

# UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería  
Departamento de Ciencias de la Computación

*“Análisis y diseño de un sistema de actualización entre un sistema SPC y un Sistema de acceso remoto para el control de movimientos de Activos Fijos.”*

Antonio Juárez López

Guatemala  
2006



***“Análisis y diseño de un sistema de actualización entre un sistema SPC y un Sistema de acceso remoto para el control de movimientos de Activos Fijos.”***

# UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería  
Departamento de Ciencia de la Computación

*“Análisis y diseño de un sistema de actualización entre un sistema SPC y un Sistema de acceso remoto para el control de movimientos de Activos Fijos.”*

Trabajo de graduación presentado por  
Antonio Juárez López  
para optar al grado académico de  
Licenciado en Ingeniería en Ciencia de la Computación

Guatemala  
2006

Vo.Bo.:

(f) \_\_\_\_\_  
M.Sc. Francisco Saravia Altolaquirre

Tribunal:

(f) \_\_\_\_\_

(f) \_\_\_\_\_

(f) \_\_\_\_\_

Fecha de aprobación:

## **RECONOCIMIENTOS**

Agradezco a todos aquellos que me han ayudado a realizar este reporte de Proyecto Profesional:

- A mi mamá, por enseñarme el buen camino.
- A mi papá, por enseñarme tantas cosas.
- A mi compañero de prácticas y de cursos, Bidkar Alexander Pojoy Corzo.
- A mi Director de Departamento, Luis Furlán, por guiarnos durante el desarrollo de este Proyecto Profesional.
- A Francisco Saravia, por ofrecernos la idea de realizar este Proyecto en su departamento, y por su apoyo.
- A Jaime Rojas, quien nos ofreció su ayuda constantemente.
- A Carlos Fuentes, por ayudarnos a entender la Base de Datos del SPC.
- A la Fundación Juan Bautista Gutiérrez, por apoyar mis estudios y permitirme realizar este Trabajo de Proyecto Profesional.
- A todas aquellas personas que de una u otra forma colaboraron en la realización de este trabajo.

## CONTENIDO

LISTA DE TABLAS .....	viii
LISTA DE FIGURAS.....	ix
RESUMEN .....	x
<b>I INTRODUCCIÓN.....</b>	<b>1</b>
<b>II ANTECEDENTES.....</b>	<b>2</b>
<b>III JUSTIFICACIÓN.....</b>	<b>3</b>
<b>IV OBJETIVOS.....</b>	<b>4</b>
<b>A Objetivos generales.....</b>	<b>4</b>
<b>B Objetivos específicos.....</b>	<b>4</b>
<b>V MARCO TEÓRICO.....</b>	<b>5</b>
<b>A Base de Datos.....</b>	<b>5</b>
<b>VI MATERIALES Y METODOLOGÍA.....</b>	<b>16</b>
<b>A Materiales.....</b>	<b>16</b>
<b>B Metodología.....</b>	<b>16</b>
<b>VII RESULTADOS.....</b>	<b>18</b>
<b>A Análisis de la empresa.....</b>	<b>18</b>
<b>B Concepción del nuevo sistema de Control de Activos Fijos.....</b>	<b>28</b>
<b>C Descripción de la nueva aplicación.....</b>	<b>30</b>
<b>VIII DISCUSIÓN.....</b>	<b>33</b>
<b>IX CONCLUSIONES.....</b>	<b>39</b>
<b>X RECOMENDACIONES.....</b>	<b>40</b>
<b>XI BIBLIOGRAFÍA.....</b>	<b>41</b>
<b>A Libros consultados.....</b>	<b>41</b>
<b>B Consultas en línea.....</b>	<b>41</b>
<b>XII GLOSARIO.....</b>	<b>43</b>
<b>XIII APÉNDICES.....</b>	<b>44</b>
<b>A Definición de Base de Datos SPC.....</b>	<b>44</b>
<b>B Definición de Base de Datos de Aplicación.....</b>	<b>54</b>
<b>C Migración y actualización de datos entre Bases de Datos.....</b>	<b>56</b>

## LISTA DE TABLAS

1. Diferencias entre tipos de Microsoft SQL Server y Sybase Data.....	9
2. Hardware y servidores disponibles en la empresa.....	23
3. Software y sistemas disponibles en la empresa.....	24
4. Relación entre tareas de Activos Fijos y procedimientos almacenados del SPC	27
5. Equivalencia entre tablas de tipos de Activos Fijos.....	34
6. Equivalencia entre tablas de Activos Fijos.....	35
7. <i>TRASLADO ACTIVOS_FIJOS</i> .....	57
8. <i>TRASLADO TIPOS ACTIVOS</i> .....	57

## LISTA DE FIGURAS

1. Sistema actual de control de Activos Fijos.....	19
2. Ciclo de vida de un Activo Fijo .....	22
3. Estructura de Red de la Empresa.....	22
4. Acceso remoto y VPN en la empresa.....	22
5. Diseño propuesto de nuevo sistema de control de Activos Fijos.....	29
6. Diagrama Entidad Relación, SPC.....	53
7. Diagrama Entidad Relación, aplicación.....	54
8. Diagrama de migración masiva.....	56
9. Diagrama de sincronización periódica.....	57

## RESUMEN

El *Sistema de Actualización entre un sistema SPC y un Sistema de acceso remoto para el control de movimientos de Activos Fijos* es la segunda parte de un proyecto mayor compuesto por dos partes: el *Sistema Remoto de Control de Movimientos de Activos Fijos* y el presente.

El propósito del proyecto es desarrollar un sistema de acceso remoto y distribuido para controlar los movimientos de los Activos Fijos de una empresa con muchas extensiones localizadas en distintas regiones. La empresa analizada es una cadena de restaurantes con una aplicación de Activos Fijos bastante completa, pero el proceso externo de ingreso de datos y actualización pasa a través de un solo usuario, muestra ineficiencia, y no cumple con los resultados esperados por la empresa.

De ello, se realizó un estudio del sistema actual y de la infraestructura tecnológica de la empresa para proponer un diseño de proyecto que mejoraría el desempeño del sistema, sin perder ninguno de los resultados logrados con el sistema actual. Este diseño incluye una aplicación central llamada *Sistema de Control de Activos Fijos*, o *SCAF*, a ser instalada en un servidor central de aplicaciones, y accesada remotamente por cualquier número de terminales conectadas a este servidor. *SCAF* cuenta con una Base de Datos propia, diseñada durante el proyecto. Sin embargo, la Base de Datos de la aplicación anterior realiza procesos contables críticos que deben ser alimentados periódicamente. Por esta razón, se requiere de una interacción regular entre las Bases de Datos de *SCAF* y el *Sistema SPC* de la empresa. Este documento reporta las fases de Análisis y Diseño de esta interacción.

Este proyecto fue realizado con la autorización del Gerente General y del Gerente del Departamento de Informática de la empresa analizada. La implementación del sistema completo y el mantenimiento del mismo quedó en manos del Departamento de Informática de la empresa analizada, que posee los manuales técnicos, código fuente, y diseños necesarios.

# I. INTRODUCCIÓN

En el 1er semestre de 2006, se tuvo la oportunidad de realizar un proyecto dentro de una empresa guatemalteca, cuyo negocio principal es una cadena de restaurantes. Estos restaurantes están distribuidos a través de todo el país, lo que hace a la comunicación un elemento muy importante para la administración de la empresa.

Esta empresa, al igual que muchas otras, se está adaptando a las nuevas tecnologías; para mantenerse actualizados y para solucionar problemas que hasta ahora se pueden solucionar. El control de Activos Fijos dentro de la empresa es uno de estos problemas; la empresa posee miles de Activos Fijos distribuidos a través de todas sus oficinas y restaurantes, y debe asegurarse que reciban mantenimiento regular, saber sobre su paradero y su estado, y minimizar pérdidas de los mismos.

Esta empresa posee un módulo auxiliar que mantiene registros de sus Activos Fijos; sin embargo, sus resultados no han sido satisfactorios. El ingreso de datos del sistema actual está centralizado en un solo usuario, y los procesos de papeleo no son tan rápidos como para verificar la veracidad de los datos que el sistema guarda.

Por eso se decidió realizar un nuevo sistema de Control de Activos Fijos, que debía interactuar con el módulo existente para no perder ciertas funcionalidades críticas, pero con una interfaz distribuida, de acceso remoto, y multi-usuario, para que éste fuera accesible desde todas las sucursales y oficinas de la empresa.

## II. ANTECEDENTES

En la empresa analizada, el control de los movimientos de los activos comprados se realiza en el sistema contable de dicha empresa, cuya aplicación central está instalada en el departamento de contabilidad. Estos activos son ingresados mensualmente al módulo auxiliar de activos fijos y conciliados con el módulo de contabilidad. En el momento de realizar algún movimiento de estos activos entre las distintas localidades, la información y autorización de dichos movimientos se realizan por escrito al departamento de contabilidad, el cual actualiza el módulo de Activos Fijos en un periodo mensual. Este esquema de control de activos fijos, presenta varios inconvenientes para la empresa:

- El personal de distintos departamentos tiene acceso y autorización para mover un activo de su localidad asignada.
- No existe un control sobre las hojas de información de traslado y, por lo tanto, no se actualiza el estado del activo.
- La descripción de los Activos Fijos no es homogénea, ya que ésta depende del usuario que haya ingresado el activo al sistema.
- No se tiene conocimiento que activos se encuentran en mantenimiento.
- La notificación de los movimientos de ingreso, traslado y baja de los activos fijos se realiza de forma escrita, causando demoras en la actualización del sistema contable e inconsistencias en el inventario de Activos Fijos.
- El sistema contable cuenta con un proceso de inventario, que solamente refleja información respecto a un ciclo de proceso mensual, lo que da como resultado diferencias entre el inventario del sistema y el inventario real de una localidad.

Aunque en la empresa se cuente con toda una infraestructura completa de redes, con acceso remoto hacia los restaurantes y distintas localidades, y un esquema de servidores de bases de datos completo, no es posible aprovechar estas tecnologías con el sistema contable debido a que esta es una aplicación central y no cuenta con módulos de acceso remoto para el registro de los movimientos diarios de los Activos Fijos en cada localidad.

### **III. JUSTIFICACIÓN**

El sistema de Activos Fijos ya existente posee ciertas características que justifican este proyecto. El sistema existente es parte de otro sistema mayor orientado a aplicaciones de contabilidad. Una de las funciones principales del sistema existente consiste en calcular las depreciaciones de los activos mensualmente. Esta función es de necesidad crítica, e interactúa profundamente con una Base de Datos central, de la cual dependen muchas otras aplicaciones de la empresa.

Este proyecto surge de la necesidad de agregar una interfaz entre el ingreso de datos y el procesamiento contable del sistema de Activos Fijos. Debido a que esta nueva capa debe de interactuar con el sistema ya existente, el nuevo sistema exige una interfaz automatizada entre esta nueva capa de interfaz para el usuario y el sistema de Activos Fijos ya existente.

Esta interacción debe realizar ciertas funciones entre el sistema existente y la nueva aplicación, tales como resolver distintas codificaciones de activos entre Bases de Datos, mantener integridad referencial entre distintas Bases de Datos, y solucionar cualquier incompatibilidad entre los tipos de datos de los registros. Es necesaria una interacción con el SPC para permitir que los procesos contables de la empresa se realicen correctamente.

## **IV. OBJETIVOS**

### **A. Objetivos generales**

1. Diseñar un sistema de interacción entre el sistema de SPC (Sistema de Planificación y Control) existente en la empresa analizada y la nueva aplicación de Control de Movimientos de Activos Fijos.
2. Especificar la tecnología adecuada para lograr una interacción adecuada entre las distintas Bases de Datos involucradas en el proyecto.
3. Lograr una interacción funcional entre los distintos sistemas sin alterar los procesos de la Base de Datos ya existente.

### **B. Objetivos específicos**

1. Analizar detalladamente un Sistema de Control de Activos Fijos y sus problemas para la empresa de restaurantes Pollo Campero, S.A.
2. Analizar el sistema existente SPC de la empresa analizada y su Base de Datos.
3. Diseñar y generar una Base de Datos apropiada para la nueva aplicación de Control de Movimientos de Activos Fijos.
4. Encontrar puntos de interacción clave para alterar y tomar información referente a los Activos Fijos de la empresa en la Base de Datos del SPC.
5. Identificar puntos clave de interacción desde la nueva aplicación desarrollada.
6. Identificar las tareas necesarias de interacción entre la nueva aplicación de Control de Movimientos de Activos Fijos y el SPC.
7. Para cada tarea identificada, diseñar una serie de pasos específicos que permita llevar a cabo estas tareas.
8. Identificar incompatibilidades entre los distintos manejadores Bases de Datos.
9. Identificar diferencias entre la codificación de Activos Fijos en ambas Bases de Datos.
10. Proveer propuestas de soluciones para cada una de las incompatibilidades y diferencias encontradas.

## V. MARCO TEÓRICO

### A. Base de datos

#### 1. Diseño de Bases de Datos relacionales

<<En general, el objetivo del diseño de una Base de Datos relacional es generar un conjunto de esquemas de relaciones que nos permitan almacenar información sin redundancia innecesaria, pero que a la vez nos permitan recuperar información fácilmente.>><sup>1</sup>

##### a. Peligros en el diseño de Bases de Datos relacionales

1) Repetición de información. <<La repetición de información en una Base de Datos no es conveniente. Además de desperdiciar espacio, la repetición de información complica la actualización de la base de datos.>><sup>2</sup> La redundancia causa peligros de inconsistencia y, en el mejor de los casos, desperdicia valioso tiempo de procesamiento.

2) Incapacidad para representar cierta información. Otro problema con un mal diseño de Base de Datos consiste en la incapacidad de representar cierta información a menos que se den ciertas condiciones. Si los objetos representados en las tablas no se diseñan apropiadamente, es posible que los objetos no puedan representarse a menos que exista suficiente información para llenar un registro completo de una tabla.

<<Una solución a este problema es introducir *valores nulos*,... para tratar las actualizaciones por medio de vistas. Sin embargo, los valores nulos son difíciles de manejar... Está claro que esto no es conveniente.>><sup>3</sup>

3) Pérdida de información. Algunos malos diseños sugieren que se debe descomponer un esquema de relaciones con muchos atributos en varios esquemas con menos atributos. Sin embargo, si la descomposición se hace sin cuidado puede llegarse a otra forma de diseño defectuoso. Una descomposición con pérdida de información es, en general, un mal diseño de Base de Datos.

---

<sup>1</sup> Korth, p.183

<sup>2</sup> Korth, p.186

<sup>3</sup> Idem

b. Normalización por media de dependencias funcionales. En el diseño de sistemas de Bases de Datos, se debe intentar no incurrir en los problemas descritos en la sección anterior. Durante este diseño, puede llegar a ser necesario descomponer una relación en varias más pequeñas. Usando dependencias funcionales, podemos definir varias formas normales que representan diseños “buenos” de bases de datos. Existe un gran número de formas normales. Las tratadas aquí son BCNF y 3NF.

1) Forma normal de Boyce-Codd. <<Una de las formas normales más deseables que podemos obtener es la forma normal Boyce-Codd>> (BCNF). Un esquema de relaciones R está en BCNF con respecto a un conjunto F de dependencias funcionales si para todas las dependencias funcionales en  $F^+$  de la forma  $\alpha \rightarrow \beta$ , donde  $\alpha$  y  $\beta$  son subconjuntos de R, por lo menos se cumple una de las siguientes condiciones:

- $\alpha \rightarrow \beta$  es una dependencia funcional trivial (es decir,  $\beta$  es subconjunto de  $\alpha$ )
- $\alpha$  es una **superclave** del esquema R.

<<Un diseño de Base de Datos está en BCNF si cada uno de los miembros del conjunto de los esquemas de relación que comprende el diseño está en BCNF.>><sup>4</sup>

2) Tercera forma normal. <<En aquellos casos en los que no puede satisfacerse BCNF, se acepta una forma normal más débil llamada *tercera forma normal* (3NF). Siempre es posible encontrar una descomposición de producto sin pérdida que conserve las dependencias que estén en 3NF.>><sup>5</sup>

<<3NF es un poco menos estricta que BCNF, permitiendo las dependencias funcionales no triviales cuyo lado izquierda no sea una superclave. Un esquema de relaciones R está en 3NF con respecto a un conjunto F de dependencias funcionales si para todas las dependencias funcionales en  $F^+$  de la forma  $\alpha \rightarrow \beta$ , donde  $\alpha$  y  $\beta$  son subconjuntos de R, por lo menos se cumple una de las condiciones siguientes:

- $\alpha \rightarrow \beta$  es una dependencia funcional trivial.
- $\alpha$  es una superclave de R.
- Cada atributo A en  $\beta - \alpha$  está contenido en una clave candidata de R.

---

<sup>4</sup> Korth, p.195

<sup>5</sup> Korth, p.199

La definición de 3NF permite ciertas dependencias funcionales que no se permiten en BCNF. Una dependencia  $\alpha \rightarrow \beta$  que satisface sólo la tercera condición de la definición de 3NF no se permite en BCNF, aunque sí se permite en 3NF. Estas dependencias se llaman dependencias transitivas.>>

3) Comparación de BCNF y 3NF. <<Hemos visto dos formas normales de esquemas de bases de datos relacionales: 3NF y BCNF. 3NF tiene la ventaja de que sabemos que siempre es posible obtener un diseño 3NF sin sacrificar un producto sin pérdida o la conservación de las dependencias. No obstante, 3NF tiene una desventaja. Si no eliminamos todas las dependencias transitivas, puede ser necesario utilizar valores vacíos para representar algunas de las posibles relaciones significativas entre los datos, y está el problema de la repetición de la información.>>

<<Si nos vemos obligados a elegir entre BCNF y la conservación de las dependencias con 3NF, generalmente es preferible optar por 3NF. Si no podemos probar la conservación de las dependencias eficientemente, pagamos un alto precio en el rendimiento del sistema o un riesgo en la integridad de los datos de la base de datos. Ninguna de estas alternativas resulta atractiva.>>

<<Para resumir lo anterior, el objeto de un diseño de base de datos relacional es:

- BCNF.
- Producto sin pérdida.
- Conservación de las dependencias.
- Si no se puede lograr esto, aceptamos:
- 3NF.
- Producto sin pérdida.
- Conservación de las dependencias.>><sup>6</sup>

---

<sup>6</sup> Korth, p.202-203

2. Manejadores de Bases de Datos. <<Entre la base de datos física misma (es decir, los datos tal y como están almacenados en realidad) y los usuarios del sistema existe un nivel de programas, el manejador de base de datos (manejador de BD) o, en la mayoría de los casos, el sistema de administración de bases de datos (DBMS, database management system). El DBMS maneja todas las solicitudes de acceso a la base de datos formuladas por los usuarios,... las facilidades para la adición y eliminación de archivos (o tablas), la obtención y puesta al día de datos de esos archivos o tablas, etcétera, están todas incluidas en el DBMS. Así, una de las funciones generales del DBMS es distanciar a los usuarios de la base de datos de detalles al nivel del equipo (de manera muy similar a la forma como los sistemas de lenguajes de programación evitan a los programadores de aplicaciones la necesidad de ocuparse de detalles al nivel de la máquina).>><sup>7</sup>

Las funciones de un DBMS deberían incluir por lo menos las siguientes:

- Definición de datos
- Manipulación de datos
- Seguridad e integridad de los datos
- Recuperación y concurrencia de los datos
- Diccionario de datos
- Desempeño

Como conclusión, el DBMS constituye la interfaz entre el usuario y el sistema de bases de datos. Las implementaciones específicas de DBMS relevantes a este proyecto son SQL Server 2000 y Sybase Adaptive Server Enterprise 11.9.2.

a. SQL Server 2000<sup>8</sup>. Las características de Microsoft® SQL Server™ 2000 incluyen:

- Integración a Internet: SQL Server 2000 incluye soporte XML integrado. También tiene la escalabilidad, disponibilidad, y seguridad requerida para operar como el componente de almacenamiento de datos de los sitios Web más grandes. El modelo de programación SQL Server 2000 está integrado con la arquitectura DNA de Windows para desarrollar aplicaciones Web.

---

<sup>7</sup> Date, p.8

<sup>8</sup> Ref. B.1

- Escalabilidad y disponibilidad: El mismo motor de Base de Datos puede usarse en plataformas desde portátiles con Microsoft Windows® 98 hasta servidores multiprocesadores con Microsoft Windows 2000 Data Center Edition. La edición SQL Server 2000 Enterprise soporta servidores federados, vistas indexadas, y gran soporte de memoria que le permite escalar a los niveles de desempeño requeridos por los más grandes sitios Web.
- Características de Base de Datos empresarial: SQL Server 2000 soporta ambientes demandantes de procesamiento de datos. El motor protege la integridad de los datos mientras minimiza el costo de manejar a miles de usuarios modificando la base de datos concurrentemente. Los queries distribuidos de SQL Server 2000 le permiten referir a la data de múltiples fuentes como si fuera parte de una base de datos SQL Server 2000. Al mismo tiempo, las transacciones distribuidas protegen la integridad de las actualizaciones de la data distribuida. La replicación le permite mantener múltiples copias de la data, y asegurarse que las copias permanezcan sincronizadas. Puede replicar un conjunto de data hacia múltiples usuarios desconectados, y luego unir sus modificaciones en la base de datos.
- “Data warehousing”: SQL Server 2000 incluye herramientas para extraer y analizar data resumida para procesamiento analítico en línea. Además, incluye herramientas para diseñar bases de datos visualmente y analizar data usando preguntas de lenguaje natural.

b. Sybase Adaptive Server Enterprise. Sybase Adaptive Server Enterprise (Sybase ASE) es un manejador de Base de Datos de alto rendimiento y calidad industrial utilizado por algunas de las mayores corporaciones a nivel mundial. Es considerado por algunos expertos como <<uno de los servidores de base de datos más fáciles para aprender y trabajar.>><sup>9</sup> Tiene un amplio rango de opciones y de replicación extremadamente avanzadas.

---

<sup>9</sup> Spongsveen, Kian

Existe una versión disponible para Linux gratuita aún para uso de producción. Este servidor comparte una herencia común con Microsoft SQL Server, pues Sybase vendió su código a Microsoft.

La última versión (15) de ASE incluye características nuevas como:

- Un optimizador de consultas más inteligente.
- Técnicas de particionamiento semántico de tablas que aumentan la velocidad de acceso a los datos.
- Columnas encriptadas para mayor seguridad de los datos.
- Columnas computadas "virtuales" y materializadas.
- Índices funcionales.
- Mejoras al lenguaje Transact-SQL.
- Mejoras a los servicios de Java y XML en la base de datos.
- Mejoras a los servicios para consumo y publicación de servicios Web.
- Herramientas mejoradas para la administración y el monitoreo.
- Más rendimiento y menor costo total de propiedad.

c. Diferencias entre tipos de datos de Microsoft SQL Server y Sybase Data<sup>10</sup>

**Tabla VI.1**

<b>SQL Server Data Type</b>	<b>Sybase Data Type</b>	<b>Descripción</b>	<b>Conversión</b>
char( <i>n</i> ) 1<= <i>n</i> <=8000	char( <i>n</i> ) 1<= <i>n</i> <=255	Datos de caracteres de longitud fija	Si <i>n</i> >255, el <b>char(<i>n</i>)</b> de SQL Server debe ser convertido al texto de Sybase
nchar( <i>n</i> ) 1<= <i>n</i> <=4000	nchar( <i>n</i> ) 1<= <i>n</i> <=255	Datos de caracteres de longitud fija	Si <i>n</i> >255, el <b>nchar(<i>n</i>)</b> de SQL Server debe ser convertido al texto de Sybase

<sup>10</sup> Ispirer Systems, Ref. B.3

		(Unicode, Multibyte)	
<code>varchar(n)</code> $1 \leq n \leq 8000$	<code>varchar(n)</code> $1 \leq n \leq 255$	Data de caracteres de longitud variable	Si $n > 255$ , el <b>varchar(n)</b> de SQL Server debe ser convertido al texto de Sybase
<code>nvarchar(n)</code> $1 \leq n \leq 4000$	<code>nvarchar(n)</code> $1 \leq n \leq 255$	Datos de caracteres de longitud variable (Unicode, Multibyte)	Si $n > 255$ , el <b>nvarchar(n)</b> de SQL Server debe ser convertido al texto de Sybase
<code>bigint</code>	-	Entero, 64-bits	<b>Bigint</b> de SQL Server debe ser convertido a <b>decimal(19,0)</b> de Sybase
<code>int</code>	<code>int</code>	Entero, 32-bits	No requerida
<code>smallint</code>	<code>smallint</code>	Entero, 16-bits	No requerida
<code>tinyint</code>	<code>tinyint</code>	Entero, 8-bits	No requerida
<code>datetime</code>	<code>datetime</code>	Date and time	No requerida
<code>smalldatetime</code>	<code>smalldatetime</code>	Date and time	No requerida
<code>money</code>	<code>money</code>	Data monetaria	No requerida
<code>smallmoney</code>	<code>smallmoney</code>	Data monetaria	No requerida
<code>decimal(p,s)</code> $1 \leq p \leq 38;$ $0 \leq s \leq p;$	<code>decimal(p,s)</code> $1 \leq p \leq 38;$ $0 \leq s \leq p;$	Data numérica de punto fijo	No requerida
<code>numeric(p,s)</code> $1 \leq p \leq 38;$ $0 \leq s \leq p;$	<code>numeric(p,s)</code> $1 \leq p \leq 38;$ $0 \leq s \leq p;$	Data numérica de punto fijo	No requerida
<code>float(n)</code> $1 \leq n \leq 53$	<code>float(n)</code> $1 \leq n \leq \text{machine}$	Data numérica de punto flotante	No requerida

	dependent		
-	double precision	Data numérica de punto flotante, 64-bits	<b>Double precision</b> de Sybase debe ser convertida a <b>float(53)</b> de SQL Server
real	real	Data numérica de punto flotante, 32-bits	No requerida
text	text	Data de caracteres de longitud variable, hasta 2GB	No requerida
ntext	-	Data de caracteres Unicode de longitud fija, hasta 1GB	<b>Ntext</b> de SQL Server debe ser convertido a <b>text</b> de Sybase
binary(n) 1<=n<=8000	binary(n) 1<=n<=255	Data binaria de longitud fija	Si n>255, <b>binary(n)</b> de SQL Server debe ser convertido a <b>image</b> de Sybase
varbinary(n) 1<=n<=8000	varbinary(n) 1<=n<=255	Data binaria de longitud variable	Si n>255, <b>varbinary(n)</b> de SQL Server debe ser convertido a <b>image</b> de Sybase
image	Image	Data binaria de longitud variable, hasta de 2GB	No requerida
bit	bit	Entero de valor 0 o 1	<b>Bit</b> de SQL Server puede ser NULL, y el <b>bit</b> de Sybase no puede serlo. Si el <b>bit</b> de SQL Server podría ser NULL, convertir a <b>tinyint</b> o <b>char(1)</b> de

			Sybase
uniqueidentifier	-	Identificador único global	<b>uniqueidentifier</b> de SQL Server debe ser convertido a <b>char(36)</b> de Sybase
Timestamp	timestamp	Guarda el tiempo de la última actualización del registro	No requerida

3. Procedimientos almacenados. Los procedimientos almacenados (también llamados módulos almacenados) son instrucciones definidas en un manejador de Base de Datos. Éstos tienen un nombre, una lista de parámetros, y uno o más valores de retorno.

a. Definición estándar. Según la definición de SQL estándar, <<un módulo de servidor SQL es un objeto persistente definido en un esquema e identificado por un <nombre de módulo SQL estándar>. Los módulos de servidor SQL son creados con <definiciones de módulo de SQL Server> y destruidos con <instrucciones de “drop module”> y por <instrucciones de “drop schema”> que destruyen a los esquemas que los contienen.>><sup>11</sup>

La definición completa de un procedimiento almacenado incluye:

- El nombre del procedimiento.
- El descriptor del conjunto de caracteres en el que el procedimiento se representa.
- El identificador de autorización del procedimiento.
- El descriptor de tabla de toda tabla temporal local descrita en el procedimiento.
- El descriptor de cada rutina SQL invocada desde el procedimiento.
- El texto (las instrucciones) del procedimiento almacenado.

---

<sup>11</sup> Melton, p.7

b. Historia de los procedimientos almacenados. En las implementaciones iniciales de las Bases de Datos, todo el procesamiento se realizaba en grandes estaciones de trabajo, que enviaban la salida a terminales tontas. No había interés sobre el procesamiento de la data por parte del cliente, pues la data siempre residía en el servidor. Este proceso cambió cuando las Bases de Datos comenzaron a aparecer en máquinas UNIX de rango medio, donde el cliente y el servidor eran comúnmente máquinas distintas. Los procedimientos almacenados fueron creados con los siguientes objetivos:

- Para permitir que el procesamiento de data ocurriera en los servidores más rápidos.
- Para reducir la carga de trabajo y los cuellos de botella de **CPU** en los clientes más lentos.
- Para reducir la cantidad de data enviada a través de la red.

Los procedimientos almacenados pertenecen a una clase de objetos llamados objetos de aplicación de Base de Datos. Los objetos de aplicación encapsulan lógica de aplicación en el servidor de Base de Datos en vez de mantenerla en el código de la aplicación cliente. El uso de los objetos de aplicación ayuda a reducir el tráfico de la red. En el caso de los procedimientos almacenados, por ejemplo, solamente la primera petición y la salida final necesitan ser transmitidas entre el cliente y el servidor.<sup>12</sup>

c. Ventajas ofrecidas. Siendo una tecnología relativamente nueva, la implementación de los procedimientos almacenados aún difiere bastante a través de los distintos manejadores. Aún así, las ventajas que los procedimientos almacenados ofrecen, sin embargo, son básicamente las mismas:

- Facilidad de programación (lado del cliente).
- Mejor manejo de Base de Datos.
- Asegurar la integridad y la portabilidad de la Base de Datos.
- Facilitar el manejo de permisos de los usuarios.
- Encapsular y facilitar la administración de permisos de usuario.

---

<sup>12</sup> Janmohamed, p.2

- Esconder instrucciones SQL, y así mantener confidencial la estructura de la Base de Datos.
- Encapsular el manejo de transacciones, y así asegurarse de la integridad de los datos.
- Disminuye el tráfico de instrucciones SQL a través de la red.
- En algunas implementaciones, como en Microsoft SQL Server, los procedimientos almacenados se guardan en forma precompilada y optimizada. Esto incrementa la velocidad de las peticiones a la Base de Datos.

## **VI. MATERIALES Y METODOLOGÍA**

### **A. Materiales**

1. Computadora personal. Para el diseño del sistema, se utilizó una computadora personal con procesador Intel Pentium IV con 256 MBytes de memoria RAM. Esta computadora estuvo conectada a una LAN 100Mbps con acceso a los servidores de Bases de Datos para realizar parte del Análisis remotamente. En esta computadora, también se diseñaron los procesos del sistema.
2. Servidor de Base de Datos. La Base de Datos del SPC, que fue analizada, reside en una computadora con 4 GB de memoria RAM y con un procesador Intel Pentium de 2.4 GHz.
3. Sistema Operativo Microsoft Windows XP Service Pack 2. La estación de trabajo personal funcionaba con el Sistema Operativo Microsoft Windows XP Service Pack 2
4. Servidor de Base de Datos Microsoft SQL Server 2000. Este DBMS es el manejador anfitrión de la nueva aplicación de Control de Movimientos de Activos Fijos.
5. Documentación en línea. Referencias en páginas Web. Documentación publicada en Internet sobre los DBMS, productos, y tecnologías utilizadas.
6. Documentación impresa. Libros de texto y referencia sobre temas varios.

### **B. Metodología**

El proyecto se llevó a cabo dentro de una empresa guatemalteca de restaurantes como parte de un proyecto general de Inventario y Control de Movimientos de Activos Fijos, contando con la autorización del Gerente General de la empresa y con la supervisión del Gerente del Departamento de Informática de la misma.

Se llevaron a cabo las siguientes tareas durante la realización del proyecto:

1. Análisis
  - a. Entrevistas a personal responsable del control de los activos fijos.
  - b. Reunión con personal del Departamento de Operaciones.
  - c. Reunión con los administradores del Departamento de Informática.
  - d. Análisis de la Base de Datos del SPC, con el propósito de comprender su estructura y funcionamiento.
  - e. Reunión con los creadores de la Base de Datos, discusión sobre su funcionamiento general.
  - f. Obtención de la estructura específica y de las interfases (procedimientos almacenados) disponibles en ella en archivos de texto.
  - g. Análisis de la estructura y de los procedimientos almacenados.
  - h. Redacción del Departamento de Especificación de Requerimientos para la empresa analizada.
  - i. Aprobación final del Documento de Especificación de Requerimientos.
2. Diseño
  - a. Investigación acerca del funcionamiento de los manejadores de Bases de Datos Sybase 11 y SQL Server 2000.
  - b. Creación de diagramas funcionales.
  - c. Determinar las distintas tareas que el sistema debe realizar.
    - 1) Migración inicial de datos desde la Base de Datos SPC hacia la Base de Datos de la aplicación central.
    - 2) Almacenaje y actualización de los datos compatible con la estructura existente en la Base de Datos SPC.
    - 3) Actualización de la Base de Datos SPC con los datos temporales disponibles en el inventario de la aplicación central.
  - d. Creación de algoritmos transaccionales para cada uno de estas tareas.
  - e. Traslado de estos algoritmos hacia diagramas.
  - f. Reunión con los encargados del Departamento de Informática para aprobar el diseño.

## **VII. RESULTADOS**

### **A. Análisis de la empresa**

#### 1. Sistema de Control de Activos Fijos actual

a. Proceso general. La empresa analizada cuenta con más de 80 oficinas y sucursales a través de todo el país. Estas serán referidas como localidades de aquí en adelante. En estas localidades se encuentran más de 20000 Activos Fijos, propiedad de la empresa. Un Activo Fijo se define como una propiedad de la empresa con un valor de al menos Q5000.00, y una vida útil de al menos tres años.

Actualmente, existe un módulo auxiliar en el SPC de la empresa que almacena los registros de los Activos Fijos en la Base de Datos del SPC. Este módulo se encarga de calcular depreciaciones de los Activos Fijos y de aplicar los costos a las partidas contables respectivas de cada localidad. Este cálculo de depreciaciones y distribución del costo a través de las partidas contables funciona actualmente, pero no se tiene documentación detallada sobre el funcionamiento interno de este proceso (algoritmos que utiliza, tablas que afecta). Para no perder esta funcionalidad con el desarrollo del nuevo sistema, se decidió mantener este módulo auxiliar como parte del sistema.

La Base de Datos del SPC no contiene registros de los proveedores de cada Activo Fijo, de las personas responsables de cada Activo, o de los usuarios del sistema. Aunque posee la capacidad de mantener registros con esta información, estos campos no se utilizan.

Cuando un Activo Fijo es adquirido o es movilizado entre localidades, el encargado del activo o localidad debería llenar un formulario reportando este movimiento al departamento de Contabilidad, donde debe ser validado y autorizado para ser ingresado al sistema. Cuando el movimiento ha sido autorizado, el encargado del módulo de Activos Fijos ingresa los datos del movimiento manualmente en el sistema. Muchas veces los movimientos no son reportados, lo que causa inconsistencias e irregularidades en los registros de los Activos. Creemos que esto sucede porque no existe una

responsabilidad registrada o documentada de mantener los Activos Fijos en buen estado y en orden. La falta de responsabilidad de parte de los operarios es una de las razones por la que se decidió realizar un nuevo sistema.

Cuando un movimiento se reporta, se estima que tarda un promedio de dos semanas para que sea ingresado al sistema. Algunos movimientos de activos son muy rápidos, y muchas veces se da el caso de inconsistencias e irregularidades en los registros de los Activos.

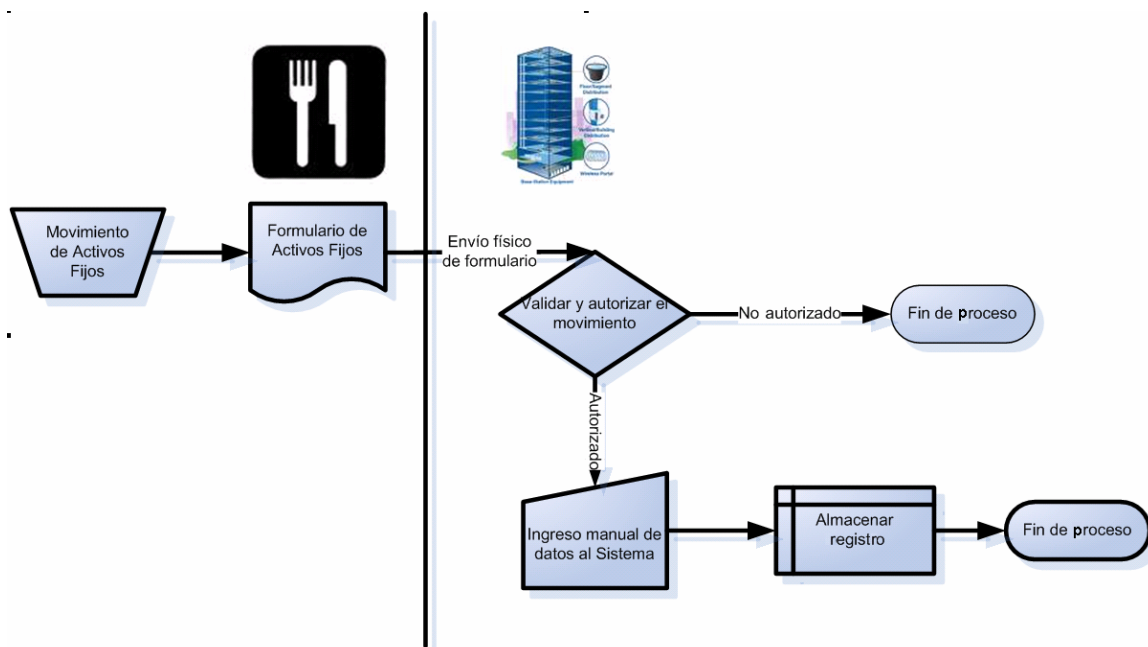
Por otro lado, los Activos no poseen una codificación satisfactoria. No existe un campo único como llave primaria de los Activos. Los procesos del sistema actual son incapaces de actualizar la localidad de un Activo Fijo, y además, los nombres de los Activos Fijos son ingresados textualmente, lo que dificulta mucho la generación de reportes de costos y de localidades agrupados por Tipo de Activo o por localidad.

El departamento de Auditoría requiere los datos de los Activos Fijos en algunos de sus procesos. Estos datos deben ser precisos y actualizados para que sirvan a Auditoría de la mejor manera. Se determinó, sin embargo, que los datos disponibles de los Activos Fijos no cumplían los requisitos necesarios para ser útiles a Auditoría. El departamento de Contabilidad tiene un sistema muy limitado para llevar a cabo el control de más de 20000 activos en más de 80 localidades en todo el país.

b. Tareas identificadas. El sistema actual de Control de Activos Fijos consta de los siguientes procesos interactivos (ilustración en la figura VI.1):

- Envío de formularios de ingresos o movimientos de Activos a la oficina central.
- Análisis y filtrado manual de formularios.
- Ingreso manual de los formularios al sistema.
- Proceso “batch”, cálculo de depreciaciones.

Figura VI.1



El envío de formularios y el ingreso manual de datos son procesos de traslados de información. El segundo proceso es una autorización evaluada por directivas de negocio; éste requiere de interacción con un autorizador válido. El cuarto proceso es independiente de los otros tres, y además, es el único completamente automatizado, pero de funcionamiento interno no documentado.

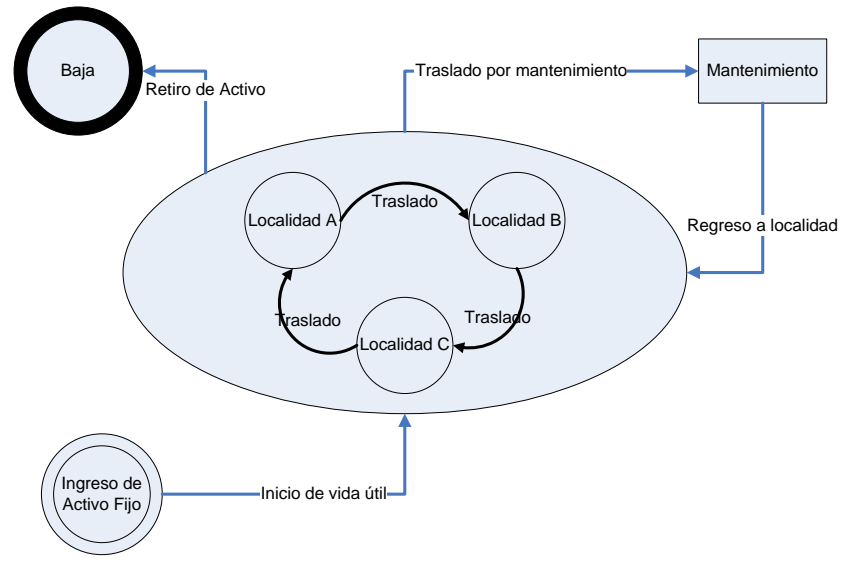
c. **Objetos identificados.** De las entrevistas realizadas al personal de los distintos departamentos, se determinaron los objetos que tienen algún rol en el sistema. Estos objetos fueron utilizados como base para la creación de las tablas en la base de datos.

- 1) **Proveedores:** Entidades que proveen Activos Fijos a la empresa, y/o que proveen reparaciones a los mismos.
- 2) **Transacciones:** Procesos realizados sobre un activo (Adquisición, traslado, mantenimiento, retiro)
- 3) **Autorizaciones:** La autorización de un usuario respecto a una transacción específica.
- 4) **Tipos de Activos:** Categorías generales en que se divide a los Activos Fijos (terrenos, maquinaria, equipo de cómputo, etc.)

- 5) **Nombres de Activos:** Identificación de un Tipo de Activo específico. Es la clasificación más fina de un activo (dos vehículos iguales son dos Activos Fijos distintos, pero tienen el mismo Nombre de Activo).
- 6) **Activos:** Objetos tangibles, propiedad de la empresa, de vida útil mínima de tres años, y valor mínimo de Q5000.00.
- 7) **Usuarios:** Personal de la empresa que cumpla con una de dos o ambas funciones: Usuario de la aplicación /Responsable de al menos un Activo Fijo.
- 8) **Bitácora:** Conjunto de registros de acciones realizadas sobre el sistema. Permite un control administrativo del sistema.
- 9) **Roles:** Conjuntos de permisos del sistema que representan la función de un usuario en el sistema (Gerente de Área de Restaurantes, Contabilidad, Administrador)
- 10) **Localidades:** Área o departamento de trabajo particular, en la que hay y/o se utilizan Activos Fijos de la empresa.
- 11) **Cuentas:** Cuentas definidas en contabilidad, con presupuestos y gastos propios.
- 12) **Centros o Sub-Divisiones:** Subconjunto de una localidad que realiza una función específica. Tiene una cuenta de gastos asignada.

d. Ciclo de vida de los Activos Fijos. Basados en las entrevistas y las conclusiones de las reuniones con los usuarios del sistema contable dentro de la empresa, se pudo especificar el diagrama del ciclo de vida de un activo fijo específico. Este se ilustra en la figura VI.2.

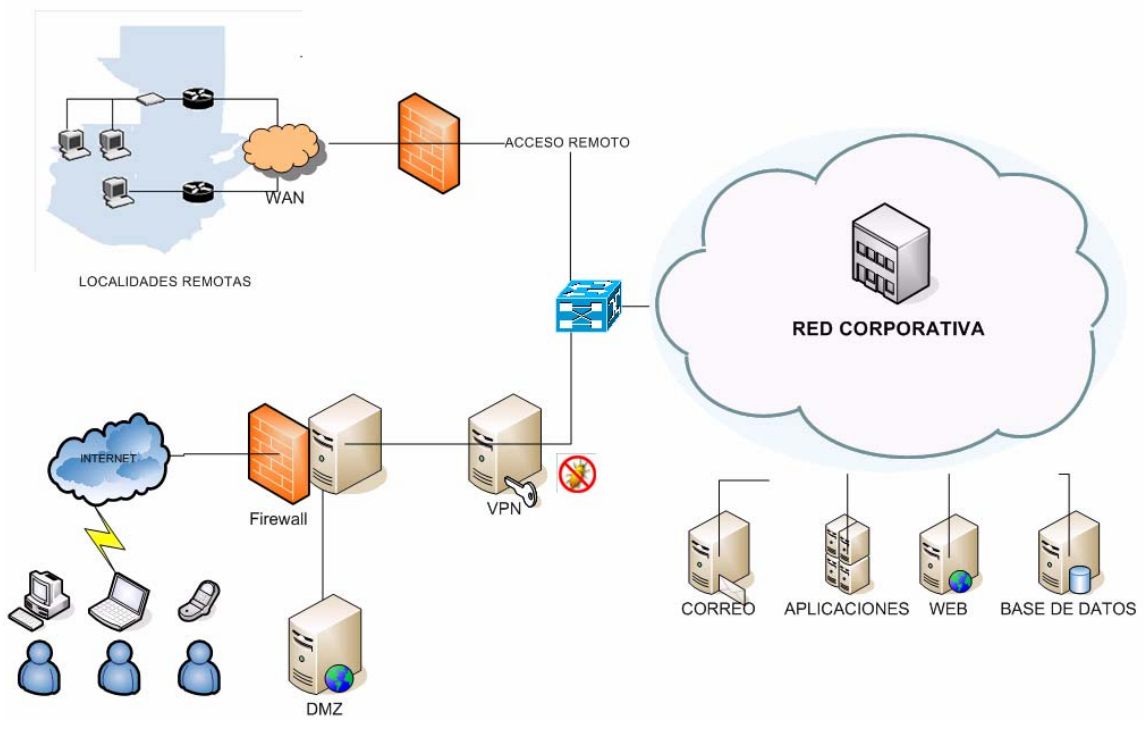
Figura VI.2



2. Infraestructura de cómputo

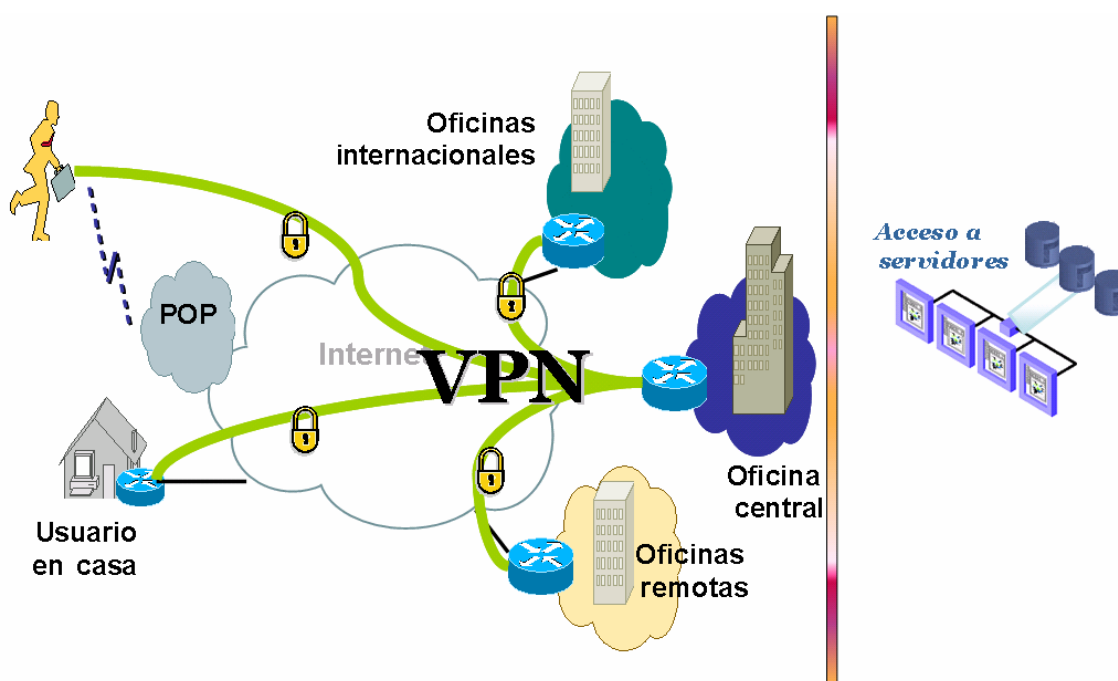
a. Estructura de Red. La empresa analizada ya posee implementada una estructura de red entre sus oficinas centrales y existe el acceso remoto por medio de "Terminal Services". hacia las localidades (restaurantes) alrededor del país. El diagrama general de la estructura de la red es el siguiente:

Figura VI.3



Las localidades remotas (restaurantes) pueden utilizar las aplicaciones en la central por medio de “*Terminal Services*”. Este servicio es lo que permite el acceso remoto a la aplicación nueva de Control de Activos Fijos. Estas localidades remotas se encuentran dentro de la red interna corporativa por medio de enlaces dedicados que les ofrece un proveedor de servicios de telecomunicación. Los usuarios remotos que no se encuentran dentro de la red interna, pueden acceder a los servicios de la red corporativa por medio de Internet a través de una VPN (Virtual Private Network) o Red Privada Virtual.

**Figura VI.4**



b. Servidores. Dentro de la empresa analizada se encuentra el siguiente hardware:

**Tabla VI.2**

<b>HARDWARE</b>					
<b>Aplicación</b>	<b>Marca</b>	<b>Modelo</b>	<b>Disco</b>	<b>RAM</b>	<b>Procesador</b>
Nóminas	HP	BLADE	280	4 GB	4 XEON 3200
	COMPAQ	PROLIANT	GB		MHz
		BL20PG2			
Contabilidad	HP	PROLIANT	150	4 GB	4 XEON 3065

	COMPAQ	BL20PG2	GB		MHz
Suministros y distribución	HP	PROLIANT	480	4 GB	4 XEON 3056
Ejecución	COMPAQ	ML350 G3	GB		MHz
Presupuestaria	HP	PROLIANT	550	4 GB	4 XEON 3056
Servidor	COMPAQ	ML350 G3	GB		MHz
Aplicaciones	HP	BLADE	195	2 GB	2 XEON 3200
Varias	COMPAQ	PROLIANT	GB		MHz
		BL20PG3			
Servidor	HP	BLADE	137	4 GB	2 PENTIUM 2400
Aplicaciones	COMPAQ	PROLIANT	GB		MHz
Remotas		BL20PG3			
Recursos Humanos	IBM	NETFINITY	40 GB	2 GB	2 PENTIUM 700
					MHz

Como ya se contaba con esta infraestructura dentro de la empresa, se utilizó el Servidor de Aplicaciones Remotas como anfitrión de la aplicación desarrollada.

Además se encuentran los siguientes manejadores de Base de Datos para el almacenamiento de información de la empresa.

**Tabla VI.3**

<b>SOFTWARE</b>		
<b>Aplicación</b>	<b>Bases de Datos</b>	<b>Sistema</b>
Nóminas	SYBASE 11.9.2	SPC (Sistema Planeación y Control)
Contabilidad	SYBASE 11.9.2	SPC (Sistema Planeación y Control)
Suministros y distribución	SYBASE ASE 11.9.2	SPC (Sistema Planeación y Control)
Ejecución presupuestaria	SQL 2000	Sales Analysis Module
Servidor Base Datos	SQL 2000	Bases de Datos - Aplicaciones Varias (SCAF)
Aplicaciones varias		
Recursos Humanos	SQL 2000	Sistema Administración RRHH

Como se observa, el Sistema SPC contiene la aplicación de Contabilidad. Uno de los módulos de la aplicación de Contabilidad es el módulo auxiliar de Control de Activos Fijos actual, con el cual este proyecto debe interactuar.

### 3. Sistema SPC

#### a. Módulo auxiliar de Activos Fijos

1) Funcionamiento general. El propósito del módulo auxiliar de Activos Fijos es controlar los registros de Activos Fijos dentro de la Base de Datos del SPC. Para lograrlo, el módulo resalta distintos casos de uso:

a) Ingreso de nuevos activos. Para ingresar un activo nuevo al sistema, el usuario debe ingresar al sistema los datos del activo. Los datos relevantes del registro son su localidad inicial, su código correlativo dentro de la localidad, el proveedor, el tipo de activo, y el valor inicial del activo.

b) Modificación de la cuenta de Activos Fijos. Cuando un activo fijo es trasladado a otra localidad, la cuenta contable a la que se carga su depreciación debe cambiar también. El módulo auxiliar puede cambiar la referencia de un activo hacia su cuenta contable. Sin embargo, esta función no puede cambiar la referencia a la localidad en la que el activo está realmente.

c) Baja de activos. Cuando se determina que un activo ya no será utilizado por la empresa, su estado debe cambiar hacia “Baja”. Las razones representables por las que un activo puede darse de baja son las siguientes:

- Desechado
- Donado
- Vendido

d) Generación de reportes. El Departamento de Auditoría de la empresa requiere reportes periódicos y actualizados sobre el estado y las localidades de los Activos Fijos.

e) Cálculo de depreciaciones. El departamento de Contabilidad realiza un proceso de cierre al final de cada mes. Este proceso incluye adjudicar costos de depreciación de activos a todas aquellas cuentas que utilicen o sean responsables del uso de un conjunto de Activos Fijos.

El módulo auxiliar de Activos Fijos del SPC tiene una función automatizada en la que se calculan las depreciaciones de cada activo, se adjudican como costos a las cuentas respectivas de cada activo, y se anexan a las partidas contables generales de la empresa. No se encontró documentación sobre el mecanismo interno de esta función.

2) Base de Datos. El módulo auxiliar de Activos Fijos del SPC trabaja directamente en la Base de Datos del SPC.

a) Manejador. El DBMS utilizado por la Base de Datos del SPC es Sybase 11.9.2 Adaptive Server.

b) Estructura. La Base de Datos del SPC contiene cientos de tablas, pero sólo 31 están relacionadas con los Activos Fijos. Las tablas que contienen registros de los activos fijos o que se refieren o son referidas por los activos fijos, se describen por completo en el Apéndice A.

La tabla que contiene los registros de los activos fijos es ACTIVOS\_FIJOS.

c) Codificación de Activos. En esta Base de Datos, un Activo Fijo individual no posee un código identificador único. Incluso, el campo llamado CODIGO\_DE\_ACTIVO parece ser un identificador, pero no lo es. Este campo en realidad hace referencia a la localidad que el activo tenía cuando fue ingresado al sistema (pues una vez ingresada, es imposible modificarla en la Base de Datos).

Sin embargo, es posible identificar a un Activo Fijo con la unión de dos de sus campos: CODIGO\_DE\_ACTIVO y MEJORA\_CORRELATIVO.

d) Procedimientos almacenados. La Base de Datos del SPC cuenta con miles de procedimientos almacenados, pero de éstos, solo unos pocos interactúan con las tablas de Activos Fijos. Estos procedimientos se describen por completo en el Apéndice A.

En particular, se identificó que ciertos procedimientos almacenados realizaban las tareas listadas en la sección anterior. Esta relación entre procedimientos y tareas del módulo auxiliar está descrita en la tabla VI.4.

Tabla VI.4

Tarea	Procedimiento Almacenado
Ingreso de activos	sp_Alta_TIPO_MOV_ACTIVOS
Modificación de activos	sp_Cambio_TIPO_MOV_ACTIVOS
Baja de activos	sp_Baja_TIPO_MOV_ACTIVOS

4. Conclusiones del análisis. De acuerdo al análisis del sistema actual de Control de Activos Fijos, se determinó que sus funciones principales son:

- Reportar movimientos, ingresos y bajas de Activos Fijos desde las localidades hacia las oficinas centrales.
- Validar y autorizar los movimientos reportados.
- Registrar los movimientos en la Base de Datos del SPC.
- Generar reportes

Los problemas determinados fueron los siguientes:

- El ingreso de datos del sistema es realizado por un solo usuario, lo que causa un cuello de botella.
- Papeleo de formularios y el traslado físico de la información.
- Doble ingreso manual de datos.
- No existe campo único como código, o codificación adecuada para los Activos Fijos de la empresa.
- El sistema actual no registra varios datos requeridos por Auditoría (proveedor, responsable de activo, usuarios).
- No es posible cambiar la localidad referida por un Activo Fijo.

## **B. Concepción del nuevo sistema de control de Movimientos de Activos Fijos**

Los propósitos que se tuvieron al realizar el proyecto de Activos Fijos en general son los siguientes:

- Eliminar o reducir los problemas que el sistema actual de control de Activos Fijos posee.
- Mantener las mismas funcionalidades que ofrece el sistema actual.

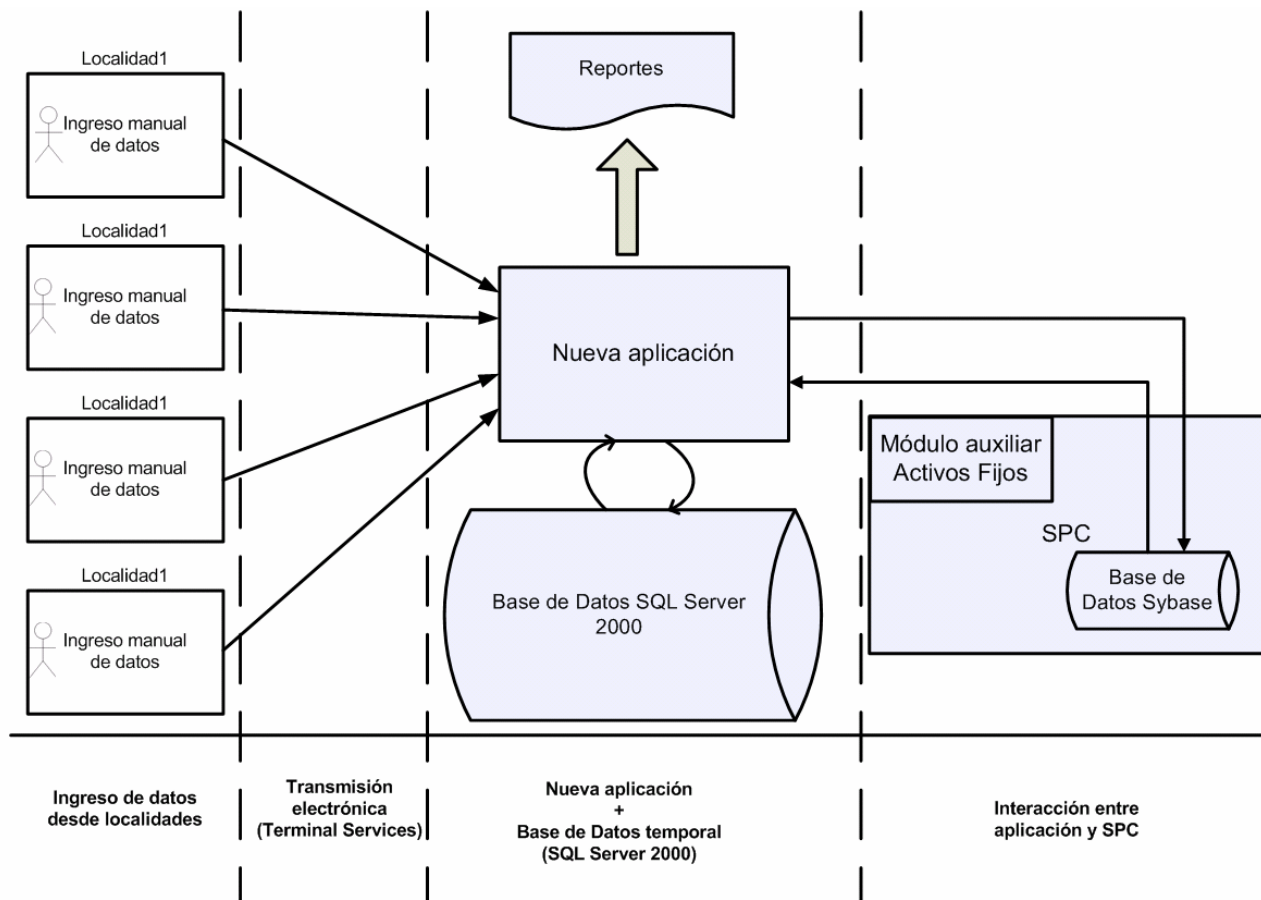
Según el análisis realizado, las tareas realizadas por el sistema actual de control de Activos Fijos son las siguientes:

- Envío de reportes de movimientos de las distintas localidades hacia las oficinas centrales.
- Validación y autorización de los movimientos.
- Cálculo de depreciaciones.
- Generación de reportes

Basados en el Análisis del sistema, se decidió desarrollar una aplicación que funcionara de manera adjunta al sistema actual (véase la figura VI.5). Esta aplicación sirve como una interfaz más moderna. Sus distintas pantallas permiten reportar todos los movimientos reportados anteriormente, y más detalles, como la persona responsable del Activo reportado. La aplicación funciona desde un servidor de aplicaciones central, y así, todas las localidades pueden acceder a ésta remotamente.

Cuando un reporte se ingresa al sistema, se inicia una transacción multi-usuario en la que un usuario autorizador decide si autorizar la transacción o no. Este proceso imita la validación del sistema actual.

Figura VI.5



Esta aplicación utiliza una Base de Datos propia, diseñada durante la realización de este proyecto. El fin de esta Base de Datos es proveer los mismos procesos realizados manualmente en el sistema actual, de manera electrónica. Esta Base de Datos permanecerá constantemente actualizada, y provee de datos a la aplicación para generar reportes dinámicos y configurables al Departamento de Auditoría sobre los Activos Fijos.

Sin embargo, se determinó que esta aplicación no podría realizar una de las funciones que el sistema actual realiza: El cálculo de depreciaciones. Esto sucede porque no se tiene documentación sobre el funcionamiento interno de esta función. Por eso, se decidió diseñar un sistema de interacción entre la Base de Datos de la aplicación y la Base de Datos del SPC. El diseño de esta interacción es el tema central de este reporte.

## C. Descripción de la nueva aplicación

1. Funcionamiento general. Tomando en cuenta los problemas que se enfrentan debido al sistema actual de Activos Fijos, se decidió desarrollar una aplicación que no tuviera los problemas identificados, y que mantuviera la misma funcionalidad del Sistema de Control de Activos Fijos y del SPC.

Con este fin, se planificó el desarrollo de un sistema con las siguientes características:

- Ingreso de datos distribuido desde cada una de las localidades.
- Traslado de datos electrónico a través de líneas dedicadas y seguras.
- Validación y autorización electrónica, no habrá necesidad de ingresar los datos una vez autorizado cada movimiento.
- Base de Datos temporal que guarde los activos y las transacciones que aún no están en estados aceptables para la Base de Datos del SPC.
- Campo identificador único para cada activo fijo.
- Campos de proveedor y persona responsable se incluyen en la tabla de Activos.
- Registro de todos los movimientos que los usuarios realizan en una **bitácora**.
- La posibilidad de modificar cualquier campo de los Activos Fijos, mientras siga el procedimiento determinado por la aplicación.
- Las funciones de ingreso, modificación, y baja de Activos Fijos se implementan de manera centralizada y transaccional.
- Sistema de solicitudes, autorizaciones, y estados intermedios durante una transacción.
- Contendrá un proceso que actualice la Base de Datos del SPC con los datos ingresados en la Base de Datos temporal.

a. Procesos. Esta nueva aplicación realizará las mismas tareas que realiza actualmente el módulo auxiliar de Control de Activos Fijos del SPC, además de la interacción con la Base de Datos del SPC:

- 1) Ingreso de nuevos activos
- 2) Movimiento de activos
- 3) Baja de activos
- 4) Actualización de SPC

## 2. Detalles técnicos

a. Lenguaje de programación. La aplicación desarrollada (SCAF) se programó en el ambiente de desarrollo Microsoft Visual Studio .NET 2005, bajo el lenguaje Visual Basic .NET. Se eligió este ambiente de programación debido a que el manejador de Base de Datos comúnmente utilizado en la empresa, Microsoft SQL Server, se ajusta fácilmente a las librerías de este ambiente.

### b. Base de Datos

1) DBMS. La aplicación desarrollada utiliza Microsoft SQL Server 2000 como DBMS para su Base de Datos principal. El ambiente de desarrollo utilizado, Microsoft Visual Studio .NET 2005, tiene librerías especiales que facilitan y optimizan la conexión al DBMS elegido.

2) Estructura. (Nota: Aquí sólo se refiere a ciertas tablas que tienen relevancia a este proyecto. Para el diagrama completo, refiérase al Apéndice C y al Proyecto Profesional de Bidkar Pojoy).

La Base de Datos incluye las siguientes tablas, columnas, con sus respectivos tipos:

- Activos
  - Cod\_Activo (bigint) [PK]
  - Cod\_Nombre (varchar[50])
  - FechaAdquisicion (datetime)
  - FechaRetiro (datetime)
  - Estado (tinyint)

- Cod\_Proveedor (bigint)
- Username (varchar[30])
- FechaRetorno (datetime)
- Centro (bigint)
- Transacciones
  - Cod\_Activo (bigint) [PK]
  - Username (varchar[30])
  - Accion (tinyint)
  - FechaInicio (datetime)
  - FechaFin (datetime)
  - UsernameOrigen (varchar[30])
  - UsernameDestino (varchar[30])
  - CentroOrigen (bigint)
  - CentroDestino (bigint)
  - Descripcion (varchar[200])
  - Estado (tinyint)

3) Codificación de Activos. Debido a las dimensiones de la empresa analizada, el número de Activos a controlar es considerable, y es preferible tener un campo único para su identificación.

La sección anterior mostró el campo Cod\_Activo dentro de la tabla Activos. Este campo es el identificador único de cada Activo Fijo. Su tipo de dato respectivo, bigint, utiliza 8 bytes para su almacenamiento. Este tipo de dato permite un máximo de  $1.8 \cdot 10^{19}$  registros únicos. Para el propósito del proyecto actual, se supone seguramente que no habrá necesidad de ampliar este rango.

c. Conectividad distribuida. La conectividad entre las localidades remotas a la oficina central se realiza por medio del protocolo “*Terminal Services*”. Se utiliza la infraestructura de red existente en la empresa como medio de comunicación.

## VIII. DISCUSIÓN

El Análisis del módulo de Control de Activos Fijos reveló que un nuevo sistema de Activos Fijos requeriría inevitablemente de un sistema de interacción entre la Base de Datos del nuevo sistema y la Base de Datos del SPC. La primera funcionaría simplemente como un repositorio temporal desde donde se generarían reportes y se almacenarían las transacciones incompletas. La segunda Base de Datos, sin embargo, es la utilizada por el módulo auxiliar del SPC para calcular depreciaciones y adjudicarlas como costos a las cuentas de contabilidad respectivas.

Es por eso que se decidió realizar una interacción entre ambas Bases de Datos. Este sistema de interacción tiene como objetivo mantener la funcionalidad actual del cálculo de depreciaciones. Para este fin, su principal proceso es actualizar la Base de Datos del SPC con los constantes nuevos movimientos de Activos Fijos que se almacenan en la Base de Datos de la nueva aplicación.

### 1. Tareas identificadas

a. Migración inicial de datos (diagrama). La Base de Datos del sistema actual contiene más de veinte mil registros de Activos Fijos. Para que estos activos sean incluidos en el nuevo sistema, se debe hacer una migración masiva inicial desde la Base de Datos del SPC hacia la Base de Datos de la nueva aplicación.

Esta migración masiva, en teoría, se debe realizar una sola vez. Después de esta primera migración, todo nuevo registro y movimiento se almacenará en la Base de Datos de la aplicación, y luego se actualizará en la Base de Datos del SPC por medio del proceso de actualización.

Como las tablas de ambas Bases de Datos no poseen la misma estructura, es necesario hacer una transformación a los registros del SPC para que sean compatibles con la Base de Datos de la aplicación. Para realizar el diseño de migración, se analizó la

estructura de las tablas relevantes. Una descripción completa de estas se incluye en el Apéndice A.

Adicionalmente, los especialistas de la empresa determinaron que los catálogos de cuentas y de localidades son importantes para otras aplicaciones que la empresa pudiera desarrollar en el futuro. Por esta razón, se decidió que estos catálogos fueran accedidos como vistas de una Base de Datos en un servidor central de la empresa. Esta Base de Datos se llama SAM, y sus características se tratan con mayor detalle en el Reporte de Proyecto Profesional de Bidkar Pojoy.

Los objetos cuyos registros deben migrarse de Bases de Datos externos son los siguientes:

1) Localidades. El catálogo de localidades no se migró hacia la Base de Datos de la aplicación, técnicamente. En vez de eso, la Base de Datos de la aplicación solo accede una vista de la Base de Datos SAM para obtener los datos de las localidades.

2) Cuentas contables. Al igual que el catálogo de localidades, el catálogo que representa cuentas contables (o Subdivisiones en la Base de Datos central) se accede mediante una vista a la Base de Datos SAM.

3) Tipos de Activos. La tabla que representa los tipos de activos en la Base de Datos del SPC tiene como nombre TIPOS\_DE\_ACTIVOS. La tabla VI.5 muestra la equivalencia definida entre los campos de las Bases de Datos del SPC y de la aplicación.

4) Activos Fijos. La tabla que representa los tipos de activos en la Base de Datos del SPC tiene como nombre ACTIVOS\_FIJOS. La tabla VI.6 muestra la equivalencia definida entre los campos de las Bases de Datos del SPC y de la aplicación.

Tabla VI.5

<b>TIPOS DE ACTIVOS</b>		
<b>Campo en Base de Datos de la aplicación</b>	<b>Valor asignado al campo (desde la tabla <i>TIPOS_DE_ACTIVOS</i>)</b>	<b>Tipo de conversión</b>
Codigo	Entero positivo incremental único (1, 2, 3...)	Entero incremental
Descripción	<i>DESCRIP_TIPO_ACTIVO</i>	Ninguna
Tasa Depreciación	<i>PORCENTAJE_DEPREC</i>	Ninguna

Tabla VI.6

<b>ACTIVOS FIJOS</b>		
<b>Campo en la aplicación</b>	<b>Valor asignado al campo (desde la tabla <i>ACTIVOS_FIJOS</i>)</b>	<b>Tipo de conversión</b>
Codigo_Activo	Entero positivo incremental único (1, 2, 3...)	Entero incremental
Codigo_SPC	<i>CODIGO_DE_ACTIVO</i>	Ninguna
Correlativo_SPC	<i>MEJORA_CORRELATIVO</i>	Ninguna
Nombre	<i>DESCRIP_CORTA_ACTIVO</i>	Ninguna
FechaAdquisicion	<i>FECHA_ADQUISICION</i>	Ninguna
FechaRetiro	<i>FECHA_BAJA_ACTIVO</i>	Ninguna
Serie	<i>ACTIVOS_FIJOS_GEN.NUMERO_SERIE_ACTIVO*</i>	Ninguna
Modelo	<i>ACTIVOS_FIJOS_GEN.MODELO_DEL_ACTIVO*</i>	Ninguna
Chasis	<i>ACTIVOS_FIJOS_GEN.CHASIS_ACTIVO</i>	Ninguna
Precio	<i>VALOR_ORIG_ACT_USADO</i>	Ninguna
Estado	<i>ESTATUS</i>	*** ‘D’->4 ‘N’->4 ‘P’->2 ‘U’->2 ‘V’->4

Cod_Proveedor	NULL	Ninguna
CodResponsableA signado	NULL	Ninguna
Cod_Cuenta	CODIGO_DIVISION->SAM.SUBDIVISIONES	**
FechaRetorno	NULL	Ninguna

\*La tabla ACTIVOS\_FIJOS\_GEN contiene campos adicionales a ACTIVOS\_FIJOS. Se referencia a la tabla ACTIVOS\_FIJOS porque ambos tienen la misma llave: la concatenación de CODIGO\_DE\_ACTIVOS y MEJORA\_CORRELATIVO.

\*\*Generar una correspondencia entre los códigos de DIVISIONES de SPC y SUBDIVISIONES de SAM, y buscar el código equivalente para cada una de las Cuentas.

\*\*\*Ver Apéndices A y B

b. Actualización periódica de Base de Datos (diagrama). Así como la migración inicial llena las tablas de la Base de Datos de la aplicación con los datos iniciales, debe existir una actualización periódica hacia la Base de Datos del SPC para poder alimentar con datos reales al proceso de cálculo de depreciaciones, realizado solamente por el módulo auxiliar de Activos Fijos del SPC.

Esta actualización requiere que los datos de la Base de Datos de la aplicación sean filtrados adecuadamente para representar los mismos objetos (los Activos Fijos) en una tabla con distinta estructura y con distinta representación de sus características. Por ejemplo, aunque la Base de Datos del SPC almacena el estado de un Activo Fijo (pendiente, actualizado, donado, vendido, etc...), esta no posee campos para definir estados intermedios que representen que un Activo Fijo aun no ha concluido un proceso transaccional, como un traslado o una baja.

1) Actualizaciones a realizar. Se identificó que la información que se debe actualizar sobre la Base de Datos del SPC para que el cálculo de depreciaciones sea correcto y consistente es la siguiente:

a) Referencias. La tabla de tipos de Activos está duplicada en ambas Bases de Datos. Cuando ésta se modificara en la Base de Datos de la aplicación, sus

cambios deben reflejarse en la Base de Datos del SPC para mantener integridad referencial.

Por ende, la tabla TIPOS\_DE\_ACTIVOS del SPC debe sincronizarse con la tabla TiposActivos antes de actualizar los mismos Activos Fijos.

b) Ingreso de nuevos activos. Si se han ingresado nuevos activos desde la última actualización, estos deben ingresarse en la Base de Datos del SPC. Nótese que la información acerca del activo que debe ingresarse no es la información con que el activo se ingresó al sistema, sino que la información del estado actual del activo.

c) Traslado de activos. Todo activo que haya sido trasladado de localidad debe ser actualizado en el SPC con el campo CODIGO\_DIVISION de la tabla ACTIVOS\_FIJOS.

d) Baja de activos. Todo activo que haya sido dado de baja debe actualizarse en el SPC modificando el campo ESTATUS de la tabla ACTIVOS\_FIJOS al valor 'D', 'N', o 'V', dependiendo del motivo de la baja.

e) Activos en estados transaccionales. Si un Activo Fijo no ha concluido aún una transacción en el momento en que se decida sincronizar las Bases de Datos, el estado del Activo en el SPC debe reflejar el último estado “estable” (con transacciones completas) en que el Activo estuvo. Este estado se averigua por medio del historial de transacciones en la tabla Transacciones de la aplicación, e incluye datos como su localidad, el usuario responsable, la cuenta contable a la que se aplicará el costo de depreciación, y su estatus en el ciclo de vida de los Activos Fijos.

2) Métodos de actualización. Para cada una de las actualizaciones identificadas en la sección anterior, se identificaron procedimientos almacenados específicos que las ejecutan. La descripción de estos procedimientos almacenados y de los valores que se les envía como argumentos, se incluye en el Apéndice A.

a) Referencias. Para sincronizar las tablas TIPOS\_DE\_ACTIVOS y TiposActivos, se deben utilizar tres procedimientos almacenados, según se necesite ingresar, modificar los datos de, o eliminar algún registro de la tabla TIPOS\_DE\_ACTIVOS:

- *sp\_Alta\_TIPOS\_DE\_ACTIVOS*
- *sp\_Cambio\_TIPOS\_DE\_ACTIVOS*
- *sp\_Baja\_TIPOS\_DE\_ACTIVOS*

b) Ingreso de nuevos activos. El procedimiento almacenado que ingresa un nuevo activo es *sp\_Alta\_Activos\_FIJOS*.

c) Traslado de activos. El procedimiento almacenado que ingresa un nuevo activo es *sp\_Cambio\_Activos\_FIJOS*.

d) Baja de activos. El procedimiento almacenado que ingresa un nuevo activo es *sp\_Baja\_Activos\_FIJOS*.

e) Activos en estados transaccionales. Este tipo de actualización requiere que antes se determine el último estado actual del Activo Fijo. Una vez se haya determinado, una serie de condicionales determinará si el activo debe ingresarse, trasladarse, y/o darse de baja en el SPC. La acción resultante depende del estado del Activo Fijo en la Base de Datos del SPC y en la Base de Datos de la aplicación.

Los algoritmos completos en detalle que sincronizan ambas Bases de Datos se explican con mayor detalle en el Apéndice C.

## **IX. CONCLUSIONES**

1. Se realizó un análisis completo del sistema de control de Activos Fijos, y con éste se identificaron sus funciones.
2. Basados en los requisitos del proyecto, se determinó que el nuevo sistema debe estar compuesto por una aplicación distribuida y remota multi-usuario, una Base de Datos con soporte para estados transaccionales, y un sistema de interacción entre esta Base de Datos y la Base de Datos del sistema SPC de la empresa.
3. La nueva aplicación desarrollada cumple los requisitos definidos por el análisis del sistema original de activos fijos.
4. La implementación del nuevo sistema contendrá toda la funcionalidad ofrecida por el sistema actual.
5. El sistema de interacción con el sistema SPC debe implementarse como un módulo de la nueva aplicación desarrollada, en el mismo lenguaje de programación Visual Basic .NET.
6. El diseño del sistema de interacción soluciona las incompatibilidades y diferencias de codificación, y crea una unión entre ambas Bases de Datos.

## **X. RECOMENDACIONES**

1. Implementar el diseño presentado en este reporte adjunto a la aplicación ya desarrollada.
2. Antes de poner el sistema en producción, realizar pruebas extensivas en una Base de Datos de pruebas para verificar su funcionalidad y la integridad de los datos.
3. Una vez se implemente y pruebe el sistema, poblar los catálogos de la Base de Datos de la aplicación con datos reales, como los usuarios y proveedores.
4. Colocar el sistema en producción en el servidor de aplicaciones.
5. Capacitar al personal de la empresa en el correcto uso del sistema.
6. Mantener las Bases de Datos de la aplicación y del sistema SAM actualizados.

## XI. BIBLIOGRAFÍA

### A. Libros consultados

1. Date, C.J. 2001. *Introducción a los Sistemas de Bases de Datos*. 7ª. ed. Pearson Educación México. 936 págs.
2. Pressman, Roger. *Ingeniería del Software. Un enfoque práctico*. 5ª ed. McGrawHill España. 601 págs.
3. Russel, Charlie y Crawford Sharon. 2000 *Running Microsoft Windows 2000 Server*. 1ª. ed. Washington, E.E.U.U. 1212 págs.
4. Melton, Jim. Agosto, 2003. *(ISO-ANSI Working Draft) Persistent Stored Modules (SQL/PSM)*. **WG3:HBA-005 H2-2003-307**. ISO (International Organization for Standardization), ANSI (American National Standards Institute). 172 págs.
5. Alvarez Feldmar, Julio Manuel. 2001. *DESARROLLO DE UN SISTEMA INTEGRADOR DE ENSEÑANZA EN LA UNIVERSIDAD DEL VALLE DE GUATEMALA VÍA INTERNET: NÚCLEO FUNCIONAL*". Universidad del Valle de Guatemala, Guatemala. 107 págs.

### B. Consultas en línea

1. MSDN Library. *SQL Server Architecture. Features of SQL Server 2000*. <[http://msdn.microsoft.com/library/default.asp?url=/library/en-us/architec/8\\_ar\\_00\\_11bn.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/architec/8_ar_00_11bn.asp)> [Consulta: Septiembre 2006]
2. Spongsveen, Kian. 2003. *Sybase Adaptive Server Enterprise HOWTO*. <<http://www.tldp.org/HOWTO/Sybase-ASE-HOWTO/introduction.html>> [Consulta: Septiembre 2006]
3. Ispirer Systems. *SQL Ways. Microsoft SQL Server and Sybase Data Type Differences*. <<http://www.ispirer.com/doc/sqlways38/Output/SQLWays-1-135.html>> [Consulta: Septiembre 2006]
4. Janmohamed, Zamil; Liu, Clara; Bradstock, Drew; Chong, Raul F.; Gao, Michael; McArthur, Fraser; Yip, Paul. *DB2® SQL PL: Essential Guide for DB2® UDB on Linux™, UNIX®, Windows®, i5/OS™, and z/OS®*. 2a Edición.

<[http://www.ibmpressbooks.com/content/images/0131477005/samplechapter/0131477005\\_ch01.pdf#search=%22%22history%20of%20stored%20procedures%22%22](http://www.ibmpressbooks.com/content/images/0131477005/samplechapter/0131477005_ch01.pdf#search=%22%22history%20of%20stored%20procedures%22%22)>

[Consulta: Septiembre 2006]

5. Microsoft Developer Network [E.E.U.U.] <<http://msdn2.microsoft.com>>

[Consulta: Septiembre 2006]

## **XII. GLOSARIO**

- Base de Datos: Conjunto de datos persistentes que es utilizado por los sistemas de aplicación de alguna empresa dada.
- Base de Datos Relacional: Base de Datos basada en la teoría formal denominada modelo relacional.
- Bitácora: Conjunto de los registros sobre las transacciones que se realizan en un sistema
- Clave candidata: Superclave para la cual ningún subconjunto propio es superclave.
- CPU (Central Processing Unit): Unidad donde se ejecutan las instrucciones de los programas y se controla el funcionamiento de los distintos componentes de la computadora. Suele estar integrada en un chip denominado microprocesador.
- Cuello de botella: “Embotellamiento” de los paquetes que circulan por una conexión por algún motivo. Este fenómeno suele causar pérdidas o demora de la información.
- DBMS (Database Management System): Sistema de administración de base de datos.
- Dependencia funcional: Restricción al conjunto de relaciones legales, que permiten expresar hechos acerca de la empresa que se modela con la base de datos.
- Esquema de relaciones: Diseño global de la Base de Datos.
- Modelo relacional: Modelo de base de datos, según el cual los datos están representados como filas de tablas y cuentan con operadores que manejan directamente el proceso de inferir proposiciones verdaderas adicionales a partir de las ya dadas.
- SPC: Sistema de Planificación y Control
- Superclave: Conjunto de uno o más atributos que, considerados conjuntamente, permiten identificar de forma única a una entidad en el conjunto de entidades.
- Vista: Cualquier relación que no es parte del modelo conceptual, pero se hace visible al usuario como una “relación virtual”.

## XIII. APÉNDICES

### A. Definición de Base de Datos SPC

#### 1. Estructura

TABLAS	DESCRIPCION	SIS ORIGEN
ACTIVOS_FIJOS	MAF - Maestro de Activos Fijos	ACTI
ACTIVOS_FIJOS_GEN	AFG - Informacion Gral. de Activos Fijos	ACTI
TIPOS_DE_ACTIVOS	TAC - Tipos de Activos Fijos	ACTI

#### a. ACTIVOS\_FIJOS

TABLA: ACTIVOS\_FIJOS - MAF - Maestro de Activos Fijos

CAMPO	NOMBRE DEL CAMPO	TIPO	LONG	DEC	LLAVE	NULO	COD
CODIGO_DE_ACTIVO	CODIGO ACTIVO FIJO - OPCIONAL	char	15	0	PK		***
MEJORA_CORRELATIVO	CODIGO DE ACTIVO FIJO - OPCIONAL	int	4	0	PK		***
CODIGO_PROVEEDOR	CODIGO DE PROVEEDOR	int	10	0		X	***
CODIGO_TIPO_ACTIVO	TIPO DE ACTIVO SELECCIONADO PA REVALUAR	char	5	0			***
CODIGO_DIVISION	CODIGO DIVISION O DEPARTAMENTO	char	10	0		X	***
DESCRIP_CORTA_ACTIVO	DESCRIPCION CORTA DEL ACTIVO	varchar	40	0		X	***
VALOR_ORIGINAL_ACTIVIV	VALOR ORIGINAL DEL ACTIVO	money	13	2			***
VALOR_ACTUAL_ACTIVIVO	VALOR ACTUAL DEL ACTIVO	money	13	2			***
VALOR_RESCATE_ACTIVIVO	VALOR DE RESCATE DEL ACTIVO	money	13	2			***
TIPO_DEPRECIACION	TIPO DE DEPRECIACION A REALIZAR A ACTIVO	char	1	0			***
TIEMPO_DEPRECIACION	TIEMPO A EJECUTAR DEPRECIACION (MESES)	int	5	0			***
PORCENTAJE_ANUAL_DEP	PORCENTAJE ANUAL DE DEPRECIACION	real	8	4			***
FECHA_ADQUISICION	FECHA ADQUISICION	datetime	20	0			***
FECHA_BAJA_ACTIVIVO	FECHA DE BAJA DEL ACTIVO	datetime	20	0		X	***
VALOR_DEP_ACUM_ACT	VALOR DEPRECIACION ACUMULADA DE ACTIVO	money	13	2			***
MONTO_ULTIMA_REVAL	MONTO DE LA ULTIMA REVALUACION	money	13	2			***
FECHA_ULTIMA_REVAL	FECHA DE LA ULTIMA REVALUACION	datetime	20	0		X	***
MONTO_ULTIMA_DEPREC	MONTO DE LA ULTIMA DEPRECIACION	money	13	2			***
FECHA_ULTIMA_DEPREC	FECHA DE LA ULTIMA DEPRECIACION	datetime	20	0		X	***
USUARIO_OPERACION	USUARIO OPERACION	char	10	0			***
FECHA_OPERACION	FECHA OPERACION	datetime	20	0			***
ESTATUS	ESTADO EN QUE SE ENCUENTRA EL ACTIVO	char	1	0			***
TASA_CAMBIO_COMPRA	TASA DE CAMBIO EN COMPRA DEL ACTIVO	float	13	5			***
TASA_CAMBIO_ULT_REV	TASA DE CAMBIO ULTIMA REVALUACION	float	13	5			***
CODIGO_DE_MONEDA	Moneda	char	2	0			***
FECHA_ULT_INVENTARIO	FECHA DEL ULTIMO INVENTARIO	datetime	20	0		X	***
NO_ORDEN	NUMERO DE ORDEN DE COMPRA	int	10	0		X	***
USR_ACTUALIZACION	USUARIO QUE ACTUALIZO ACTIVO	char	10	0		X	***
FEC_ACTUALIZACION	FECHA DE ACTUALIZACION	datetime	20	0		X	***
SALDO_MESES_DEPRE	MESES RESTANTES DE DEPRECIACION	int	3	0			***
FECHA_ULT_MANTMTO	FECHA ULTIMO MANTENIMIENTO	datetime	20	0		X	***
COSTO_ACUM_MNTMTO	COSTO ACUMULADO MANTENIMIENTO	money	14	2			***
FRECUENCIA_MNTMTO	FRECUENCIA MANTENIMIENTO EN DIAS	int	5	0			***
ANO_PERIODO	AÑO DEL PERIODO CONTABLE	int	4	0			***
NO_PERIODO	NUMERO DE PERIODO CONTABLE	int	2	0			***
DEPRECIABLE	EL ACTIVO ES DEPRECIABLE S/N	char	1	0			***
ESTADO_SERVICIO	MARCA SI ESTA EN USO O EN MANTENIMIENTO	char	1	0			***
VALOR_ORIG_ACT_USADO	VALOR ORIGINAL DEL ACTIVO COMPRADO USADO	money	14	2		X	***
TASA_DEDUCIBLE_ISR	PORC DEDUCIBLE ISR SOBRE VALOR ORIGINAL	float			7	4	X ***

VALOR_NO_DEDUCIBLE	VALOR DEL ACTIVO NO DEDUCIBLE	money	14	2	X	***
CODIGO_DE_BARRAS	CODIGO DE BARRAS QUE IDENTIFICA ACTIVO	varchar	30	0	X	***
VALOR_MONEDA_LOCAL	VALOR DEL ACTIVO EN MONEDA LOCAL	money	14	2	X	***
REFERENCIA	REFERENCIA	varchar	15	0	X	***
FECHA_INICIO_DEPREC	FECHA DE INICIO DE DEPRECIACION	datetime	20	0	X	***
CODIGO_MARCA	CODIGO DE MARCA DE PRODUCTOS	char	5	0	X	***
RESPONSABLE_ID	IDENTIFICADOR DEL RESPONSABLE DEL ACTIVO	char	5	0	X	***
CATEGORIA_ID	IDENTIFICADOR DE CATEGORIA DE ACTIVO	char	5	0	X	***

TABLAS PADRES	DESCRIPCION DE LA TABLA	DEP MENSAJE DE BORRADO
ACT_CATEGORIAS	CDA - Categorías de activos fijos	Ref ACT_CATEGORIAS tiene ACTIVOS_FIJOS
ACT_RESPONSABLES	RAF - Responsables de Activos Fijos	Ref ACT_RESPONSABLES tiene ACTIVOS_FIJOS
DIVISIONES	DIV - Divisiones	Ref DIVISION UTILIZADA EN ACTIVOS_FIJOS
MARCAS	MAR Marcas	Ref MARCAS tiene ACTIVOS_FIJOS
MONEDAS	MON - Monedas	Ref MONEDAS tiene ACTIVOS_FIJOS
ORDENES_MAESTRO	ORD Maestro de Ordenes	Ref ORDENES_MAESTRO tiene ACTIVOS_FIJOS
PERIODOS_CONTABLES	PER Tabla de Periodos Contables	Ref PERIODOS_CONTABLES tiene ACTIVOS_FIJOS
PROVEEDORES	PRV - Proveedores	Ref PROVEEDOR UTILIZADO EN ACTIVOS_FIJOS
TIPOS_DE_ACTIVOS	TAC - Tipos de Activos Fijos	Ref TIPO DE ACTIVO UTILIZADO EN ACTIVOS

TABLAS HIJAS	DESCRIPCION DE LA TABLA	DEP MENSAJE DE BORRADO
ACTIVOS_FIJOS_GEN	AFG - Informacion Gral. de Activos Fijos	Dep ACTIVOS_FIJOS tiene ACTIVOS_FIJOS_GEN
ACT_CAMPOS_VAL	VAC - Valor en activo campos definidos	Dep ACTIVOS_FIJOS tiene ACT_CAMPOS_VAL
ACT_TRASLADO_DET	DTA - Detalle traslado activos fijos	Dep ACTIVOS_FIJOS tiene ACT_TRASLADO_DET
LINEA_GARANTIA_ACT	LGA- Activos que garantizan linea	Dep ACTIVOS_FIJOS tiene LINEA_GARANTIA_ACT
NEGOCIO_PROPUUESTA	PDN - Propuesta de negocio	Ref ACTIVOS_FIJOS tiene NEGOCIO_PROPUUESTA
PLANTA_MAQUINA	PLM - Maquinas por planta	Ref ACTIVOS_FIJOS tiene PLANTA_MAQUINA
REVALUACION_DET	RVD - Detalle de la revaluacion activos	Dep ACTIVOS_FIJOS tiene REVALUACION_DET
SEGUROS_DET	SED - Activos Asegurados por la Poliza	Dep ACTIVOS_FIJOS tiene SEGUROS_DET
SERVICIOS_DE_ACTIVOS	SEA - Servicios de Activos	Dep ACTIVOS_FIJOS tiene SERVICIOS_DE_ACTIVOS
TALLER_MAQUINA	MXT - Maquinas utilizadas por taller	Ref ACTIVOS_FIJOS tiene TALLER_MAQUINA

INDICE: 00 ACCESO PRINCIPAL NO permite duplicados

CAMPOS: CODIGO\_DE\_ACTIVO CODIGO ACTIVO FIJO - OPCIONAL  
 MEJORA\_CORRELATIVO CODIGO DE ACTIVO FIJO - OPCIONAL

CAMPO: DEPRECIABLE EL ACTIVO ES DEPRECIABLE S/N  
 VALORES: N No es Depreciable  
 S Si es Depreciable

CAMPO: ESTADO\_SERVICIO MARCA SI ESTA EN USO O EN MANTENIMIENTO  
 VALORES: M MANTENIMIENTO  
 U USO

CAMPO: ESTATUS ESTADO EN QUE SE ENCUENTRA EL ACTIVO  
 VALORES: D Desechado  
 N Donado  
 P Pendiente  
 U Actualizado  
 V Vendido

CAMPO: TIPO\_DEPRECIACION TIPO DE DEPRECIACION A REALIZAR A ACTIVO  
 VALORES: D Doble Saldo Declinante  
 L Línea Recta  
 S Suma de Años Digitos

## b. ACTIVOS\_FIJOS\_GEN

TABLA: ACTIVOS\_FIJOS\_GEN - AFG - Informacion Gral. de Activos Fijos

CAMPO	NOMBRE DEL CAMPO	TIPO	LONG	DEC	LLAVE	NULO	COD
-----	-----	----	----	----	----	----	----
CODIGO_DE_ACTIVO	CODIGO ACTIVO FIJO - OPCIONAL	char	15	0	PK		***
MEJORA_CORRELATIVO	CODIGO DE ACTIVO FIJO - OPCIONAL	int	4	0	PK		***
DESCRIP_LARGA_UNO	DESCRIPCION LARGA UNO	varchar	60	0			***
DESCRIP_LARGA_DOS	DESCRIPCION LARGA DOS	varchar	60	0		X	***
DESCRIP_LARGA_TRES	DESCRIPCION LARGA TRES	varchar	60	0		X	***
MARCA	MARCA DEL ACTIVO	varchar	30	0		X	***
NUMERO_SERIE_ACTIVO	NUMERO DE SERIE DEL ACTIVO	varchar	30	0		X	***
MODELO_DEL_ACTIVO	MODELO DEL ACTIVO	varchar	15	0		X	***
ESTILO_DEL_ACTIVO	ESTILO_DEL_ACTIVO	varchar	15	0		X	***
CHASIS_ACTIVO	CHASIS_ACTIVO	varchar	30	0		X	***
UBICACION_ACTIVO	UBICACION DEL ACTIVO	char	15	0		X	***
RESPONSABLE_ACTIVO	USUARIO RESPONSABLE DEL ACTIVO	varchar	30	0		X	***
DESC_GARANTIA_1	1ERA. DESCRIPCION GARANTIA DEL ACTIVO	varchar	60	0		X	***
DESC_GARANTIA_2	2DA. DESCRIPCION GARANTIA DEL ACTIVO	varchar	60	0		X	***
DESC_GARANTIA_3	3RA. DESCRIPCION GARANTIA DEL ACTIVO	varchar	60	0		X	***
DESC_GARANTIA_4	4TA. DESCRIPCION GARANTIA DEL ACTIVO	varchar	60	0		X	***

TABLAS PADRES                      DESCRIPCION DE LA TABLA                      DEP MENSAJE DE BORRADO

-----                      -----                      -----

ACTIVOS\_FIJOS                      MAF - Maestro de Activos Fijos                      Dep ACTIVOS\_FIJOS tiene

ACTIVOS\_FIJOS\_GEN

INDICE: 00 ACCESO PRINCIPAL                      NO permite duplicados

CAMPOS: CODIGO\_DE\_ACTIVO                      CODIGO ACTIVO FIJO - OPCIONAL  
 MEJORA\_CORRELATIVO                      CODIGO DE ACTIVO FIJO - OPCIONAL

## c. TIPOS\_DE\_ACTIVOS

TABLA: TIPOS\_DE\_ACTIVOS - TAC - Tipos de Activos Fijos

CAMPO	NOMBRE DEL CAMPO	TIPO	LONG	DEC	LLAVE	NULO	COD
-----	-----	----	----	----	----	----	----
CODIGO_TIPO_ACTIVO	TIPO DE ACTIVO SELECCIONADO PA REVALUAR	char	5	0	PK		***
DESCRIP_TIPO_ACTIVO	DESCRIPCION DEL TIPO DE ACTIVO	char	40	0			***
TIPO_DEPRECIACION	TIPO DE DEPRECIACION A REALIZAR A ACTIVO	char	1	0			***
TIEMPO_DEPRECIACION	TIEMPO A EJECUTAR DEPRECIACION (MESES)	int	5	0			***
PORCENTAJE_DEPREC	PORCENTAJE_DEPREC	float	8	4		X	***
MONTO_MINIMO_ACTIVO	MONTO MINIMO DE ACTIVO PARA PODER DEPREC	money	13	2			***
FORZAR_FECHA_ACTIVO	DIAS PARA FORZAR FECHA DE DEPRECIACION	int	2	0			***
TIPO_CATALOGO	TIPO DE CATALOGO CONTABLE	char	3	0		X	***
CODIGO_CENTRO_ACTIVO	CENTRO RESPONSAB. QUE PERTENECE ACTIVO	char	15	0		X	***
CTA_ACTIVO	CUENTA DE ACTIVO	char	30	0		X	***
CTA_DEPREC_ACUM	CUENTA DE DEPRECIACION ACUMULADA	char	30	0		X	***
COD_CENTRO_GASTO_ACT	CODIGO DEL CENTRO DE GASTOS DEL ACTIVO	char	15	0		X	***
CTA_GASTO_DEPREC	CUENTA DE GASTO DE DEPRECIACION	char	30	0		X	***
COD_CENTRO_RESULTADO	CODIGO CENTRO INGRESOS EGRESOS	varchar	15	0		X	***
CTA_GAN_VTA_ACTIVO	CUENTA DE GANANCIA EN VENTA ACTIVOS	char	30	0		X	***
CTA_PER_VTA_ACTIVO	CUENTA DE PERDIDA EN VENTA DE ACTIVOS	char	30	0		X	***
COD_CENTRO_COMPRA	CODIGO CENTRO DE COMPRA	varchar	15	0		X	***
CTA_COMPRA_ACTIVO	CUENTA CONT. POR COMPRA/MEJORA DE ACTIVO	char	30	0		X	***
COD_CENTRO_BAJA	CODIGO CENTRO PARA BAJA DE ACTIVO	varchar	15	0		X	***
CTA_BAJA_ACTIVO	CUENTA CONTABLE POR BAJA DEL ACTIVO	char	30	0		X	***
COD_CENTRO_REVALUA	CODIGO CENTRO PARA REVALUACION ACTIVO	varchar	15	0		X	***



```

    @CODIGO_DE_ACTIVIVO char(15) = ' '
, @MEJORA_CORRELATIVO int = null
, @CODIGO_PROVEEDOR int = null
, @CODIGO_TIPO_ACTIVIVO char(5) = ' '
, @CODIGO_DIVISION char(10) = null
, @DESCRIP_CORTA_ACTIVIVO varchar(40) = null
, @VALOR_ORIGINAL_ACTIVIVO money = 0
, @VALOR_ACTUAL_ACTIVIVO money = 0
, @VALOR_RESCATE_ACTIVIVO money = 0
, @TIPO_DEPRECIACION char(1) = ' '
, @TIEMPO_DEPRECIACION int = 0
, @PORCENTAJE_ANUAL_DEP real = 0
, @FECHA_ADQUISICION datetime = ' '
, @FECHA_BAJA_ACTIVIVO datetime = null
, @VALOR_DEP_ACUM_ACT money = 0
, @MONTO_ULTIMA_REVAL money = 0
, @FECHA_ULTIMA_REVAL datetime = null
, @MONTO_ULTIMA_DEPREC money = 0
, @FECHA_ULTIMA_DEPREC datetime = null
, @USUARIO_OPERACION char(10) = ' '
, @FECHA_OPERACION datetime = ' '
, @ESTATUS char(1) = ' '
, @TASA_CAMBIO_COMPRA float = 0
, @TASA_CAMBIO_ULT_REV float = 0
, @CODIGO_DE_MONEDA char(2) = ' '
, @FECHA_ULT_INVENTARIO datetime = null
, @NO_ORDEN int = null
, @USR_ACTUALIZACION char(10) = null
, @FEC_ACTUALIZACION datetime = null
, @SALDO_MESES_DEPRE int = 0
, @FECHA_ULT_MANTMTO datetime = null
, @COSTO_ACUM_MNTMTO money = 0
, @FRECUENCIA_MNTMTO int = 0
, @ANO_PERIODO int = 0
, @NO_PERIODO int = 0
, @DEPRECIABLE char(1) = ' '
, @ESTADO_SERVICIO char(1) = ' '
, @VALOR_ORIG_ACT_USADO money = 0
, @TASA_DEDUCIBLE_ISR float = 0
, @VALOR_NO_DEDUCIBLE money = 0
, @CODIGO_DE_BARRAS varchar(30) = null
, @VALOR_MONEDA_LOCAL money = 0
, @REFERENCIA Varchar(15) = null
, @FECHA_INICIO_DEPREC Datetime = null
, @CODIGO_MARCA char(5) = null
, @RESPONSABLE_ID char(5) = null
, @CATEGORIA_ID char(5) = null
as
/*--CODIGO DEL PROCEDIMIENTO ALMACENADO--*/

```

## b. sp\_CAMBIO\_ACTIVOS\_FIJOS

```

create procedure sp_Cambio_ACTIVOS_FIJOS
    @CODIGO_DE_ACTIVIVO char(15) = ' ',

```

```

@MEJORA_CORRELATIVO    int          = 0,
@CODIGO_PROVEEDOR      int          = null,
@CODIGO_TIPO_ACTIVOS   char(5)      = ' ',
@CODIGO_DIVISION       char(10)     = null,
@DESCRIP_CORTA_ACTIVOS varchar(40) = null,
@VALOR_ORIGINAL_ACTIVOS money       = 0,
@VALOR_ACTUAL_ACTIVOS  money       = 0,
@VALOR_RESCATE_ACTIVOS money       = 0,
@TIPO_DEPRECIACION     char(1)      = ' ',
@TIEMPO_DEPRECIACION   int          = 0,
@PORCENTAJE_ANUAL_DEP  real         = 0,
@FECHA_ADQUISICION     datetime     = '',
@FECHA_BAJA_ACTIVOS    datetime     = null,
@VALOR_DEP_ACUM_ACT    money       = 0,
@MONTO_ULTIMA_REVAL    money       = 0,
@FECHA_ULTIMA_REVAL    datetime     = null,
@MONTO_ULTIMA_DEPREC   money       = 0,
@FECHA_ULTIMA_DEPREC   datetime     = null,
@USUARIO_OPERACION     char(10)     = ' ',
@FECHA_OPERACION       datetime     = ' ',
@ESTATUS               char(1)      = ' ',
@TASA_CAMBIO_COMPRA    float        = 0,
@TASA_CAMBIO_ULT_REV    float        = 0,
@CODIGO_DE_MONEDA      char(2)      = ' ',
@FECHA_ULT_INVENTARIO  datetime     = null,
@NO_ORDEN              int          = null,
@USR_ACTUALIZACION     char(10)     = null,
@FEC_ACTUALIZACION     datetime     = null,
@SALDO_MESES_DEPRE     int          = 0,
@FECHA_ULT_MANTMTO     datetime     = null,
@COSTO_ACUM_MNTMTO     money       = 0,
@FRECUENCIA_MNTMTO     int          = 0,
@ANO_PERIODO           int          = 0,
@NO_PERIODO            int          = 0,
@DEPRECIABLE           char(1)      = ' ',
@ESTADO_SERVICIO       char(1)      = ' ',
@VALOR_ORIG_ACT_USADO  money       = 0,
@TASA_DEDUCIBLE_ISR    float        = 0,
@VALOR_NO_DEDUCIBLE    money       = 0,
@CODIGO_DE_BARRAS      varchar(30) = null,
@VALOR_MONEDA_LOCAL    money       = 0,
@REFERENCIA            varchar(15) = null,
@FECHA_INICIO_DEPREC   datetime     = null,
@CODIGO_MARCA          char(5)      = null,
@RESPONSABLE_ID        char(5)      = null,
@CATEGORIA_ID          char(5)      = null,
@timestamp             timestamp    = null
as
/*--CODIGO DEL PROCEDIMIENTO ALMACENADO--*/

```

### c. sp\_BAJA\_ACTIVOS\_FIJOS

```

create procedure sp_Baja_ACTIVOS_FIJOS
    @CODIGO_DE_ACTIVOS char(15) = ' '

```

```

, @MEJORA_CORRELATIVO int = null
, @CODIGO_PROVEEDOR int = null
, @CODIGO_TIPO_ACTIVOS char(5) = ' '
, @CODIGO_DIVISION char(10) = null
, @DESCRIP_CORTA_ACTIVOS varchar(40) = null
, @VALOR_ORIGINAL_ACTIVOS money = 0
, @VALOR_ACTUAL_ACTIVOS money = 0
, @VALOR_RESCATE_ACTIVOS money = 0
, @TIPO_DEPRECIACION char(1) = ' '
, @TIEMPO_DEPRECIACION int = 0
, @PORCENTAJE_ANUAL_DEP real = 0
, @FECHA_ADQUISICION datetime = ' '
, @FECHA_BAJA_ACTIVOS datetime = null
, @VALOR_DEP_ACUM_ACT money = 0
, @MONTO_ULTIMA_REVAL money = 0
, @FECHA_ULTIMA_REVAL datetime = null
, @MONTO_ULTIMA_DEPREC money = 0
, @FECHA_ULTIMA_DEPREC datetime = null
, @USUARIO_OPERACION char(10) = ' '
, @FECHA_OPERACION datetime = ' '
, @ESTATUS char(1) = ' '
, @TASA_CAMBIO_COMPRA float = 0
, @TASA_CAMBIO_ULT_REV float = 0
, @CODIGO_DE_MONEDA char(2) = ' '
, @FECHA_ULT_INVENTARIO datetime = null
, @NO_ORDEN int = null
, @USR_ACTUALIZACION char(10) = null
, @FEC_ACTUALIZACION datetime = null
, @SALDO_MESES_DEPRE int = 0
, @FECHA_ULT_MANTMTO datetime = null
, @COSTO_ACUM_MNTMTO money = 0
, @FRECUENCIA_MNTMTO int = 0
, @ANO_PERIODO int = 0
, @NO_PERIODO int = 0
, @DEPRECIABLE char(1) = ' '
, @ESTADO_SERVICIO char(1) = ' '
, @VALOR_ORIG_ACT_USADO money = 0
, @TASA_DEDUCIBLE_ISR float = 0
, @VALOR_NO_DEDUCIBLE money = 0
, @CODIGO_DE_BARRAS varchar(30) = null
, @VALOR_MONEDA_LOCAL money = 0
, @REFERENCIA Varchar(15) = null
, @FECHA_INICIO_DEPREC Datetime = null
, @CODIGO_MARCA char(5) = null
, @RESPONSABLE_ID char(5) = null
, @CATEGORIA_ID char(5) = null
as
/*--CODIGO DEL PROCEDIMIENTO ALMACENADO--*/

```

#### d. sp\_ALTA\_TIPOS\_DE\_ACTIVOS

```
create procedure sp_Alta_TIPOS_DE_ACTIVOS
```

```

@CODIGO_TIPO_ACTIVO char(5) = null,
@DESCRIP_TIPO_ACTIVO char(40) = ' ',
@TIPO_DEPRECIACION char(1) = ' ',
@TIEMPO_DEPRECIACION int = 0,
@MONTO_MINIMO_ACTIVO money = 0,
@FORZAR_FECHA_ACTIVO int = 0,
@TIPO_CATALOGO char(3) = null,
@CODIGO_CENTRO_ACTIVO char(15) = null,
@CTA_ACTIVO char(30) = null,
@CTA_DEPREC_ACUM char(30) = null,
@COD_CENTRO_GASTO_ACT char(15) = null,
@CTA_GASTO_DEPREC char(30) = null,
@COD_CENTRO_RESULTADO varchar(15) = null,
@CTA_GAN_VTA_ACTIVO char(30) = null,
@CTA_PER_VTA_ACTIVO char(30) = null,
@COD_CENTRO_COMPRA varchar(15) = null,
@CTA_COMPRA_ACTIVO char(30) = null,
@COD_CENTRO_BAJA varchar(15) = null,
@CTA_BAJA_ACTIVO char(30) = null,
@COD_CENTRO_REVALUA varchar(15) = null,
@CTA_REVALUA_ACTIVO char(30) = null,
@COD_CENTRO_VENTA varchar(15) = null,
@CTA_VENTA_ACTIVO char(30) = null,
@CLA_CONTABILIZACION char(1) = ' ',
@CALCULO_PRIMERA_DEPR varchar(1) = 'M',
@COD_CENTRO_DONACION char(15) = null,
@CTA_DONACION_ACTIVO char(30) = null,
@COD_CENTRO_ACT_NODED char(15) = null,
@CTA_ACTIVO_NODEDUCIB char(30) = null,
@FORMA_DEPRECIACION char(1) = null,
@NO_DIAS_ANIO int = null
as
/*--CODIGO DEL PROCEDIMIENTO ALMACENADO--*/

```

### e. sp\_CAMBIO\_TIPOS\_DE\_ACTIVOS

```

create procedure sp_Cambio_TIPOS_DE_ACTIVOS
@CODIGO_TIPO_ACTIVO char(5) = ' ',
@DESCRIP_TIPO_ACTIVO char(40) = ' ',
@TIPO_DEPRECIACION char(1) = ' ',
@TIEMPO_DEPRECIACION int = 0,
@MONTO_MINIMO_ACTIVO money = 0,
@FORZAR_FECHA_ACTIVO int= 0,
@TIPO_CATALOGO char(3) = null,
@CODIGO_CENTRO_ACTIVO char(15) = null,
@CTA_ACTIVO char(30) = null,
@CTA_DEPREC_ACUM char(30) = null,
@COD_CENTRO_GASTO_ACT char(15) = null,
@CTA_GASTO_DEPREC char(30) = null,
@COD_CENTRO_RESULTADO varchar(15) = null,
@CTA_GAN_VTA_ACTIVO char(30) = null,
@CTA_PER_VTA_ACTIVO char(30) = null,
@COD_CENTRO_COMPRA varchar(15) = null,

```

```

@CTA_COMPRA_ACTIVO char(30) = null,
@COD_CENTRO_BAJA varchar(15) = null,
@CTA_BAJA_ACTIVO char(30) = null,
@COD_CENTRO_REVALUA varchar(15) = null,
@CTA_REVALUA_ACTIVO char(30) = null,
@COD_CENTRO_VENTA varchar(15) = null,
@CTA_VENTA_ACTIVO char(30) = null,
@CLA_CONTABILIZACION char(1) = ' ',
@CALCULO_PRIMERA_DEPR varchar(1) = 'M',
@COD_CENTRO_DONACION char(15) = null,
@CTA_DONACION_ACTIVO char(30) = null,
@COD_CENTRO_ACT_NODED char(15) = null,
@CTA_ACTIVO_NODEDUCIB char(30) = null,
@FORMA_DEPRECIACION char(1) = null,
@NO_DIAS_ANIO int = null,
@timestamp timestamp = null
as
/*--CODIGO DEL PROCEDIMIENTO ALMACENADO--*/

```

### f. sp\_BAJA\_TIPOS\_DE\_ACTIVOS

```

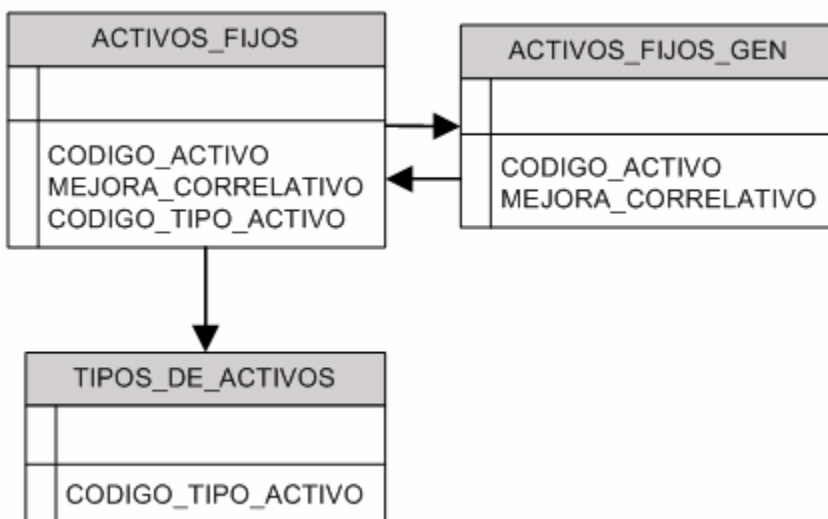
create procedure sp_Baja_TIPOS_DE_ACTIVOS
@CODIGO_TIPO_ACTIVO char(5) = ' ',
@DESCRIP_TIPO_ACTIVO char(40) = ' ',
@TIPO_DEPRECIACION char(1) = ' ',
@TIEMPO_DEPRECIACION int = 0,
@MONTO_MINIMO_ACTIVO money = 0,
@FORZAR_FECHA_ACTIVO int =0,
@TIPO_CATALOGO char(3) = null,
@CODIGO_CENTRO_ACTIVO char(15) = null,
@CTA_ACTIVO char(30) = null,
@CTA_DEPREC_ACUM char(30) = null,
@COD_CENTRO_GASTO_ACT char(15) = null,
@CTA_GASTO_DEPREC char(30) = null,
@COD_CENTRO_RESULTADO varchar(15) = null,
@CTA_GAN_VTA_ACTIVO char(30) = null,
@CTA_PER_VTA_ACTIVO char(30) = null,
@COD_CENTRO_COMPRA varchar(15) = null,
@CTA_COMPRA_ACTIVO char(30) = null,
@COD_CENTRO_BAJA varchar(15) = null,
@CTA_BAJA_ACTIVO char(30) = null,
@COD_CENTRO_REVALUA varchar(15) = null,
@CTA_REVALUA_ACTIVO char(30) = null,
@COD_CENTRO_VENTA varchar(15) = null,
@CTA_VENTA_ACTIVO char(30) = null,
@CLA_CONTABILIZACION char(1) = ' ',
@CALCULO_PRIMERA_DEPR varchar(1) = 'M',
@COD_CENTRO_DONACION char(15) = null,
@CTA_DONACION_ACTIVO char(30) = null,
@COD_CENTRO_ACT_NODED char(15) = null,
@CTA_ACTIVO_NODEDUCIB char(30) = null,
@FORMA_DEPRECIACION char(1) = null,
@NO_DIAS_ANIO int = null
as

```

/\*--CODIGO DEL PROCEDIMIENTO ALMACENADO--\*/

### 3. Diagrama Entidad-Relación (sólo se muestran tablas relevantes)

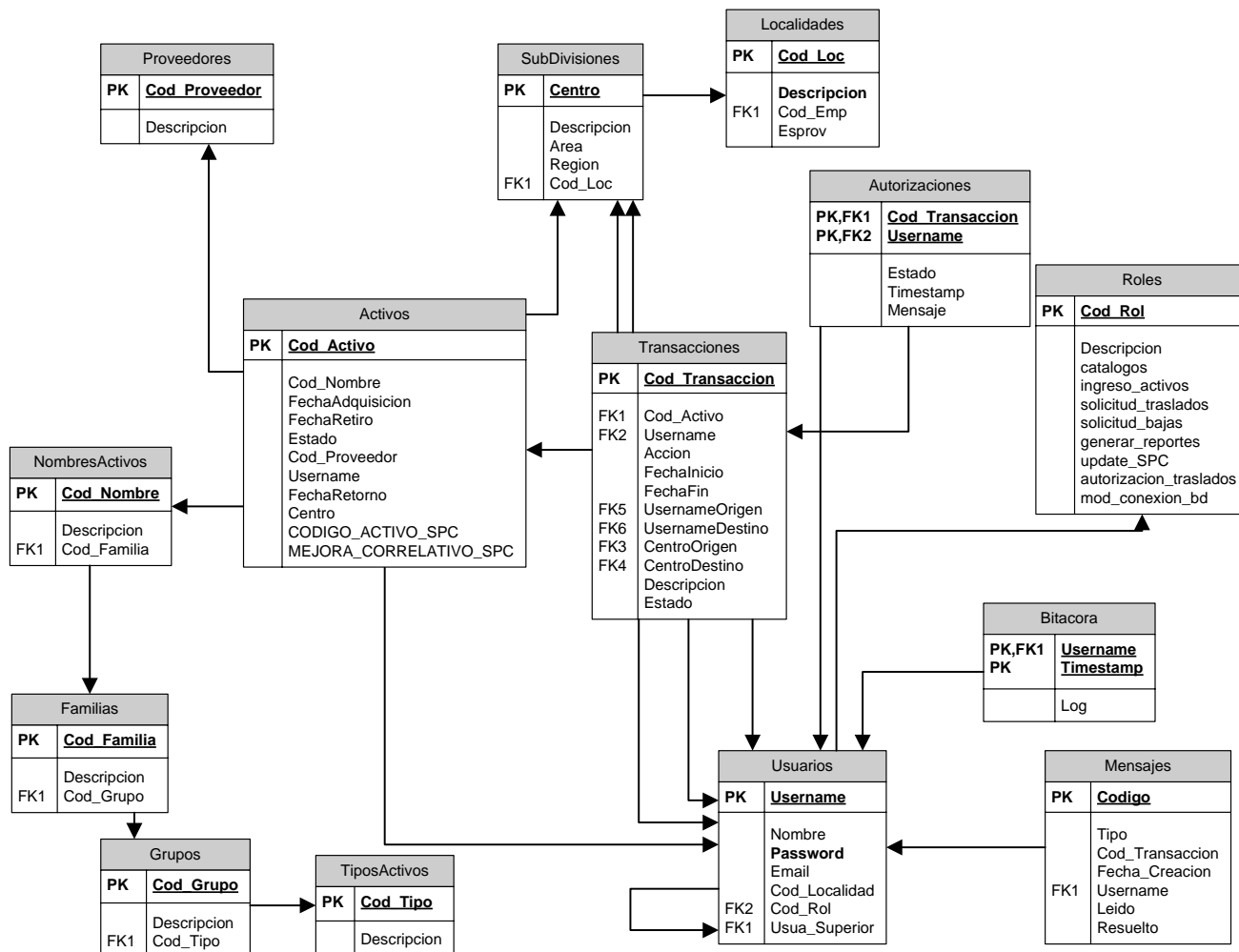
**Figura XI.1**



## B. Definición de Base de Datos de aplicación

### 1. Diagrama E-R

Figura XI.2



- Posibles valores de **ACTIVOS.ESTADO**:
  1. DE ALTA
  2. EN ESPERA DE TRASLADO O BAJA
  3. DE BAJA
- Posibles valores de **TRANSACCIONES.ACCION**:
  1. TRASLADO DE ACTIVO
  2. INGRESO DE ACTIVO
  3. BAJA DE ACTIVO
- Posibles valores de **TRANSACCIONES.ESTADO**

1. PENDIENTE
  3. CANCELADA
  4. COMPLETA
- Posibles valores de MENSAJES.TIPO:
    1. SOLICITUD DE TRASLADO A SUPERIOR
    2. SOLICITUD DE BAJA A SUPERIOR
    3. CONFIRMACIÓN DE AUTORIZACIÓN DE TRASLADO
    4. CONFIRMACIÓN DE AUTORIZACIÓN DE BAJA
    5. CANCELACIÓN DE TRASLADO
    6. CANCELACIÓN DE BAJA
    7. NUEVO ACTIVO A SU DISPOSICION
    8. RETIRO DE RESPONSABILIDAD DE ACTIVO
  - Posibles valores de AUTORIZACIONES.ESTADO:
    1. PENDIENTE
    2. AUTORIZADA
    3. DENEGADA
  - Posibles valores de MENSAJES.LEIDO: {0, 1}
  - Posibles valores de MENSAJES.RESUELTO: {0, 1}

### C. Migración y actualización de datos entre Bases de Datos

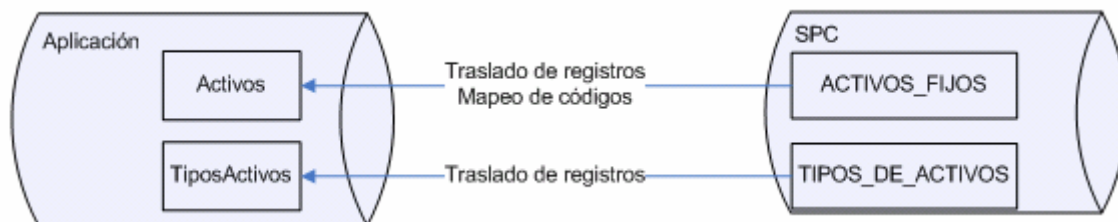
La sincronización entre la Base de Datos de la aplicación de control de movimientos de Activos Fijos y la Base de Datos del SPC consta de dos procesos independientes: la migración inicial y la sincronización periódica.

La migración inicial es un proceso a realizar una única vez, y se encarga de poblar la Base de Datos de la aplicación con todos los registros de Activos Fijos en la Base de Datos del SPC.

La sincronización periódica está diseñada para ejecutarse como un proceso auxiliar de la nueva aplicación. Su función principal es modificar los registros del SPC para que reflejen el contenido de la otra Base de Datos.

1. Migración inicial. El proceso de migración inicial se debe realizar una sola vez, antes que el nuevo sistema se ponga en marcha. Se recomienda que este proceso no sea desarrollado en conjunto con la aplicación, sino como un proceso independiente, con el fin de que se ejecute una sola vez.

**Figura XI.3**



La lógica del proceso está ilustrada en la figura XI.3. El traslado de registros de Activos Fijos requiere de un mapeo para mantener una correlación entre los códigos entre la Aplicación y el SPC. Los registros de tipos de activos no requieren de mapeo, pues el código a utilizar en cada Base de Datos es numérica y es la misma.

El traslado de campos se especifica en las tablas XI.1 y XI.2.

Tabla XI.1

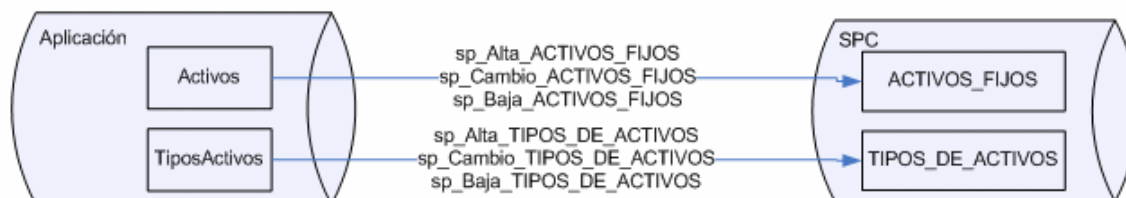
TRASLADO ( <i>ACTIVOS_FIJOS</i> => <i>Activos</i> )				
Campo <i>ACTIVOS_FIJOS</i>	Tipo de dato	Campo <i>Activos</i>	Tipo de dato	Conversión
N/A	N/A	Cod_Activo	bigint	Auto-generado
FECHA_ADQUISICION	datetime	FechaAdquisicion	smalldatetime	Ninguna
FECHA_BAJA_ACTIVIVO	datetime	FechaRetiro	smalldatetime	Ninguna
ESTATUS	char(1)	Estado	tinyint	'D' -> 4 'N' -> 4 'P' -> 2 'U' -> 2 'V' -> 4
CODIGO_DIVISION	char(10)	Centro	varchar(10)	Ninguna
CODIGO_ACTIVIVO	char(15)	CODIGO_ACTIV O_SPC	char(15)	Ninguna
MEJORA CORRELATIVO	int(4)	MEJORA_CORR ELATIVO_SPC	Int	Ninguna

Tabla XI.2

TRASLADO ( <i>TIPOS_DE_ACTIVOS</i> => <i>TiposActivos</i> )				
Campo <i>TIPOS_DE_ACTIVOS</i>	Tipo de dato	Campo <i>TiposActivos</i>	Tipo de dato	Conversión
CODIGO_TIPO_ACTIVIVO	char(5)	Codigo	bigint	String -> Int
DESCRIP_TIPO_ACTIVIVO	char(40)	Descripción	varchar(50)	Ninguna

2. Sincronización periódica. El proceso de sincronización periódica debería implementarse como un módulo auxiliar de la aplicación, de manera que un usuario con suficientes privilegios pueda ejecutarlo. La figura XI.4 ilustra la lógica del proceso.

Figura XI.4



Como muestra la figura, la transferencia de datos puede expresarse exclusivamente por medio de llamadas a procedimientos almacenados en el SPC. El proceso debe encontrar todos aquellos registros en la Base de Datos de la aplicación que hayan cambiado después de la última sincronización. El momento de la última sincronización se almacena en la configuración de la aplicación.

La lógica del proceso se describe a continuación:

- a. Seleccionar la última transacción para cada activo fijo distinto que haya ocurrido después de la última sincronización (0 o 1 transacciones por activo) y que ya haya sido finalizada (estado = 4).
- b. Para cada transacción seleccionada:
  - 1) Si es un Ingreso de Activo:
    - a) Actualizar el valor del campo `Activos.CODIGO_ACTIVOS_SPC` con el valor de `Activos.CODIGO_DIVISION`.
    - b) Seleccionar el máximo valor del campo `ACTIVOS_FIJOS.MEJORA_CORRELATIVO` entre los registros cuyo campo `CODIGO_ACTIVOS` sea igual al valor asignado en `Activos.CODIGO_ACTIVOS_SPC`.
    - c) Asignar el valor seleccionado al campo `Activos.MEJORA_CORRELATIVO_SPC`.
    - d) Llamar a `sp_Alta_ACTIVOS_FIJOS` con los argumentos adecuados.
  - 2) Si es un Traslado de Activo:
    - a) Si el registro del activo aun no existe en la Base de Datos del SPC, realizar el proceso de “Ingreso de Activo”, descrito en la sección anterior.
    - b) Llamar a `sp_Cambio_ACTIVOS_FIJOS` con los argumentos adecuados.
  - 3) Si es una Baja de Activo:
    - a) Si el registro del activo aun no existe en la Base de Datos del SPC, realizar el proceso de “Ingreso de Activo”, descrito en una sección anterior.
    - b) Llamar a `sp_Cambio_ACTIVOS_FIJOS` con los argumentos adecuados.