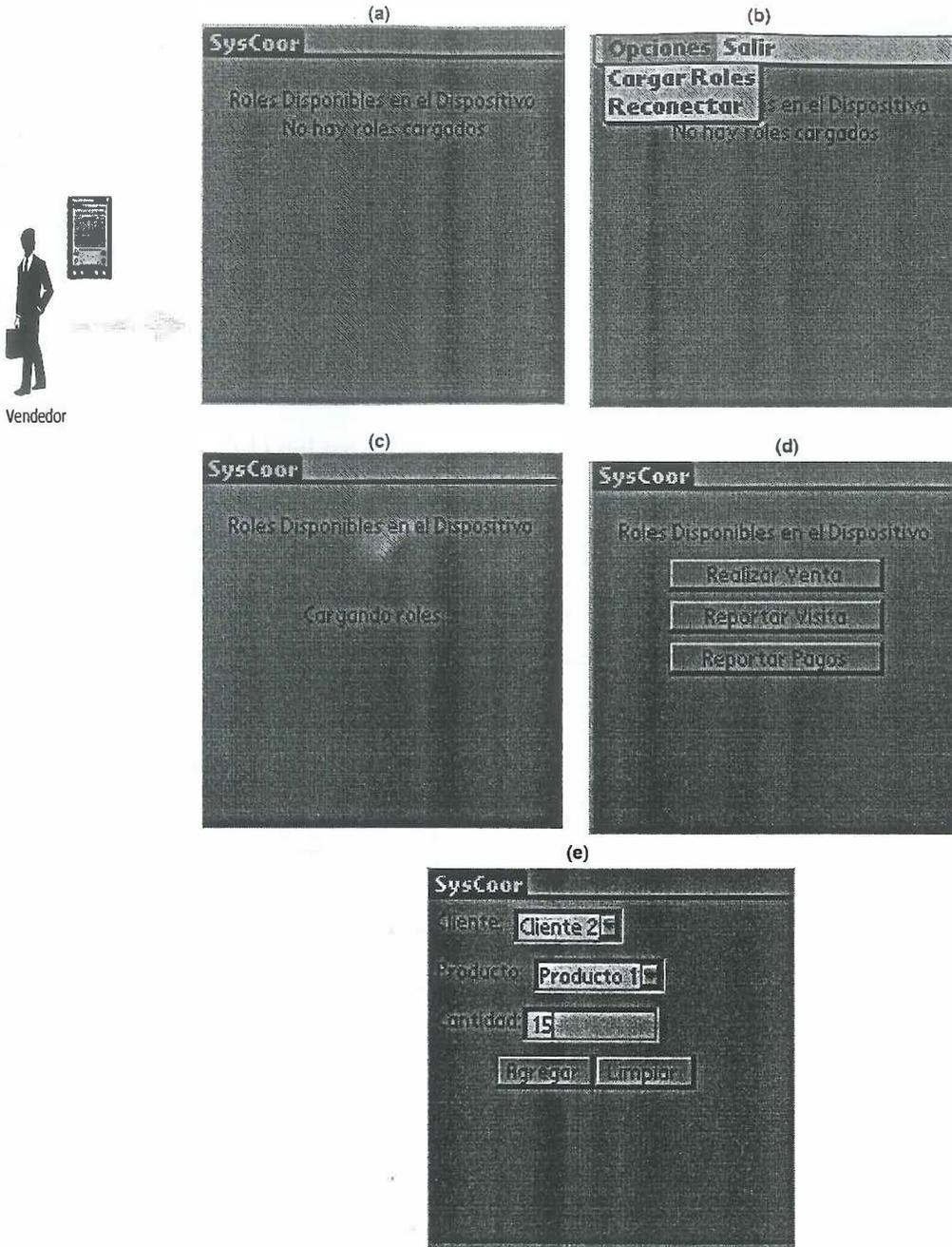


Figura 39. Escenarios en el dispositivo PDA en el orden en el que ocurren. Primero se (a) consultan los roles que se encuentran en el dispositivo, para cargar nuevos roles (b) se toma la opción "cargar roles", el dispositivo se conecta (c) y carga los roles que han sido preparados (d) para terminar con la ejecución de alguno de estos roles (e)

V.6.6 Accesar la herramienta SysCoor en el Web con el fin de ejecutar un rol que depende de la salida de uno que se encuentra en desconexión

Para verificar que el AGP siga manteniendo una coordinación adecuada de los recursos y las actividades que participan en la ejecución de un sistema de workflow, intentamos realizar la actividad “revisar pedido” del rol “Solicitando Mercancía”, la cual depende de la salida del rol “Realizando venta”. Observamos los mensajes en la figura 40.

Grupo de Ingeniería de Procesos
CICESE

Rol	Agente	Estado Actual	Conectado	
Realizando Venta	Vendedor	1 Recibir solicitud	En desconexión	Desconecta
Solicitando Mercancía	Cliente	3 Revisar pedido	Si	

Agente: Cliente
Actividad: Revisar pedido
Esperando Respuesta.....

Mensajes:
Mensaje: El rol Realizando Venta no esta conectado
Mensaje: actividad Revisar pedido agente Cliente no puede realizarse
Mensaje: espera actividad Finalizar proceso agente Vendedor

El agente Cliente, solicita continuar con la siguiente actividad

La actividad que el agente cliente solicita ejecutar depende de la salida del proceso que un proceso que se encuentra en desconexión

Para salir del proceso

Figura 40. Pantalla principal de SysCoor (AGP) indicando al usuario que la actividad “revisar pedido” no puede llevarse a cabo, ya que depende de otra que aun no termina

Como se puede observar, el AGP continua manteniendo un adecuado manejo del proceso a pesar de que uno de los roles que lo componen se encuentra en desconexión.

V.6.7 Accesar la herramienta SysCoor en el dispositivo con el fin de solicitar la reconexión de un rol

Para finalizar, procedemos a reconectar el rol “Realizando venta” a la máquina central de workflow (AGP). Con esta finalidad, en el dispositivo seleccionamos el rol para ser descargado. La figura 41 muestra los pasos para realizar esta actividad.

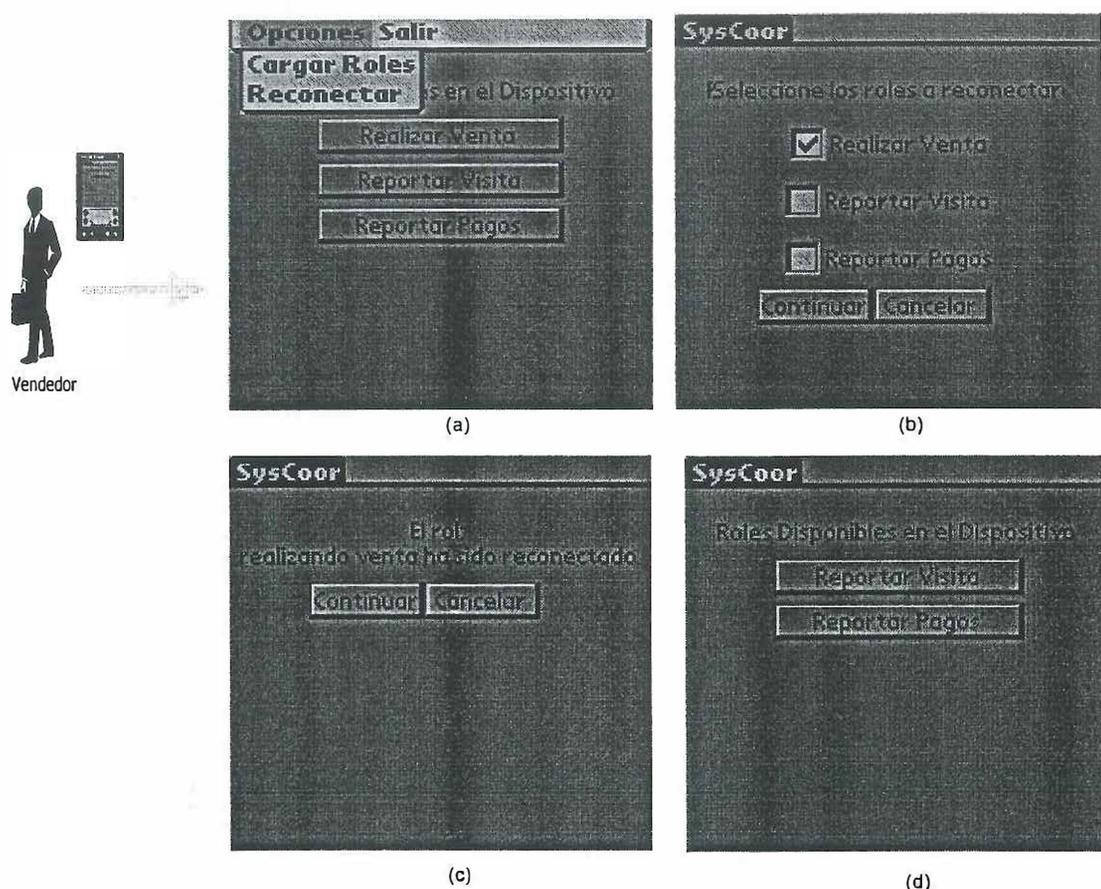


Figura 41. Escenarios en el dispositivo PDA en el orden en el que ocurren cuando se requiere reconexión. Primero se toma la opción “reconectar” (a), el dispositivo muestra los roles que se encuentran cargados (b) de los cuales el usuario debe elegir aquellos que se reconectarán, el coordinador local notifica la reconexión (c), por último, se muestran los roles que aún están en el dispositivo (d)

Observamos los mensajes que emite el *mecanismo de transferencia de documentos XML* y el cambio de estados en entidades de información y actividades que realizó el *mecanismo de sincronización* (figura 42).

Rolet del proceso ventas.dat - Microsoft Internet Explorer

Address: http://coo1ab5/coordnacion/proceso.jsp?proceso=ventas.dat

**Grupo de Ingeniería de Procesos
CICESE**

Rol	Agente	Estado Actual	Conectado	
Realizando Venta	Vendedor	5 Final proceso	Si	Desconecta
Solicitando Mercancia	Cliente	3 Final proceso	Si	

Rol: Realizando Venta
Rol: Solicitando Mercancia

Mensajes:

Mensaje: El rol Realizando Venta fue reconectado
Mensaje: actividad Obtener requerimientos de mercancia
Mensaje: actividad Negociar pedido
Mensaje: actividad Llenar formato de pedido

Area de mensajes en la que se indica, que el rol fue reconectado y las actividades que fueron completadas

Se actualizan correctamente los estados de las actividades en los roles

Para salir del proceso

Figura 42. Pantalla principal de SysCooor (AGP) indicando al usuario la reconexión de un rol

V.7 Resumen

En este capítulo se presentó la implementación de las extensiones para el apoyo a la operación desconectada en la herramienta SysCooor.

El punto de inicio para extender la funcionalidad de SysCooor fue la generación de una nueva representación de un sistema de workflow por medio del *modelo base WFD* el

cual permite definir un sistema de workflow con los elementos necesarios para ejecutarse sin necesidad de consultar una máquina central de workflow.

Se describió el desarrollo llevado a cabo para apoyar la operación desconectada en sistemas de workflow, el cual consistió en modificar algunos elementos de la arquitectura básica (previa) de *SysCoor*, implementar extensiones para el servidor y el desarrollo de una arquitectura ligera para un cliente de forma que pueda ser implementada en dispositivos PDA.

Se mostró también la funcionalidad de las extensiones por medio del proceso de “Realizando Venta” del caso de estudio realizado en la compañía vitivinícola “Chateau Camou”. Este escenario nos permitió probar las fases en la generación del sistema de workflow desconectado, su transferencia al dispositivo, la funcionalidad durante la desconexión y la fase de reconexión a la máquina central de workflow.

En el siguiente capítulo se presentan las conclusiones obtenidas a partir de la realización de esta tesis. Así mismo, se enlista una serie de beneficios obtenidos a partir de las extensiones propuestas y se señala el trabajo futuro en el área de workflow y workflow desconectado.

Capítulo VI. Conclusiones, aportaciones y trabajo futuro

VI.1 Conclusiones

Los Sistemas Administradores de Workflow (WFMS) han proveído de apoyo a la automatización en la ejecución de tareas permitiendo de esta forma que las organizaciones logren un ambiente de trabajo más coordinado y controlado. Recientemente, estas mismas organizaciones se han estado desplazando hacia ambientes que requieren movilidad, dando lugar al uso de dispositivos de cómputo móvil que presentan la necesidad de trabajar desconectados de una red de comunicación. Por medio de este trabajo, hemos presentado el concepto de *workflow desconectado (operación desconectada en sistemas de workflow)* como un enfoque para apoyar la movilidad de usuarios de este tipo de sistemas.

A través de la realización de dos casos de estudio, pudimos observar escenarios reales de trabajo donde las personas se mueven de un ambiente donde cuentan con todos los elementos para llevar a cabo sus actividades, a uno en donde los recursos pueden no estar a la mano. Los escenarios observados en los casos de estudio, nos permitieron definir algunos requerimientos iniciales para una herramienta que provea soporte a las actividades cuando para ejecutarlas, se requiere de movilidad y alejamiento de un área de trabajo (trabajo desconectado). Estos casos de estudio se llevaron a cabo en la compañía vitivinícola “Chateau Camou” en el área de ventas y en la Clínica Hospital General de Zona IV No. 8 del Instituto Mexicano del Seguro Social (IMSS) en las áreas de medicina interna y urgencias.

Con el fin de conocer los esfuerzos que se han realizado para apoyar el trabajo desconectado en sistemas de workflow, se llevó a cabo una revisión de la literatura existente en el área, la cual permitió identificar los requerimientos que han sido previamente establecidos e implementados para apoyar este tipo de operación.

El resultado obtenido a partir de los casos de estudio y de la revisión de la literatura, fue el definir cuatro áreas generales que se deben considerar para el apoyo a la operación desconectada en sistemas de workflow. Estas áreas son: evaluación de la factibilidad en la desconexión de un rol, provisión de información al cliente previo a la desconexión, sincronización de la información generada o modificada durante la desconexión cuando ésta finaliza y el control de la coordinación mientras los clientes se encuentran desconectados.

Antes de proponer las extensiones que apoyan el trabajo desconectado, se describió el WFMS *SysCoor*, cuya arquitectura provee soporte a la generación semiautomática de sistemas de workflow a través del uso de modelos de procesos organizacionales descritos en el Lenguaje eXtensible de Marcado (XML por sus siglas en Inglés). Este WFMS es la base de la propuesta tecnológica presentada.

Basándonos en las cuatro áreas mencionadas, se propusieron las extensiones a la arquitectura de *SysCoor* para el apoyo a clientes móviles que trabajan en desconexión. El resultado fue la definición de cuatro componentes: un *transformador de código* dentro del *generador de sistemas de workflow* existente, una base de datos para almacenar definiciones de los sistemas de workflow, un *módulo para la operación desconectada* y la arquitectura para clientes PDAs. Esta arquitectura ligera se compone de cuatro componentes: el *coordinador local*, el *generador de interfaz gráfica del usuario*, el *rol base* y un *contenedor de datos*. La arquitectura cliente fue desarrollada utilizando el lenguaje de programación *SuperWaba*, sin embargo, ésta puede ser implementada utilizando otros lenguajes de programación para dispositivos PDAs

La implementación realizada fue evaluada en términos de funcionalidad, por medio del desarrollo de un sistema de workflow desconectado para el proceso “*Realizar Venta*” del caso de estudio llevado a cabo en la compañía Chateau Camou. Al cumplir con la funcionalidad definida en el análisis y diseño de las extensiones, el resultado de la

evaluación concluyó que las extensiones realizadas pueden apoyar de una forma adecuada la operación desconectada en actividades de workflow.

VI.2 Aportaciones

La principal aportación realizada a través de la realización de esta tesis, fue la de definir los mecanismos necesarios para el apoyo a la operación desconectada en sistemas de workflow. Estos mecanismos fueron establecidos en base a las fases en las que se desarrolla la operación desconectada y tomando en cuenta los elementos involucrados en este tipo de sistemas, por lo tanto, pueden considerarse de aplicación general para esta clase de operación en sistemas de workflow, de forma que pueden ser implementados en otras arquitecturas tomando en cuenta las características propias de éstas.

Además de la aportación principal que acabamos de establecer, se realizaron otras aportaciones las cuales se presentan a continuación.

1. *Identificación de cuatro áreas básicas que deben ser consideradas para el apoyo a la operación desconectada en sistemas de workflow.* A través de la realización de dos casos de estudio y de la revisión de la literatura existente en esta área, obtuvimos una serie de requerimientos que nos llevaron a definir cuatro áreas generales las cuales se presentaban recurrentemente en los escenarios. Estas áreas nos permitieron diseñar mecanismos que apoyan la operación desconectada en sistemas de workflow, a lo largo de las fases que este tipo de operación presenta.
2. *Definición de un sistema de workflow con los elementos necesarios para trabajar de forma desconectada.* La definición del *modelo base WFD* permite crear una representación completa de un sistema de workflow para ser ejecutado sin necesidad de consultar una máquina central de workflow. Este modelo base permite representar el sistema de workflow en forma estructurada y utilizarlo en la arquitectura cliente para construir un sistema de workflow ejecutable. Este modelo tiene la propiedad de ser

portable, ya que se puede utilizar en cualquier plataforma que tenga soporte de XML, debido a que es un archivo de texto en forma estructura de acuerdo a las reglas establecidas en un documento conocido como DTD.

3. *Interoperabilidad de interfaces gráficas de usuario por medio de su definición a través del lenguaje XML.* Este aspecto fue introducido para solucionar el problema de las diferencias de despliegue gráfico entre las computadoras de escritorio (plataforma original de *SysCoor*) y los dispositivos PDA. Mientras que los sistemas de workflow de *SysCoor* para el Web utilizan páginas HTML, se hizo necesario redefinir los elementos de interfaz gráfica presentes en estas páginas a elementos de interfaz gráfica comunes a cualquier lenguaje de programación para PDAs. El resultado fue el *modelo base GUI*, el cual permite definir diferentes interfaces gráficas para las actividades de acuerdo a necesidades específicas de estas, que pueden ser interpretadas e implementadas por lenguajes de programación para PDAs.
4. *Introducción del concepto de instancia en la herramienta de SysCoor.* El desarrollo llevado a cabo por García Carrillo y Ramírez Fernández estableció las bases de la herramienta *SysCoor*, si embargo, en éste no se incluía la capacidad de ejecutar más de una instancia de un rol, es decir, el rol se desarrollaba como un hilo único, iniciaba y no podía crearse una nueva instancia del rol antes que la anterior terminara. En este trabajo introducimos el manejo de múltiple instancias de un rol, de forma que un usuario puede iniciar varias instancias del mismo rol. Este concepto de múltiples instancias de un rol fue solo probado con roles que se desconectan, y permite que los usuarios puedan ejecutar diferentes instancias del mismo rol.

VI.3 Trabajo futuro

El área de la operación desconectada en sistemas de workflow, y en general, de los sistemas de workflow, establece retos que pueden llevar a aumentar la funcionalidad de éstos. A continuación presentamos algunas de las áreas de investigación y trabajo futuro

relacionadas con los retos existentes para apoyar la operación de sistemas de workflow en ambientes que requieren de movilidad.

1. *Apoyo a desconexiones involuntarias.* Las desconexiones involuntarias ocurren cuando una falla inesperada en el sistema de comunicación impide al usuario continuar con su trabajo, mientras que las desconexiones voluntarias son aquellas en las que el usuario toma explícitamente sus tareas y las desconecta de un sistema de comunicación. En esta tesis limitamos la operación de sistemas de workflow a ambientes de desconexiones voluntarias. Sin embargo en ciertos ambientes, como en aquellos que involucran el uso de redes inalámbricas, el apoyo a desconexiones involuntarias permite a los agentes en movilidad continuar con la ejecución de un sistema de workflow a pesar de estas desconexiones. Un enfoque posible es el de mantener en un *caché* porciones de los roles en los que participa un usuario y de las entidades de información utilizada. Se puede pensar en la elección de estas porciones de acuerdo a un enfoque probabilístico basado en la frecuencia de utilización de roles por parte de un usuario, si embargo, esta implementación requiere de un dispositivo con recursos suficientes para mantener este *caché*.

2. *Apoyar la movilidad a través del uso de agentes autónomos de software.* Un agente puede ser visto como una entidad que se encarga de tratar con un problema específico observando autonomía, sociabilidad e iniciativa [Wooldrige y Jennings,1995] Actualmente el AGP mantiene un control centralizado de los estados de las actividades y de las entidades de información utilizadas por un usuario ya sea en conexión o en desconexión. El enfoque con el uso de agentes permitiría a un cliente que se desconecta, dejar a un agente en su representación que continuara con el trabajo y con capacidad suficiente para tomar decisiones. Este enfoque es útil en organizaciones que implementan un número grande de procesos ya que permite una nueva distribución de las actividades.

3. *Uso de un enfoque optimista para la desconexión de tareas.* La solución implementada en este trabajo hace uso de un *enfoque pesimista* para la evaluación de la desconexión de un rol. Este enfoque permite asegurar que las actividades y datos que se desconectan no podrán ser utilizados por ningún otro rol, sino hasta que termine la desconexión. El implementar un *enfoque optimista* implica que a pesar de que una entidad de información o actividad no se encuentre disponible, pueda ser desconectada, aún si está siendo utilizada por otro rol. Este enfoque permitiría lograr mayor flexibilidad en cuanto a que los usuarios podrían desconectar roles en cualquier momento. Por otro lado, este enfoque requiere de mecanismos robustos para el manejo en el control de la concurrencia tanto de entidades de información como de estados de las actividades.

4. *Manejo de aplicaciones externas.* La arquitectura de *SysCoor* permite interactuar con algunas de las aplicaciones de *Microsoft Office (Excel y Word)*. El desarrollo realizado para apoyar la operación desconectada no incluyó este aspecto con el fin de mantener la compatibilidad entre diferentes plataformas ya que no todos los dispositivos PDA cuentan con las mismas aplicaciones. Sin embargo el integrar el uso de estos programas como también de sistemas de información, a la ejecución de los sistemas de workflow en un dispositivo PDA, permitiría al usuario realizar un número mayor de tareas, como la edición de documentos y actividades definidas en sistemas de información.

VI.4 Logros adicionales

La realización del presente trabajo generó tres artículos los cuales se listan a continuación:

1. "Supporting Disconnected Workflow in PDA Devices". Presentado en el *International Conference on Information Systems and Engineering 2003 (ISE 2003)* en la ciudad de Montreal Canadá en Julio del 2003 y publicado en memorias de dicho congreso.

2. “An Architecture for Supporting Disconnected Operation in Workflow: A XML-Based Approach”, por presentarse en el *9th International Workshop on Groupware (CRIWG 2003)* y por publicarse en las memorias de este evento en *Lecture Notes of Computer Science*.
3. “Una Arquitectura para el apoyo al Workflow Móvil”. Por presentarse en el taller de cómputo móvil dentro del *Tercer Encuentro Nacional de Ciencias de la Computación (ENC03)* en Septiembre del 2003.

Referencias

Alonso, G., Gunthor, R., Kamath, M., Agrawal, D., El Abbadi, A. y Mohan, C. 1996. "Exotica/FMDC: Handling Disconnected Clients in a Workflow Management System". *Journal of Distributed and Parallel Databases: Special Issue on Databases and Mobile Computing*. 4(3):229-248 p.

Alonso, G., Agrawal, D., El Abbadi, A. y Mohan, C. 1997. "Functionalities and Limitations of Current Workflow Management Systems". Research Report, IBM Almaden Research Center. 20 pp.

Arroyo-Sandoval, P., Martínez-García, A. y Ramírez-Fernández, C. 2003. "An Architecture for Supporting Disconnected Operation in Workflow: A XML-Based Approach". Por publicarse en las memorias de CRIWG'03 en *Lecture Notes of Computer Science*.

Barbara, D., Mehrotra, S., y Rusinkiewicz, M. 1996. "INCAS: A computation model for dynamic workflows in autonomous distributed environments". *Journal for Database Management, Special Issue on Multidatabases*. 7(1): 5-15 p.

Bolcer, G. 2000. "Magi: An Architecture for Mobile and Disconnected Workflow". *IEEE Internet Computing*. 4(3): 46-54 p.

Booch, G., Rumbaugh, J. y Jacobson, I. 1999. "The unified modeling language user guide". Addison-Wesley. 230 pp.

Curtis, B., Kellner, M. y Over, J. 1992. "Process Modeling". *Communications of the ACM*. 35(9):74-89 p.

Deitel, H. y Deitel, P. 1997. "Java How to Program", Prentice -Hall, Englewood Cliffs, N.J. 680 pp.

Fernström, C. y Lennart, F. 1991. "Integration Needs in Process Enacted Enviroments". *Proceedings of the First International Conference on the Software Process. Manufacturing Complex Systems*. 142-158 p.

Fischer, L. 1995. "New Tools for New Times: The Workflow Paradigm". Future Strategies Inc Book Division. Segunda edición. Alameda, CA, USA. 225 pp.

Flores Ríos, B. 2002. "Diseño y desarrollo de una herramienta de soporte para el estudio de procesos organizacionales". Tesis de Maestría en el Centro de Investigación Científica

y de Educación Superior de Ensenada. Departamento de Ciencias de la Computación. Ensenada Baja California México. 219 pp.

García Carrillo, F. 2001. "Desarrollo de una arquitectura de coordinación de procesos organizacionales en Internet". Tesis de Maestría en el Centro de Investigación Científica y de Educación Superior de Ensenada. Departamento de Ciencias de la Computación. Ensenada Baja California México. 171 pp.

García Carrillo, F., Ramírez Fernández, C., Martínez García, A. y Rojas Iñíguez, F. 2002. "An Architecture for Generating Organizational Coordination Systems using Reusable Models in XML". CICESE. Borrador.

Greenberg, S., Boyle M., Laberge J. 1999. "PDAs and Shared Public Displays: Making Personal Information Public, and Public Information Personal". *Personal Technologies*. 3(1):54-64 p.

Greenwood, R.M., Robertson, I., Snowdon, R. A. y Warboys, B.C. 1998. "Active Models in Business". *Proceedings of the 5th Annual Conference on Business Information Technology*. Manchester Metropolitan University. 141-152 p.

Hall, M. 2000. "Core Servlets and JavaServer Pages". Prentice Hall. New Jersey. 575 pp.

Humphrey, W.S. y Feiler, P.H. 1992. "Software Process Development and Enactment: Concepts and Definitions". *Proceedings of the Second International Conference on the Software Process*. 28-40 p.

Kawalek, P. 1998. "A Method for Designing the Software Support for Coordination". Tesis doctoral. Universidad de Manchester, Inglaterra. 197 pp.

Kistler, J. 1993. "Disconnected Operation in a Distributed File System". Tesis doctoral. Departamento de Ciencias de la Computación. Universidad Carnegie Mellon. 240 pp.

Litiu, R. y Prakash, A. 2000. "Developing Adaptive Groupware Applications Using a Mobile Component Framework". *Proceedings of CSCW 2000. ACM Conference on Computer Supported Cooperative Work*. 107-116 p.

Miers, D. 1996. "Use of Tools and Technology within a BPR Initiative in Business Process Reengineering". Coulson-Thomas Colin. 165 pp.

Nixon, P. y Cahill, V. 1998. "Mobile Computing: Technologies for a Disconnected Society". *IEEE Internet Computing*. 4(1): 65-74 p.

Ould, M. 1995. "Business Processes". John Wiley and Sons. West Sussex, Inglaterra. 216 pp.

Pepin, C. y Kriger, C. 2002. "Creating a Lotus Notes application with Domino Everywhere Enterprise". The Technical Resource for Lotus Software. 155 pp.

Plesums, C. 2002. "An Introduction to Workflow". Computer Sciences Corporation Financial Services Group. 512 pp.

Pressman, Roger S. 1997. "*Ingeniería de software*". Editorial McGraw-Hill. 601 pp.

Ramírez Fernández, C. 2002. "Generación de sistemas de coordinación a través de modelos de procesos representados en XML". Tesis de Maestría en el Centro de Investigación Científica y de Educación Superior de Ensenada. Departamento de Ciencias de la Computación. Ensenada Baja California México. 179 pp.

Rojas, F. y Martínez, A. I. 1998. "From Process Modeling to Enactment and Simulation". Proceedings of the 12th European Simulation Multiconference, Manchester, Inglaterra. 844-848 p.

Rumbaugh, J., Jacobson, I. y Booch G. 2000. "El lenguaje unificado de modelado. Manual de referencia". Editorial Pearson educación, S. A. Madrid. 552 pp.

Seungil, L., Dongsoo, H. y Dongman, L. 2001. "Supporting Voluntary Disconnection in WFMSs". Proceedings of the Third International Symposium on Cooperative Database Systems for Advanced Applications (CODAS 2001). Beijing, China. 147- 154 p.

Silver, B. 1999. "Domino Workflow White Paper. Automating Real-World Business Processes". Bruce Silver Associates. Industry Trend Reports. 125 pp.

van der Aalst, W.M.P. 1998. "The Application of Petri Nets to Workflow Management". The Journal of Circuits, Systems and Computers. 8(1):21-26 p.

Warboys, B., Kawalek, P., Robertson, I., y Greenwood, M. 1999. "Business Information Systems: A process approach". McGraw Hill. 262 pp.

Workflow Management Coalition. 1995. "The Workflow Reference Model". Documento TC00-1003. Hampshire, Inglaterra. 55 pp.

Workflow Management Coalition. 1999. "Terminology and Glossary". Documento TC00-1011. Hampshire, Inglaterra. 65 pp.

Workflow Management Coalition. 2002. "Workflow Process Definition Interface – XML process definition Language". Documento WfMC TC-1025. Hampshire, Inglaterra. 87 pp.

Yung, Y. 2002. "Enabling Cost-effective Lightweight Disconnected Workflow for Web-based Teamwork Support". *Journal of Applied Systems Studies*, Cambridge, Inglaterra. 3(2):437-453 p.

Apéndice A. Caso de estudio del “*proceso de ventas*” de la compañía vitivinícola Chateau Camou

A.1 Introducción

La Ingeniería de Procesos otorga una oportunidad potencial para ejercer su práctica en procesos de diversa índole. Como *parte* de sus actividades, se realiza el análisis y evaluación de procesos en búsqueda de encontrar propuestas de mejora y planteamientos para fortalecer aspectos débiles de coordinación y comunicación. Buscando por medio de la Tecnología de Información (TI) se lleven a cabo procesos eficientes y acordes a las necesidades de la organización. Este apéndice presenta un breve estudio del *proceso de ventas* llevado a cabo en la compañía vitivinícola Chateau Camou en la Ciudad de Ensenada Baja California.

El *proceso de ventas* es parte fundamental en la estructura organizacional de la compañía Chateau Camou ya que el servicio y el tiempo de respuesta a las solicitudes de los clientes deben ser los mejores. Las fallas en el servicio son de consecuencias graves para la economía de la empresa.

En este trabajo se aplica la metodología de Ingeniería de Procesos (IP) PADM. El estudio se realiza hasta la fase de propuestas de rediseño y soporte, que respaldado en el modelado, se considera que pueden representar mejoras en proceso.

La documentación de este trabajo está estructurada de la siguiente forma: primeramente la definición del problema y la identificación de sus objetivos. Enseguida un resumen de los agentes involucrados y los roles que desempeñan, la descripción textual del proceso y la gráfica rica de su estructura. La gráfica rica ejemplifica el proceso completo de forma simple y facilita su entendimiento. Se describen también los problemas técnicos y

sociales descubiertos, así como el modelado formal a través de diversos diagramas que proporcionan información representativa de los estados, actividades, roles, flujos de información involucrados. Se concluye con la evaluación del proceso, propuestas a la estructura del proceso, el uso de TI y conclusiones sobre la experiencia de haber realizado este estudio.

A.2 Antecedentes

La compañía vitivinícola Chateau Camou fue establecida en 1994 y se encuentra situada en Baja California, México en el valle llamado "Valle de Guadalupe". El viñedo incluye 37.5 hectáreas de Sauvignon Blanc, Chardonnay, Cabernet Sauvignon, Merlot y de Cabernet Franc, con una producción anual de 15.000 cajas. El plantar se compone de 6.000 plantas por hectárea para las uvas rojas y 5.500 para las uvas blancas. Las vides son de 20 años de edad, con un tercio de los vinos blancos que hacen un promedio total de 60 años.

Los productos de la compañía se agrupan en tres categorías: Chateau Camou, Viñas de Camou y Flor de Guadalupe.

A.3 Objetivos

Dentro de los objetivos que se han planteado para el desarrollo de este estudio se encuentran:

- Seguir la metodología PADM como guía para la definición, captura, evaluación y elaboración de propuestas.
- Documentar apropiadamente el proceso de ingeniería.
- Generar conclusiones sobre la experiencia y resultados obtenidos.

A.4 Captura

Para lograr el entendimiento del *proceso de ventas*, se realizaron una serie de entrevistas con las personas identificadas como clave y que forman parte importante para la realización satisfactoria del mismo. Fueron 4 entrevistas en total, realizadas a los siguientes agentes:

- Gerente de la compañía.
- Encargado de facturación.
- Los 2 vendedores con los que cuenta la compañía.

Además 7 documentos relacionados con el proceso, los cuales son:

- Pedidos.
- Facturas.
- Contrarrecibos.
- Reportes de visitas obligadas.
- Hojas de cata.
- Hojas descriptivas del producto.
- Reportes de actividades.

Con la información obtenida de las entrevistas y los documentos analizados podemos dar una breve descripción del proceso.

A.5.1 Descripción textual del proceso

El proceso inicia cuando un cliente solicita un pedido de mercancía. El pedido se puede llevar a cabo cuando el vendedor visita al cliente o, el cliente puede solicitarlo por medio del teléfono directamente al vendedor o a la encargada de facturación. El siguiente paso es entregar el pedido a la encargada de facturación para que lo procese (en el caso de que no haya sido ella quien atendió el pedido). Cuando el pedido se realiza durante las visitas al cliente, el pedido no se entrega a la encargada de facturación sino hasta el día siguiente. A continuación, la encargada de facturación debe verificar el estado del crédito

del cliente. Si no existe ningún problema con el crédito, el pedido es procesado. Si existe algún problema, se contacta al cliente para llegar a un acuerdo. Una vez que no existe problema con el crédito del cliente, la encargada de facturación en la oficina debe de verificar la mercancía que se establece en el pedido y verificar la existencia de la misma en el almacén. Si no existe suficiente mercancía para surtir el pedido, la encargada de facturación debe de contactar a las bodegas en el Valle de Guadalupe para solicitar mercancía. Cuando se tiene la mercancía en el almacén, la encargada de facturación captura el pedido en el sistema de contabilidad y genera la factura, por lo general esta actividad se lleva a cabo en las mañanas. Es por las mañanas que la encargada de facturación entrega la factura y la mercancía por surtir al cliente a los vendedores quienes son los que entregan la mercancía al cliente. Una vez entregada la mercancía, los vendedores reciben de la encargada de facturación e un comprobante de la mercancía entregada. Así mismo, la encargada de facturación entrega un listado de las facturas las cuales deben cobradas. El vendedor durante las visitas a los clientes debe de levantar pedidos, entregar mercancía, entregar facturas y recibir pagos, estos pagos son reportados en el reporte de pagos. Además, debe llenar un reporte llamado reporte de visitas, en el que describe las actividades realizadas con cada uno de los clientes. A la mañana siguiente, el vendedor entrega a la encargada de facturación, el reporte de los pagos con el efectivo o cheques recibidos por parte del cliente, los pedidos nuevos, las facturas entregadas y el reporte de visitas. La encargada de facturación entonces, debe de procesar los pagos recibidos y archivar los reportes de visitas.

Después de concluir con las entrevistas, nos dimos a la tarea de modelar el proceso anteriormente descrito, para ello utilizamos una *gráfica rica*; esta nos permite representar partes del proceso a diferentes niveles de detalle, también permite visualizar a las personas involucradas en el proceso, su propósito, sus deseos y miedos (denotados por burbujas de pensamientos). El concepto de *gráfica rica* fue desarrollado por Peter Checkland como parte de la Metodología de Sistemas Suaves (*Soft Systems Methodology –SSM*) [Checkland y Sholes, 1990].

Con esta información podemos realizar la gráfica rica del proceso que nos permite tener una visión global del mismo.

A.5.2 Gráfica rica de la estructura del proceso

Durante la primera fase de descripción y definición de un proceso, es recomendable generar una gráfica rica del mismo, en ella se incluyen los agentes participantes, los flujos de comunicación y datos, así como todos los factores que interactúan utilizando elementos gráficos de uso común.

En el caso de *proceso de ventas* resaltan las interacciones de múltiples agentes (generadores de información, capturistas, quienes procesan la información obtenida, clientes internos y externos), cargas fuertes de trabajo sobre ciertos agentes, así como la cantidad de formatos que se utilizan.

La figura 43 muestra la gráfica rica de la estructura del *proceso de ventas*. Es evidente la abundancia de actividades para un agente en particular: el *vendedor*.

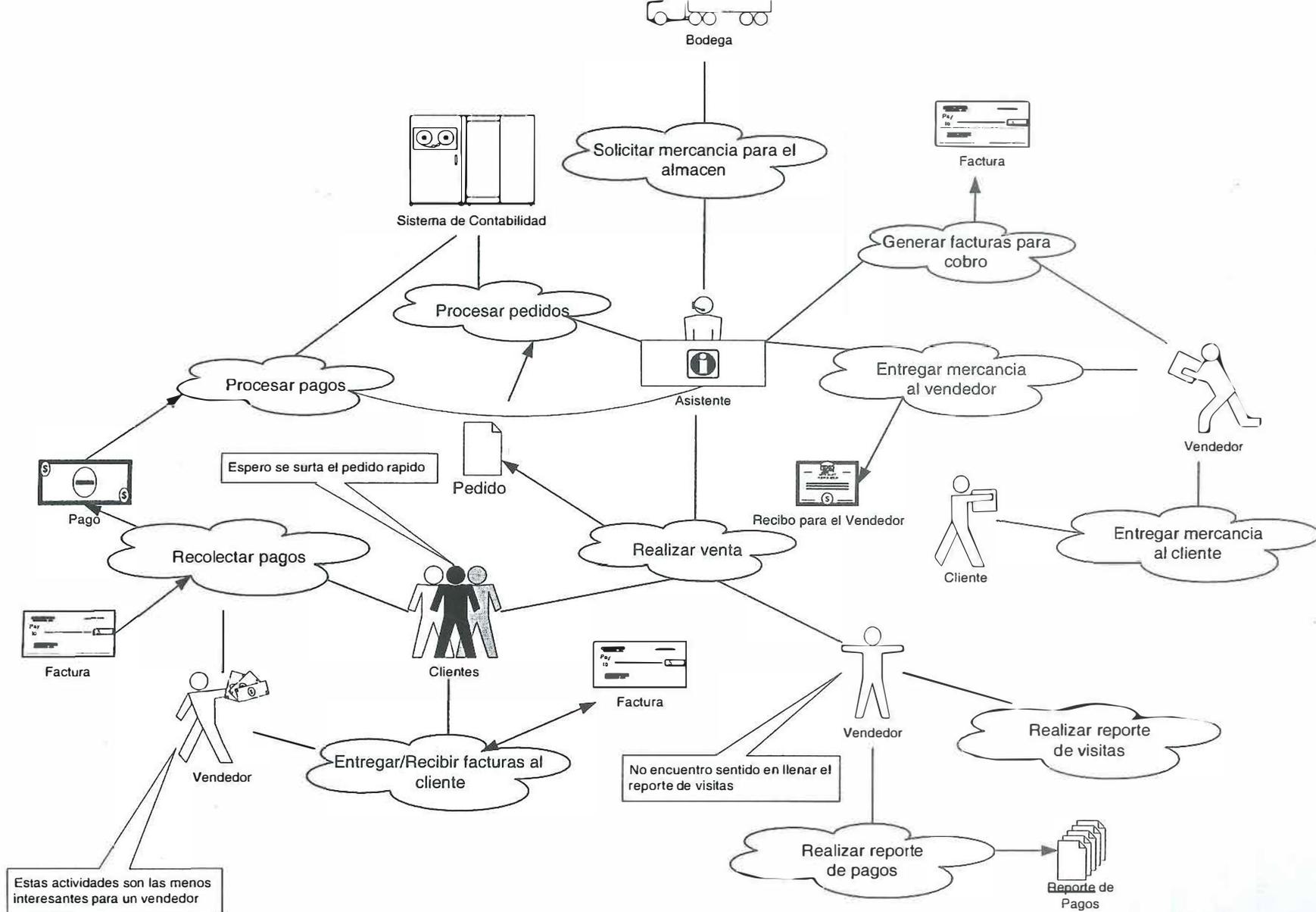


Figura 43. Gráfica rica de la estructura del proceso de ventas

A.6 Aspectos sociales y técnicos

Durante la captura del proceso se empiezan a detectar aspectos problemáticos o áreas que necesitan cierto grado de atención para mejorar la eficiencia del proceso. Estos aspectos pueden ser clasificados en sociales y técnicos. Los aspectos sociales son aquellos relacionados con factores humanos como son la comunicación, coordinación, cultura organizacional, etc. Por otro lado los aspectos técnicos reflejan la problemática relacionada con la manera en que se ejecuta el proceso y la TI que se utiliza como soporte al mismo.

Aspectos tales como: actividades redundantes, duplicidad de información, procedimientos y políticas no definidas, al uso de la TI (falta de uso o abuso de la misma), etc. A continuación se presentan algunos de los aspectos sociales y técnicos identificados durante la fase de captura.

A.6.1 Sociales

- a) Los *vendedores* reconocen que la empresa no los apoya en sus necesidades de tecnología para llevar a cabo su trabajo.
- b) Los *vendedores* consideran que las actividades de entrega, cobranza y pedidos son rutinarios.
- c) Los *vendedores* ven en actividades como la presentación de nuevos productos y degustaciones, satisfacciones personales como la oportunidad para mostrar sus conocimientos.

A.6.2 Técnicos

- a) Toda la información del proceso es llenada en formas de papel de muchos tipos a través de una máquina de escribir o simplemente con letra de molde.
- b) Cuando hace falta algún tipo de formato, se utilizan hojas en blanco.
- c) Solo la encargada de facturación tiene acceso al sistema de ventas.
- d) Los *vendedores* no cuentan con ningún apoyo tecnológico para realizar presentaciones de los productos de la compañía a los clientes.

A.7 Modelado de Procesos

Con el objeto de facilitar el entendimiento del proceso, de apoyar su definición a través de técnicas diagramáticas que permitan analizar objetivamente y visualizar los puntos débiles, se presentan en esta sección el catálogo de usuarios y una serie de diagramas que permitan realizar el modelo aplicando las perspectivas de modelado. Con los diferentes diagramas elaborados se describe qué elementos del proceso se llevan a cabo y sus flujos de información, cuándo se llevan a cabo, dónde y por qué agentes se desarrollan, la comunicación entre ellos y las entidades de información involucradas. Esta actividad es la base para las propuestas de mejora y propuestas de soporte técnico resultado de este trabajo.

A.7.1 Catálogo de usuarios

A continuación se listan los usuarios del proceso entrevistados en la fase de captura de este proyecto.

Favela Vara

Gerente de la compañía

Gerencia

Carmen

Encargada de facturación

Oficina de ventas

Manuel

Vendedor

Oficina de ventas

Jesús

Vendedor

Oficina de ventas

A.7.3 Modelo del sistema

En el modelo del sistema se describen los agentes que participan en el proceso y se describen las interacciones entre ellos. Es un diagrama simple que refleja la visión global del proceso. En la Figura 44 se encuentra el Modelo del Sistema del proceso de ventas, como puede observarse que los agentes tienen interacciones sencillas entre cada uno de ellos.

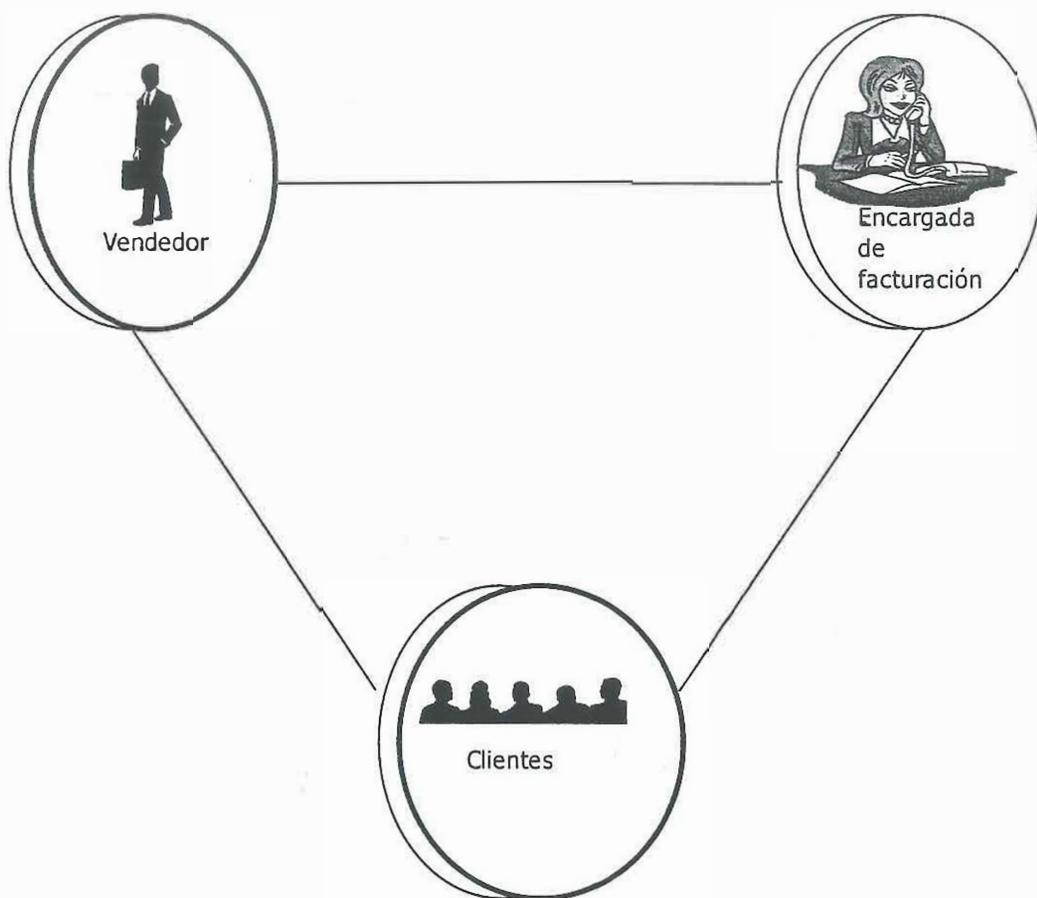


Figura 44. Modelo del sistema del proceso ventas

A.7.4 Modelo de los objetivos

Los modelos de los objetivos se aplican para definir de forma particular sobre un objetivo, los agentes que interactúan, el objetivo que cada agente cumple con su participación y la multiplicidad de sus interacciones. La Figura 45 presenta el modelo de los objetivos del proceso de ventas.



Figura 45. Modelo de los objetivos del proceso de ventas

A.7.5 Diagramas de rol actividad

Los Diagramas de Rol y Actividad (RAD) [Miers, 1996, Ould, 1995] son considerados como los más productivos y útiles en el proceso de modelado de procesos, presentan un método para capturar los aspectos más relevantes de un proceso: actividades, orden, decisión, roles, responsabilidades, interacciones, objetivos, reglas. A continuación se presentan los diagramas RAD realizados. La parte en la que nos enfocamos fue en la interacción entre la encargada de facturación, el *vendedor* y el *cliente*, y sus actividades correspondientes.

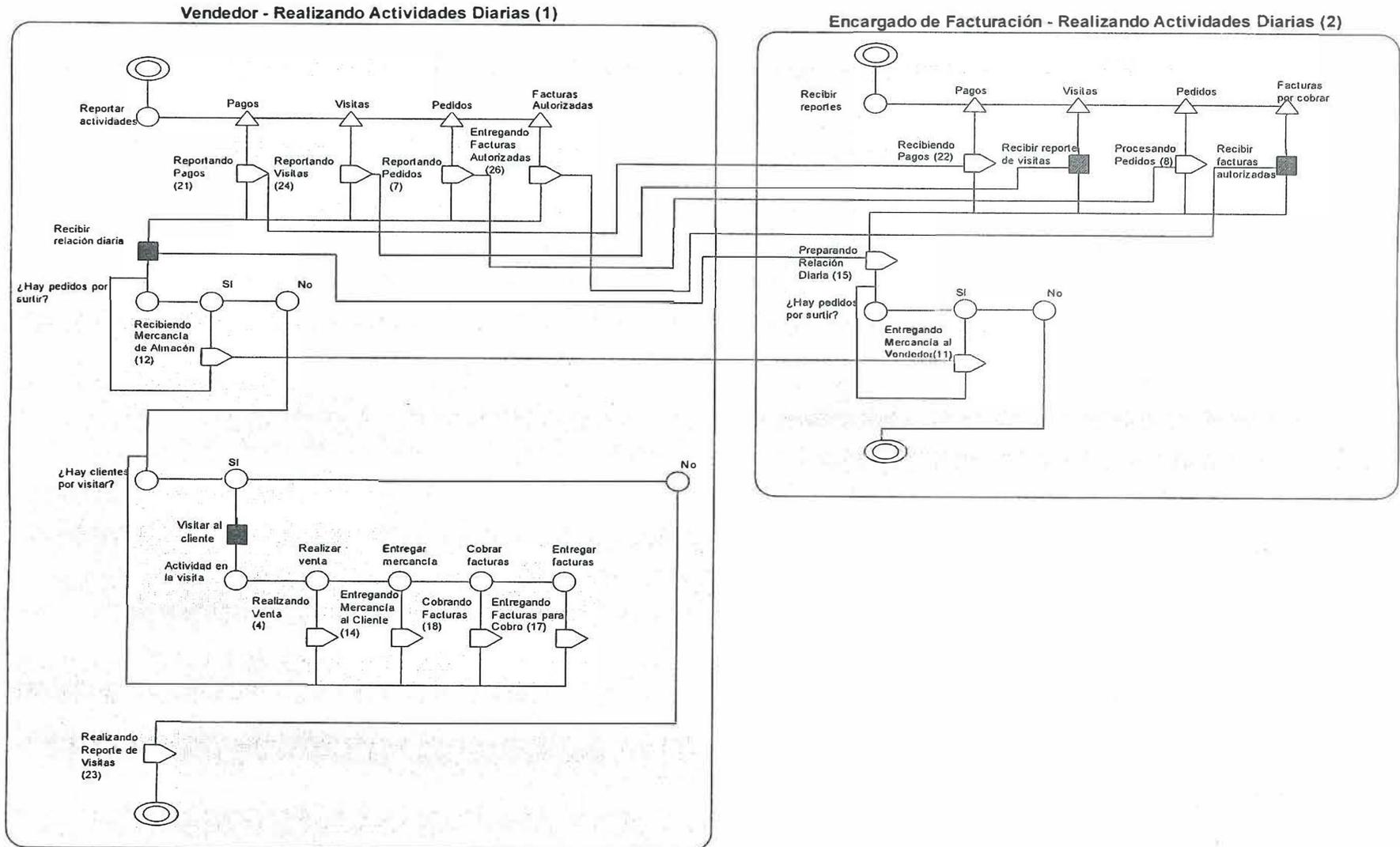


Figura 46. Diagramas RAD globales de las actividades del *vendedor* y del *encargado de facturación*

Cliente - Solicitando Mercancia (3)

Vendedor - Realizando Venta (4)

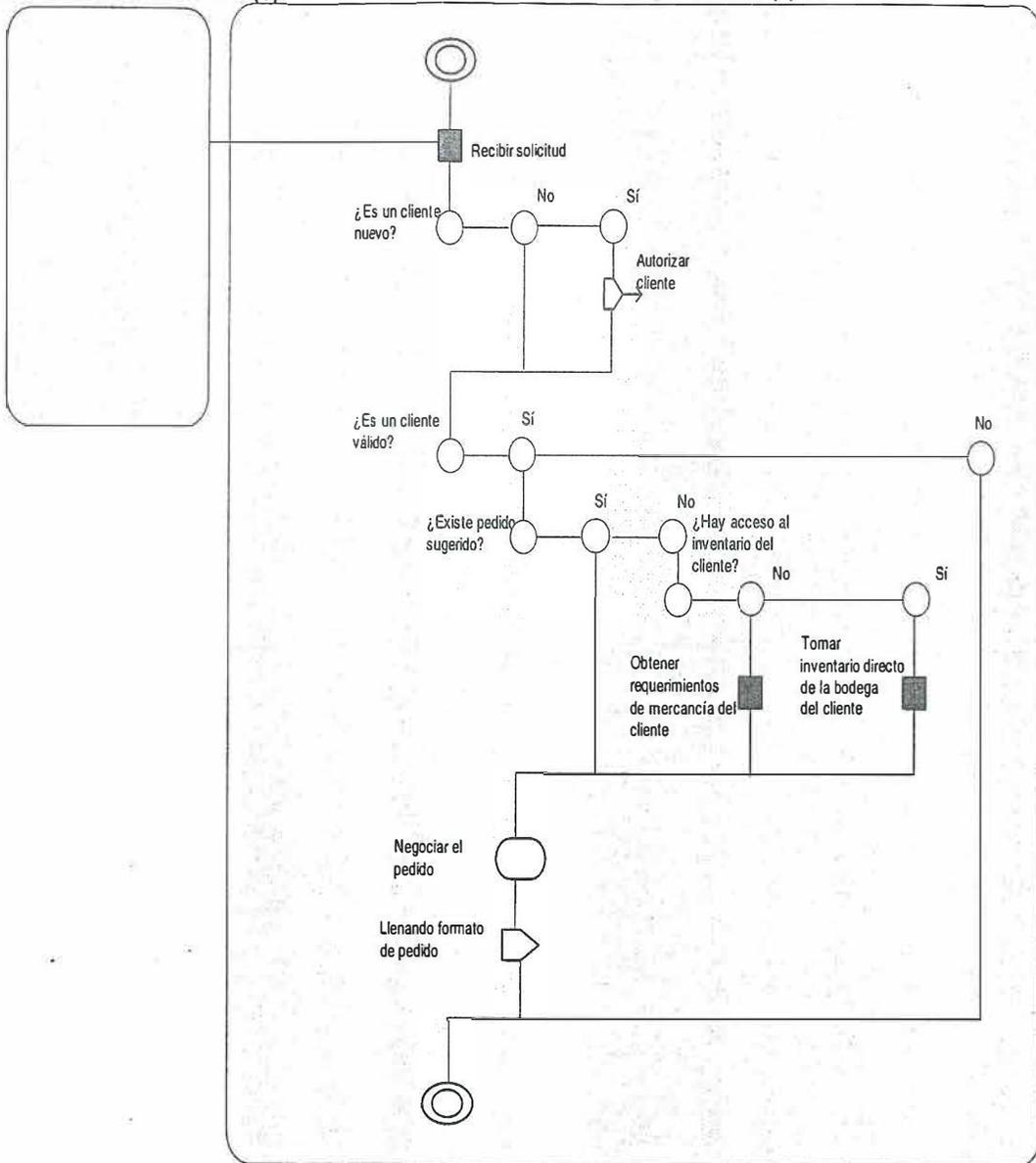


Figura 47. Diagrama RAD del subproceso realizando la venta

Vendedor - Llenando Formato de Pedido (5) Encargado de Facturación- Generando Factura (9)

Cliente - Autorizando Pedido (27)

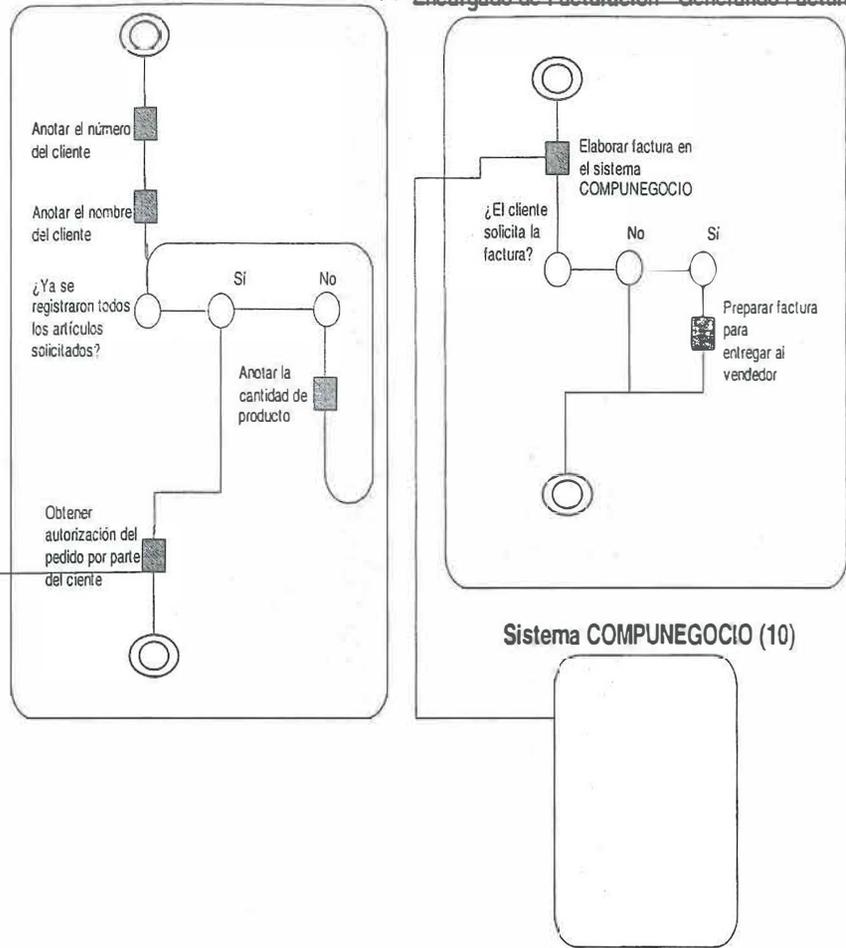


Figura 48. Diagramas RAD que forman el subproceso *llenando formato de pedido*

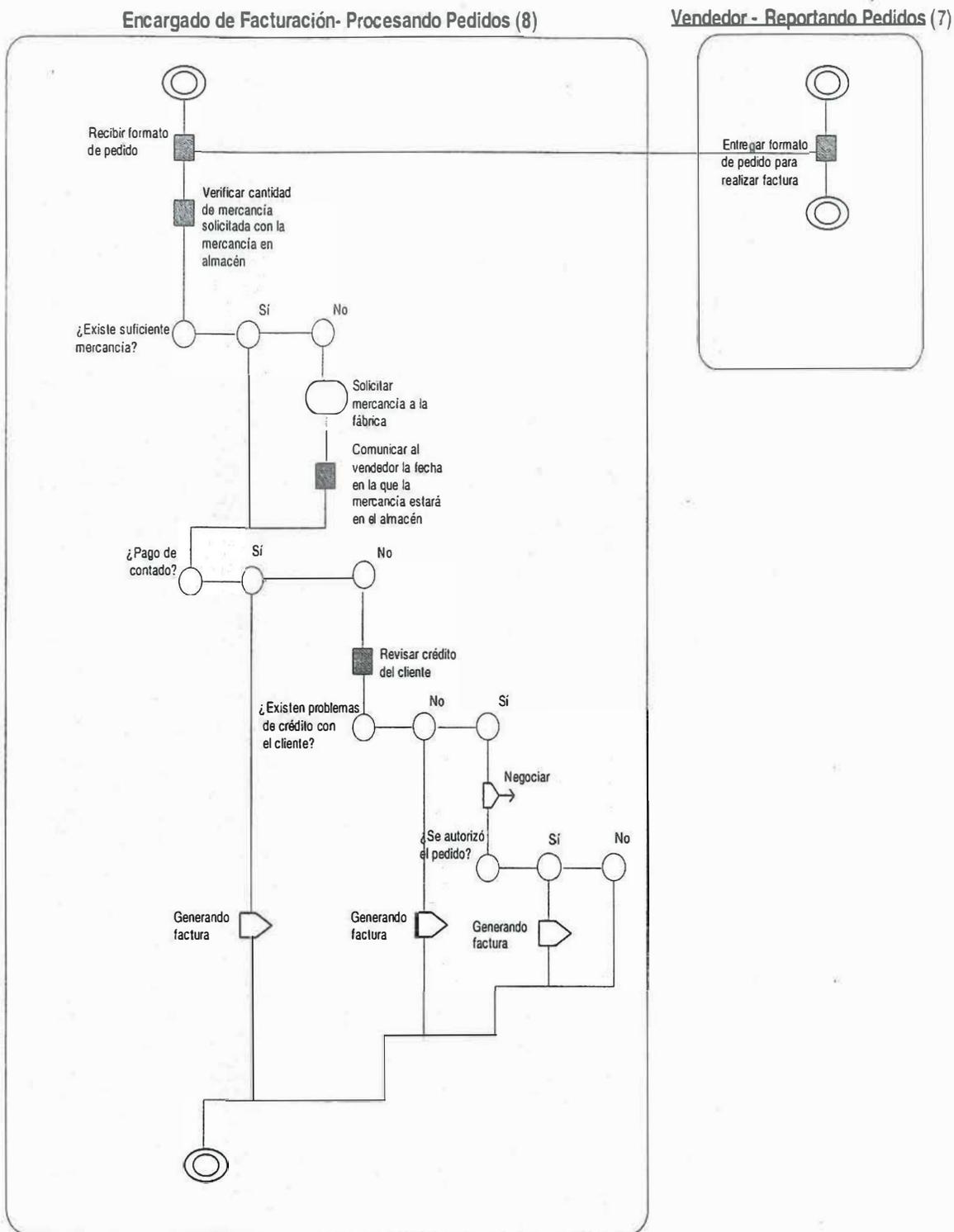
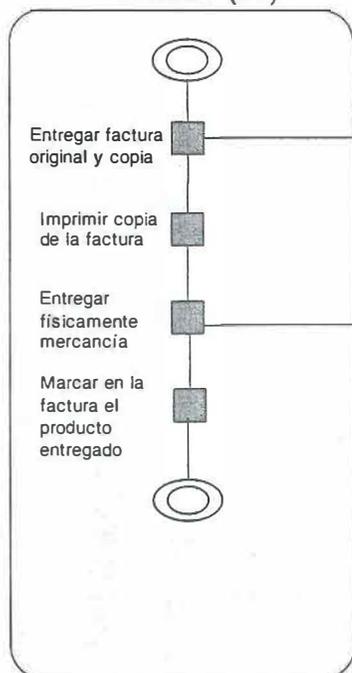


Figura 49. Diagramas RAD del subproceso procesar pedido

Encargado de Facturación- Entregando Mercancía al Vendedor (11)



Vendedor - Recibiendo Mercancía del Almacén (12)

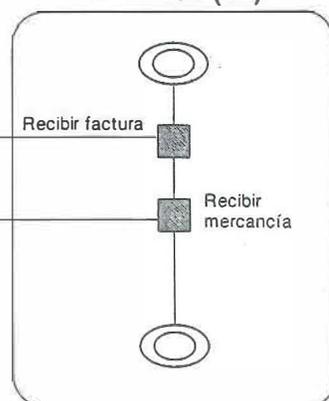
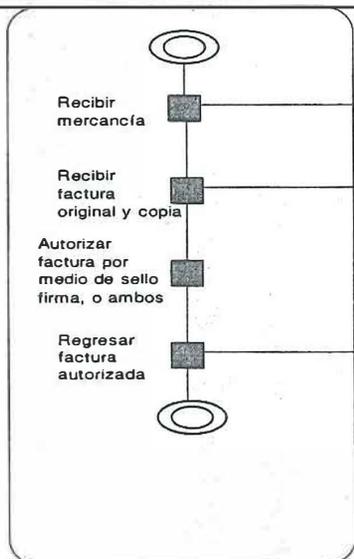


Figura 50. Diagramas RAD del subproceso de recibir y entregar mercancía entre *vendedor* y el *encargado de facturación*

Cliente - Recibiendo Mercancía (13)



Vendedor - Entregando Mercancía al Cliente(14)

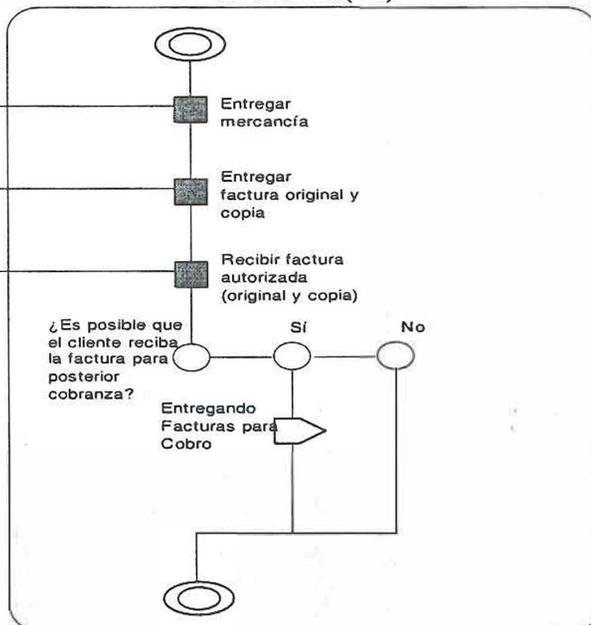


Figura 51. Diagramas RAD del subproceso de recibir y entregar mercancía entre *vendedor* y *cliente*

Facturación - Preparando Relación Diaria (15)

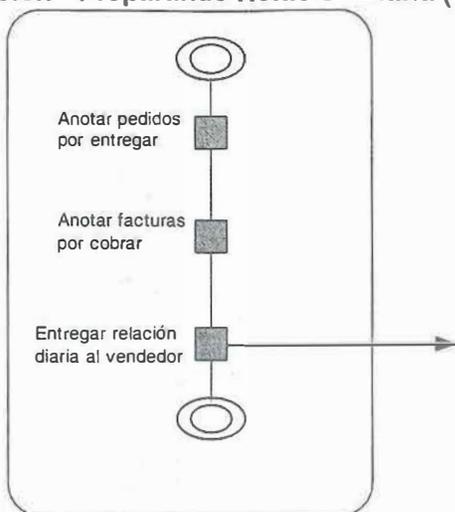
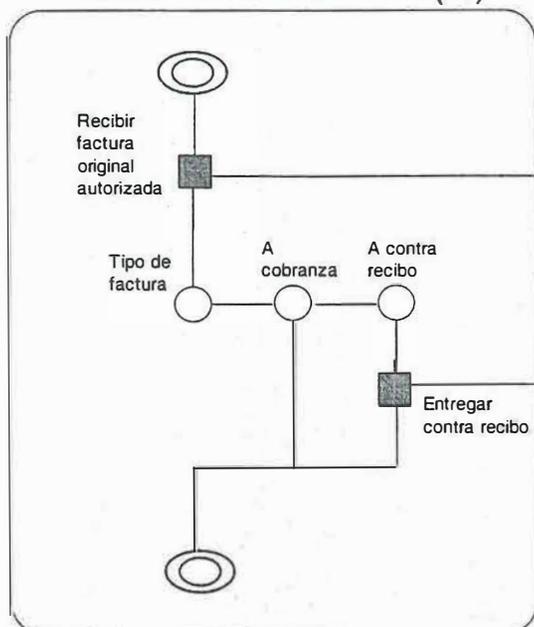


Figura 52. El *vendedor* preparando la relación diaria

Cliente - Atendiendo Visita (16)



Vendedor - Entregando Facturas para Cobro (17)

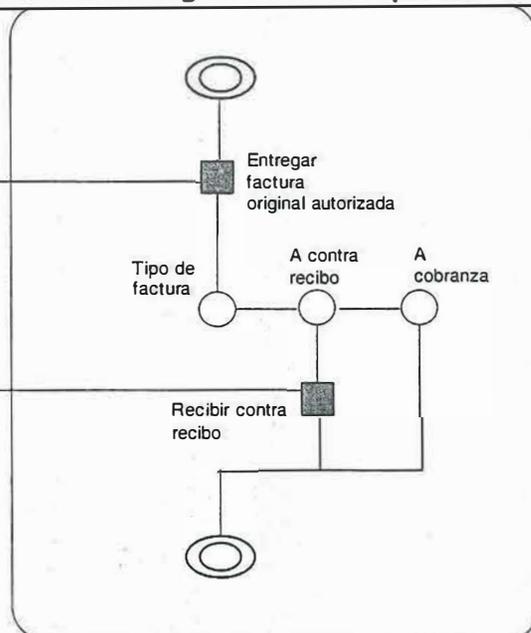


Figura 53. Diagrama RAD del subproceso *Entregando Facturas para Cobro*

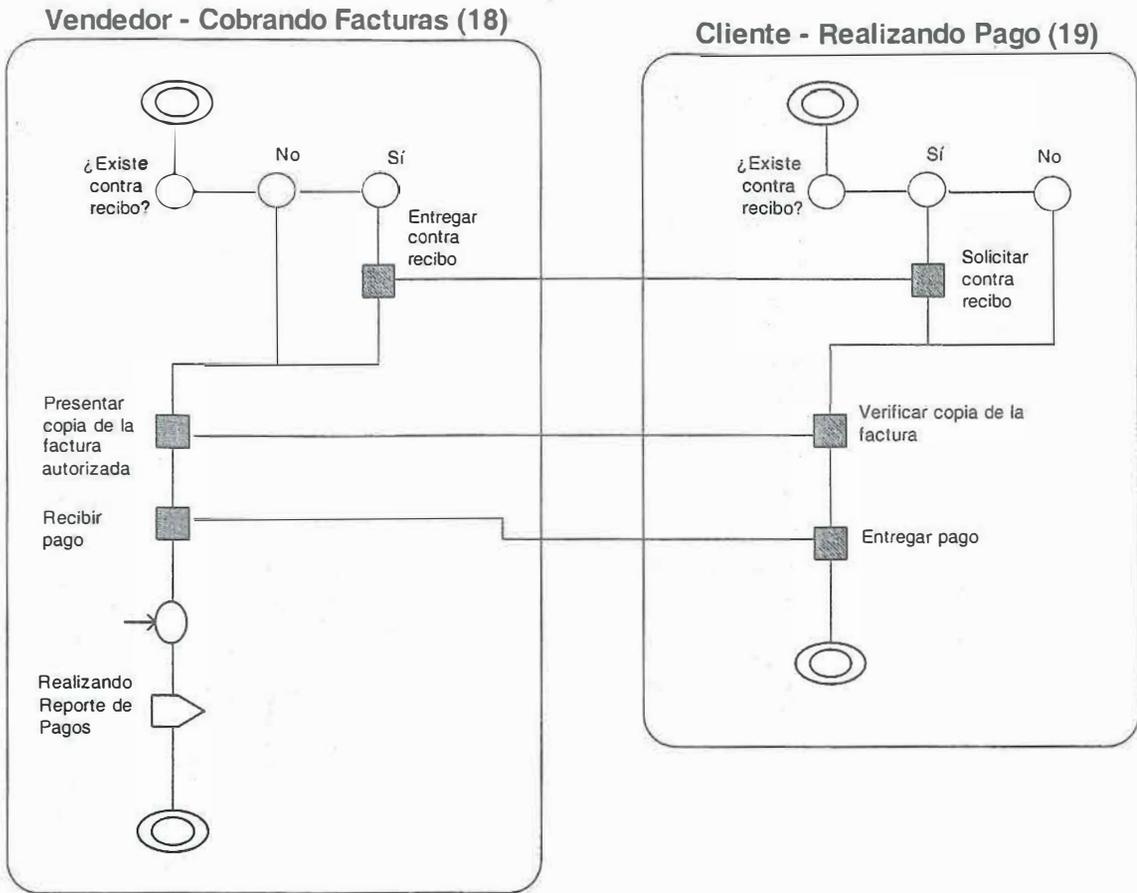


Figura 54. Diagramas RAD del subproceso de cobrar facturas

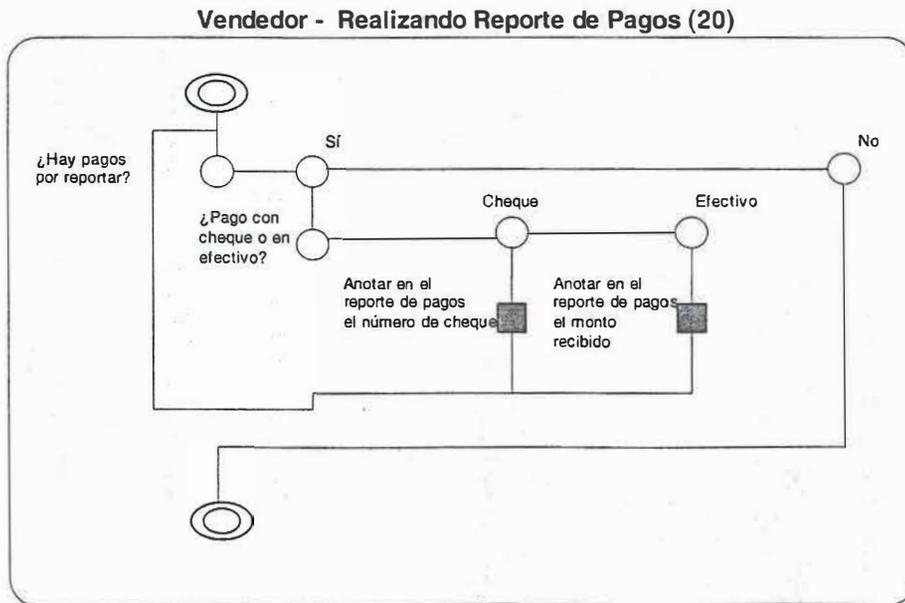


Figura 55. El *vendedor* realizando reporte de pagos

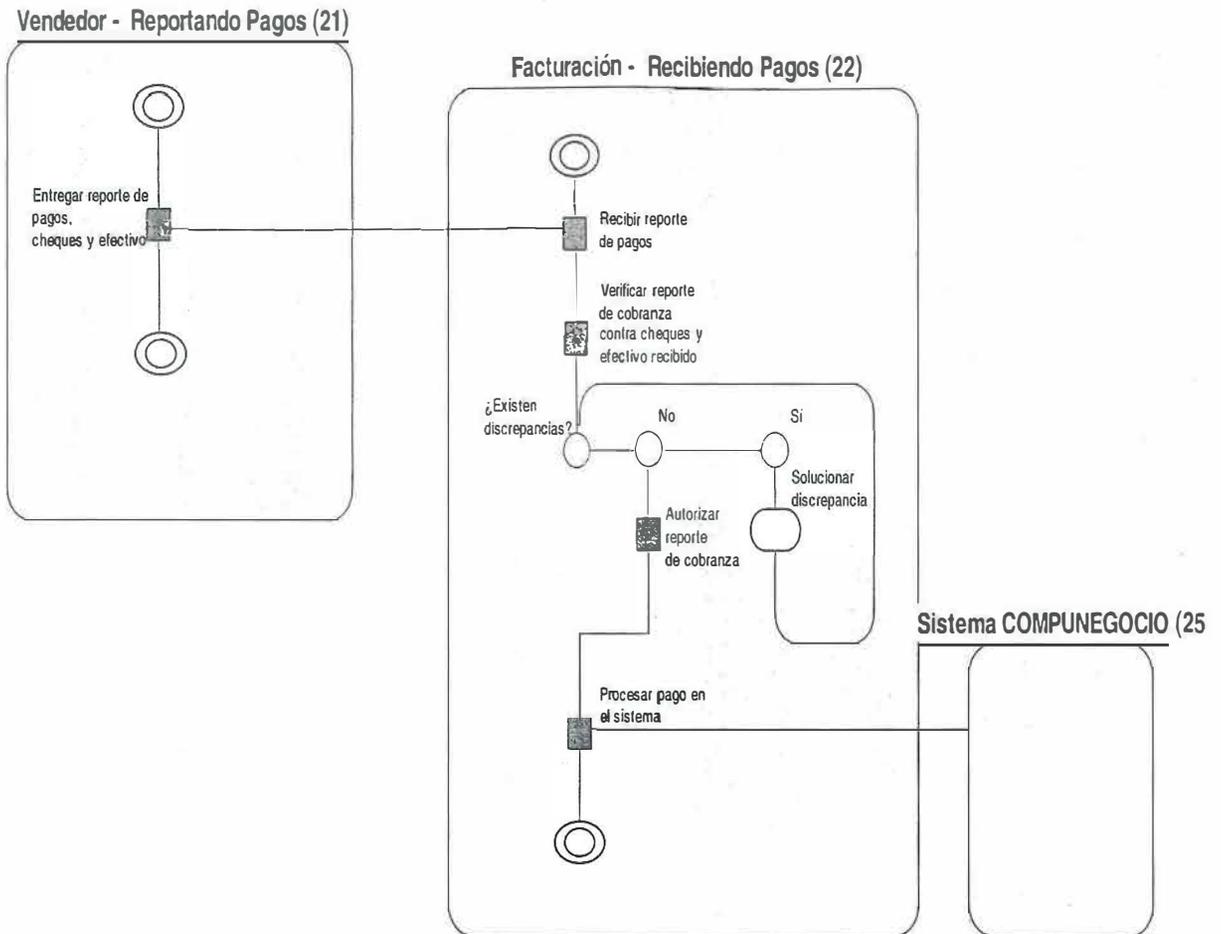
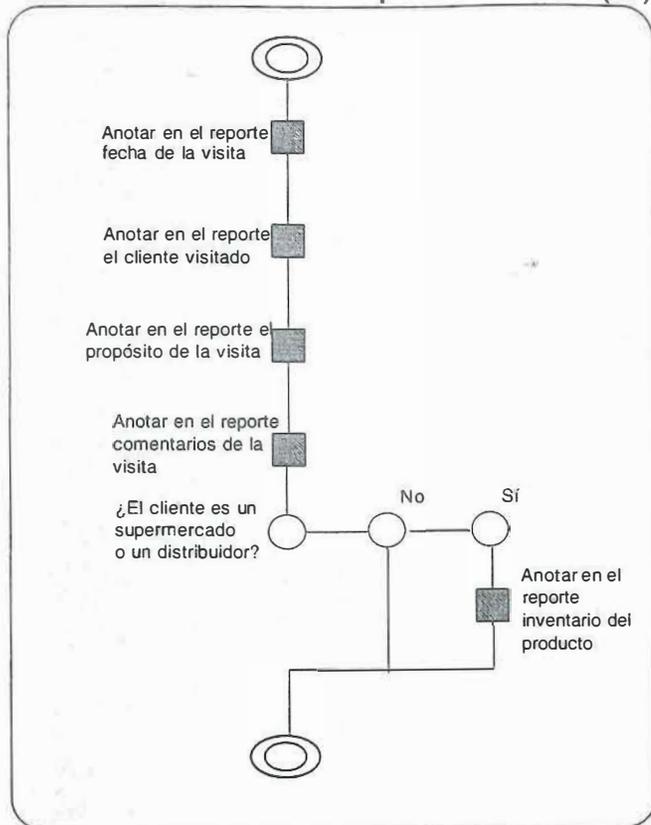


Figura 56. Diagramas RAD del subproceso *recibiendo pagos*

Vendedor - Realizando Reporte de Visitas (23)



Vendedor - Reportando Visitas (24)



Figura 57. Diagramas RAD del *vendedor* realizando el reporte de visitas

Vendedor - Entregando Facturas Autorizadas (26)

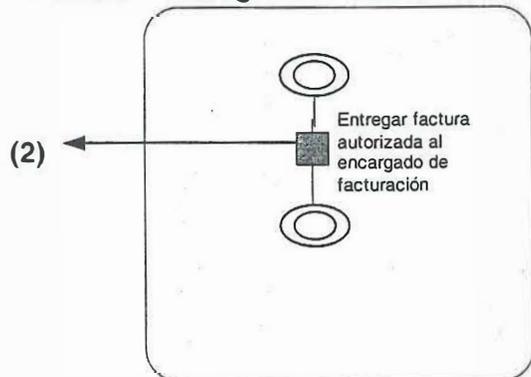


Figura 58. El *vendedor* entregando facturas autorizadas

Apéndice B. Modelos de procesos en XML

B.1 Introducción

En este apéndice, se explican los elementos que conforman a los DTDs responsables de validar a los modelos de procesos en XML en RAD, DTE, entidades y de *sistemas de workflow* completos.

B.2 Elementos del DTD del *modelo base de transición de estados (modelo base DTE)*

El DTD que valida al *modelo base de transición de estados* se presenta en la figura 59, esta estructurado por el elemento principal *estados* que esta formado por uno o más *rol_estados*. Cada *rol_estados* tiene un *nombre* y uno o más *estado*. Un *estado* lo forma un *evento* que tiene cero o una *acción* y una *transición* de estado. El *estado* tiene un identificador (*id*) y nombre (*nombre*) que lo hacen diferente de los demás. De igual forma, un *evento* tiene asociado un identificador (*act_id*) relacionado al identificador de una actividad (comparándolo con el *modelo base en RAD*), el nombre del evento (*nom_act* o *nom_resp*, comparándolo con el *modelo base en RAD*, dependiendo si ocurre una actividad simple, condicional o paralela). Una *acción* esta formada por el identificador de origen de la entidad a enviar y la correspondiente a la de destino, así como el identificador de la actividad y agente receptor.

```

<?xml version='1.0' encoding="UTF-8"?>
<!--Definiendo el evento que ocasiona que se realice una transicion de estado -->
<!ELEMENT evento (accion?,transicion)>
<!ATTLIST evento act_id (CDATA) #REQUIRED
                 nom_act (CDATA) #IMPLIED <!--Realizar una actividad en el RAD-->
                 nom_resp (CDATA) #IMPLIED <!--Realizar una respuesta-->
<!ELEMENT accion EMPTY > <!-- indica cuando hay inter. con info.-->
<!ATTLIST accion Oclave_info (CDATA) #REQUIRED
                 Dclave_info (CDATA) #IMPLIED
                 cual (CDATA) #REQUIRED
                 agente (CDATA) #REQUIRED> <!--id agente. Corresponde a una
interacción entre roles en el RAD -->
<!ELEMENT transicion EMPTY >
<!ATTLIST transicion id_sig_edo (CDATA) #REQUIRED>
<!--Definicion de los elementos de cada estado -->
<!ELEMENT estado (evento*)>
<!ATTLIST estado id (CDATA) #REQUIRED
                 nom (CDATA) #REQUIRED>
<!--Elemento raiz -->
<!ELEMENT estados (rol_estados+)
<!ELEMENT rol_estados (nombre,estado+)
<!ELEMENT nombre (PCDATA) <!-- nombre_diagrama_estados por rol-->

```

Figura 59. Documento base de transición de estados

B.3 Elementos del DTD del *modelo base de información (modelo base EI)*

El DTD que valida al modelo base de información presenta en la figura 60. El elemento principal es entidad_informacion formado por cero o más rol_informacion. Un rol_informacion tiene un nombre y cero o más documentos (docto) o datos (dato). Los elementos docto y dato están asociados al elemento identificador de la actividad (act_id) en la que se utilice de acuerdo al modelo base en RAD. El elemento docto también con los atributos: clave, nombre (nom) y un si o no si dependiendo de la existencia de dicho documento (en existente). Además de lo anterior, después de especificar el final de la definición del elemento docto, deberá declararse la aplicación con la cual se podrá editar dicho documento (docto) o si es que se encuentra en un sistema de información (SI). La aplicación se define con un atributo con el que se clasifica como procesador de texto (PT), hoja de cálculo (HC) u hoja de presentaciones (HP). Por su parte el elemento dato, cuenta con su: clave, nombre (nom), tipo (sólo: entero, flotante, texto o double), valor por defecto (valor_defecto), valor mínimo (valor_minimo), y la operación a realizar cuando se esté enviando dicho dato en una interacción con otro rol (suma, resta, división, multiplicación o

ninguna). En el elemento SI, deben especificarse: el lugar, usuario y clave del usuario (clave_usuario) origen de donde se tomará la información.

```

<?xml version='1.0' encoding="UTF-8"?>

<!--Definiendo el tipo de informacion relacionada a una actividad -->

<!ENTITY % informacion "(docto|dato)+">

<!ELEMENT aplicacion (EMPTY)>
<!ATTLIST aplicacion tipo (PT|HC|HP) #REQUIRED>

<!-- Tipo documento y la aplicacion relacionada -->
<!ENTITY % aplicacion "(apli_tipo|SI)">

<!ELEMENT SI EMPTY>
<!ATTLIST SI lugar (CDATA) #REQUIRED
           usuario (CDATA) #REQUIRED
           clave_usuario (CDATA) #REQUIRED>
<!ELEMENT docto (%aplicacion:)>
<!ATTLIST docto act_id (CDATA) #REQUIRED
           clave (CDATA) #REQUIRED
           nom (CDATA) #REQUIRED
           existente (si|no) #REQUIRED>

<!--Tipo dato y la operacion relacionada -->
<!ELEMENT dato EMPTY>
<!ATTLIST dato act_id (CDATA) #REQUIRED
           clave (CDATA) #REQUIRED
           nom (CDATA) #REQUIRED
           tipo (entero|floatante|texto|double) #REQUIRED
           valor_defecto (CDATA) #REQUIRED
           valor_minimo (CDATA) #IMPLIED
           oper_tipo (suma|resta|division|multiplicacion|ninguna) #REQUIRED>

<!--Elemento raíz -->
<!ENTITY % informacion "(docto|dato)+">

<!--Elemento raíz -->
<!ELEMENT entidad_informacion (rol_informacion+)>
<!ELEMENT rol_informacion (nombre,%informacion:)>
<!ELEMENT nombre (PCDATA)>

```

Figura 60. Documento base de información

B.4 Elementos del DTD del *modelo base de workflow (modelo base WF)*

El DTD que valida a los *modelos base del sistema de workflow* se encuentra en la figura 61. Está formado por el elemento raíz llamado *roles* que tiene uno o más *rol_coordinacion*. Cada *rol_coordinacion* cuenta con su *nombre*, uno o más *agente* que lo desempeñan y uno o más elemento *estado*. Este último, tiene definidos una actividad simple (*act*), *condicional* o *paralela*, así como cero o más información (*dato* o *docto*). El

resto de la definición de dichos elementos contenidos en *estado*, ha sido realizada en la sección B2 y B3 de éste apéndice.

Los atributos del elemento estado *plantilla* y *escenario*, son los que se agregan en este DTD. De acuerdo al número que el *modelador* especifique en éstos, es la interfaz y el orden que le corresponde en el *sistema de workflow* a generar.

```

<?xml version='1.0' encoding ="UTF-8"?>
<!--Declaracion de las entidades de información -->
<!ELEMENT aplicacion (EMPTY)>
<!ATTLIST aplicacion tipo (PT|HC|HP) #REQUIRED>

<!ENTITY % aplicacion "(apli_tipo|SI)">
...
<!ELEMENT docto (%aplicacion;)>
<!ATTLIST docto act_id (CDATA) #REQUIRED
               clave (CDATA) #REQUIRED
               nom (CDATA) #REQUIRED
               existente (si|no) #REQUIRED>
<!ELEMENT dato EMPTY>
<!ATTLIST dato act_id (CDATA) #REQUIRED
               clave (CDATA) #REQUIRED
               nom (CDATA) #REQUIRED>
<!ENTITY % informacion "(docto|dato)+">

<!--Declaracion del comportamiento -->
<!--Definiendo el evento que ocasiona que se realice una transicion de estado -->
<!ELEMENT evento (accion?,transicion)>
<!ATTLIST evento act_id (CDATA) #REQUIRED
               nom_act (CDATA) #IMPLIED
               nom_resp (CDATA) #IMPLIED> <!--Realizar una respuesta-->
<!ELEMENT accion EMPTY > <!-- indica cuando hay inter. con envio de informacion-->
<!ATTLIST accion Oclave_info (CDATA) #REQUIRED
               Dclave_info (CDATA) #IMPLIED
               cual (CDATA) #REQUIRED
               agente (CDATA) #REQUIRED><!--id agente. Corresponde a una interaccion entre roles en el RAD -->
<!ELEMENT transicion EMPTY >
<!ATTLIST transicion id_sig_edo (CDATA) #REQUIRED>

<!ENTITY % respuesta "(resp,evento)*">
<!ELEMENT resp (#PCDATA)>
<!ATTLIST resp op (CDATA) #REQUIRED>

<!ELEMENT act (iral?|terminador*|interaccion*|evento*)>
<!ATTLIST act id (CDATA) #REQUIRED nom (PCDATA) #REQUIRED>
<!ELEMENT interaccion (#PCDATA)>
<!ATTLIST interaccion cual CDATA #REQUIRED
               tipo (E|R) #REQUIRED>
<!ELEMENT iral (#PCDATA)>
<!ELEMENT terminador (#PCDATA)>
<!ATTLIST terminador desc CDATA #REQUIRED>

<!ENTITY % alterna "(condicional |paralela )">
<!ELEMENT condicional
  (%respuesta;,( %alterna; | %respuesta;)+)>
<!ATTLIST condicional que (CDATA) #REQUIRED>
<!ELEMENT paralela
  (%respuesta;,( %alterna; | %respuesta;)+)>
<!ATTLIST paralela que (CDATA) #REQUIRED>

<!--Definicion de los elementos de cada estado -->
<!ELEMENT estado (act* |(%alterna; %informacion;)*)>
<!ATTLIST estado id (CDATA) #REQUIRED
               nom (CDATA) #REQUIRED
               platilla (CDATA) #REQUIRED
               escenario (CDATA) #REQUIRED> <!--Numero de escenario en el proceso -->

<!ENTITY % elemento "(estado+)">

<!ELEMENT roles (rol_coordinacion+)>
<!ELEMENT rol_coordinacion (nombre,agente+,(%elemento;)+)>
<!ELEMENT nombre (#PCDATA)>
<!ELEMENT agente (#PCDATA)>
<!ATTLIST agente quien (CDATA) #REQUIRED>

```

Figura 61. Documento base del workflow

B.5 Elementos del DTD del *modelo base de workflow desconectado (modelo base WFD)*

El DTD que valida a los *modelos base del sistema de workflow desconectado* se encuentra en la figura 62. Está formado por el elemento raíz llamado *roles* que tiene uno o más *rol_coordinacion*. Cada *rol_coordinacion* cuenta con su *nombre*, uno o más *agente* que lo desempeñan y uno o más elemento *estado*. Este último, tiene definidos una actividad simple (*act*), *condicional* o *paralela*, así como cero o más información (*dato* o *docto*). El resto de la definición de dichos elementos contenidos en *estado*, ha sido realizada en la sección B2 y B3 de éste apéndice.

Los atributos del elemento estado *plantilla* y *escenario*, son los que se agregan en este DTD. La principal diferencia de este DTD con el del *modelo base WF* es que el modelo base WFD es un modelo auto contenido, es decir, en lugar de solo mostrar las referencias a los datos y a las pantallas usadas durante la ejecución del workflow, agrega la definición explícita de estos dos elementos.

```

<?xml version='1.0' encoding="UTF-8"?>
<!--Declaracion de las entidades de información -->
<!ELEMENT aplicacion (EMPTY)>
<!ATTLIST aplicacion tipo (PT|HC|HP) #REQUIRED>

<!ENTITY % aplicacion "(apli_tipo|SI)">
***
<!ELEMENT docto (%aplicacion;)>
<!ATTLIST docto act_id (CDATA) #REQUIRED
               clave (CDATA) #REQUIRED
               nom (CDATA) #REQUIRED
               existente (si|no) #REQUIRED>
<!ELEMENT dato EMPTY>
<!ATTLIST dato act_id (CDATA) #REQUIRED
               clave (CDATA) #REQUIRED
               nom (CDATA) #REQUIRED>
<!ENTITY % informacion "(docto|dato)+">

<!--Declaracion del comportamiento -->
<!--Definiendo el evento que ocasiona que se realice una transicion de estado -->
<!ELEMENT evento (accion?,transicion)>
<!ATTLIST evento act_id (CDATA) #REQUIRED
                 nom_act (CDATA) #IMPLIED
                 nom_resp (CDATA) #IMPLIED> <!--Realizar una respuesta-->
<!ELEMENT accion EMPTY > <!-- indica cuando hay inter. con envio de informacion-->
<!ATTLIST accion Oclave_info (CDATA) #REQUIRED
                 Dclave_info (CDATA) #IMPLIED
                 cual (CDATA) #REQUIRED
                 agente (CDATA) #REQUIRED> <!--id agente. Corresponde a una interacciOn entre roles en el RAD -->
<!ELEMENT transicion EMPTY >
<!ATTLIST transicion id_sig_edo (CDATA) #REQUIRED>

<!ENTITY % respuesta "(resp,evento)">
<!ELEMENT resp (#PCDATA)>
<!ATTLIST resp op (CDATA) #REQUIRED>

<!ELEMENT act (iral?|terminador*|interaccion*|evento*)>
<!ATTLIST act id (CDATA) #REQUIRED nom (PCDATA) #REQUIRED>
<!ELEMENT interaccion (#PCDATA)>
<!ATTLIST interaccion cual CDATA #REQUIRED
                       tipo (E|R) #REQUIRED>
<!ELEMENT iral (#PCDATA)>
<!ELEMENT terminador (#PCDATA)>
<!ATTLIST terminador desc CDATA #REQUIRED>

<!ENTITY % alterna "(condicional |paralela )" >
<!ELEMENT condicional
  (%respuesta;,( %alterna; | %respuesta;)+)>
<!ATTLIST condicional que (CDATA) #REQUIRED>
<!ELEMENT paralela
  (%respuesta;,( %alterna; | %respuesta;)+)>
<!ATTLIST paralela que (CDATA) #REQUIRED>

<!--Definicion de los elementos de cada estado -->
<!ELEMENT estado (act* |(%alterna; %informacion;)*)>
<!ATTLIST estado id (CDATA) #REQUIRED
                 nom (CDATA) #REQUIRED
                 platilla (CDATA) #REQUIRED
                 escenario (CDATA) #REQUIRED> <!--Numero de escenario en el proceso -->

<!ENTITY % elemento "(estado+)">

<!ELEMENT roles (rol_coordinacion+)>
<!ELEMENT rol_coordinacion (nombre,agente+,(%elemento;)+)>
<!ELEMENT nombre (#PCDATA)>
<!ELEMENT agente (#PCDATA)>
<!ATTLIST agente quien (CDATA) #REQUIRED>

```

Figura 62. Documento base del workflow desconectado