

Universidad Central “Marta Abreu” de Las Villas

Facultad de Matemática, Física y Computación



Diseño y Modelado de un Sistema de Adquisiciones

**Tesis presentada en opción al Título Académico
de Master en Computación Aplicada**

Autor: Ing. J. Israel De León Sánchez

Tutores: Dr. Luisa M. González González

M.Sc. Carlos E. García González

Jalisco, México 2003

RESUMEN

El objetivo principal de este trabajo, es el demostrar la efectividad y calidad con que se puede hacer un análisis y diseño de un sistema tipo Cliente-Servidor. Debido a las necesidades que el Municipio de Guadalajara tiene, es imperativo desarrollar un Sistema de Adquisiciones que cubra por lo menos los puntos mas críticos, en lo que se refiere a flujo de dinero, y lo que se intenta hacer con este trabajo es el desarrollar un modelo de base de datos óptimo, para que la información que brinde sea lo mas fidedigna y confiable posible. El modelado de la base de datos se basará en una metodología brindada por David A. Ruble[1], pues con ella se puede demostrar lo fácil y seguro que este modelo es, cuando se hace de una manera correcta. Por supuesto no será la única bibliografía mencionada, pues el modelado Cliente-Servidor es un tema muy extenso y del cual se pueden desprender muchas opciones y caminos a tomar, pero la metodología de Ruble será el centro de nuestro estudio.

ABSTRACT

The main objective of this work, is demonstrating the effectiveness and quality with which one can make an analysis and design of a system type Client-Server. Due to the necessities that the Municipality of Guadalajara has, it is imperative to develop a System of Acquisitions that covers at least the points but critical, in what refers to flow of money, and what is tried to make with this work is developing a good database model, so that the information that toasts is it but trustworthy and reliable possible. The modeling of the database will be based on a methodology toasted by David A. Ruble[1], because with her it can demonstrate himself the easy thing and for sure this model is, when one makes in a correct way. Of course it won't be the only mentioned bibliography, because the modeling Client-Server is a very extensive topic and of which can come off many options and roads to take, but the methodology of Ruble will be the center of our study.

INDICE

INTRODUCCIÓN	3
OBJETIVOS:	4
ESTRUCTURA DE LA TESIS	4
CAPÍTULO 1: REVISIÓN BIBLIOGRÁFICA	6
Antecedentes de la arquitectura Cliente/Servidor	6
¿Cuál es el mejor de los servidores?	10
Metodologías para el análisis/diseño de un sistema Cliente-Servidor	10
Generaciones de metodologías de análisis/diseño.	11
Metodología aplicada al modelado de éste sistema.	12
CAPÍTULO 2: DESCRIPCIÓN DEL PROBLEMA	14
Requerimientos del Sistema	14
Solicitudes	14
Invitaciones a Cotizar.	14
Cotizaciones	15
Cuadros Comparativos	15
Ordenes de Compra/Servicio.	16
Casos Especiales.	16
Talleres Municipales	16
Dirección Municipal de Salud.	18
Patrimonio	18
Tesorería y el presupuesto de adquisiciones.	19
CAPÍTULO 3: SOLUCIÓN (DESARROLLO DEL MODELADO DEL SISTEMA)....	21
Plan general del proyecto	21
Declaración de la meta	22
Objetivos	26
Meta	27
Modelo de Contexto	28
Modelo de Eventos	33
Lista de Eventos	34
Diccionario de Eventos	35
Modelo de Información	47
Diagrama Entidad-Relación	47
Lista de Atributos por Entidad	51
Diagramas de Transiciones de Estado	58
Matrices de Entidad	61
Modelo Arquitectónico	65
Diseño Relacional	68
CONCLUSIONES	69
REFERENCIAS BIBLIOGRAFICAS	71
BIBLIOGRAFÍA	72

INTRODUCCIÓN

El Sistema Municipal de Adquisiciones será una herramienta de gran utilidad, pues va a permitir la automatización del proceso de adquisiciones del Municipio de Guadalajara. Estará implementado principalmente en la dependencia de Proveeduría, quienes serán los dueños del proceso principal, y quienes a su vez estarán interactuando con otras dependencias como Tesorería, (quienes estarán encargados de proporcionar la validación del presupuesto dentro del sistema), con la Dirección de Patrimonio (quienes estarán encargados de inventariar los bienes patrimoniales adquiridos), con la Dirección de Taller Municipal, (para el control de la compra de refacciones y el pago de servicios de los vehículos cuando salgan a talleres externos), y la Dirección Municipal de Salud, (o Cruz Verde, para la adquisición de Medicamentos y Material de Laboratorio). La interacción será de la siguiente manera: SIMA traerá del sistema de tesorería la carga de las partidas presupuestales, para que puedan validarse dentro del sistema, si existe presupuesto suficiente o no para la adquisición de los diferentes bienes y servicios que necesita el Municipio. Así mismo, cuando se haga la adquisición de bienes muebles, el sistema estará encargado de dar de alta el inventario adquirido dentro del sistema de Patrimonio, en la parte de Bienes Muebles. También, llevará a cabo el proceso de adquisiciones de refacciones y servicios para los vehículos utilitarios del Municipio, pues este tipo de compras lleva un proceso diferente al del resto de las adquisiciones, además de estar interactuando dinámicamente con el sistema de Taller Municipal. Y ya por último, la compra de medicamentos para la Cruz Verde, pues es vital llevar un proceso de compra especial, debido a la urgencia con que se pueden necesitar los medicamentos y material de laboratorio.

Desafortunadamente, la red municipal existente no cubre a todas las dependencias involucradas en este proyecto, e incluso, el medio de transmisión no es del todo óptimo en algunas de ellas, pues una gran parte ya están conectadas a través de fibra óptica, pero en otras están a través de coaxial o incluso antena.

Este Sistema será de gran utilidad para la dependencia y para el municipio en general, pues actualmente en la dependencia de proveeduría no existe ninguno implementado, y el proceso lo llevan en papel directamente. Se intentó implementar un

software desarrollado dentro del municipio, pero sin éxito pues no existió la planeación adecuada. La opción más rápida, fue implementar un software diseñado por Gobierno del estado, pues era imperativo tener información de manera digital, mientras la creación de esta base de datos se llevaba a cabo, pues será la base del software que se diseñe próximamente.

Los objetivos de este trabajo plantean cumplir con lo siguiente:

OBJETIVOS:

- 1.- Diseñar un sistema de adquisiciones para el municipio de Guadalajara
- 2.- Diseñar la base de datos del SIMA con un modelado Cliente/Servidor utilizando la metodología sugerida por David A. Ruble

ESTRUCTURA DE LA TESIS

Para el cumplimiento de los objetivos se planeó estructurar la tesis con los siguientes capítulos:

Capítulo 1: Revisión Bibliográfica

En éste capítulo se analiza brevemente los antecedentes de los sistemas Cliente-Servidor, mostrando brevemente los conceptos básicos. Se menciona de manera rápida, como influyen los servidores en este tipo de tecnología. La necesidad de tener metodologías para el análisis y diseño es esencial y se observa que no mucha gente le da la importancia necesaria a este aspecto, por lo que se analizaron las generaciones de metodologías de análisis y diseño de sistemas y cuáles son algunas ventajas que tienen unas y otras. Por último se dará un vistazo a la metodología que Ruble propone y que se expondrá más a fondo en este trabajo.

Capítulo 2: Descripción del Problema

Este capítulo explica de manera detallada cual es la razón de que el municipio de Guadalajara tenga la inquietud de desarrollar un sistema de adquisiciones integral, detallando cual es el proceso que se lleva actualmente. Cabe destacar que es una

problemática bastante urgente, y se pretende con este trabajo iniciar un proceso de análisis y diseño mediante un sistema que controle los activos del municipio.

Capítulo 3: Solución (Análisis y Modelado del SIMA v 1.0)

Se hace un modelado inicial del SIMA siguiendo la metodología de Ruble. Este es el objetivo de la primera etapa del desarrollo del SIMA para el municipio de Guadalajara y el objetivo principal en la Tesis. Aquí cabe mencionar que no es objetivo de este trabajo llegar a la etapa de desarrollo, sino únicamente mostrar, como una primera etapa, la facilidad con que se puede hacer un análisis y diseño de la base de datos.

CAPÍTULO 1: REVISIÓN BIBLIOGRÁFICA

Antecedentes de la arquitectura Cliente/Servidor

La computación Cliente / Servidor es el procesamiento cooperativo de información de negocios mediante un conjunto de procesadores, en donde clientes múltiples geográficamente distribuidos inician peticiones que son procesadas por uno o más servidores centrales. Principalmente se usa el término cliente / servidor para describir al software que se ejecuta en más de un dispositivo de hardware para lograr una tarea de negocios. La separación del hardware es la norma para las aplicaciones cliente / servidor, aunque algunos han usado el término para describir componentes de software distintos que se comunican entre ellos aunque se ejecuten en la misma máquina. La distancia entre procesadores remotos varía desde computadoras que se encuentran en el mismo cuarto o edificio hasta aquellas que están ubicadas en diferentes edificios, diferentes ciudades o hasta repartidas por todo el planeta.

El uso más común de la arquitectura cliente / servidor es para explotar la potencia que tienen las PC para manejar la interfaz gráfica de usuario y, al mismo tiempo, proteger la integridad de los datos del negocio en una máquina *host* central. La mayor preocupación es que las aplicaciones parezcan ser de una sola pieza por complicadas que sean, para que los usuarios permanezcan sin darse cuenta cuál procesador está funcionando en cualquier momento dado. En una forma menos complicada, una opción para la arquitectura cliente / servidor involucra a varios clientes haciendo peticiones a un solo servidor, esto es, una arquitectura de hardware de dos niveles. Una arquitectura cliente / servidor de tres niveles, muestra a las máquinas cliente conectadas por medio de una red de área local a un servidor de aplicaciones local, el cual a su vez se comunica con un servidor de base de datos central. En el modelo arquitectónico de tres niveles las fronteras entre cliente y servidor comienzan a desvanecerse. La PC que aloja la aplicación de interfaz es ciertamente un cliente, y el host de base de datos central que aloja a los datos con seguridad es un servidor, pero el servidor de aplicaciones local a veces es cliente y a veces es servidor dependiendo de la dirección de las comunicaciones . ¿Cómo sería una arquitectura de n niveles?, Las PC se pueden conectar directamente al servidor de la base de datos, haciendo a un lado al servidor local.

La arquitectura de n niveles está llegando rápidamente a ser la norma en la mayoría de las organizaciones, conforme las redes de área local, las redes de área amplia, Internet y World Wide Web están entrelazadas. La distinción entre cliente puro y servidor puro casi se elimina en tales ambientes, haciendo que cliente / servidor sea más un patrón conceptual que es aplicado a cada transacción al momento de su ejecución.

La aplicación de software cliente / servidor puede ser dividida en capas. La capa de presentación se usa para manejar la mecánica de interfaz. La capa lógica del negocio contiene el código que hace cumplir las políticas del negocio, tal como se describen en la sección de actividad del modelo de eventos (que se explicará y mostrará posteriormente en este trabajo). La capa de administración de datos maneja el acceso simultáneo a la base de datos, así como la sincronización de datos distribuidos. El término *cliente pesado (servidor delgado)* se usa para describir una aplicación en términos generales en donde la mayor parte de la capa lógica del negocio se ejecuta en la maquina cliente. *Servidor pesado (cliente delgado)* describe a una aplicación en donde la parte importante del procesamiento sucede en el servidor y el cliente queda confinado principalmente a labores de presentación.

La primera generación de muchas herramientas de desarrollo cliente / servidor populares asumieron una arquitectura de software de dos niveles cuando se desarrollaron por primera vez. Las primeras versiones de paquetes tales como Power Builder®, y Visual Basic®, motivaron un enfoque de cliente completamente pesado. La lógica del negocio estaba íntimamente atada a la capa de presentación de la aplicación. Mediante la introducción de conceptos orientados a objetos, tales como la herencia, muchas herramientas han sido muy buenas para la explotación de la reutilización de estructuras de objetos, las cuales administran la presentación de interfaz. Respondiendo a las demandas del mercado sobre lenguajes de desarrollo más flexibles, las herramientas de desarrollo GUI de segunda generación reconocen la necesidad de separar a la presentación de la lógica del negocio. La tendencia en los lenguajes de desarrollo es permitir que el proyecto balancee las demandas de la aplicación contra la arquitectura de hardware [1].

En 1997, el INEI (Instituto Nacional de Estadística e Informática) de Perú, presentó los antecedentes de la arquitectura Cliente/Servidor [6]. Ahí se menciona que las computadoras personales y los paquetes de software de aplicaciones proliferan comercialmente. Estos ordenadores, también conocidos como estaciones de trabajo programables, están conectados a las Redes de Área Local (LAN), mediante las cuales, los grupos de usuarios y profesionales comparten aplicaciones y datos. Las nuevas tecnologías de distribución de funciones y datos en una red, permiten desarrollar aplicaciones distribuidas de una manera transparente, de forma que múltiples procesadores de diferentes tipos (ordenadores personales de gama baja, media y alta, estaciones de trabajo, minicomputadoras o incluso mainframes), puedan ejecutar partes distintas de una aplicación. Si las funciones de la aplicación están diseñadas adecuadamente, se pueden mover de un procesador a otro sin modificaciones, y sin necesidad de retocar los programas que las invocan. Si se elige una adecuada infraestructura de sistemas distribuidos y de herramientas de desarrollo, las aplicaciones resultantes podrán trasladarse entre plataformas de distintos proveedores. Dos años atrás (1995), aún cuando en aquel momento se hablaba mucho y se hacía muy poco sobre el tema, se decía que el desarrollo de aplicaciones Cliente/Servidor era inevitable por un conjunto de razones:

- En muchas situaciones es más eficiente que el procesamiento centralizado, dado que éste experimenta una "des-economía" de escala cuando aumenta mucho la cantidad de usuarios.
- Existían ya en ese momento servidores razonablemente eficientes y confiables.
- Se había establecido un estándar de hecho para una interface Cliente/Servidor (el ODBC SQL, adoptado por todos los fabricantes importantes de servidores).
- Era imprescindible, para apoyar con información a la creciente cantidad de ejecutivos de nivel medio que necesitan tomar decisiones ante el computador, ayudándose con las herramientas "*front office*", que utilizan con toda naturalidad (planillas electrónicas, procesadores de texto, graficadores, correos electrónicos, etc.).

Sin embargo, existía tecnología para esta arquitectura desde hacía ya bastantes años, sin que nada ocurriera. Los primeros trabajos conocidos para la arquitectura Cliente/Servidor los hizo Sybase, que se fundó en 1984 pensando en lanzar al mercado únicamente productos para esta arquitectura. A fines de la década de los 80's, el producto fue lanzado para el voluminoso segmento "*low-end*" del mercado, en conjunción con Microsoft, teniendo como soporte de la base de datos un servidor OS/2, y como herramienta "*front end*" básica el dbase IV de Ashton Tate. El dbase IV no se mostró como una herramienta adecuada, y los desencuentros comerciales entre Sybase, Microsoft e IBM (en aquel momento socia de Microsoft para el OS/2) hicieron el resto. La situación era muy diferente en 1994, cuando los principales fabricantes tradicionales (Informix, Oracle, Sybase) habían lanzado al mercado poderosos servidores y, a ellos, se agregaba IBM que estaba lanzando su producto DB2 para, prácticamente, todos los sistemas operativos importantes (además de sus clásicos MVS y VM, ahora anunciaba AIX, OS/2, Windows NT, Hewlett Packard's UNIX, Sun's UNIX, Siemens' UNIX, etc.) y Microsoft que, luego de finalizar su acuerdo con Sybase, partió para su propio SQL Server para Windows NT.

Existía un conjunto de herramientas "*front end*" como, por ejemplo, Delphi, Foxpro, Powerbuilder, SQL Windows, Visual Basic, etc. Se decía en aquel momento que Visual Basic, más allá de sus méritos intrínsecos como lenguaje, era el favorito para dominar el mercado, cosa que está ocurriendo. Por otra parte, en la comunidad informática existían muchas dudas sobre la calidad de los optimizadores de los sistemas de gerencia de base de datos, cuyas fallas del pasado habían sido causantes de verdaderas historias de horror.

¿Qué ocurrió en esos dos años?. Que los servidores se mostraron sólidos y eficientes, que sus optimizadores probaron, en general, ser excelentes. Que una cantidad muy importante de empresas, en todo el mundo, ha encarado aplicaciones Cliente / Servidor, y quienes lo están haciendo con los planes necesarios y con las herramientas adecuadas, están obteniendo éxitos muy importantes, mientras los que lo hicieron desaprensivamente, han cosechado fracasos.

¿Cuál es el mejor de los servidores?

Esta es una cuestión muy complicada. Se pueden tomar *bechmarks* publicados por cada uno de los fabricantes, o hacer los nuestros específicos, pero su importancia siempre es relativa. La respuesta, además, depende del momento en que se la formula. Para aplicaciones pequeñas y medias, todos han probado ser muy buenos, las diferencias se darán cuando se necesiten altísimos regímenes transaccionales, y dependerán de cómo cada uno vaya incorporando nuevas características como paralelismo, "*read ahead*", etc. Cada nueva versión puede modificar las posiciones y los principales fabricantes están trabajando al ritmo de una gran versión nueva por año. En general, la tecnología de los servidores de base de datos ha evolucionado mucho en los últimos años y todos los fabricantes trabajan con tecnología sensiblemente equivalente. Parecen, mucho más importantes para la elección, elementos que están fuera de la tecnología: la confianza que despierta el fabricante, su compromiso con el producto, su tendencia a mantenerse siempre actualizado, su situación económico/financiera, las garantías que brinda el soporte local y, en menor medida, el precio. Aunque inicialmente fueron los propios usuarios quienes impulsaron esta nueva tecnología, la situación ha cambiado drásticamente. Hoy en día, el modelo Cliente/Servidor se considera clave para abordar las necesidades de las empresas. El proceso distribuido se reconoce actualmente como el nuevo paradigma de sistemas de información, en contraste con los sistemas independientes. Este cambio fundamental ha surgido como consecuencia de importantes factores (negocio, tecnología, proveedores), y se apoya en la existencia de una gran variedad de aplicaciones estándar y herramientas de desarrollo, fáciles de usar que soportan un entorno informático distribuido.

Metodologías para el análisis/diseño de un sistema Cliente-Servidor.

A pesar de contar hoy en día con múltiples herramientas de desarrollo de aplicaciones en los ambientes Cliente / Servidor (i.e. herramientas para generar interfaces gráficas, herramientas para manejar datos u objetos multimediales y herramientas de conectividad), un problema que dificulta el desarrollo de sistemas de información modernos es no contar con metodologías claras para su diseño.

Las metodologías tradicionales de desarrollo de sistemas de información se quedan cortas cuando se tratan de aplicar a los sistemas que se requieren hoy en día. [1]

Generaciones de metodologías de análisis/diseño.

David Ruble[1], se refiere a las generaciones de metodologías de análisis/diseño. En los “viejos tiempos”, cuando los sistemas se desarrollaban en lenguaje ensamblador en mainframes primitivas, la suposición simplista (aunque algo irreal) era que los desarrolladores no tenían ninguna metodología. Luego, desde mediados de los setenta hasta finales de los ochenta, los desarrolladores tendieron a enfocarse sobre las técnicas de análisis/diseño estructurado (SA/SD) combinadas con un enfoque hacia el desarrollo de ciclo de vida en “cascada” secuencial. A finales de los ochenta y a principios de los noventa, la transición de arquitecturas de mainframe hacia cliente/servidor coincidió con una transición de los enfoques de ciclo de vida en cascada hacia los enfoques de ciclo de vida RAD/prototipos/espiral. Junto con eso, el énfasis sobre las GUIs (Interfaces Gráficas de Usuario) inició una transición de los métodos estructurados, a los métodos OO (Orientados a Objetos). Es interesante hacer notar que la nueva generación de desarrolladores de software, frecuentemente rechazan al SA/SD por la misma razón que rechazan a las computadoras mainframe y al COBOL: por lo que a ellos concierne, es irremediamente obsoleto. También es interesante notar que en muchos casos las razones para tal rechazo es que el SA/SD requiere un enfoque de cascada, y como el enfoque de cascada es obsoleto suponen que el SA/SD también lo es. David Ruble muestra que éste no es el caso. Los sistemas desarrollados para la corriente “actual” de arquitecturas, la combinación cliente/servidor y GUI, pueden ser desarrollados con un enfoque de análisis/diseño estructurado robusto. Todo esto puede hacerse con un enfoque iterativo inteligente de sentido común para construir el sistema. Aunque las generaciones más recientes de metodologías OO pueden creer que son quienes inventaron a los “eventos” (mencionados frecuentemente como “casos de uso”), Ruble muestra que un modelo basado en eventos es práctico para los sistemas cliente/servidor, y que puede utilizarse bastante bien con un enfoque de métodos estructurados. Este no es un concepto completamente revolucionario. Steve McMenamin y John Palmer[2], presentaron la noción de un enfoque basado en eventos para el análisis estructurado en 1984, pero su

discusión no tuvo la oportunidad de ilustrar las ideas dentro del contexto moderno de los sistemas cliente/servidor y basados en GUI.

Metodología aplicada al modelado de éste sistema.

Esta metodología, consiste de una serie de pasos que se describirán a continuación.

El plan general del proyecto, inicio del proceso analítico con el marco de trabajo para la toma de decisiones sobre el proyecto, técnica que va a ayudar a los miembros del municipio (Sistemas y dependencias involucradas) a determinar los verdaderos objetivos de su nuevo sistema.

El modelado de contexto, una técnica importante para la exploración y definición del alcance del sistema y sus interfaces externas.

El modelado de eventos define el comportamiento observable del sistema en términos del negocio y documenta la política y reglas que comprenden los requerimientos del proceso. Es un modelo crucial, el cual guía el desarrollo de la interfaz gráfica de usuario manejada por eventos.

El modelo de información, crea un mapa organizacional de los datos que requiere recordar el sistema. Ésta es una técnica vital para un buen diseño de bases de datos relacionales y el modelado de objetos.

Después se cambia de la construcción de modelos a la asimilación de éstos. El prototipo de interfaz es nuestra primera incursión hacia el diseño. Se puede usar la creación de prototipos para validar modelos, diseñar la interfaz o hasta descubrir requerimientos.

A partir de éste punto se lleva a cabo el diseño del sistema de manera que se dé un enfoque cliente/servidor con una interfaz gráfica de usuario.

El modelo arquitectónico muestra la manera de usar los modelos esenciales del análisis para determinar la arquitectura técnica más deseable (o la menos desfavorable) para el sistema.

Después se tratan los puntos básicos de la transformación de un modelo de información hacia un diseño de base de datos relacional.

Posteriormente vienen las especificaciones escritas del diseño de la interfaz externa. Una especificación escrita es una herramienta de administración vital para dividir el trabajo de desarrollo entre varios programadores y para concebir planes de prueba adecuados.

Y por último, viene el diseño de componentes internos (enfocado al concepto Orientado a Objetos).

Las técnicas empleadas en éste sistema, pretenden probar que mediante ésta metodología se pueden construir sistemas confiables que hacen realidad el objetivo cliente/servidor. Haciendo esto en una forma organizada y racional evitaremos la anarquía que prevalece en los departamentos correspondientes del municipio, y el esfuerzo resultante producirá un sistema de información que será lo suficientemente flexible para satisfacer las necesidades expuestas anteriormente, y al mismo tiempo, mantener la custodia segura de los activos de datos del Ayuntamiento.

CAPÍTULO 2: DESCRIPCIÓN DEL PROBLEMA

La Dirección de Proveduría, actualmente no cuenta con un sistema que les permita automatizar su proceso, así como resguardar su información de manera digital, por lo que su necesidad de implementar un sistema es uno de los proyectos estratégicos de ésta administración. ¿Qué es lo que se pretende resolver con éste sistema?.

Requerimientos del Sistema.

Solicitudes.

Todas las dependencias del municipio generan una solicitud de mercancía o servicio (conocida como requisición), y la envían a Proveduría. Ahí es entregada en ventanilla, se revisa, se pone un sello de recibida, y dependiendo de si es una solicitud de compra o de servicio es asignada al jefe de departamento correspondiente. Este a su vez, verifica que lo que se esté solicitando, esté especificado de manera correcta, y entonces coloca la firma de aceptada. Posteriormente la asigna a alguno de los cotizadores para que le dé seguimiento.

Invitaciones a Cotizar.

El cotizador, recibe la solicitud, verifica a qué familia pertenece la solicitud, ve cuáles son los artículos o servicios que contiene, y busca a qué proveedores puede invitar a cotizar. Después, hace una invitación para cada proveedor de la siguiente manera: llena un formato que se llama carátula de invitación, poniendo los datos del proveedor, y el número de solicitud. La carátula también contiene la manera en que debe de contestar esa invitación (o sea, con una cotización), entregada en cierto lugar en cierta fecha y con ciertas características (sobre cerrado, dirigido al Proveedor Municipal, etc.).

Cuando todas las invitaciones están listas, se llama por teléfono a los representantes de cada proveedor, y se les envía uno a uno la invitación por fax. Hasta aquí el proceso tarda de 2 a 4 días, dependiendo de la carga de trabajo que se tenga en el momento.

A partir de aquí, una vez entregada la invitación al proveedor, el tiempo de respuesta por parte de él, ya no depende del proceso, y oscila entre 7 y 10 días.

Cotizaciones.

Cuando llega ésta respuesta (la cotización), se recibe en ventanilla, en sobre cerrado y en el caso de exceder una cantidad (\$250,000 como total de la cotización) se separan para ser evaluadas en un comité de adquisiciones, en caso contrario se continúa con el proceso normal. Cuando una solicitud se manda a comité, éste decidirá al proveedor ganador a través de un cuadro comparativo, previamente capturado, dando la justificación necesaria del porque resultó seleccionado, que puede ser, por ser el precio más bajo, por tener la calidad más alta, la mayor garantía o el menor tiempo de entrega. Posteriormente, la cotización ganadora se manda al departamento de Ordenes de Compra/Servicio, para que emitan la orden correspondiente al proveedor. En caso de que la solicitud no se vaya a comité se sigue con el proceso normal. Se revisa que la cotización coincida con lo que se está pidiendo, tanto en artículos o servicios, como en las especificaciones de cada uno de ellos, para verificar que lo que se solicita, es lo mismo que lo que se cotiza. En el caso de existir diferencias, en cantidades, presentación, especificaciones, artículos o servicios, se necesita de una carta por parte del proveedor, especificando el porqué de esos cambios, para que esa cotización pueda ser tomada en cuenta en el proceso de adquisición.

Cuadros Comparativos.

Llegando la fecha límite de recepción de cotizaciones, se concentran todas las que pertenezcan a la solicitud, y se capturan en un cuadro comparativo (actualmente un formato en Excel), y se manda a cada cotizador para que él determine el proveedor ganador. Una vez seleccionado, se manda la cotización ganadora al jefe de departamento para que valide el resultado y se manda la cotización ganadora al departamento de Ordenes de Compra/Servicio.

Ordenes de Compra/Servicio.

Una vez ingresada la Cotización ganadora al departamento de Ordenes de Compra/Servicio, se genera la orden en una terminal conectada a Tesorería (un sistema hecho en AS400), esto para hacer la afectación del presupuesto e imprimir la orden. Desafortunadamente, si la partida presupuestal (o sea, la cuenta donde radica el dinero destinado a la familia a que pertenece la solicitud), no cuenta con los fondos suficientes, la orden es rechazada, por lo que todo el proceso anteriormente descrito se cancela, y se le notifica a la dependencia del rechazo, para que ella sea la encargada de hacer los movimientos presupuestales necesarios, y volver a generar otra solicitud para poder efectuar la adquisición. Se puede notar que un sistema desarrollado con una buena metodología de diseño y modelado, puede solucionar este proceso en muchos sentidos, incluyendo los tiempos de respuesta que la dependencia puede brindar.

Esto es la problemática principal de las adquisiciones en el municipio, aunque existen casos especiales.

Casos Especiales.

Talleres Municipales.

El primero es el caso de Talleres Municipales. Ahí, el proceso es ligeramente diferente al resto de las adquisiciones del municipio. Físicamente, las oficinas de la Dirección de Proveeduría, están separadas de las oficinas de Proveeduría de Talleres, además tomando en cuenta que existen 2 talleres municipales, uno separado del otro (a extremos opuestos de la ciudad), y cada uno tiene su departamento de adquisiciones de refacciones y servicios. Cabe mencionar que los servicios que se pagan aquí son servicios a talleres externos, y resultan cuando un vehículo tiene garantía en una agencia, o cuando la falla requiere de una reparación mas especializada, pues en el municipio se cuenta con vehículos especializados como barredoras, taladores de árboles, etc. los cuales solo pueden repararse en centros de servicio autorizados por la marca del vehículo. Así mismo, la compra de refacciones sólo puede hacerse en estos talleres.

Aquí las diferencias que existen son las siguientes: En la generación de Ordenes de Compra, (o adquisición de refacciones), es muy común que el proveedor cotice una pieza genérica, en vez de la pieza original, pues ambas sirven aunque no con la misma calidad que la pieza original, y es necesario un sistema que permita registrar piezas “equivalentes”. Otra diferencia que existe con el proceso de adquisiciones en la Dirección de Proveeduría, es que en la Dirección es contra la ley solicitar un artículo de una marca específica, a menos que esté bien justificado el por qué de esa marca. Pero esto no se refleja en el proceso de adquisición normal, pues además de estar prohibido, no son casos que se den con frecuencia. Pero en la parte de las refacciones vehiculares pasa todo lo contrario, pues aquí siempre se tiene que cotizar por marca y modelo, e inclusive, existen las mismas refacciones para diferentes marcas de vehículos y para diferentes modelos.

También el proceso de selección del proveedor es diferente, pues la marca influye muchísimo, ya que hay proveedores especializados en diferentes marcas. En la parte de Servicios también éste es un punto importante. Además, cuando se manda un vehículo a un taller externo a algún servicio, se genera una PreOrden de Servicio, con la que se manda el vehículo, especificando que es lo que requiere, pero el proveedor regresa esa misma orden con más servicios o incluso refacciones, dependiendo de lo que haya necesitado el vehículo, y los costos de cada uno, para posteriormente, en proveeduría generarles su respectiva Orden de Servicio. O sea, siempre por cada Orden de Servicio, existe una PreOrden que la antecede. También los reportes que solicitan son diferentes, pero la Dirección de proveeduría le solicita a talleres esos reportes para anexarlos a los reportes que ellos generan, y entregar sus informes conjuntamente. En este sentido sería ideal contar con la misma base de datos en la dirección y en talleres municipales, pues este paso se evitaría, ahorrando tiempo y esfuerzo del personal que se dedica a crear esos informes. Así mismo las Ordenes de Compra y de Servicio resultantes, se manejan al principio por separado, y después los gastos se anexan. Este es un paso que también podría ser evitado si las órdenes se guardaran en un solo lugar y no en 3 como actualmente se está realizando. La información generada por la parte de proveeduría de talleres es compartida con el sistema de Taller Municipal, que lleva el control de las reparaciones, asignación de mecánicos, almacén de refacciones en talleres, y por supuesto los resultados de las adquisiciones. O sea, las refacciones las agrega al *stock* existente en

almacén o a veces a un vehículo en específico. Y de las Preordenes de Servicio, ellos llevan el histórico por vehículo de las reparaciones o servicios que éste lleva, y los costos de cada uno de ellos, para evaluar si un vehículo es viable dejarlo en el padrón o darlo de baja.

Dirección Municipal de Salud.

Otro caso interesante es el de la Dirección Municipal de Salud, o mejor conocida como Cruz Verde. Ellos hacen sus adquisiciones a través de la Dirección de Proveduría como todas las demás dependencias. Pero cuando se trata de medicamentos, material de laboratorio, material quirúrgico, y artículos especializados, se maneja un poco diferente. Al igual que en taller, los proveedores son especializados, y no cualquiera puede cotizar estos artículos y medicamentos. Es un caso similar. Pero aquí la diferencia radica en que sus compras son cada 3 meses, y la cantidad de artículos manejados en su solicitud es bastante grande, y por lo general de urgencia. A veces ellos ya saben a quien se le va a comprar cada artículo. Pero lo interesante es que en los artículos que caducan, como medicamentos, se hace una programación de entregas por fechas y por lugares, pues son 4 unidades médicas y se tienen que entregar en fechas diferentes.

Patrimonio.

En la parte de Bienes Patrimoniales, existe un punto digno de comentarse. En la actualidad cuando se hace la adquisición de un bien patrimonial (una silla, mesa, librero, etc.), se asignan a la dependencia, y posteriormente se inventarían para asignarles un número patrimonial, con etiqueta de código de barras. Los jefes se han quejado de hacer esto con el proceso actual, debido a que pasan meses antes de que se inventaríe algo en el municipio, y muchas veces es porque el registro de la compra sólo existe en papel. Esto trae como consecuencia que los resguardos de cada trabajador, nunca esté actualizado, y por consiguiente a cada rato se estén perdiendo artículos o bienes de valor para el municipio. Se solicitó a la dirección de proveduría la implementación de un módulo en el sistema de adquisiciones de manera que cuando se haga la compra de un bien para alguna dependencia, el sistema lo dé de alta en el inventario municipal. Así sólo quedaría ir y pegar la etiqueta, única y exclusivamente.

Tesorería y el presupuesto de adquisiciones.

Por último y la parte más importante de ésta problemática es el presupuesto.

La dirección de tesorería es la encargada de controlar el presupuesto de todo el municipio y es quien le paga a los proveedores por medio de las Ordenes de compra o Servicio que la Dirección de Proveeduría genera. En éste sentido, no existe actualmente un control del presupuesto y el gasto ejercido en el municipio, pues la información no está en línea y es muy común ver Ordenes de Compra que son rechazadas al momento de ser generadas porque no hay dinero para esa partida presupuestal. Es imperativo que el sistema de adquisiciones valide desde el momento de la generación de la Solicitud, si existe presupuesto o no disponible para llevar a cabo ese proceso, pero el problema no termina aquí. ¿Qué pasa cuando en una partida si hay dinero suficiente para darle entrada a una solicitud, y durante el proceso se termina?. Cuando se llegue al punto de generar la orden de compra o de servicio, no va a ser posible, pues no existe una partida disponible para hacer la afectación de éste gasto. Es por eso que el control del presupuesto es muy importante, pues además de validar el monto disponible, tiene que validar el monto comprometido, y el monto ejercido, durante el proceso, y estar tomando esa información del sistema de Tesorería. ¿Por qué entonces no se actualiza el presupuesto cada determinado tiempo?. Porque el dinero disponible en una partida, depende de varios factores, no solamente del proceso de adquisiciones. Existen movimientos contables, como traspasos de cuenta, deducción de dinero por honorarios, nómina de empleados, servicios extraordinarios, etc. Así que la compra de bienes o servicios es sólo un factor de varios existentes en la contabilidad para el manejo del dinero.

Además de esto, existen dos maneras de llevar el gasto ejercido en el municipio. Uno se denomina Presupuesto por Proyecto (PP), y consiste en distribuir los montos disponibles por programa, en vez de por partidas presupuestales. O sea, se asigna una cantidad a un programa (ejemplo, el desarrollo e implementación del sistema de adquisiciones en el municipio), y al monto de ese programa, se le pueden cargar, el pago de algún servicio, la compra de equipo de computo, la nomina del personal que estará involucrado, y cualquier gasto que ese programa genere. La otra manera de llevar el gasto es como se explicó anteriormente.

Es debido a ésta problemática que el Municipio de Guadalajara, necesita un Sistema de Adquisiciones para llevar el control de la información que genera éste proceso.

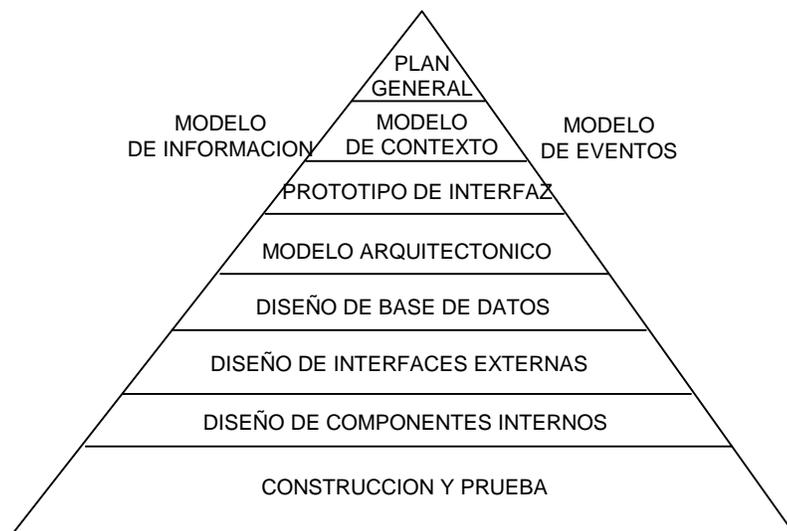
CAPÍTULO 3: SOLUCIÓN (DESARROLLO DEL MODELADO DEL SISTEMA)

En éste capítulo se expondrá el diseño del SIMA v. 1.0 siguiendo el modelado que propone Ruble[1]. Con este trabajo pretendemos presentar un modelo que sea verificable por el público, tanto técnico como no técnico.

Debido a la magnitud del proyecto, las características de éste, y la prontitud de resultados necesarios, se le dará un enfoque en cascada, dividiéndolo en fases e integraremos un grado de iteración de ajuste controlado.

Plan general del proyecto

Se presentará una técnica valiosa llamada el marco de trabajo para la toma de decisiones del proyecto. Esta técnica es particularmente útil para dirigir a los miembros del negocio a través de las etapas necesarias para comprender sus problemas actuales y reconocer nuevas oportunidades que anteriormente no se habían aprovechado.



Se utiliza la pirámide como metáfora geométrica principal para organizar las actividades de desarrollo de este sistema. Hay un significado asociado a la pirámide. También se podría usar un cuadrado, un círculo o un conjunto de nubes sin forma, pero el usar una pirámide tiene varias razones: La pirámide nunca permite olvidar que el código que se construye es simplemente la base de una estructura que esta especificada para que alcance un conjunto de objetivos del negocio. En la parte superior de la pirámide está el plan general del proyecto. Esto incluye la meta del proyecto y los objetivos que la soportan. Estas son las razones por las que el proyecto existe en primer lugar. Bajo el plan, en una forma descendente, se encuentran todas las actividades que necesitan suceder entre la identificación de los objetivos del proyecto y la colocación de los unos y ceros en la parte inferior de la pirámide que son el software resultante.

Declaración de la meta

Todo proyecto de sistema de información tiene una meta. Necesita ser clara, no ambigua, concisa y medible. Todos necesitan estar de acuerdo sobre lo que es y saber cuando se ha logrado. Una mejor forma para definir la meta es a través de la lista completa de objetivos individuales que deben lograrse para que el proyecto sea un éxito. Los objetivos se determinan descubriendo todos los problemas que hay en la forma actual de hacer negocios y proporcionando ideas sobre oportunidades no aprovechadas. Después de que la lista de objetivos ha sido ratificada por el grupo, puede ser destilada en una declaración de meta resumida que presente al conjunto de los objetivos del proyecto. Se pasará a discutir los problemas, oportunidades y objetivos para poder declarar la meta.

El inicio del proceso analítico comienza preguntándole a la gente qué hay de malo en el ambiente actual. Un sistema de información nuevo es una solución. Para diseñar la solución adecuada se debe comprender el problema que se está tratando de resolver. Existe un problema en cualquier momento en que alguien no esté satisfecho con el comportamiento o capacidades de su sistema de información existente y puede expresar lo que piensa que debiera ser el comportamiento o capacidad adecuado.

Los Doctores Kenneth Blanchard y Spencer Jonson [3], dan una definición magnífica de un problema en términos de comportamiento:

“Un problema existe si hay una diferencia entre lo que está sucediendo actualmente y lo que se desea que suceda. Si no puede decirme lo que le gustaría que estuviera sucediendo, todavía no tiene un problema. Simplemente se está quejando.”

Ahora bien, se analizarán cuales son los problemas que el municipio tiene respecto a la adquisición de bienes o servicios.

El primer problema que el municipio tiene en éste sentido es que no cuenta con un sistema de información para el control de los diferentes tipos de adquisiciones que se hacen. Todo el proceso es realizado en papel. Los equipos de cómputo son utilizados como máquinas de escribir y cuando mucho utilizan Excel para hacer los cuadros comparativos. La cantidad de información que se maneja durante el día es excesivamente abrumadora y muy difícil de controlar. Por esto un sistema es la mejor solución para resolver éste problema, pues se puede aprovechar la tecnología con que cuenta el municipio y una base de datos que sirva para distribuir la información puede acelerar tiempos y disminuir costos, personal y esfuerzos.

El segundo problema que tenemos presente es que la información está descentralizada y esparcida en varias dependencias, lo que hace más difícil concentrar los datos de las dependencias involucradas y por lo tanto la toma de decisiones es mucho más complicada. Nuevamente el tener conjuntada toda la información en una base de datos centralizada, resuelve el problema, tomando en cuenta las diferentes opciones que existen para el modelado de la base de datos. En éste sentido se decidió que un modelo Cliente/Servidor es el más óptimo, debido a la forma en que se va a compartir la información (Centralizada en el servidor de la dependencia responsable de ella, que en este caso es la Dirección de Proveeduría Municipal).

Otro problema es el tener la información en línea. La toma de decisiones en el municipio es muy importante cuando se trata de recursos monetarios y desafortunadamente no se cuenta con una herramienta que proporcione dicha información en el momento. Esto es debido a que cuando las compras exceden de cierta cantidad se

van a un comité de adquisiciones, y piden información de presupuesto, historial del proveedor, compras anteriores, etc., y todo en el mismo instante que se celebra la junta. Esto ocasiona que varias personas estén buscando toda la información que requiere el comité en ese instante y se desperdicia personal, tiempo y es demasiado esfuerzo el trabajar con papel, pues duran casi todo el día trabajando en esta actividad. Nuevamente esto se resuelve con un sistema que pueda brindar la información en línea de las dependencias involucradas en el proceso de compra y con el mínimo de esfuerzo humano.

Agregado a todo esto, también se tiene la generación de las invitaciones a cotizar, pues además de que se capturan en papel, todos los cotizadores (que son alrededor de 15), como parte de su actividad diaria, tienen que buscar a que proveedores invitar y capturar una por una las invitaciones correspondientes. Este sistema sería de ayuda considerable si tomamos en cuenta que las puede generar de manera automática. Este punto también es igual con los cuadros comparativos.

Respecto a la dependencia de Tesorería, básicamente existen 2 problemas. El primero es que no se tiene el presupuesto en línea para validar la aceptación de las solicitudes e incluso la generación de las ordenes de compra. Una vez ya generadas, y en el momento de que se autorizan por proveeduría, cuando no tienen presupuesto es cuando son rechazadas, pero hasta este punto. El otro problema que se tiene es que cuando se generan las ordenes de compra, se tiene que mandar una relación en papel a Tesorería para capturarlas en su sistema. Es necesario una conexión con éste sistema, pues las ordenes que se generen en el SIMA, pueden aparecer en línea en tesorería sin necesidad de recapturarlas.

En patrimonio, existe sólo un problema, pero es bastante considerable. En ésta dependencia hacen un inventario de los bienes patrimoniales con que cuenta el municipio al principio de la administración. Pero conforme se van adquiriendo y dando de baja estos bienes, el inventario ya no concuerda y tienen que estar haciendo inventarios constantemente para tener un aproximado cercano a lo que es la realidad. El estar actualizando en el sistema lo que se adquiere les ahorra el 50 % del trabajo que estaban haciendo para este proceso, pues se enfocarían a verificar lo que se ha dado de baja únicamente.

En los talleres municipales existen varios puntos a tratar. Primero, a diferencia del proceso normal, ellos generan algo que se llama preorden de Servicio. Este documento sirve para mandar a los vehículos a talleres externos, cuando éstos no pueden ser reparados dentro del mismo taller, ya sea por que el costo es menor, porque no hay espacio físico en el taller, o porque no se tiene la capacidad para hacer ese trabajo. Las preordenes se hacen en otro sistema, y esa información se tiene que recapturar nuevamente para la generación de las ordenes de servicio finales. Además, no se lleva un control de los precios que los proveedores externos le cobran a los talleres, por lo que a veces cobran lo que quieren. Igualmente sucede con las refacciones, pues cuando se hacen los servicios externamente, las cobran a precios demasiado elevados. También es necesario llevar el control de las solicitudes de aprovisionamiento de refacciones para el almacén de los talleres. El control de los movimientos de éste almacén, así como de el procesos de reparación y control de los talleres se llevan acabo en otro sistema.

En las Unidades Medicas existe un solo problema: Ellos hacen sus pedidos dos o tres veces al año, concentrando todos los requerimientos de las unidades en una sola solicitud. Pero cuando llega esa solicitud a Proveeduría, se necesita atender en el instante y darle seguimiento con carácter de urgente, por lo que es necesario implementar en el sistema una manera de atender esa solicitud sin todo el tramite que necesitan el resto de las solicitudes.

Ahora bien, una vez expuestos los problemas que tenemos, hay que tomar en cuenta cuales son las oportunidades y posibles soluciones que se pueden encontrar.

Como se mencionó antes, un sistema de información es una posible solución, y debido a que existen pequeñas islas de información, el tener una herramienta que pueda centralizar la información es la solución más eficaz.

A continuación se especificarán los objetivos del proyecto, para de ahí desglosar, posteriormente, la meta.

Objetivos

- 1.- Automatizar el proceso de la dependencia de Proveduría.
 - 1.1.- Generar de manera automática las invitaciones a cotizar.
 - 1.2.- Evitar la recaptura de información entre los procesos.
 - 1.3.- Generar de manera automática los cuadros comparativos.
 - 1.4.- Generar la orden de compra y de servicio más fácil, evitando la recaptura de información.
- 2.- Establecer una comunicación mas directa con el sistema de Tesorería.
 - 2.1.- Tomar la información del presupuesto, para validarlo en el proceso de adquisiciones.
 - 2.2.- Actualizar la información de la orden de compra y de servicio en el sistema de Tesorería para su posterior pago.
- 3.- Registrar los bienes patrimoniales en el sistema de Patrimonio.
 - 3.1.- Registrar en el inventario patrimonial, la adquisición de bienes.
- 4.- Llevar el control del proceso de adquisiciones en los Talleres Municipales.
 - 4.1.- Generar automáticamente, las preordenes y ordenes de servicio a talleres externos.
 - 4.2.- Proveer de información de costos de refacciones adquiridas y servicios externos al sistema de Taller Municipal.
 - 4.3.- Generar las solicitudes de adquisiciones para la reposición de refacciones en almacén.
- 5.- Facilitar el proceso de adquisiciones de medicamentos para la Dirección Municipal de Salud
 - 5.1.- Implementar un mecanismo de adquisiciones de medicamentos y material de curación, en el que se puedan hacer adquisiciones de manera normal, y de manera urgente.

Meta

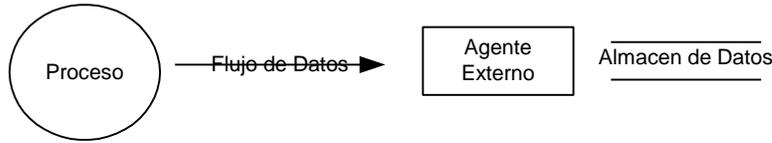
La declaración de la meta podría quedar de la siguiente manera:

“ La meta es concentrar la información de las dependencias involucradas en las adquisiciones en un solo servidor, y automatizar la distribución de la información de ese servidor al personal directivo para facilitar la toma de decisiones, y que éstas sean tomadas a tiempo, además de llevar el control estadístico e histórico de las adquisiciones del municipio, desde la solicitud, hasta la entrega del producto o servicio a la dependencia”

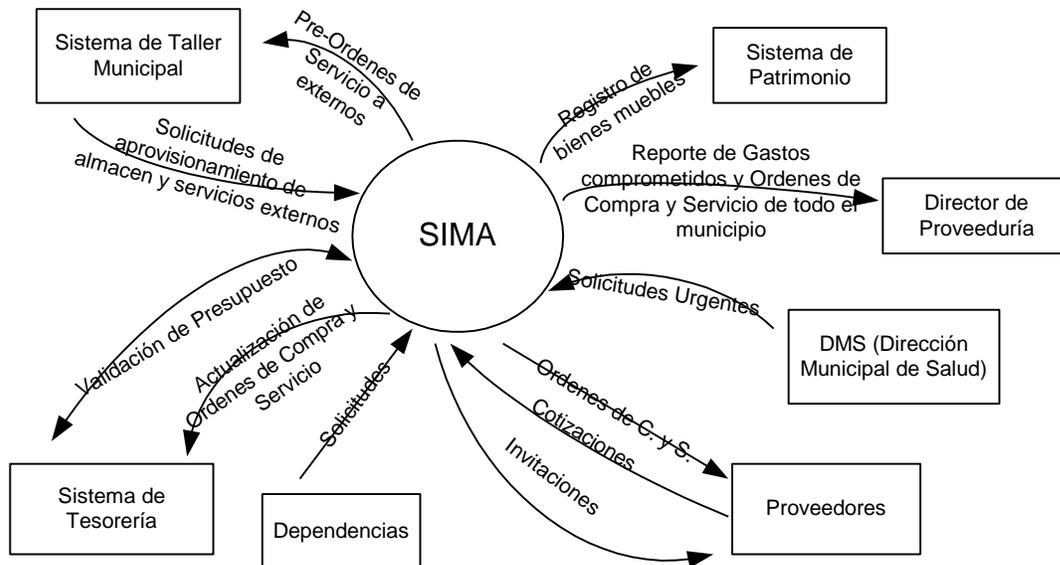
Esta meta es clara, no ambigua, concisa y mensurable. Pueden existir varias soluciones para ésta problemática, pues cuando se habla de “concentrar la información”, se pueden juntar todas las bases de datos en un solo servidor y así se está concentrando toda la información que se necesita, pero también se observa la siguiente parte de la meta: “y automatizar la distribución de la información de ese servidor al personal directivo para facilitar la toma de decisiones y que éstas sean tomadas a tiempo, además de llevar el control estadístico e histórico de las adquisiciones del municipio, desde la solicitud, hasta la entrega del producto o servicio a la dependencia”. Tomando en cuenta que existen varias soluciones que ayudan a procesar la información de varios sistemas como la minería de datos, en proveeduría también tendría que existir un sistema para juntar toda la información y desafortunadamente no es así. La dependencia de proveeduría no tiene ningún sistema de computación que le permita llevar el control de las adquisiciones, y tomando en cuenta que es necesario para el municipio el llevar el control y automatización de éste proceso, se decidió que al desarrollar éste sistema de información se tomara en cuenta todo el proceso global tomando en cuenta los puntos expuestos anteriormente.

Modelo de Contexto

El modelo de contexto utiliza la notación de diagramación de flujo de datos clásica.

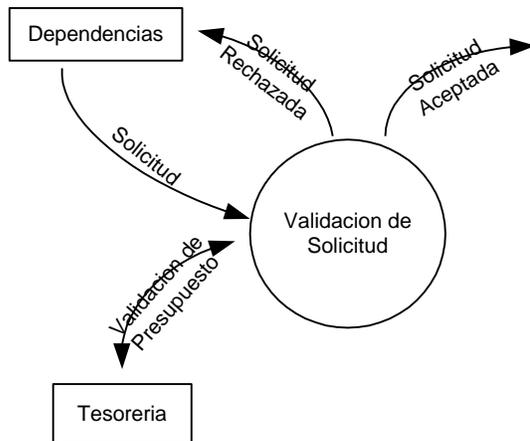


Los DFD (Diagramas de Flujo de Datos) fueron introducidos por primera vez en 1979 por Tom DeMarco [4]. Los diagramas de flujo de datos son modelos que muestran la ruta que toman los datos a través de una organización, sin ninguna tendencia a causa de una implementación específica. Es presentado básicamente para mostrar, en un esquema global, el alcance que va a tener nuestro proyecto. Abajo mostramos el Modelo General de Contexto de nuestro Sistema.

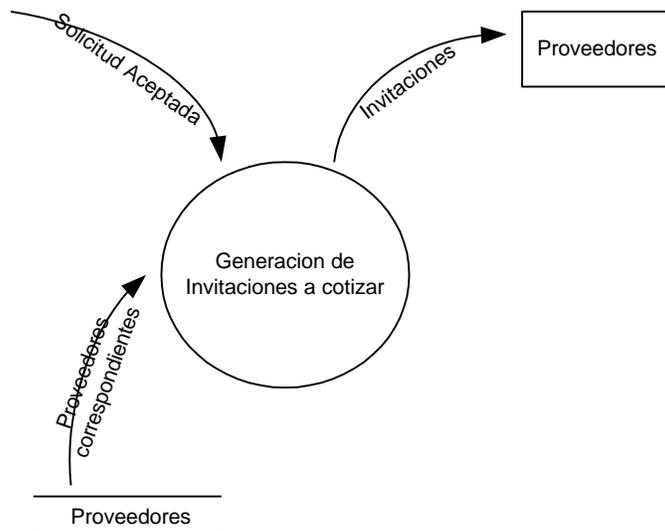


Una vez mostrado el alcance que va a tener nuestro sistema, mostraremos el siguiente nivel de nuestro diagrama de contexto, mostrando con un poco mas de detalle los procesos involucrados en las adquisiciones del municipio.

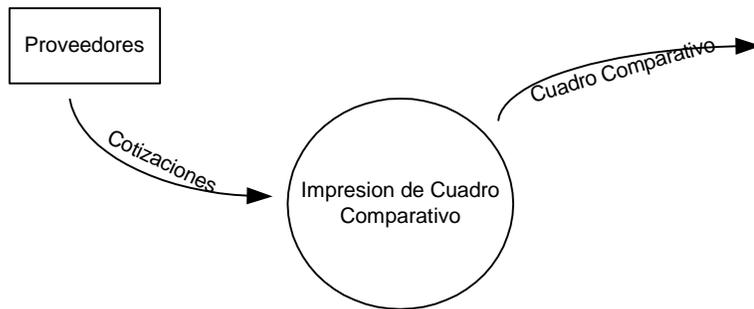
DFD 1 (Validación de Solicitud)



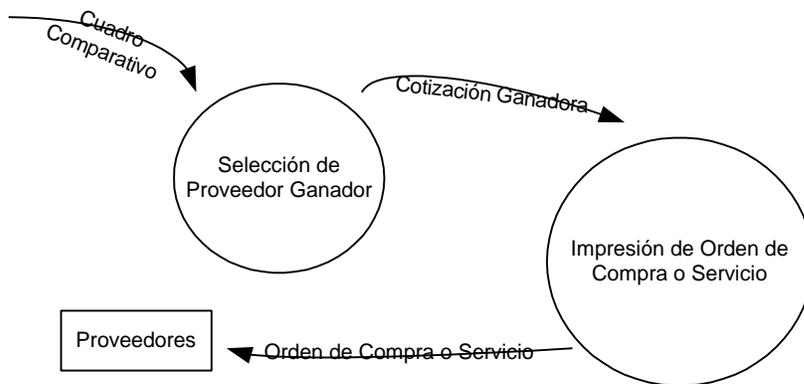
DFD 1 (Generación Automática de Invitaciones a Cotizar)



DFD 1 (Recepción de Cotizaciones e Impresión de Cuadro Comparativo)

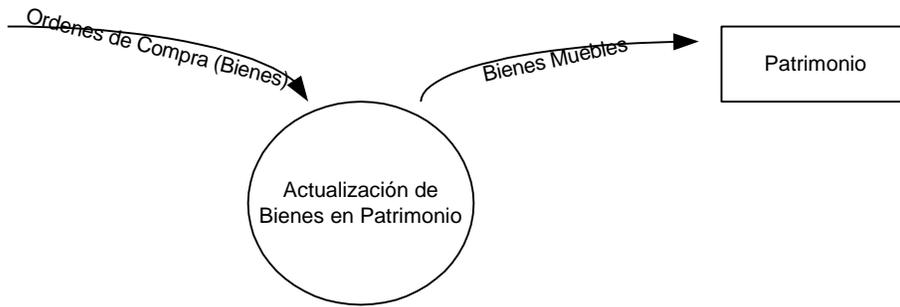


DFD 1 (Selección de Proveedor Ganador y Generación de Ordenes de Compra y Servicio)

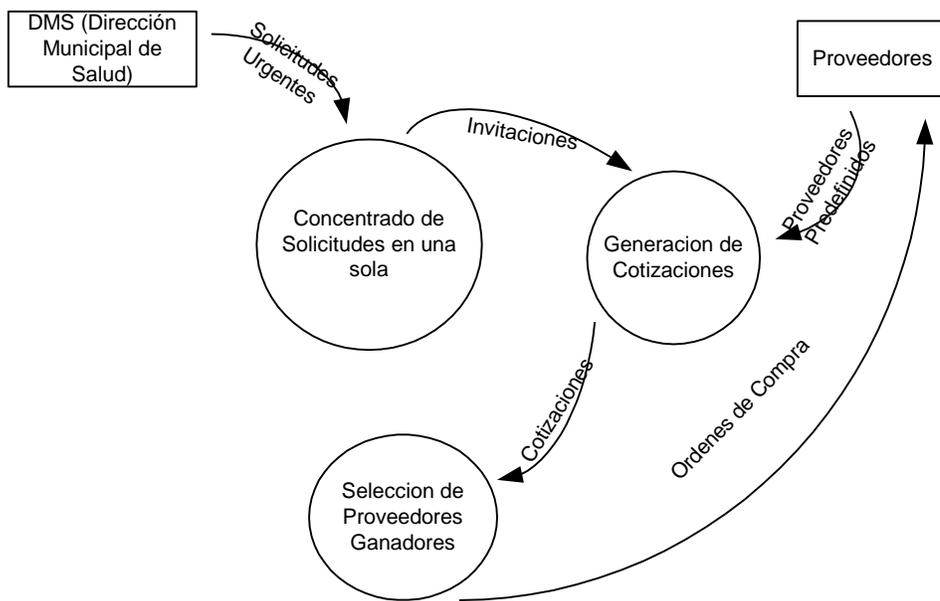


Hasta este punto se ha mostrado el proceso normal de proveeduría respecto a las adquisiciones de la mayoría de las dependencias. A continuación se muestran las diferentes variantes existentes con respecto a Taller Municipal, Unidades Medicas y Patrimonio.

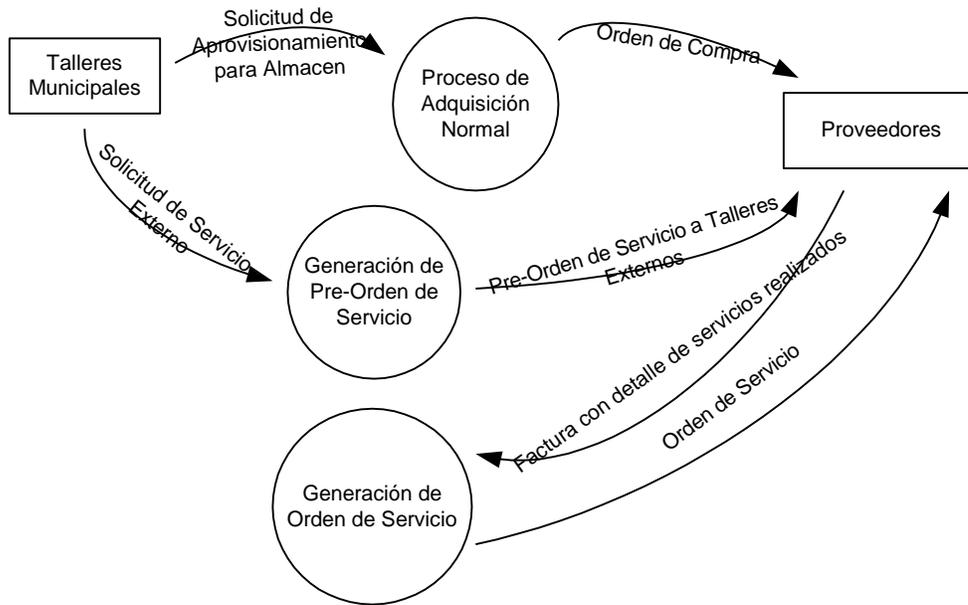
DFD 1 (Patrimonio)



DFD 1 (Unidades Medicas)

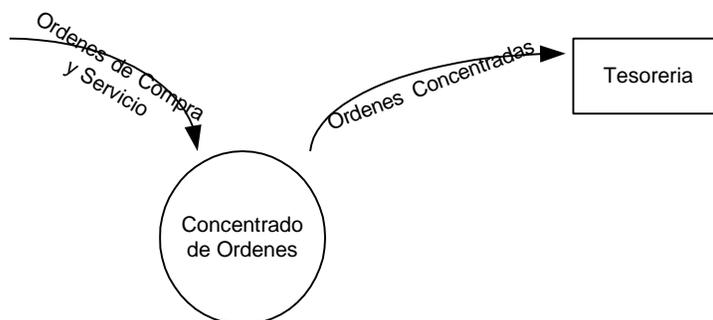


DFD 1 (Talleres Municipales)



Posteriormente, todas las ordenes de compra y servicio que se generan en el sistema, independientemente de la dependencia que lo generó, se concentran y se manda un reporte diario al sistema de Tesorería, para llevar el control del pago a los proveedores.

DFD 1 (Tesorería)



Ahora, el modelo de contexto, sirve cada vez menos en la medida que se avanza en el modelado del sistema, pero cuidado, el brincar éste paso, puede poner en aprietos el modelado sin tener una idea clara de los límites que el sistema debe tener, hubiera sido necesario modelar posteriormente, cosas y situaciones no contempladas, y por un tiempo indefinido.

A continuación, se puede observar que el modelo de contexto está directamente relacionado con el modelo de eventos, el cual forma la base de gran parte de las tareas de diseño de interfaz que se van a realizar. En cualquier momento en que se mueva la frontera del contexto también se alterará el paisaje del modelo de eventos.

Modelo de Eventos

El modelo de eventos contiene una lista de eventos y un diccionario de eventos. Siempre que sea posible, los eventos de la lista se mencionarán en una sintaxis sujeto-verbo-objeto. El diccionario de eventos proporciona la parte medular de la especificación del proceso para el sistema. Para cada evento de la lista, se define su significado en texto claro y conciso. La parte técnica del diccionario de eventos contiene una definición de los datos de estímulo requeridos para activar el evento, la actividad realizada por el sistema para formular la respuesta adecuada y los datos contenidos en esa respuesta.

El modelo de eventos es un documento poderoso que se aprovechará completamente a lo largo del proyecto. Cabe mencionar que éste tiene sus raíces en el análisis estructurado tradicional. La técnica se ha puesto al frente actualmente debido a que los sistemas que estamos construyendo en la actualidad son mucho más aplanados que las estructuras jerárquicas de antaño. La lista de eventos y el diccionario de eventos han probado ser modelos de desarrollo GUI poderosos. La clave para el buen diseño GUI es la comprensión de la manera en que el sistema debe comportarse cuando responde a los eventos de negocios e incorporarle la suficiente flexibilidad al diseño para manejar varios eventos y secuencias no tradicionales. Esto no quiere decir que el diseño de ventanas GUI permita que cualquier cosa ocurra. Una tarea significativa del diseñador GUI es decidir lo que el usuario no puede hacer en un momento dado. Un modelo de eventos sólido ayuda a organizar esta tarea.

Lista de Eventos

A continuación se muestra la lista de todos los eventos que afectan al sistema. Hay Algunas reglas que determinan si cualquier frase antigua en la sintaxis sujeto-verbo-objeto califica, de hecho, como un evento para nuestro sistema. Para lograr la membresía en la gran orden fraternal de los eventos, un candidato debe pasar cada una de las siguientes pruebas:

- 1.- Un evento sucede en un momento específico.
- 2.- Un evento sucede en el ambiente y no dentro del sistema.
- 3.- La ocurrencia del evento está bajo control del ambiente y no del sistema.
- 4.- El sistema debe ser capaz de detectar que el evento sucedió.
- 5.- Se supone que el sistema hará algo con respecto a él, significando esto que es relevante para el plan general del proyecto.

Si un evento no cumple cualquiera de las reglas es motivo suficiente para descalificarlo como candidato a evento.

Se ordenará la lista conforme al tiempo, esto es, conforme van sucediendo, para que sea mas entendible por el usuario.

- 1.- Proveduría recibe las solicitudes de las dependencias
- 2.- DMS envía solicitudes urgentes (DMS solamente)
- 3.- Tesorería valida el Presupuesto para las solicitudes
- 4.- El usuario imprime las solicitudes
- 5.- El usuario cancela las solicitudes
- 6.- El usuario genera las invitaciones a cotizar
- 7.- El usuario imprime las invitaciones a cotizar
- 8.- El usuario captura las cotizaciones
- 9.- El usuario cancela las cotizaciones
- 10.- El usuario genera el cuadro comparativo
- 11.- El usuario imprime el cuadro comparativo
- 12.- El usuario selecciona proveedor(es) ganador(es)
- 13.- El usuario imprime la PreOrden de Servicio (Taller solamente)

- 14.- El usuario genera las Ordenes de Compra o Servicio
- 15.- El usuario imprime las Ordenes de Compra o Servicio
- 16.- Se actualiza la información de las Ordenes en Tesorería
- 17.- El usuario cancela Ordenes de Compra o Servicio
- 18.- Se actualiza la información de las Ordenes Canceladas en Tesorería
- 19.- El usuario actualiza en el sistema de bienes muebles los bienes adquiridos por el municipio (Patrimonio solamente)

Diccionario de Eventos

El diccionario de eventos consta de varias secciones que se explicarán a continuación.

Identificador (ID) del evento puede ser un número, pero se recomienda que ese número no sea significativo de alguna manera. Un identificador asignado en forma aleatoria permite que se cambie el nombre del evento sin tener que cambiar su identificador.

El **nombre del evento** es una oración clara del evento en palabras que el usuario puede comprender. Está especificada en la sintaxis sujeto-verbo-objeto cada vez que es posible.

La **descripción del evento** informa al lector en términos claros y simples, cuáles son las políticas del negocio para el evento. Si los usuarios no leen más que la descripción, deberán ser capaces de verificar si se capturó la esencia de su actividad dentro del negocio. La descripción del evento también se vuelve a usar posteriormente en la documentación del diseño.

La sección de **estímulos** del diccionario de eventos identifica los datos que se requieren para activar el evento. Todos los eventos necesitan algún tipo de flujo de datos de entrada. Éste puede tomar la forma de un flujo de datos clásico, con elementos de datos discretos, o de un flujo de control, el cual no contiene datos sino solamente un mensaje del ambiente que le dice al sistema “hazlo”.

La **actividad** sucede dentro del sistema. Este es todo el procesamiento que el sistema debe hacer para convertir la entrada del estímulo en una respuesta adecuada para el evento. Es una mini especificación del proceso. Hay muchas formas para documentar la actividad de un evento.

La sección de **respuesta** del diccionario de eventos identifica los datos requeridos por el usuario para lograr el efecto deseado en el ambiente de negocios.

El **efecto** es la poscondición deseada del ambiente después de que ha recibido la respuesta. Al igual que el evento, el efecto también sucede en el ambiente. El efecto no es parte del sistema automatizado, pero es de gran importancia para los usuarios. El único propósito del sistema es producir el efecto deseado en el mundo real.

ID del evento: 1

Nombre del evento: Proveeduría recibe las solicitudes de las dependencias

Descripción del evento: Las dependencias envían las solicitudes a proveeduría. El usuario de ventanilla la captura en el sistema.

Estímulo: Id de solicitud, Id de Dependencia, Id de Subdependencia, Fecha de elaboración de Solicitud, Id de usuario que la capturó, Tipo de solicitud, Id de cuenta, Id de SubCuenta, Justificación de la Solicitud, Observaciones de la Solicitud, Id grupo, Id familia, Status de la Solicitud, Id artículo, Cantidad solicitada, Especificaciones del artículo.

Actividad: Crear una instancia de Solicitud usando el Id de Solicitud, Id de Dependencia, Id de Subdependencia, Fecha de elaboración de solicitud, , Id de usuario que la capturó, Tipo de solicitud, Id de cuenta, Id de SubCuenta, Justificación de la Solicitud, Observaciones de la Solicitud, Id grupo, Id familia.

Por cada Detalle de Solicitud Recibida

Crear una instancia de detalle de Solicitud usando el Id del artículo, Cantidad solicitada y Especificaciones del artículo.

El Estatus de la Solicitud se pone en “capturada”.

El usuario ingresa su firma de autorización de captura.

Respuesta: Solicitud de Aprovisionamiento

Efecto: El capturista informa a la dependencia cuando se le notificará, si la solicitud fue aceptada o rechazada por presupuesto, y en el caso de ser aceptada, una fecha probable de entrega de mercancía o de servicio.

ID del evento: 2

Nombre del evento: DMS envía solicitudes urgentes (DMS solamente)

Descripción del evento: DMS (Dirección Municipal de Salud) envía las solicitudes urgentes a proveeduría. El usuario de ventanilla las captura en el sistema.

Estímulo: Id de solicitud, Id de Dependencia, Id de Subdependencia, Fecha de elaboración de Solicitud, Id de usuario que la capturó, Tipo de solicitud, Id de cuenta, Id de SubCuenta, Justificación de la Solicitud, Observaciones de la Solicitud, Urge Solicitud, Id grupo, Id familia, Status de la Solicitud, Id artículo, Cantidad solicitada, Especificaciones del artículo.

Actividad: Crear una instancia de Solicitud usando el Id de Solicitud, Id de Dependencia, Id de Subdependencia, Fecha de elaboración de solicitud, , Id de usuario que la capturó, Tipo de solicitud, Id de cuenta, Id de SubCuenta, Justificación de la Solicitud, Observaciones de la Solicitud, Urge Solicitud, Id grupo, Id familia.

Por cada Detalle de Solicitud Recibida

Crear una instancia de detalle de Solicitud usando el Id del artículo, Cantidad solicitada y Especificaciones del artículo.

El Estatus de la Solicitud se pone en “capturada”.

El usuario ingresa su firma de autorización de captura.

Respuesta: Solicitud de Aprovisionamiento

Efecto: El capturista informa a la DMS cuando se le notificará, una fecha probable de entrega de mercancía o de servicio.

ID del evento: 3

Nombre del evento: Tesorería valida el Presupuesto para las solicitudes

Descripción del evento: Tesorería ve cuales son las solicitudes que están por autorizarse contra el presupuesto. Las que se aplican a cuentas que tienen fondos suficientes se autorizan, y las que no lo tienen se rechazan.

Estímulo: Solicitud de Aprovisionamiento

Actividad: Buscar la cuenta y subcuenta que afectan la solicitud.

Buscar en el sistema de tesorería en la tabla de presupuestos, si existe presupuesto suficiente para esa cuenta y subcuenta y dependencia solicitante.

Si hay presupuesto suficiente

Autorizar la Solicitud con la firma electrónica del usuario de presupuestos

Cambiar el estatus de la Solicitud a “Autorizada por Tesorería”

Si no hay presupuesto suficiente

Cambiar el estatus de la Solicitud a “Rechazada por falta de presupuesto”

Respuesta: Solicitud de Aprovevisionamiento autorizada

Efecto: Proveeduría informa a las dependencias que la solicitud fue autorizada por Tesorería y se les da una fecha probable de entrega de los productos o servicios que solicitaron.

ID del evento: 4

Nombre del evento: El usuario imprime las solicitudes

Descripción del evento: El usuario imprime la solicitud para cuestiones legales con contraloría, firmas de autorización y archivo.

Estímulo: Solicitud de Aprovevisionamiento

Actividad: Se manda imprimir la solicitud.

Si la Solicitud no está en estatus de “Cancelada”

Si se imprimió correctamente

El sistema marca esa solicitud como “Impresa” y cualquier copia que se

Imprima posteriormente a esa, se le anexa una leyenda de “COPIA”

El estatus de la Solicitud se cambia a “Impresa”

Respuesta: Solicitud de Aprovevisionamiento Impresa

Efecto: La solicitud es un documento legal que afecta a cuestiones de contraloría y por lo tanto es necesario tener un archivo físico histórico de las solicitudes que se recibieron y autorizaron por tesorería para llevar la recaudación de las firmas y sellos correspondientes.

ID del evento: 5

Nombre del evento: El usuario cancela las solicitudes

Descripción del evento: El usuario cancela la solicitud por falta de presupuesto, por error de captura, o porque la dependencia se dio cuenta de que no era lo que estaba solicitando. Lo puede hacer en cualquier parte del proceso.

Estímulo: Solicitud de Aprovisionamiento

Actividad: El usuario presiona el botón de cancelar

La solicitud se pone en estatus de “cancelada”.

Respuesta: Solicitud de Aprovisionamiento Cancelada

Efecto: Cuando una Solicitud se cancela, se le informa a la dependencia la razón por la cual se cancelo dicha solicitud, para que se gestione los movimientos correspondientes, dependiendo de la razón de la cancelación.

ID del evento: 6

Nombre del evento: El usuario genera las invitaciones a cotizar

Descripción del evento: El usuario genera las invitaciones a cotizar para los proveedores correspondientes al grupo y familia al que pertenece la solicitud.

Estímulo: Solicitud de Aprovisionamiento, Id de los Proveedores que corresponden al grupo y área de la solicitud, Razón Social de los Proveedores, Dirección, Teléfono, Fax, Grupo, Familia, Fecha de elaboración de la Invitación, Status de la Invitación, Id del Artículo, Cantidad, Especificaciones, Usuario que generó la Invitación,.

Actividad: Si el Proveedor corresponde al Grupo y familia de la solicitud.

Agregar una instancia de Invitación con Id de Invitación, Fecha de Elaboración, Usuario que genero la invitación, Id del Proveedor, Razón Social, Dirección, Teléfono, Fax, Solicitud de Aprovisionamiento.

El Estatus de la Invitación se pone en “Generada”

Respuesta: Invitación a Cotizar

Efecto: Se les informa a los proveedores que fueron invitados a cotizar que artículos o servicios se les solicita, y el personal de proveeduría sabe cuantas cotizaciones esperar para dicha solicitud.

ID del evento: 7

Nombre del evento: El usuario imprime las invitaciones a cotizar

Descripción del evento: El usuario imprime las invitaciones que se generaron para que los proveedores que son invitados pasen a recoger el documento a Proveeduría.

Estímulo: Invitación a Cotizar

Actividad: Se manda imprimir la Invitación a Cotizar

Si el estatus de la Invitación no esta en “Impresa”

Si se imprimió correctamente

El sistema marca esa invitación como impresa y cualquier copia que se

Imprima posteriormente a esa, se le anexa una leyenda de “COPIA”

El estatus de la Solicitud se cambia a “Invitada a cotizar”

El estatus de la Invitación se cambia a “Impresa”

Respuesta: Invitación a Cotizar Impresa

Efecto: El proveedor pasa a recoger la invitación a proveeduría para posteriormente enviar la cotización respectiva.

ID del evento: 8

Nombre del evento: El usuario captura las cotizaciones

Descripción del evento: El usuario captura las cotizaciones que el proveedor entrega en ventanilla.

Estímulo: Invitación a Cotizar, Cantidad cotizada, Precio Unitario, Garantía, Tiempo de Entrega, Condiciones de Pago, Anticipo, Estatus.

Actividad: Se modifica una instancia de la Invitación a cotizar, agregándole cantidad cotizada, Precio Unitario, Garantía, tiempo de Entrega, Condiciones de pago, Anticipo.

Se pone el Estatus de la Invitación / Cotización en “Cotización Capturada”

Se cambia el estatus de la Solicitud a “Cotizada”

Respuesta: Cotización.

Efecto: Se notifica al Proveedor una posible fecha de resolución en caso de que haya resultado ganador.

ID del evento: 9

Nombre del evento: El usuario cancela las cotizaciones

Descripción del evento: El usuario cancela la cotización por error de captura, o porque el proveedor fue sancionado, o porque la cotización perdió validez por vencimiento.

Estímulo: Cotización.

Actividad: El usuario cancela manualmente la cotización.

Se cambia el estatus de la Invitación / Cotización a “Cancelada”

Respuesta: Cotización Cancelada.

Efecto: En caso de ser necesario, se le solicita al proveedor que vuelva a cotizar. O en caso de que sea otro proveedor el que resulte ganador, se le solicita una cotización nueva.

ID del evento: 10

Nombre del evento: El usuario genera el cuadro comparativo

Descripción del evento: El usuario genera el cuadro comparativo como herramienta para la toma de decisión del proveedor ganador

Estímulo: Cotizaciones, Id cuadro, Id solicitud, Fecha de Elaboración, Usuario que lo generó.

Actividad: Se genera el cuadro comparativo

Si la cotización pertenece a la Solicitud seleccionada

Si la Invitación / Cotización no está en estatus de “Cancelada”

Agregar la Invitación / Cotización al Cuadro Comparativo

Grabar el Id del cuadro en la tabla de Invitaciones / Cotizaciones

Respuesta: Cuadro Comparativo.

Efecto: En este punto no existe ningún efecto en el ambiente, hasta que se imprime el cuadro.

ID del evento: 11

Nombre del evento: El usuario imprime el cuadro comparativo

Descripción del evento: El usuario imprime el cuadro comparativo para pasarlo al comité de adquisiciones para elegir al(os) proveedor(es) ganador(es).

Estímulo: Cuadro comparativo.

Actividad: Se imprime el cuadro comparativo

Si se imprimió bien el cuadro

Marcar el cuadro como impreso, y cualquier copia que se imprima posterior a esa se le agrega una leyenda que dice “COPIA”

El estatus de la solicitud se cambia a “Por decidir”

Respuesta: Cuadro Comparativo Impreso.

Efecto: El comité de adquisiciones tiene un documento impreso que le proporciona toda la información necesaria para poder elegir a los proveedores a los cuales se les van a asignar los productos o servicios que se están solicitando.

ID del evento: 12

Nombre del evento: El usuario selecciona proveedor(es) ganador(es)

Descripción del evento: El usuario selecciona al(os) proveedor(es) ganador(es) y les asigna el(los) producto(s) o servicio(s)

Estímulo: Cuadro comparativo.

Actividad: Por cada artículo de la Solicitud

Se asigna al proveedor ganador

Se introduce una firma electrónica de autorización (del usuario responsable)

Se cambia el estatus de la solicitud por “Asignada”

El estatus del Cuadro Comparativo se cambia a “Asignado”

Respuesta: Cuadro comparativo Asignado

Efecto: Se les notifica a los proveedores ganadores que surtan en los domicilios correspondientes los productos o servicios que se les están asignando.

ID del evento: 13

Nombre del evento: El usuario imprime la PreOrden de Servicio (Taller solamente)

Descripción del evento: Antes de Generar la Orden de Servicio final, el usuario puede asignar a un proveedor e imprimir un formato de PreOrden de Servicio para que el vehículo se vaya a taller externo.

Estímulo: Id de la PreOrden, Solicitud de Servicio, Fecha de Elaboración, Id del Proveedor, Id del Vehículo, Dependencia del Vehículo, Fecha de salida de Taller.

Actividad: Se asigna un Proveedor para la Solicitud de Servicio

Se imprime la PreOrden de Servicio.

Si se imprimió bien

Se marca como impresa, y cualquier impresión posterior a ésta se le anexa una leyenda de “COPIA”

Se guarda la asignación en un archivo de PreOrdenes

El estatus de la Solicitud se pone en “Asignada a taller Externo”

Respuesta: PreOrden de Servicio Impresa

Efecto: El Vehículo es llevado al taller donde va a ser reparado, y la PreOrden de Servicio le sirve al proveedor para saber que servicio se le va a realizar.

ID del evento: 14

Nombre del evento: El usuario genera las Ordenes de Compra o Servicio

Descripción del evento: Cuando ya esta asignado el proveedor para surtir los productos o servicios que están en la solicitud, se le genera su orden de compra o servicio.

Estímulo: Cuadro Comparativo Asignado.

PreOrden de Servicio (Sólo en el caso de talleres municipales para Ordenes de Servicio)

Actividad: Si la Orden de Servicio es de taller municipal

Por cada instancia de PreOrdenes de Servicio que no tengan Orden de Servicio

Generar Orden de Servicio

Se introduce la firma electrónica correspondiente al usuario responsable

La PreOrden se marca como “Orden de Servicio Generada”

El estatus de la Solicitud cambia a “Orden Generada”

En caso contrario

Por cada instancia de cuadro comparativo asignado que no tenga Orden de

Compra

Generar Orden de Compra o Servicio

Se introduce la firma electrónica correspondiente al usuario responsable

EL cuadro comparativo se marca como “Orden de servicio Generada”

El estatus de la Solicitud cambia a “Orden Generada”

Respuesta: Orden de Compra o Servicio

Efecto: Se le notifica al proveedor que pase a la dependencia de Proveeduría (o en el caso de talleres, a la Dirección de taller municipal) a recoger las ordenes de compra o servicio que les corresponde.

ID del evento: 15

Nombre del evento: El usuario imprime las Ordenes de Compra o Servicio

Descripción del evento: Cuando el proveedor va a recoger las ordenes de compra o servicio que les corresponden, se imprimen en ese momento para que puedan firmar de recibido y posteriormente ellos puedan hacer el cobro correspondiente en Tesorería.

Estímulo: Ordenes de Compra o Servicio

Actividad: El usuario manda imprimir la orden de compra o servicio seleccionada
El sistema solicita la firma electrónica de autorización para la impresión.

Si la orden de compra se imprimió bien

Se marca como impresa y cualquier impresión posterior a ésta se le anexa una leyenda “COPIA SIN VALIDEZ OFICIAL”

El estatus de la Solicitud se cambia a “Orden Entregada”

Respuesta: Orden de Compra o Servicio impresa.

Efecto: Se le entrega la Orden de Compra o Servicio al Proveedor para que pueda ir a cobrarla a Tesorería.

ID del evento: 16

Nombre del evento: Se actualiza la información de las Ordenes en Tesorería

Descripción del evento: Se guardan en el sistema de Tesorería todas las Ordenes que se van generando para mantener un control de todos los pagos que ésta dependencia hará en los siguientes días.

Estímulo: Ordenes de Compra o Servicio

Actividad: Por cada Orden de compra o Servicio que no esté marcada como “Actualizada en Tesorería”

Crear una Instancia en la tabla de Ordenes por Pagar del sistema de Tesorería con los datos de la Orden de compra o servicio correspondiente.

Se marca la Orden de compra o servicio como “Actualizada en Tesorería”

Respuesta: Orden de Compra o Servicio por Cobrar en el Sistema de Tesorería.

Efecto: El personal de Tesorería hace la relación correspondiente de las ordenes que se van pagar a los proveedores y en su sistema lleva la contabilidad correspondiente, afectando el presupuesto final de la Cuenta y Subcuenta que afectó la Solicitud.

ID del evento: 17

Nombre del evento: El usuario cancela Ordenes de Compra o Servicio

Descripción del evento: El usuario cancela la Orden de compra o servicio, por que el proveedor no cumplió con lo pactado, ya sea en producto, servicio, precio, garantía, tiempo de entrega, calidad del producto o servicio, o por error de captura del usuario que la generó.

Estímulo: Ordenes de Compra o Servicio

Actividad: El usuario cancela la Orden de compra o servicio

Si la Orden de compra o servicio esta marcada como “Impresa”

Mandar un mensaje al usuario: “Esta Orden no puede ser Cancelada por que ya fue entregada al Proveedor”.

Si la Orden de compra o servicio no esta marcada como “Impresa”

Se introduce la firma electrónica correspondiente del usuario responsable

Se introduce la razón de cancelación de la Orden

La orden de compra o servicio se marca como “Cancelada”

Si la Orden de Servicio pertenece a Taller

La PreOrden se marca como “Cancelada”

En caso contrario

El cuadro comparativo asignado se marca como “Cancelado”

El estatus de la solicitud se pone en “Cotizada” (para volver a generar otro cuadro

Comparativo, u otra PreOrden de Servicio en el caso de Taller)

Respuesta: Orden de Compra o Servicio Cancelada

Efecto: El comité de adquisiciones tiene que volver a escoger a otro proveedor para asignárselo a la orden de compra o servicio.

ID del evento: 18

Nombre del evento: Se actualiza la información de las Ordenes Canceladas en Tesorería

Descripción del evento: Se guardan en el sistema de Tesorería todas las Ordenes Canceladas

Estímulo: Ordenes de Compra o Servicio Canceladas

Actividad: Por cada Orden de compra o Servicio Canceladas

Agregar una instancia en la tabla de Ordenes Canceladas del sistema de Tesorería con los datos de la orden de compra o servicio.

Respuesta: Orden de Compra o Servicio Cancelada en el Sistema de Tesorería.

Efecto: El personal de Tesorería hace la relación correspondiente de las ordenes que se cancelaron para que la numeración coincida con la tabla de las ordenes por pagar. Esto para tramites de contraloría.

ID del evento: 19

Nombre del evento: El usuario actualiza en el sistema de bienes muebles los bienes adquiridos por el municipio (Patrimonio solamente)

Descripción del evento: Se guardan en el sistema de Patrimonio todos los bienes muebles que se van adquiriendo y que se tienen que inventariar

Estímulo: Id de la Orden de Compra que pertenezca al grupo y familia de Bienes Muebles., Fecha de la Orden de Compra, Id del Proveedor, No. Factura, Fecha de Factura, Id del Artículo, cantidad, Especificaciones del Artículo, Garantía.

Actividad: Si la Orden de Compra pertenece al grupo y familia de bienes muebles

Crear una nueva instancia en la tabla de bienes no inventariados del sistema de Patrimonio con Id de Orden de Compra, Fecha de la Orden de Compra, Id del Proveedor, No. Factura, Fecha de Factura, Id del Artículo, cantidad, Especificaciones del Artículo, Garantía.

Respuesta: Bienes Actualizados en el Sistema de Patrimonio

Efecto: La dirección de patrimonio utiliza esta tabla para asignarle al mueble manualmente su numero patrimonial y ponerlo en lista de espera para que se le vaya a poner su etiqueta e ingresarlo al resguardo de esa dependencia.

Modelo de Información

Un modelo de información completo que sea lo suficientemente detallado para que sea útil en el diseño subsecuente debe incluir lo siguiente:

El **diagrama entidad-relación** mostrando todas las entidades denominadas, relaciones nombradas y la cardinalidad mínima y máxima de cada relación en ambas direcciones. Los diagramas grandes deben dividirse para asegurar su legibilidad.

Un **listado de atributos** para cada entidad.

Una **definición** escrita y clara de cada entidad, relación y atributo.

Las **propiedades** de cada atributo incluyendo: opcionalidad, tipo de dato, rango, unidad de medida, precisión y valores restringidos.

Diagramas de estado-transición para cada atributo de estado o ciclo de vida de entidad relevante.

Una diversidad **de matrices de entidad**.

Este modelo proporciona las bases detalladas para todas las decisiones de diseño de datos que vienen a continuación, incluyendo el diseño de base de datos físico, la distribución de los datos y hasta la disposición de ventanas y reportes

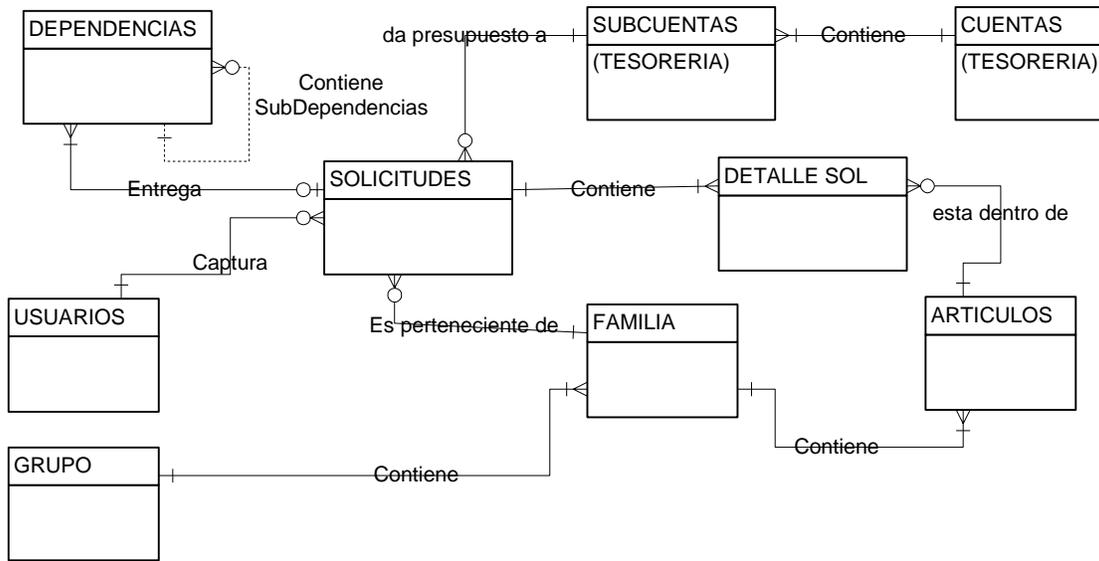
Diagrama Entidad-Relación

El elemento gráfico principal del modelo de información es el diagrama ERD (entidad-relación). Está compuesto de las entidades acerca de las cuales el sistema necesita recordar hechos específicos y las relaciones que existen entre estos grupos de hechos.

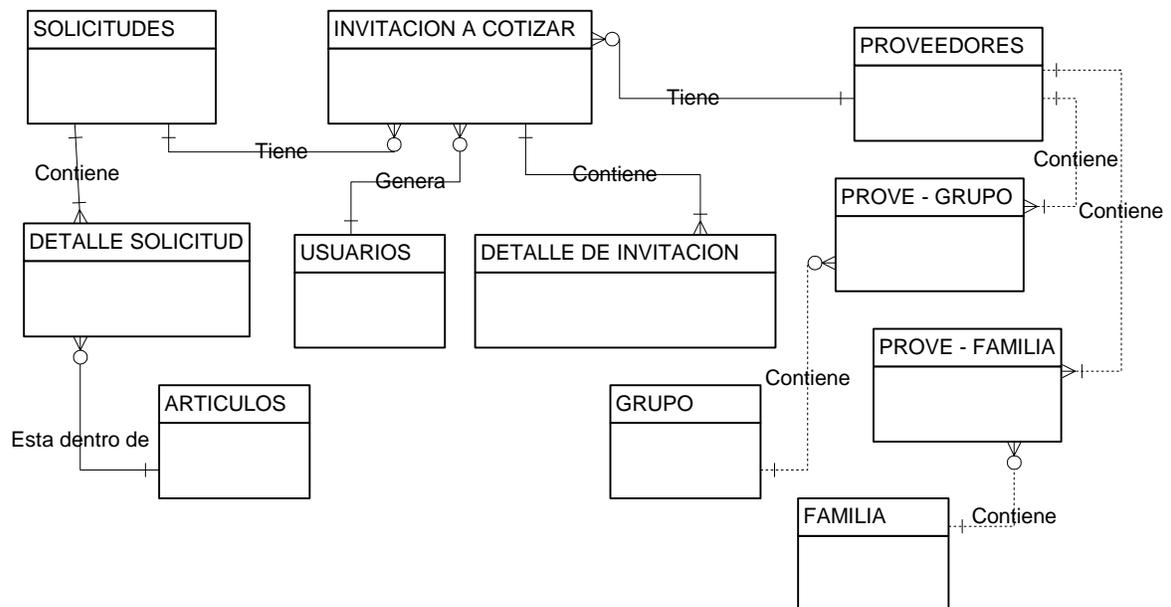
Los Diagramas Entidad-Relación de este sistema, y que mostramos a continuación ya se encuentran en la tercera forma normal, y por razones prácticas de la tesis, el proceso de normalización se obvia.

Los atributos se omiten en este diagrama por razones de espacio y legibilidad, pero se detallan después de los diagramas E-R. También se secciona el diagrama en partes, tomando en cuenta el proceso natural del sistema.

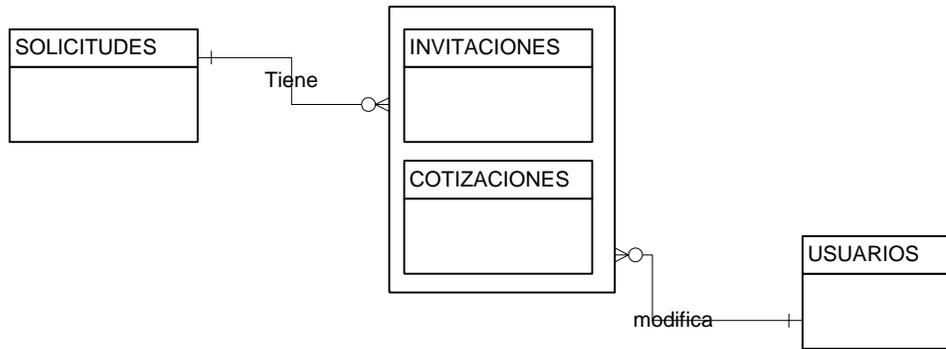
Captura de Solicitudes



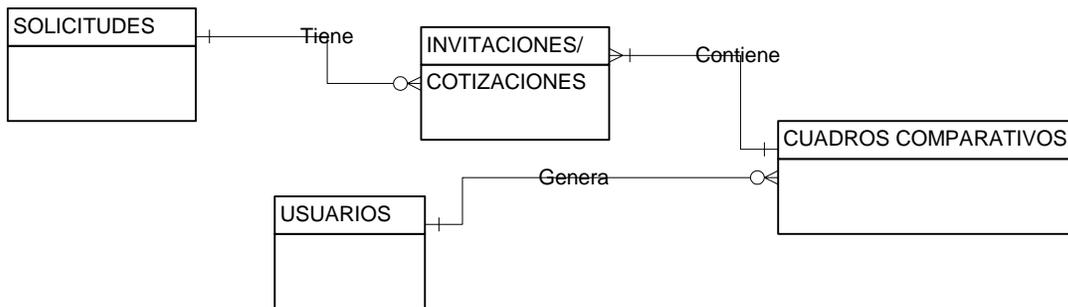
Generación de Invitaciones a Cotizar



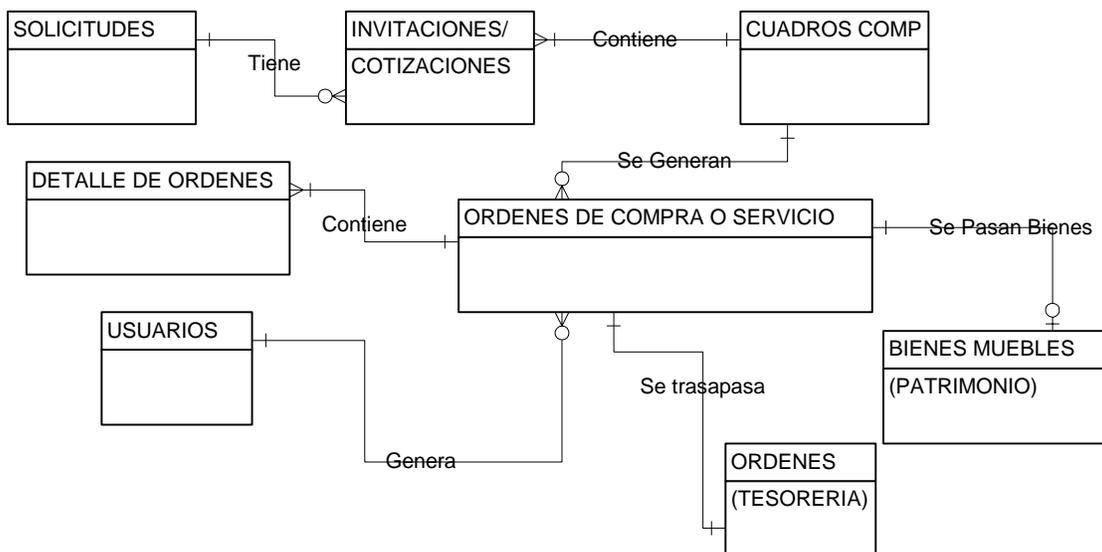
Captura de Cotizaciones



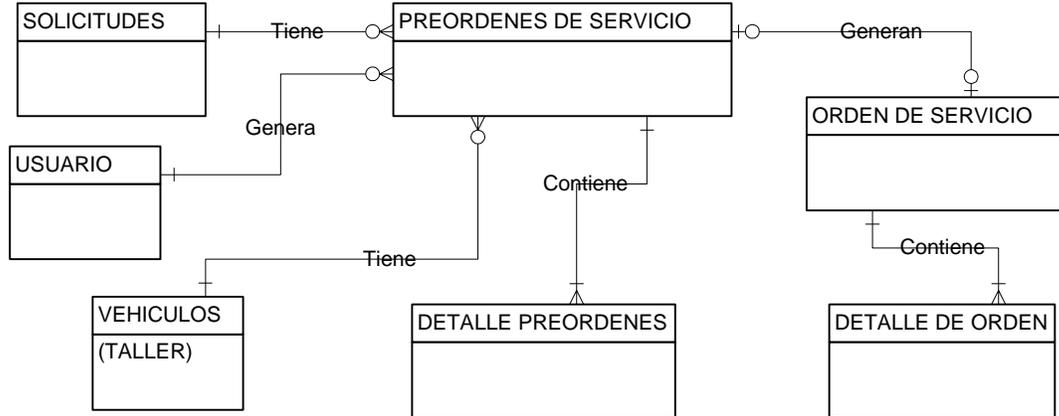
Generación de Cuadro Comparativo



Generación de Ordenes de Compra o Servicio



Generación de Preordenes de servicio a taller externo (Talleres Municipales)



Los Procesos de validación del presupuesto de parte de Tesorería, la actualización de los bienes en Patrimonio, y la actualización de las Ordenes de Compra y Servicio en la base de datos de Tesorería no se reflejan en éste diagrama, ya que las tablas entre los diferentes sistemas no tienen una relación directa. En el caso de la validación del presupuesto, únicamente se lee la tabla de presupuestos para determinada partida y dependencia, si hay o no dinero disponible, se valida o cancela la solicitud dependiendo del caso. Así mismo, en la actualización de los sistemas de Patrimonio y Tesorería únicamente se registran las adquisiciones correspondientes. Por último, el proceso de adquisición de la Dirección Municipal de Salud es similar al proceso normal, pero sólo con menos restricciones.

A continuación se muestran los atributos que están contenidos dentro de las tablas, su tipo de dato y su cardinalidad. La cardinalidad define los valores mínimo y máximo que ese campo puede tener. |

Lista de Atributos por Entidad

La cardinalidad del atributo nos indica el valor mínimo y máximo que puede tomar. Si como valor mínimo lleva un 1, quiere decir que es un valor requerido. La tercera columna nos indica el tipo de dato del atributo.

Entidad: SOLICITUDES

<u>Atributos:</u>	1-1	Id Solicitud (ID)	Ulong (Unsigned Long)
	1-1	Id Dependencia	Integer
	1-1	Id Subdependencia	Integer
	1-1	Fecha Elaboración	Date
	1-1	Id Usuario	Integer
	1-1	Tipo	Byte
	1-1	Id Cuenta	Integer
	1-1	Id SubCuenta	Integer
	1-1	Justificación	String [255]
	0-1	Observaciones	String [255]
	1-1	Id Grupo	Integer
	1-1	Id Familia	Integer
	1-1	Status	String [40]
	0-1	Urgente	Byte
	0-1	Firma Validación	Integer
	0-1	Impresa	Byte

Entidad: DETALLE SOLICITUD

<u>Atributos:</u>	1-M	Id Solicitud (ID)	Ulong
	1-M	Id artículo	Long
	1-M	Cantidad solicitada	Integer
	1-M	Especificaciones	String [255]

Entidad: INVITACIONES A COTIZAR / COTIZACIONES

<u>Atributos:</u>	1-1	Id Invitación / Cotización	Ulong
	1-1	Id Solicitud	Ulong
	1-1	Id Proveedor	Long
	1-1	Fecha Elaboración	Date
	0-1	Fecha Cotización	Date
	1-1	Id Grupo	Integer
	1-1	Id Familia	Integer
	1-1	Estatus	String [40]
	0-1	Impresa	Byte
	0-1	Tiempo de Entrega	String [50]
	0-1	Condiciones de Pago	String [50]
	0-1	Anticipo	String [50]
	1-1	Id Usuario	Integer
	0-1	Id Cuadro	Ulong

Entidad: DETALLE INVITACIÓN / COTIZACIÓN

<u>Atributos:</u>	1-M	Id Invitación / Cotización	Ulong
	1-M	Id Artículo	Long
	1-M	Cantidad solicitada	Integer
	1-M	Especificaciones	String [255]
	0-M	Cantidad Cotizada	Integer
	0-M	Precio Unitario	Long
	0-M	Garantía	String [255]

Entidad: CUADROS COMPARATIVOS

Atributos:

1-1	Id Cuadro	Ulong
1-M	Id Solicitud	Ulong
1-1	Fecha de Elaboración	Date
0-1	Impreso	Byte
1-1	Id Usuario	Integer

Entidad: PREORDEN DE SERVICIO

Atributos:

1-1	Id PreOrden	Ulong
1-1	Id Solicitud	Ulong
1-1	Fecha de Elaboración	Date
1-1	Id Proveedor	Long
1-1	Id Vehículo	Long
1-1	Fecha de Salida a Taller Ext.	Date
0-1	Fecha de Retorno de Taller Ext.	Date
1-1	Id Usuario	Integer
0-1	Impreso	Byte
1-1	Estatus	String [40]

Entidad: DETALLE DE PREORDEN DE SERVICIO

Atributos:

1-M	Id PreOrden	Ulong
1-M	Id Servicio	Ulong
1-M	Cantidad	Integer
1-M	Especificaciones	String [255]
0-M	Especificaciones Finales	String [255]
0-1	Precio Unitario	Long

Entidad: ORDENES DE COMPRA / SERVICIO

<u>Atributos:</u>	1-1	Id Orden	Ulong
	0-1	Id Cuadro	Ulong
	0-1	Id PreOrden	Ulong
	1-1	Fecha de Elaboración	Date
	1-1	Id Proveedor	Long
	0-1	Tiempo de Entrega	String [50]
	0-1	Condiciones de Pago	String [50]
	0-1	Anticipo	String [50]
	1-1	Total de la Orden	Integer
	1-1	Estatus	String [40]
	0-1	Impreso	Byte
	1-1	Id Usuario	Integer

Entidad: DETALLE DE ORDENES DE COMPRA / SERVICIO

<u>Atributos:</u>	1-M	Id Orden	Ulong
	1-M	Id Articulo	Long
	1-M	Cantidad	Integer
	1-M	Especificaciones	String [255]
	1-M	Precio Unitario	Long
	0-M	Garantía	String [255]

Entidad: CATÁLOGO DE ARTÍCULOS

<u>Atributos:</u>	1-1	Id Artículo	Long
	1-1	Nombre del Artículo	String [100]
	1-M	Id Grupo	Integer
	1-M	Id Familia	Integer

Entidad: CATÁLOGO DE DEPENDENCIAS

Atributos:

1-1	Id Dependencia	Integer
1-1	Nombre de la Dependencia	String [100]
0-1	Dirección	String [200]
0-1	Teléfono	String [20]
0-1	Fax	String [20]

Entidad: CATÁLOGO DE SUBDEPENDENCIAS (ALIAS DE DEPENDENCIAS)

Atributos:

1-1	Id Dependencia	Integer
1-1	Nombre de la Dependencia	String [100]
0-1	Dirección	String [200]
0-1	Teléfono	String [20]
0-1	Fax	String [20]

Entidad: CATÁLOGO DE GRUPOS

Atributos:

1-1	Id Grupo	Integer
1-1	Nombre del Grupo	String [100]

Entidad: CATÁLOGO DE FAMILIAS

Atributos:

1-M	Id Grupo	Integer
1-1	Id Familia	Integer
1-1	Nombre de la Familia	String [100]

Entidad: CATÁLOGO DE PROVEEDORES

Atributos:

1-1	Id Proveedor	Long
1-1	Razón social	String [100]
0-1	Dirección	String [200]

0-1	Teléfono	String [20]
0-1	Fax	String [20]
0-1	Representante	String [50]
0-1	Teléfono del Representante	String [20]

Entidad: CATÁLOGO DE USUARIOS

Atributos:

1-1	Id Usuario	Integer
1-1	Nombre del Usuario	String [100]
1-1	Password	String [10]
1-1	Firma Electrónica	String [10]
0-1	Puesto	String [50]
0-1	Teléfono	String [20]

Entidad: PROVEEDOR - GRUPO

Atributos:

0-M	Id Proveedor	Long
0-M	Id Grupo	Integer

Entidad: PROVEEDOR - FAMILIA

Atributos:

0-M	Id Proveedor	Long
0-M	Id Familia	Integer

Las siguientes entidades pertenecen a los sistemas externos que están involucrados en el proceso. Su estructura dependerá de los ajustes de sus respectivos sistemas. Esto es debido a que se está llevando a cabo una reingeniería en las dependencias involucradas en el proceso y es muy posible que se modifiquen los sistemas actualmente funcionales que participan en el proceso de adquisiciones de manera local

Entidad: BIENES MUEBLES (SISTEMA DE PATRIMONIO)

Atributos:

1-M	Id Artículo	Ulong
1-M	Especificaciones	String [255]
1-M	Cantidad	Integer
1-M	Clave del Grupo	Integer
1-M	Clave de la Familia	Integer
1-M	Id Orden de Compra	Ulong
1-M	Fecha de la Orden de Compra	Date
0-M	Id Proveedor	Long
0-M	Número de Factura	Integer
0-M	Fecha de la Factura	Date
0-M	Garantía	String [255]

Entidad: CUENTAS PRESUPUESTALES (SISTEMA DE TESORERÍA)

Atributos:

1-1	Id Cuenta	Integer
1-1	Nombre de la Cuenta	String [100]
1-1	Presupuesto Total	Ulong

Entidad: SUBCUENTAS PRESUPUESTALES (SISTEMA DE TESORERÍA)

Atributos:

1-M	Id Cuenta	Integer
1-1	Id Subcuenta	Integer
1-1	Nombre de la SubCuenta	String [100]
1-1	Presupuesto Disponible	Ulong

Entidad: CATÁLOGO DE VEHÍCULOS (SISTEMA DE TALLER)

Atributos:

1-1	Id Vehículo	Long
1-1	Marca	String [50]
1-1	SubMarca	String [50]
1-1	Placas	String [10]
1-1	Numero Patrimonial	String [10]
1-1	Clave de la Dependencia	Integer

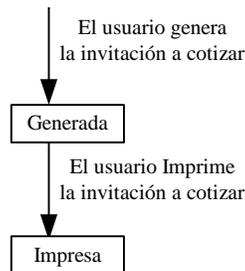
Diagramas de Transiciones de Estado

Un diagrama de transición de estado nos muestra el ciclo de vida de la instancia de una entidad. Dicho de otro modo, la instancia de una entidad nace, pasa por varias fases en su vida y eventualmente muere. En el caso del SIMA, los identificadores sobre la fase del ciclo de vida de una entidad que el sistema tiene planeado registrar es un atributo llamado Estatus.

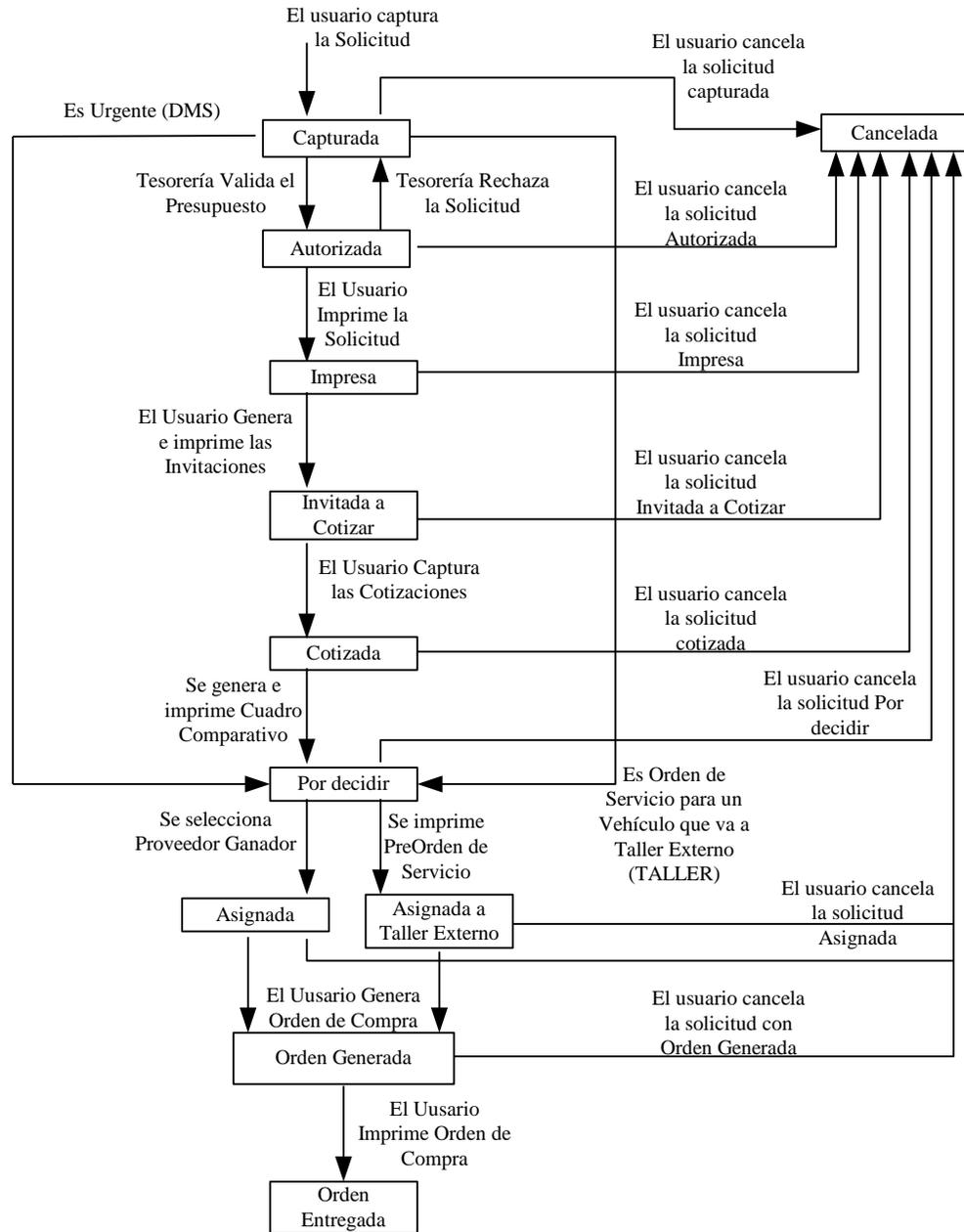
Para “lanzar” una entidad de un valor de estado al siguiente se necesita un evento. El diagrama de transición de estados hace un enlace importante entre el modelo de eventos y el modelo de información debido a que muestra gráficamente cuáles eventos cambian el valor de un atributo de estado dado.

Los valores de estado se muestran dentro del cuadro. Los eventos se muestran como flechas, las cuales indican la dirección del cambio de estado. El estado inicial es indicado por una flecha que entra al diagrama por la parte superior o por un lado

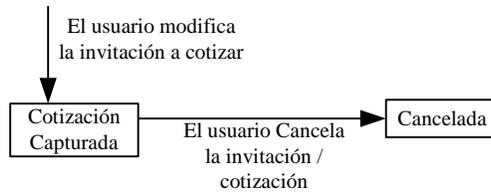
ESTADO DE LA INVITACIÓN A COTIZAR



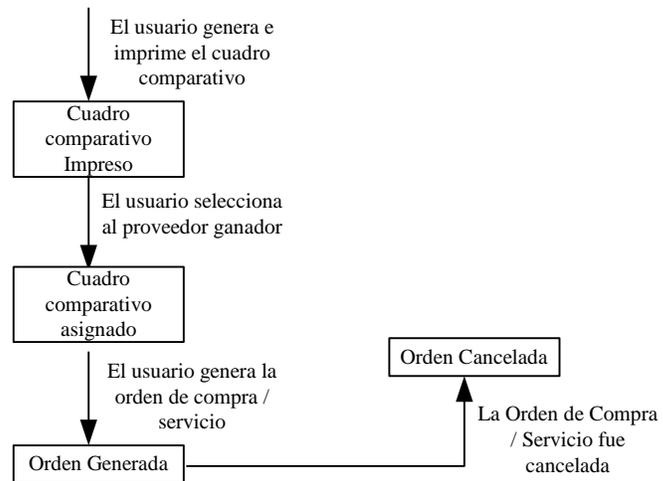
ESTADO DE LA SOLICITUD



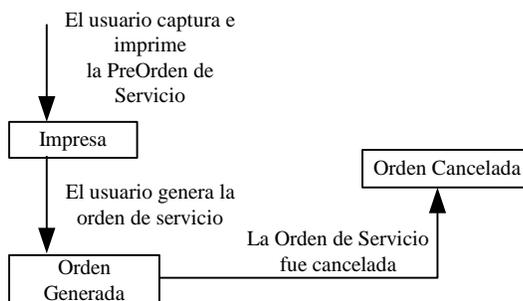
ESTADO DE LA COTIZACIÓN



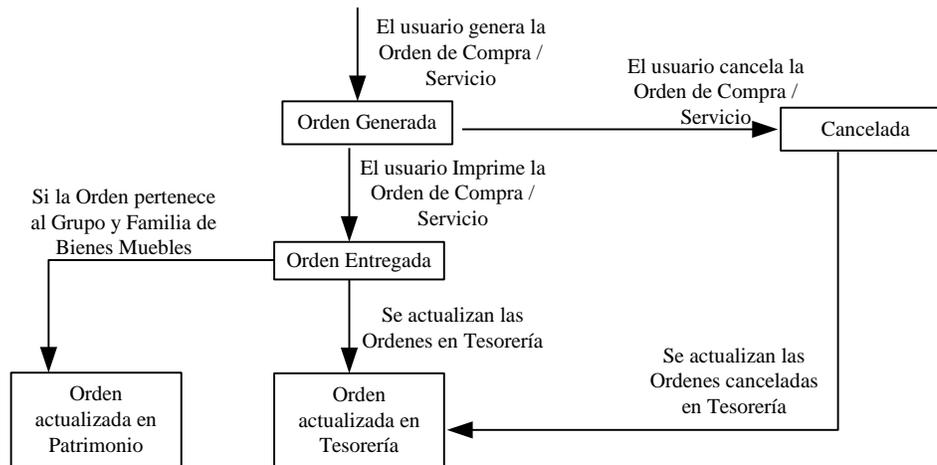
ESTADO DEL CUADRO COMPARATIVO



ESTADO DE LA PREORDEN DE SERVICIO



ESTADO DE LA ORDEN DE COMPRA / SERVICIO



Matrices de Entidad

Una vez que se tiene una lista de entidades en el sistema se pueden construir una diversidad de matrices para relacionar las entidades con otros objetos en el modelo. En la ingeniería de información tradicional [5], las matrices de entidad se usan ampliamente en la formación del plan estratégico de la información de una organización. Estas matrices y sus variaciones son también útiles para derivar los requerimientos de distribución para una arquitectura cliente / servidor general.

Los requerimientos de datos de cada evento pueden ser resumidos en una matriz CRUD (Crear, Leer, Actualizar, Borrar) evento / entidad. Esto muestra cuáles eventos crean, leen actualizan o borran instancias de las entidades en el modelo de información.

Si el evento sucede en la ubicación del negocio, entonces los datos requeridos del evento también deben estar accesibles en esa ubicación. La matriz CRUD de ubicación de negocio / entidad despliega los requerimientos de distribución de datos

MATRIZ EVENTO / UBICACIÓN

<u>Evento / ubicación</u>	Proveeduría	Tesorería	DMS	Patrimonio	Taller Municipal
Proveeduría recibe las solicitudes de las dependencias	X				
DMS envía solicitudes urgentes (DMS solamente)			X		
Tesorería valida el Presupuesto para las solicitudes		X			
El usuario imprime las solicitudes	X				X
El usuario cancela las solicitudes	X				X
El usuario genera las invitaciones a cotizar	X				X
El usuario imprime las invitaciones a cotizar	X				X
El usuario captura las cotizaciones	X				X
El usuario cancela las cotizaciones	X				X
El usuario genera el cuadro comparativo	X				X
El usuario imprime el cuadro comparativo	X				X
El usuario selecciona proveedor(es) ganador(es)	X		X		X
El usuario imprime la PreOrden de Servicio					X
El usuario genera las Ordenes de Compra o Servicio	X		X		X
El usuario imprime las Ordenes de Compra o Servicio	X		X		X
Se actualiza la información de las Ordenes en Tesorería		X			
El usuario cancela Ordenes de Compra o Servicio	X	X	X		X
Se actualiza la información de las Ordenes Canceladas en Tesorería		X			
El usuario actualiza en el sistema de bienes muebles los bienes adquiridos por el municipio				X	

MATRIZ CRUD EVENTO / ENTIDAD

Evento / entidad	Solicitudes	Invitaciones	Cotizaciones	Cuadros Comparativos	Preorden de Servicio	Orden de Compra	Artículos	Familias	Grupos	Proveedores	Dependencias	Usuarios	Cuentas	Subcuentas	Bienes Muebles	Vehículos
Proveeduría recibe solicitudes	C						CR U	CR U	CR U		CR U	R	R	R		
DMS envía solicitudes urgentes	C						R	R	R		R	R	R	R		
Tesorería valida el Presupuesto	RU										R		CR UD	CR UD		
El usuario imprime las solicitudes	RU						R				R	R	R	R		
El usuario cancela las solicitudes	RU											R				
El usuario genera las invitaciones	RU	C					R	R	R	CR U	R	R				
El usuario imprime las invitaciones	U	RU					R			R	R	R				
El usuario captura las cotizaciones	U	RU	U				R			R	R	R				
El usuario cancela las cotizaciones	U	RU	RU									R				
El usuario genera cuadro comparativo	U	R	R	C			R			R	R	R				
El usuario imprime cuadro comparativo	U	R	R	RU			R			R	R	R				
El usuario selecciona proveedor(es) ganador(es)	U		R	RU						R		R				
El usuario imprime la PreOrden de Servicio	RU				CR U		R	R	R	R	R	R	R	R		R
El usuario genera Ordenes de Compra o Servicio	U		R	RU	RU	C	R			R	R	R				
El usuario imprime Ordenes de Compra o Servicio	U						RU	R			R	R				
Se actualiza la información de las Ordenes en Tesorería							RU	R			R	R		RU	RU	
El usuario cancela Ordenes de Compra o Servicio	U						U					R				
Se actualiza la información de las Ordenes Canceladas en Tesorería							RU	R			R	R		RU	RU	

El usuario actualiza en el sistema de bienes muebles							RU	R	R	R	R	R						CR U
--	--	--	--	--	--	--	----	---	---	---	---	---	--	--	--	--	--	---------

A partir de éstas dos matrices se puede derivar una tercera. Una vez que se conoce la distribución geográfica de los eventos y los requerimientos del acceso de datos de cada evento se puede derivar la distribución geográfica de los requerimientos de acceso de datos, la matriz CRUD de ubicación / entidad. Esta importante matriz nos dice exactamente cuáles ubicaciones necesitan capacidades de crear, leer, actualizar y borrar para cada entidad del modelo.

MATRIZ CRUD UBICACIÓN / ENTIDAD

<u>Ubicación / entidad</u>	Solicitudes	Invitaciones	Cotizaciones	Cuadros Comparativos	Preorden de Servicio	Orden de Compra	Artículos	Familias	Grupos	Proveedores	Dependencias	Usuarios	Cuentas	Subcuentas	Bienes Muebles	Vehículos
Proveeduría	CR U	CR U	CR U	CR U		CRU	CRU D	CR UD	CR UD	CR UD	CR UD	CR UD	R	R		
Tesorería	RU					RU	R	R	R	R	R		CR UD	CR UD		
DMS	CR U					CRU	R	R	R	R	R	R	R	R		
Taller Municipal	CR U	CR U	CR U	CR U	CR U	CRU	R	R	R	R	R	R	R	R		R
Patrimonio						RU	R	R	R	R	R				CR U	

Ahora se tiene la información suficiente para poder tomar decisiones referentes al modelo arquitectónico de nuestro sistema.

Modelo Arquitectónico

La fase de modelado arquitectónico toma decisiones clave acerca de la distribución geográfica, tanto de los datos como de los procesos, a través de la red de área amplia. A un nivel mas detallado, los lenguajes de programación seleccionados pueden influir si se favorece un enfoque de cliente pesado o de servidor pesado para la construcción del código de la aplicación. Se debe de estar conciente que no existe una solución perfecta. Todo se refiere a mediar entre soluciones que no son óptimas. Por medio de los modelo de análisis, se puede ayudar en los esfuerzos propios para escoger la arquitectura técnica menos cuestionable.

El modelo arquitectónico mapea los requerimientos esenciales de la fase de análisis hacia una arquitectura tecnológica. Debido a que son posibles muchísimas arquitecturas diferentes, el objetivo del esfuerzo del modelado arquitectónico es escoger la configuración óptima, o tal vez “la menos mala”.

Muchas de estas decisiones ya están tomadas debido a circunstancias políticas manejadas en el municipio. Estándares de equipo y comunicaciones de red ya implementados y bases de datos y lenguajes ya existentes.

Primero, es necesario determinar si los datos van a ser distribuidos entre las dependencias involucradas en el proceso o se va a centralizar la información en un solo lugar. La respuesta a ésta pregunta, incluyen decisiones políticas exclusivamente, pues existen intereses económicos que intervienen directamente en el proceso y es debido a esto que los directivos han solicitado que se centralice la base de datos en la dependencia que lleva el control del mismo. Los beneficios son numerosos:

Es fácil respaldar los datos cuando sólo existe una copia.

El diseño del sistema general es menos complicado, por ejemplo, la seguridad se mantiene centralizada, no se requieren rutinas de sincronización.

Los datos siempre son actuales.

Ningún dato es redundante a través de las ubicaciones involucradas.

Respecto a los niveles de hardware, el sistema se desarrollará para dos niveles. La razón principal es que el municipio está limitado en lo que a presupuesto se refiere y la opción más económica es el tener a varios clientes haciendo peticiones a un solo servidor de base de datos.

Las capas de software es un punto importante a considerar. El interior de la aplicación puede ser agrupado en al menos tres categorías principales, la capa de presentación, la capa lógica del negocio y la capa de administración de datos. Aquí se tiene otra limitación debido al estándar del software de desarrollo, pues actualmente se cuentan con licencias únicamente de Clarión y Delphi. Clarión es un lenguaje de 4 generación, aunque es considerado mas bien como un generador de código. Éste lenguaje limita a dejar el desarrollo en 2 capas cuando mucho, pues la capa de presentación y la capa de la lógica del proceso están íntimamente atadas y dependientes. Delphi permite un desarrollo de tres o más capas, por lo que es casi un hecho que Borland proporcionará el software de desarrollo. Otras opciones de lenguajes con que se cuenta en el municipio es Power Builder, pero no se cuentan actualmente con las licencias necesarias. La base de datos estará en SQL Server 2000 pues es el estándar actual en el municipio. Recientemente se adquirió Informix para Unix, pero no se cuenta con la capacitación adecuada para desarrollar un sistema de ésta magnitud. En todo caso se tendría que tomar en cuenta si es que los directivos toman esa ruta como decisión definitiva, y además el replantear en el análisis la arquitectura del sistema operativo. Por lo tanto se propone el desarrollo en las 3 capas básicas necesarias para cualquier software cliente / servidor, Delphi y SQL.

Las ventajas de separar la presentación de la lógica del negocio, son reusabilidad, portabilidad y mantenibilidad. La razón mas ampliamente promovida es la reusabilidad. Las clases de presentación son extremadamente fáciles de reutilizar debido a que son muy mecánicas por naturaleza. Se menciona clase para significar una clase de objetos en un sistema orientado a objetos. Por otro lado, las clases de negocios son muy complejas. El objetivo es crear clases que hagan cumplir todas las reglas del negocio para una clase de negocio particular, y que también sean capaces de reutilizar el código en cualquier lugar de la aplicación que tenga que ver con la clase.

La portabilidad es una segunda razón imperiosa para separar el código que maneja la presentación del código que ejecuta la lógica del negocio. Una vez que es dividido de la interfaz, el código de la lógica del negocio puede ser reubicado en diversos niveles de la arquitectura cliente / servidor para lograr un desempeño óptimo. La habilidad para mover la lógica del negocio por la arquitectura cliente / servidor permite que el negocio aproveche las mejoras en velocidad de procesamiento de máquinas particulares y completamente la arquitectura de hardware con un componente más rápido sin tener que volver a escribir grandes secciones de la aplicación.

Una tercera razón para acorralar la lógica del negocio en su propio conjunto de clases es tratar de mantener las reglas del negocio en un lugar en vez de tenerlas desperdigadas por toda la interfaz. La gran ventaja de esta estrategia es la reducción del efecto de propagación de onda al hacer cambios de mantenimiento y mejoras al sistema.

Entre las opciones posibles de tener a un servidor pesado (cliente ligero) o un cliente pesado (servidor ligero) se ha decidido por la opción de Servidor Pesado. Esto con la finalidad de que la carga de trabajo recaiga sobre el servidor y los clientes se dediquen a mostrar la información exclusivamente. Desafortunadamente esto lleva a un dilema. Los usuarios del municipio, al tener conocimiento de aplicaciones GUI modernas, demandan que muchas de las reglas de negocio les sean visualmente aparentes al momento de ejecución. El municipio también tiene la responsabilidad de proteger los activos de datos, que por lo general se encuentran en el servidor, de ser actualizados por otras aplicaciones que puede ser que no estén concientes de las reglas de negocio (En éste caso Tesorería, Patrimonio, Taller Municipal). Este dilema puede forzar a codificar las reglas en dos lugares, a menos que se busque una mejor opción a éste punto como pudiera ser la programación orientada a objetos.

Diseño Relacional

De todo el análisis y diseño presentado anteriormente, resulta el diseño relacional de la base de datos.

CONCLUSIONES

Se realizó el análisis y diseño del Sistema de Adquisiciones del Ayuntamiento de Guadalajara, siguiendo la metodología propuesta por David Ruble, quedan documentados:

- el modelo de contexto,
- el modelo de eventos
- el modelo de información,
- el modelo arquitectónico.

RECOMENDACIONES

Como una segunda etapa, se recomienda utilizar este modelado para la creación de la base de datos y posteriormente de la aplicación. Es importante mencionar que durante el desarrollo pudieran salir especificaciones no contempladas en el modelo original, por lo que sería necesario ajustar este modelo a las nuevas especificaciones, pero es un paso absolutamente normal que todos los diseños de bases de datos tiene que superar. Una vez desarrollado el sistema, y pasando a la etapa de implementación, es necesario contemplar varios puntos:

- Las administraciones municipales sólo duran tres años, por lo que se recomienda llegar a un acuerdo con el titular de la dependencia para que se adquiera el compromiso de que el proyecto no se interrumpirá al término de esta administración.
- En caso de que el titular sea reemplazado dejar un contrato por escrito autorizado y firmado por el Presidente Municipal, para que el titular que entre de reemplazo le dé el seguimiento correspondiente.
- Es necesario tomar en cuenta que la arquitectura propuesta se debe respetar para que con el sistema se obtengan los resultados esperados, únicamente permitiendo o dando entrada, a mejoras de rendimiento a esa arquitectura (memoria, velocidad de procesador, espacio en disco duro, etc. en los equipos, o, en el medio físico de comunicación de la red, como migrar de transmisión coaxial a fibra óptica, etc.)
- Formar un plan de contingencia para los puntos no contemplados en éste apartado como es la rotación de personal, el cambio de ubicación de la dependencia, falla de los equipos, etc.

REFERENCIAS BIBLIOGRAFICAS

[1]Ruble, D. A., “Análisis y diseño práctico de sistemas cliente/servidor con GUI”, Prentice Hall, 1997.

[2]McMenamin S., Palmer J.; “Essential Systems Análisis”, Prentice Hall / Yourdon Press, 1984.

[3]Blanchard K., Johnson S.; “The One Minute Manager”, William Morrow and Company, Inc., 1982

[4]DeMarco T.; “Structured Analysis and System Specification”, Englewood Cliffs, Nueva Jersey: Prentice Hall, 1979.

[5]Martin J., C. Finkelstein. “Information Engineering, (vols. 1 y 2). Londres: Savant Institute, 1981.

[6] <http://www.inei.gob.pe/cpi-mapa/bancopub/libfree/lib616/INDEX.HTM>

BIBLIOGRAFÍA

Blanchard K., Johnson S.; "The One Minute Manager", William Morrow and Company, Inc., 1982

Boar, Bernard H. "Implementing Client/Server Computing: A Strategic Perspective", McGraw Hill, 1992.

Booch, G. "Object-Oriented Analysis and Design with Applications", Second Edition, Redwood City, CA: Benjamin Cummings, 1994.

Cood, Edgar F. "Further Normalization of the Data Base Relational Model", Englewood Cliffs, Nueva Jersey: Prentice Hall, 1990: 355-377.

Ceri S., Pelagatti G.; "Distributed Databases: Principles and Systems", McGraw-Hill, 1984.

Date C. J., "Introducción a los sistemas de bases de datos", 7ma. Edición, Addison Wesley. 2000.

DeMarco T.; "Structured Analysis and System Specification", Englewood Cliffs, Nueva Jersey: Prentice Hall, 1979.

De Miguel Adoración y Piattini Mario, "Conceptos y diseño de bases de datos", 2da. Edición, Editorial Ra-ma. 1999.

Elmasri R., Navathe S. B., "Sistemas de Bases de Datos", 2da. Edición, Addison Wesley Iberoamericana. 1998.

Elmasri, Ramez et al., "Sistemas de Bases de Datos - Conceptos Fundamentales" 2a edición, Addison Wesley Longman de México, 1997

- Gardarin G., "Bases de datos", Editorial Paraninfo. 1995.
- González Alvarado Carlos, "Sistema de Bases de Datos", 1era. Edición, Editorial Tecnológica de Costa Rica. 2001.
- Groff J. R., Weiberg P.N.; "Aplique SQL", McGraw-Hill, 1991.
- Jaeger Ulrike, Freytag Johann Christoph, "An Annotated Bibliography on Active Databases", Humboldt University of Berlin, Germany.
- Jonson James L., "Bases de datos: Modelos, lenguajes, diseño", 1era. Edición, Oxford University Press México. 1997.
- Klim W., "Modern database Systems. The Object Model, Interoperability and Beyond", ACM Press.
- Kroenke David, "Procesamiento de bases de datos", Prentice hall, 1995
- Lockhart Thomas, "Tutorial de PostgreSQL", Postgres Global Development Group.
- Martin J., C. Finkelstein. "Information Engineering, (vols. 1 y 2). Londres: Savant Institute, 1981.
- McFadden F. R., Hoffer J. A., Prescott M. B., "Modern Database Management", Addison Wesley. 1998.
- McMenamin S., Palmer J.; "Essential Systems Análisis", Prentice Hall / Yourdon Press, 1984.
- Meyer B., "Object-Oriented Software Construction", Prentice-Hall, 1988.

- Minker Jack, "Logia and Databases: a 20 Year Retrospective", Institute of Computer Studies, University of Maryland.
- Paton Norman W., "Active rules in database systems", Springer-Verlag New York, Inc., ISBN 0-387-98529-8, 1999
- Piattini M. y Díaz O., "Advanced Databases: Technology and Design", Artech House.
- Richter J., "Windows NT Avanzado", McGraw-Hill, 1994.
- Ruble, D. A., "Análisis y diseño práctico de sistemas cliente/servidor con GUI", Prentice Hall, 1997.
- Silberschatz A. Korth H. y Sudarshan S., "Fundamentos de Bases de Datos", 3ra. Edición, Editorial McGraw-Hill. 1997.
- Stonebraker M., Brown P., "Object- Relational DBMSs tracking the next great wave", Morgan Kauffman Publishers
- Ullman, "Principles of Database System", Editorial Computer Science Press. 1992.
- Unknown, "Mastery of Clarion for Windows - Advanced programming techniques", TopSpeed Corporation, 1996
- Vaughn William R., "Programación de SQL Server 7.0 con Visual Basic 6.0", McGrawHill, 1999.
- Widom Jennifer , Ceri Stefano, "Active Database Systems: Triggers and Rules for Advanced Database Processing", Edited by, Morgan Kaufmann Publishers, Inc., ISBN 1-55860-304-2, 1996.

Páginas de Internet.

<http://www.inei.gob.pe/cpi-mapa/bancopub/libfree/lib616/INDEX.HTM>

<http://www.ucm.es/info/Psyap/Prieto/alum9798/intranet01/cliente.htm>

<http://fractal.gaiasur.com.ar/infoteca/varios/httpyc-s.html#clienteservidor>

http://ourworld.compuserve.com/homepages/palmun/vfp_cs.htm

<http://agamenon.uniandes.edu.co/sistemas/6603.htm>

<http://www.monografias.com/trabajos/refercomp/ refercomp.shtml>

<http://alarcos.inf-cr.uclm.es/doc/bda/doc/lab/BDa-P1.pdf>

<http://www.cs.ucla.edu/~zaniolo/ papers/edbt88.pdf>

<http://freehost09.websamba.com/intitec/ articulos/BaseDatosPerfecta.htm>

<http://acha.uta.cl/yon/Programacion del semestre/CC 763/>

Nuevas_tecnologias/nuevas_tecnologias.htm

http://www.ida.his.se/ida/adc/adc_bib.html

http://www.arrakis.es/~qenda/Articles/ArticleDB/Articulo_DB_9-5-01a.htm