

Áncora: Metodología para el Análisis de Requerimientos de Software conducente al Reuso



María de los Ángeles Sumano López

México, octubre 2001

Contenido

1	Introducción.....	5
2	Fundamentos, alcances y herramientas de Áncora.	7
2.1	Fundamentos de Áncora.....	7
2.2	Alcances de Áncora.	7
2.3	Presentación general del método y sus elementos.....	9
2.4	Guiones y Diálogos.....	10
2.4.1	Guiones.....	10
2.4.2	Diálogos.....	13
2.5	Tablas y Glosarios.	14
2.5.1	Tablas.....	15
2.5.2	Glosarios.....	15
2.6	Modelo Entidad-Relación.....	15
2.6.1	Modelo de Objetos Semánticos.....	17
2.6.1.1	Definición y Características de los Objetos Semánticos.....	17
2.6.1.1.1	Tipos de Objetos Semánticos.....	19
2.6.1.2	Modelando con Objetos Semánticos.....	19
2.7	Cálculo de Puntos de Función.....	21
2.7.1	Asignación de pesos para ALI y AIE.....	22
2.7.2	Asignación de pesos para EE, SE Y CE.....	23
2.7.3	Ajuste de los Puntos de Función.....	23
2.7.3.1	Comunicación de Datos.....	24
2.7.3.2	Procesamiento Distribuido de Datos.....	24
2.7.3.3	Rendimiento.....	25
2.7.3.4	Configuración Altamente Usada.....	25
2.7.3.5	Promedio de Transacciones.....	25
2.7.3.6	Entrada de Datos en Línea.....	26
2.7.3.7	Eficiencia para el Usuario Final.....	26
2.7.3.8	Actualización en Línea.....	27
2.7.3.9	Procesamiento Complejo.....	27
2.7.3.10	Reusabilidad.....	27
2.7.3.11	Facilidad de Instalación.....	28
2.7.3.12	Facilidad de Operación.....	28
2.7.3.13	Varios Sitios.....	29
2.7.3.14	Facilidad de Cambios.....	29
2.7.4	Contabilidad final en términos de Puntos de Función.....	30
2.8	Las Redes semánticas Naturales.....	30
2.8.1	Las Redes Semánticas Naturales en la definición de constructos.....	30
2.8.2	La Red Semántica Natural del Sistema de Software.....	32
2.9	Medición de actitudes.....	34
2.9.1	Construcción de la encuesta.....	36
2.10	Técnica de Grupo Nominal.....	38
2.11	La Reunión de Reflexión y Diseño.....	39
2.11.1	El lugar para la RRD.....	39
2.11.2	Los asistentes a RRD.....	40
2.11.3	La duración de la RRD.....	40
2.11.4	La conducción de la RRD.....	40

2.11.5	Apoyos físicos para la RRD.....	40
2.11.6	Los resultados de la RRD.....	41
2.11.7	El Calendario para la RRD.....	41
3	<i>Áncora y el ciclo de vida.</i>	43
3.1	Entendimiento del dominio y contexto de la aplicación.	45
3.1.1	Construcción de un primer glosario y Guión teórico.....	46
3.1.2	Conociendo la empresa y sus metas.....	46
3.1.3	El Guión de la situación actual.....	47
3.1.4	Identificación de la problemática.....	48
3.2	Recolección y clasificación de Requerimientos.	49
3.2.1	Guión de la propuesta computacional.....	49
3.2.2	Construyendo un primer prototipo.....	50
3.2.2.1	Contenido General de un Manual de Operación.....	50
3.2.3	El Modelo de Datos.....	51
3.2.3.1	Modelo de datos con E-R.....	52
3.2.3.2	El modelo de datos con Objetos Semánticos.....	53
3.2.4	Bitácora de desarrollo del Software.....	53
3.2.4.1	Revisión de la bitácora de desarrollo.....	55
3.2.5	Primera estimación del costo del nuevo software.....	55
3.3	Solución de Conflictos, Jerarquía y Validación de requerimientos de software.	58
3.3.1	La Técnica de Grupo Nominal en la RRD.....	58
3.3.2	Reunión de Reflexión y Diseño con los involucrados.....	58
3.4	El reuso en Áncora.	60
3.4.1	Guardando en Áncora.....	61
3.4.2	Reusando con Áncora.....	62
3.5	Cierre de Áncora.	63
3.5.1	Cierre de detalles.....	63
3.5.2	Conexión con el diseño.....	63
3.5.2.1	Metodología Estructurada Moderna de Yourdon.....	63
3.5.2.2	Metodologías basadas en UML.....	65
3.5.2.3	Metodología para el Desarrollo de Sistemas de Jackson (DSJ).....	71
4	<i>Glosario.</i>	74
	<i>Apéndice A. Ejemplos de Guiones.</i>	75
A.1	Sistema Conserva.	75
A.1.1	Sistema Actual.....	75
A.1.2	Problemas principales.....	76
A.1.3	Estrategia computacional.....	76
A.2	Sistema Metropolitano.	77
A.2.1	Propuesta computacional.....	77
A.3	Sistema para el Control y Seguimiento de las Auditorias.	79
A.3.1	Situación Actual.....	80
A.3.2	Propuesta Computacional.....	81
A.4	Sistema de Creación de Curriculum Vitae.	82
A.4.1	Situación Actual.....	82
A.4.2	Propuesta Computacional.....	83
A.5	Sistema de Control Escolar.	84
A.5.1	Situación Actual.....	85
A.5.2	Propuesta Computacional.....	86

<i>Apéndice B. Formas para la documentación del Análisis de Requerimientos de Software.</i>	<i>87</i>
B.1 Formatos para el conocimiento del dominio y contexto de la aplicación.	87
B.1.1 Tarjeta para las Redes Semánticas Naturales.....	87
B.1.2 Encuesta de Actitud.	88
B.1.3 Glosario del sistema.....	89
B.2 Recolección y clasificación de requerimientos.	90
B.2.1 El Prototipo Rápido.....	90
B.3 Jerarquización, validación y resolución de conflictos de requerimientos.	92
B.3.1 La Reunión de Reflexión y Diseño.....	93

1 Introducción.

Esta guía sirve para aquellas personas que conocen de computación y que desean realizar el Análisis de Requerimientos de Software (**ARS**) siguiendo actividades que les permitan obtener una Especificación de Requerimientos de Software (**ERS**), que resulte útil para todos los involucrados en la elaboración un nuevo software.

Cabe aclarar que las personas que realizarán el ARS pueden ser novatas o experimentadas, tímidas o desenvueltas, pacientes o impacientes; y que los involucrados en el software (cliente, usuario o proveedor de información) también tienen su propia personalidad y sus actitudes hacia la elaboración de un nuevo sistema de software. Áncora impide que estos aspectos sean un impedimento grave para el avance del ARS, ya que las herramientas brindadas sortean algunas barreras.

Al finalizar el ARS con la guía Áncora, se tendrán varios modelos, incluyendo un prototipo rápido, que reflejarán el sistema de software que se le propondrá a los involucrados. Además, se contará con una aproximación del costo del nuevo sistema de software. Sin embargo, faltarán detalles para un documento de ERS completo, tales como los elementos legales que debe tener un contrato, personal necesario para la elaboración del Software y, en su caso, las ganancias monetarias que la gerencia de software desearía obtener.

Áncora cubre la primera etapa en el desarrollo de un nuevo sistema de software, esto es, la definición de qué se quiere. La primera hipótesis de Áncora es que existen diferencias sociales y psicológicas entre los usuarios de los diferentes países, por lo cual se utilizan herramientas psicosociales que han probado su efectividad en el medio nacional; la segunda es que debe definirse bien qué se quiere de un nuevo sistema de software antes de empezar a diseñarlo, para lo cual se hace uso de herramientas de Ingeniería de Software que permiten un modelado formal y la última es que no se tiene que empezar siempre desde cero para definir un nuevo software, pueden utilizarse las experiencias del pasado mediante el reuso de algunos elementos de sistemas similares.

Para hacer una lectura ordenada de este documento se divide en las siguientes partes:

- Fundamentos, alcances y herramientas de Áncora. Se muestran las herramientas que Áncora utiliza para la determinación de los modelos.
- Trabajando con Áncora. Se explica en forma clara, sin intermedios teóricos, como realizar el análisis de requerimientos de un sistema de software utilizando el método propuesto. Además, una vez que se ha realizado el análisis de requerimientos de software es necesario brindar elementos al analista y diseñador para continuar con el desarrollo del software. En esta parte se brindarán los elementos para tal conexión.

- Glosario y Apéndices con ejemplos de algunos sistemas de software y formatos utilizados en Áncora.

2 Fundamentos, alcances y herramientas de Áncora.

Para trabajar con alguna metodología, lo primero es conocer qué se puede hacer con ella, cuál es su enfoque y qué herramientas brinda. Áncora brinda varias herramientas, algunos son adaptaciones de herramientas utilizadas en otras áreas distintas a la computación, otras son del área de la computación y algunas son propias de Áncora. En este capítulo se explican estas herramientas que van desde formas de representación hasta métodos que realizan alguna tarea.

2.1 Fundamentos de Áncora.

La palabra Áncora, sinónimo de ancla, utilizada en el contexto del análisis de requerimientos, dentro del desarrollo de software, transmite la idea de que para empezar a trabajar con la elaboración de un nuevo software se deben tener buenas y sólidas bases, a saber: definir correctamente qué se quiere de un software nuevo y tener una representación clara para el usuario y que, además, permita pasar a las siguientes etapas de desarrollo de software de forma sencilla y natural.

La herramienta de modelado principal de Áncora es el Guión, mismo que trata de representar una obra de teatro, en la que los actores (usuarios) representarán uno o más papeles dentro de ella. La obra será escrita por los involucrados en el software guiados por el analista. Esta forma de trabajar resulta amena y además logra que el involucrado en el nuevo software se apropie de él y en el futuro lo utilice.

Un elemento más en Áncora es el reuso. La mayoría de los sistemas de software tienen cosas en común que pueden ser reutilizadas, por eso no siempre se debe empezar desde cero para elaborar un nuevo software. Áncora brinda formas de guardar los elementos de sistemas anteriores para utilizarlos en sistemas nuevos que resulten similares. La variedad de elementos que se pueden reutilizar van desde un conjunto de actividades hasta los costos y prototipos de sistemas anteriores.

2.2 Alcances de Áncora.

Al desarrollar un software se pasa por varias etapas, la primera de ellas es especificar los requerimientos. La Ingeniería de Requerimientos de Software, cuyas actividades se dibujan en la Figura 2-1, se encarga de guiar al desarrollador en esta primera etapa produciendo un documento de Especificación de Requerimientos de Software (**ERS**), mismo que será entregado al cliente y al diseñador para continuar con el desarrollo del software. Después vendrán el diseño, implantación, prueba y mantenimiento, cada etapa cuenta con sus propias subetapas.

Áncora va a apoyar al Analista, también conocido como Ingeniero de Requerimientos, en la etapa de Análisis de Requerimientos de Software (ver Figura 2-1). Al empezar a definir los requerimientos de un nuevo software ya se debe haber avanzado en las negociaciones con los involucrados, esto es, ya se debe tener un compromiso de parte de la gerencia de la empresa, que podría ser verbal, de apoyar la realización de la Especificación de Requerimientos del Software. Es decir, ya se realizó el estudio de factibilidad y quedan tres etapas por cubrir:

1. Análisis de Requerimientos de Software (ARS). Este es proceso de derivación de requerimientos del sistema de software a través de la observación de sistemas existentes, discusión con usuarios y proveedores de información potenciales, análisis de tareas y así sucesivamente. Esto puede involucrar el desarrollo de uno o más modelos diferentes del sistema, que ayudan al analista a entender el sistema que será especificado. También pueden desarrollarse prototipos para ayudar a entender requerimientos.
2. Definición de Requerimientos de Software. Es la traducción de información recolectada durante la actividad del ARS a un documento que define el conjunto de requerimientos. Éste debería reflejar con precisión qué es lo que el cliente desea. Debe escribirse de forma que sea entendible por usuarios finales y cliente del sistema.
3. Especificación de Requerimientos de Software. En esta actividad se detalla el documento de ERS. Conjuntamente se puede realizar un diseño de alto nivel para ayudarse a descubrir errores en definición de requerimientos, mismos que deben ser corregidos.

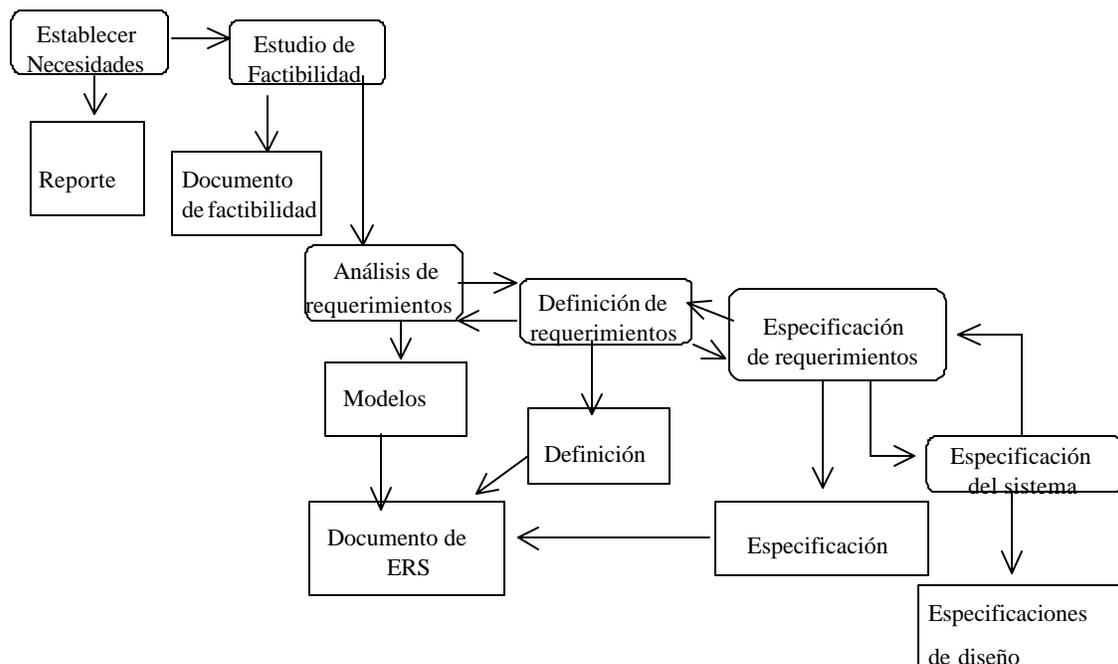


Figura 2-1. Actividades de la Ingeniería de Requerimientos de Software.

El papel del analista de software es descubrir y describir qué se quiere de un nuevo software y, al terminar, brindarle al diseñador del software elementos que le permitan empezar a definir la arquitectura del sistema.

Aunque resulta utópico pensar en un análisis de requerimientos completo (siempre habrá algo más que puede implantarse), éste debe tener por lo menos un avance sobre las metas y requerimientos para que el diseñador pueda empezar a trabajar. La forma de entregar los avances debe ser clara y sin ambigüedades y debe permitir hacer un cambio inmediato a elementos del diseño.

Muy probablemente el ARS deba negociarse varias veces, para agregar nuevos requerimientos o para modificar los ya existentes, por lo que una buena metodología de ARS debe permitir flexibilidad de modificación.

2.3 Presentación general del método y sus elementos.

En la Figura 2-2 se muestra un esquema de las actividades generales que se realizarán con Áncora y algunos artefactos que ésta produce.

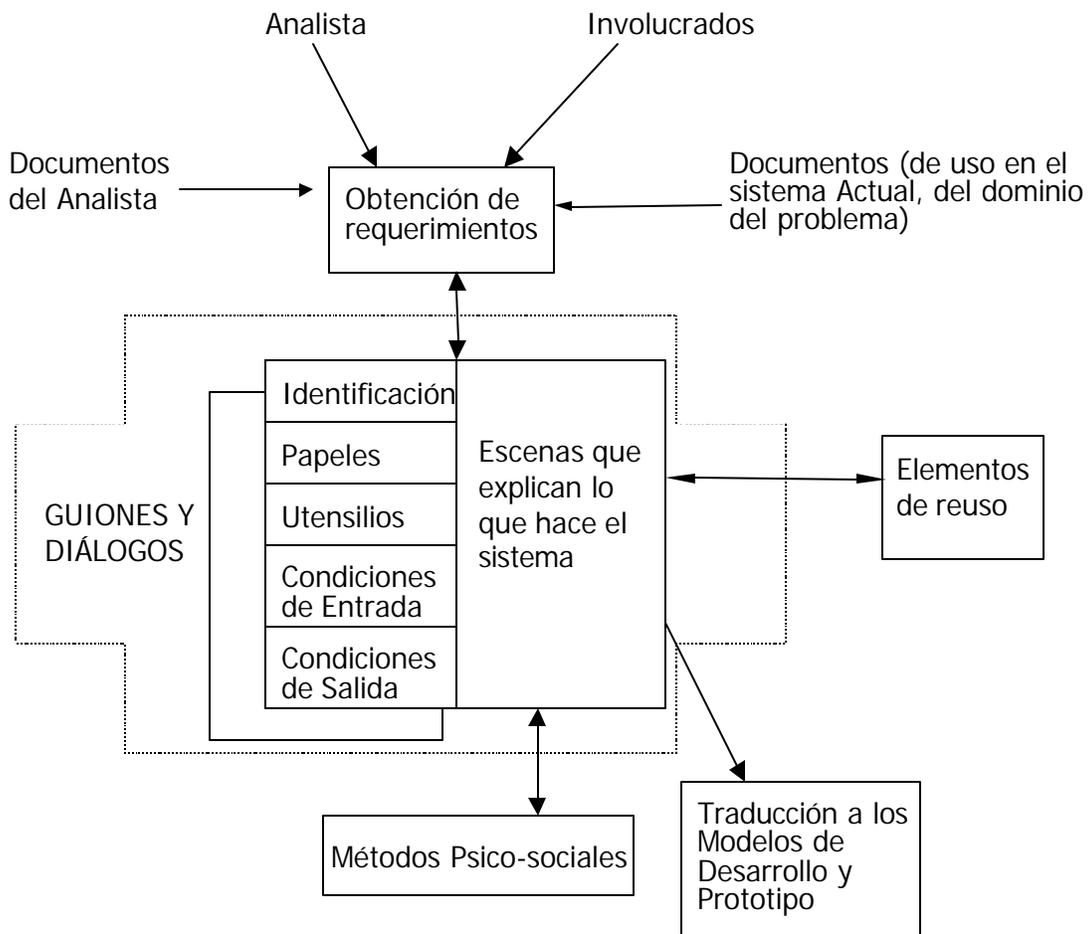


Figura 2-2 Esquema de elementos del Análisis de Requerimientos con Áncora.

En la Figura 2-2 aparece como elemento central un conjunto de Guiones y Diálogos que representan el Modelo de Requerimientos y del que se obtendrán la integración de vistas, la traducción a algunos posibles modelos desarrollo deseados (Orientación a Objetos, Flujo de datos, Estados) y que alimenta a y es alimentado por elementos de reuso de sistemas similares.

Antes de comenzar a trabajar en la construcción de modelos, es necesario saber con qué herramientas se cuenta. En el caso de Áncora se tienen:

- Guiones y Diálogos.
- Tablas y Glosarios.
- Modelo de Datos Entidad - Relación.
- Análisis de Puntos de Función.

Aunados con todas estas herramientas provenientes de ciencias computacionales se utilizan métodos psicológicos y de planeación que son:

- Redes Semánticas Naturales.
- Técnica de Grupo Nominal.
- Reunión de Reflexión y Diseño.

En las secciones restantes de éste capítulo se explicarán herramientas y métodos en forma aislada y en el capítulo siguiente se verá cómo y en qué momento aplicarlos.

2.4 Guiones y Diálogos.

Los Guiones y diálogos son dos herramientas muy importantes dentro de Áncora, la primera es indispensable y la segunda se utiliza para aclarar los “sobrentendidos” que pueda haber en lo que sería una funcionalidad requerida para el software.

2.4.1 Guiones.

El Guión es el elemento central en esta metodología (ver Figura 2-2), por lo mismo es importante que se entienda su forma de uso. Con los Guiones la forma de llevar a cabo la obtención de requerimientos será más sistemática y clara para el usuario, tratando de evitar ambigüedades de interpretación del lenguaje natural y exceso de tecnicismos. El esquema básico de un Guión se muestra en la Figura 2-3.

<p>Guión: {nombre del sistema}</p> <p>Pista: {nombre del subsistema}</p> <p>Papeles: {nombre de cada papel}</p> <p>Utensilios: {nombre de objetos, formatos, reportes}</p> <p>Condiciones de entrada: {lista de elementos necesarios para comenzar a trabajar con el sistema}</p> <p>Condiciones de salida: {lista de elementos necesarios que se obtendrán por trabajar con el sistema}</p>	<p>Escena 1: {nombre de función genérica1}</p> <p>quinteta 1.1 {resultado cierto de la condición de salto}</p> <p>quinteta 1.2</p> <p>...</p> <p>quinteta 1.n1</p> <p>Escena 2: {nombre de función genérica2}</p> <p>quinteta 2.1</p> <p>quinteta 2.2</p> <p>...</p> <p>quinteta 2.n2</p> <p>·</p> <p>·</p> <p>·</p> <p>Escena m: {nombre de función genéricam}</p> <p>quinteta m.1</p> <p>quinteta m.2</p> <p>...</p> <p>quinteta m.n2</p>
--	--

Figura 2-3 Plantilla de Guión.

Por el lado del analista, los Guiones brindan elementos que permiten una fácil comunicación tanto con el cliente o usuario como con el diseñador y por el lado del involucrado (cliente, usuario) en el software los Guiones representan una forma visual rápida y fácil de entender. Cada elemento del Guión se explica a continuación:

1. **Guión.** Se refiere sólo al nombre del sistema de software. Un Guión no debe rebasar una hoja, en caso de necesitarse más, se divide en pistas y se elabora una hoja por cada pista, todas con el nombre del Guión incluido.
2. **Pista.** El nombre del subsistema. En el caso de haber más de un subsistema, debe especificarse en un Guión separado, que también llevará el nombre del sistema.
3. **Papeles.** Se hace una lista de los papeles dentro del sistema, un papel puede ser una persona, un sistema o un equipo especial que tenga una interfaz con el sistema. Cada persona, sistema o equipo que están involucrados en el sistema puede tener uno o varios papeles.
4. **Utensilios.** Se refiere a los elementos físicos que se utilizan en el software, pueden ser: dispositivos de E/S, formas del sistema u archivos. La computadora donde corre el software puede o no considerarse un útil, en caso de ser sistemas cliente – servidor, en red o distribuidos, las computadoras se consideran como papeles.

5. **Condiciones de entrada.** En esta parte se anotan aquella lista de eventos fuera de la frontera de software que deben cumplirse para empezar a trabajar con éste.
6. **Condiciones de Salida.** Lista de eventos que provocan la terminación normal del sistema, normalmente son resultados del sistema.
7. **Escena.** Nombre de una función genérica del sistema (puede pensarse como un módulo). La escena estará detallada por las quintetas. En caso de ser muy larga se recomienda que se considere cambiarla a pista (subsistema) y se le haga su Guión separado. En general, una escena corresponde a un requerimiento.
 - 7.1. **Quinteta.** Se trata de una funcionalidad o actividad a realizar con el software. Su nombre se debe a que estará formado por un máximo de cinco elementos, aunque pueden usarse sólo los tres primeros de la siguiente lista:
 - 7.1.1. **Papel.** Se escribe la inicial o nombre con que se identificó a uno de los papeles del Guión.
 - 7.1.2. **Acción.** Debe ser un verbo que especifique una actividad.
 - 7.1.3. **Resultado.** Se refiere al elemento que recibe la acción del verbo y puede ser: un archivo, reporte o consulta. Este resultado debe estar en la lista de útiles.
 - 7.1.4. **Útil.** Este elemento es opcional en la quinteta, pero si se escribe debe existir en la lista de utilería del Guión y sirve como apoyo para la realización de la acción.
 - 7.1.5. **Periodicidad.** Se refiere al tiempo transcurrido entre ocurrencias de la actividad descrita por la quinteta.
8. **Condiciones.** En un sistema no siempre suceden las actividades de manera secuencial, por ello en un Guión es posible indicar cambios de secuencia utilizando condiciones, las cuales se representan con un enunciado entre interrogaciones en seguida de la quinteta que puede causar el salto y al lado de una línea punteada que indica el salto si la condición resulta cierta.
9. **Elementos gráficos.** Sólo son dos los elementos gráficos utilizados en un Guión: la caja que lo rodea y separa elementos y los segmentos de línea que marcan la secuencia de las acciones representadas por las quintetas. El segmento de línea continua indica secuencia (puede omitirse si no hay saltos cercanos) y el segmento de línea punteada.

Para ejemplificar como se forma un Guión, se escogió un ejemplo sencillo, pero en el Anexo A se pueden ver ejemplos de mayor complejidad.

Ejemplo. Agenda de Compromisos. Se tiene la necesidad de hacer un sistema que permita a un agente de ventas llevar una agenda de compromisos de las citas con los diferentes compradores y proveedores. Al estar el agente de ventas ausente de su oficina, sus secretarias o auxiliares pueden tomar el recado y hacer las nuevas citas basándose en la agenda de compromisos.

En la Figura 2-4 se muestra lo que sería la propuesta computacional del sistema de Agenda de Compromisos.

Guión: Agenda Pista: Compromisos	Escena 1: Generar Agenda. P anota en K compromisos permanentes P genera A cada principio de año. P imprime A usando II
Papeles: P = persona dueña de Agenda S = Secretaria	Escena 2: Anotar compromisos. P ó S pide visualizar fecha en A P ó S pide reacomodar compromiso anterior P ó S anota compromiso en A
Utensilios: A = Agenda At = Agenda temporal C = Computadora H = Hoja cualquiera K = Catálogo de compromisos permanentes. II = impresora	Escena 3: Anotar compromisos sin Agenda. P ó S recibe pedido de compromiso P ó S anota compromiso en H o en At
Condiciones de entrada: P necesita controlar sus compromisos S puede ayudar a P a llevar el control	Escena 4: Actualización batch de la Agenda. P ó S encienden equipo P ó S vizualizan aviso de actualización automática de A P ó S da orden de actualización de A con At y/o H
Condiciones de salida: P lleva su control S ayudó a P	Escena 5: Buscar compromisos pendientes. S revisa compromisos en A S puede avisar a p de compromiso P pierde compromiso P revisa compromisos en A P cumple compromiso ¿P desea imprimir? ¿intercambiar fechas? ¿P no se enteró?

Figura 2-4 Guión para la propuesta Computacional de un Control de Agenda de Compromisos de una persona ayudado por uno o más auxiliares.

2.4.2 Diálogos.

Los diálogos son una herramienta que Áncora brinda para detallar cada quinteta que sea ambigua o donde aparezcan actividades manuales sobreentendidas.

Para el caso de las quintetas ambiguas un diálogo estará formado renglones donde el que se expresa es el papel de la quinteta con verbos que aclaran los sobreentendidos en la quinteta.

Ejemplo de quinteta ambigua. En la Figura 2-4 del guión de propuesta computacional de un sistema de software para el control de Agenda de Compromisos esta la quinteta

S puede avisar a P de compromiso

En este caso resulta ambiguo el hecho de que la secretaria o auxiliar S “pueda avisar” a la persona dueña de la agenda P de tener un compromiso, por eso es necesario aclararlo con un diálogo:

- S busca a P en su escritorio o por teléfono o envía un mensaje vía Internet .
- Si S localizó a P,
- P recibe aviso de compromiso.
- Si no se localizó a P,
- P no recibe aviso de compromiso y sólo quedará anotado en la Agenda de Compromisos.

Nótese que en esta situación el problema de perder la cita no se resolvió, pero todavía puede sugerírsele al usuario que se le dé autoridad a la secretaria para posponer la cita. De esta manera la elaboración de diálogos está dando pautas para resolver el problema.

En el caso de las quintetas que incluyen acciones manuales, los diálogos estarán formados por renglones consecutivos en donde aparecerán los elementos de la quinteta unidos por verbos que representan las acciones previas y posteriores a la función automática representada por el verbo de la quinteta.

Ejemplo de quinteta que involucra acciones manuales. En la Figura 2-4 del guión de propuesta computacional de un sistema de software para el control de agenda de compromisos esta la quinteta

P imprime A usando II

claramente esto involucra acciones previas y posteriores, mismas que se detallarán con un diálogo como sigue:

- P se asegura que haya papel suficiente para la impresión.
- P se asegura que la impresora esté prendida.
- II debe avisar de encendida y lista.
- P oprime el botón “Imprimir Agenda” que aparece en la pantalla.
- *P imprime la A usando II*
- P recoge hojas conteniendo A
- P apaga la II

2.5 Tablas y Glosarios.

La idea principal del uso de Tablas y Glosarios es disminuir la ambigüedad al usar lenguaje natural ya que, en el proceso de especificación de los requerimientos de un nuevo software, se utilizan términos del área de aplicación y del área de computación. Posteriormente, servirán de apoyo para la definición del Modelo de Datos y diseño.

2.5.1 Tablas.

Las Tablas son una estructura formada por renglones y columnas misma que permite una delimitación estricta de los contenidos de los elementos que la forman. Se usarán Tablas para varios fines entre ellos están:

1. Definir cada formato que se utilice en el sistema actual. Se entiende por formato todos los siguientes: formas impresas, oficios, memorándums, faxes y hasta mensajes vía Internet.
2. Especificar cada dato: nombre, tamaño, tipo.
3. Realizar plan de pruebas de funcionalidad establecida.
4. Cálculo de costos.

Dependiendo del uso de la Tabla su formato varía y éste se irá especificando a lo largo del presente manual.

2.5.2 Glosarios.

Un Glosario es un caso especial de Tabla y se utiliza con el fin de lograr un lenguaje común para todos los involucrados en la definición del nuevo software. Para lograrlo es necesario definir términos técnicos del área de aplicación del nuevo software y del área de Ingeniería de Software (ésta última resulta necesaria para que el interesado, que es de un área diferente, entienda al analista). El contenido mínimo del Glosario es por cada término una entrada con los siguientes datos:

1. Palabra.
2. Significado de la palabra en el dominio de la aplicación o de Ingeniería de Software.
3. Formas donde aparece. Se refiere a los formatos que se utilizan en la aplicación, incluyendo pantallas.
4. Lista de posibles valores.
5. Referencia (manual, libro, persona) de donde obtuvo el significado.

2.6 Modelo Entidad-Relación.

Propuesto por P. Chen en 1976, modificado y ampliado por varios autores más, el modelo Entidad-Relación (E-R) está considerado como una forma de trabajar el modelo semántico de Bases de Datos y es muy utilizado para el modelado de datos. De hecho, una vez propuesto el modelo de datos en E-R, existen métodos para convertirlo a cualquier modelo lógico de Bases de Datos existente (relacional, jerárquico, reticular). A pesar de la inclusión del modelado de bases de datos orientadas a objetos, el modelo semántico E-R no pierde vigencia y sigue utilizándose bajo este enfoque.

Las partes que conforman el modelo E-R se describen a continuación:

- i. Entidad.- Se expresa por medio de un rectángulo que en su interior lleva el nombre de un sustantivo, el cual puede representar persona, lugar, cosa o evento de interés en el sistema. Existen también las *Entidades Débiles*, las cuales se dibujan con un rectángulo de línea doble y no pueden existir por sí solas, esto es, dependen de una Entidad Fuerte. Ejemplo: si se tiene la Entidad vehículo, la entidad automóvil sería una entidad débil.
- ii. Atributos.- Se utilizan para detallar el contenido de las entidades. Se representan por óvalos unidos al rectángulo de la entidad a la que pertenecen por medio de una línea recta. Existen dos tipos de atributos:

los descriptivos y los identificadores. Éstos últimos solo los tienen las Entidades Fuertes y sirven para identificar un ejemplar de la entidad de otro. Los atributos identificadores van subrayados y son conocidos como llaves primarias o simplemente llaves.

- iii. Relación.- Se representa por medio de un diamante que lleva dentro el nombre de una asociación entre dos o más entidades. Una relación puede o no contener atributos.

Para establecer la relación semántica entre una entidad y otra se debe utilizar una relación (diamante) y la ocurrencia de esta conectividad (uno a uno, uno a muchos y muchos a muchos), se explicita poniendo números encima de las líneas de unión. Cabe aclarar que esta notación ha variado con el tiempo y una de las más utilizadas es la de poner punta de flecha a la línea que va del diamante a la entidad en lugar de ir un número uno sobre la raya.

Para aclarar la notación se utiliza el ejemplo que se ve en la Figura 2-5 y la cual representa las relaciones entre alumnos, las materias que llevaron y las calificaciones que obtuvieron. La letra n puesta sobre la raya representa el tipo de relación que guarda la entidad alumno con la entidad materia mediante la relación "llevó". Lo que quiere decir que varios alumnos pueden llevar una o más materias y que a su vez la materia puede ser llevada por muchos alumnos.

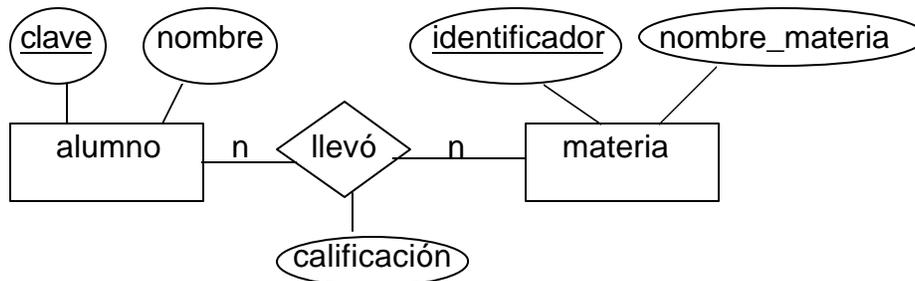


Figura 2-5. Ejemplo del modelo Entidad-Relación.

Dentro de las extensiones que tiene el modelo Entidad-Relación están los conceptos de Jerarquía de Generalización (generalización) y Jerarquía de subconjunto (especialización), las cuales se describen a continuación:

- i. Jerarquía de generalización.- Una entidad E es la generalización de las entidades E_1, E_2, \dots, E_n si cada ocurrencia de E es también una ocurrencia de una y sólo una de las entidades E_1, E_2, \dots, E_n . Ejemplo: La entidad EMPLEADO es generalización de SECRETARIA, TÉCNICO e INGENIERO.
- ii. Jerarquía de subconjunto.- Una entidad E_1 es un subconjunto de otra entidad E_2 si para toda ocurrencia de E_1 hay también una ocurrencia de E_2 . Ejemplo: una entidad CUENTA puede incluir CUENTA_AHORRO, CUENTA_INVER como especializaciones que constituyen subconjuntos de

la primera. Toda cuenta de ahorros o cuenta de inversión es también una cuenta.

2.6.1 Modelo de Objetos Semánticos

El Modelo de Objetos Semánticos (MOS) es utilizado para interpretar los requerimientos del usuario y construir el modelo de datos, mismo que se transformará en una Base de Datos (BD) durante las siguientes fases de desarrollo software.

El utilizar MOS ayudará a lograr un diseño de BD Orientada a Objetos (BDOO), mientras que si se utiliza el modelo Entidad Relación (E-R), se obtendrán diseños de Bases de Datos basados en entidades y relaciones.

2.6.1.1 Definición y Características de los Objetos Semánticos

Un *objeto semántico* representa una identidad en el medio ambiente del mundo del usuario, el objeto semántico debe tener suficientes propiedades que lo identifiquen claramente.

Los objetos semánticos se agrupan en clases. Una *clase* de objeto tiene un nombre que la distingue de otras clases y corresponde a la identidad que está representando. Los nombres de las clases se anotan con letras mayúsculas. Una instancia de una clase es un objeto semántico particular.

Los objetos semánticos tienen una *colección de propiedades* que sirven para identificarlos claramente como *identidades diferentes* del mundo real. Además, puede haber identidades que sirvan al usuario para obtener resultados y que mantengan juntas a pesar de no representar una identidad física.

Ejemplos.

1. De instancias: si se tiene la clase ESTUDIANTE, una *instancia* será ESTUDIANTE Juan Rodríguez.
2. Objeto que representa una identidad física. ESTUDIANTE.
3. Objeto necesario para el usuario, pero no es una identidad física. CARDEX ELECTRÓNICO.

Los objetos semánticos existen de manera independiente; es decir, no puede haber un objeto cuya existencia dependa de otro objeto. En el modelo E-R si existe esta situación al declarar la existencia de entidades débiles.

Las propiedades de los objetos¹ pueden ser variadas, van desde propiedades simples, como número de matrícula de un ESTUDIANTE; propiedades compuestas, como la dirección del estudiante que incluye calle, número, colonia, localidad, municipio y estado; hasta otros objetos, como podría ser el objeto MATERIA.

¹ Se utiliza el término objeto en lugar de objeto semántico para simplificar el texto.

Las propiedades que no son valores simples son propiedades multivaluadas, se abreviará como MV. Así, una propiedad MV sería la dirección del estudiante y la propiedad MATERIA. Un ejemplo del objeto semántico ESTUDIANTE con sus diferentes propiedades se puede ver Figura 2-6, donde se observan sólo dos propiedades MV: la propiedad dirección, que es un *no objeto*, y la propiedad *objeto* MATERIA.

En el caso de una propiedad objeto, ésta es MV si existen varias instancias (múltiples valores) de esta propiedad en el objeto que se está definiendo. Ejemplo, el objeto ESTUDIANTE tiene la propiedad objeto MATERIA multivaluada porque el ESTUDIANTE puede llevar varias materias, como se ve en la 2-6; por el contrario, el caso de la propiedad objeto ESCUELA no es MV ya que el ESTUDIANTE sólo puede inscribirse en una sola ESCUELA.

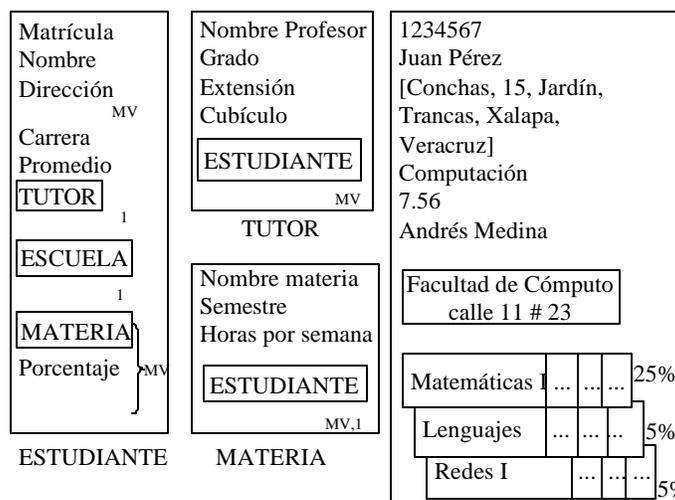


Figura 2-6. Objetos semánticos ESTUDIANTE, TUTOR Y MATERIA. Una instancia del objeto ESTUDIANTE.

El dominio de las propiedades en el modelo de objetos semánticos se define como sigue:

- i. Si propiedad es no objeto, el dominio de la propiedad debe contener dos elementos: sus propiedades físicas y sus propiedades semánticas. Ejemplo: la propiedad promedio *debe ser un número real formado por dos dígitos enteros y dos dígitos decimales y representa el promedio de calificaciones que el estudiante ha obtenido*. La primera parte del enunciado habla de las propiedades físicas de la propiedad y la segunda parte su característica semántica.
- ii. Si la propiedad es un objeto, el dominio de la propiedad está definido por las propiedades que lo componen. Ejemplo: el dominio de la propiedad objeto MATERIA está formado por las propiedades nombre de la materia, semestre, créditos y el número de horas a la semana que se debe impartir.

Los objetos semánticos pueden presentarse con más o menos propiedades para mostrar diferentes vistas. Es decir; en algunas vistas importan todas las propiedades, en otras sólo algunas.

2.6.1.1.1 Tipos de Objetos Semánticos.

Existen varios tipos de objetos semánticos se listan a continuación acompañados de una breve explicación:

- i. Objetos simples. Los que tienen una sola propiedad y es un valor simple.
- ii. Objetos compuestos. Aquellos con varias propiedades, pero todas son simples.
- iii. Objetos combinados. Los que tienen propiedades con valores simples y propiedades objetos.
- iv. Objeto de asociación. Es un objeto que relaciona dos o más objetos y que contiene propiedades que son peculiares a esa relación.
- v. Objetos híbridos. Son aquellos que contienen propiedades de objetos simples, compuesto y/o combinados. En estos objetos pueden existir propiedades asociadas a alguna propiedad objeto y no al objeto híbrido que se está definiendo. Ejemplo: si se tiene que cada ESTUDIANTE paga un porcentaje diferente para cada materia, se tiene que agregar esta propiedad al objeto ESTUDIANTE pero la propiedad estará asociada a la MATERIA y no al ESTUDIANTE. Para denotar que la propiedad objeto MATERIA y la propiedad precio van juntas dentro del objeto ESTUDIANTE se utiliza la forma de una llave como se ve en la Figura 2-6.
- vi. Objetos de generalización y subtipos. Se utilizan para la jerarquización de tipos de objetos. Ejemplo: suponga el objeto EMPLEADO de una universidad; claramente todos los empleados tienen atributos comunes como son Nombre y Número de empleado, pero dependiendo de sus actividades pueden tener otra clasificación como EMPLEADO_INTENDENTE, EMPLEADO_DOCENTE ó EMPLEADO_ADMINISTRATIVO, que vienen a ser *subtipos* del objeto *generalizado* EMPLEADO.

2.6.1.2 Modelando con Objetos Semánticos.

Existen dos formas de modelar la BDOO utilizando objetos semánticos: top-down y bottom-up², se da una breve descripción de cómo funcionan ambos enfoques.

En el método top-down el analista se concentra en la meta general de la aplicación y pregunta a los usuarios cuáles podrían ser los objetos, trabajando de manera conjunta.

En el método bottom-up se define la interfaz de la aplicación (pantallas y reportes), entonces, ya que se sabe qué se quiere y cuáles propiedades hay que tomar en cuenta, se definen los objetos semánticos.

² No se traducen los términos top-down y bottom-up por resultar muy utilizados en computación.

Ambos enfoques tienen problemas, en el top-down se requiere de un analista con mucha experiencia en diseño de sistemas de BDOO y aún así puede haber propiedades de los objetos que no se tomen en cuenta. En el caso del enfoque bottom-up el tiempo que se invierte en la definición de la interfaz de la aplicación puede ser extenso.

Lo que se recomienda es una combinación de ambos métodos como sigue:

- i. Se empieza listando los objetos semánticos potenciales considerando las metas de la aplicación.
- ii. Se examinan los reportes, formas y vistas disponibles para determinar la posible estructura del conjunto inicial de objetos y sus propiedades.
- iii. Se plantean las asociaciones entre los objetos, mismas que se obtendrán al analizar las pantallas y reportes de lo que se quiere del software.
- iv. Usando los diagramas de objetos revisados se sugieren al usuario nuevos reportes, formas y vistas con base en las metas de la aplicación.

Los pasos anteriores se repiten tantas veces como sea necesario hasta que los usuarios estén de acuerdo con la información que la aplicación producirá.

Ya que en el MOS no se incluyen objetos relación ni objetos débiles, la forma de establecer las relaciones es como sigue:

- i. Si la relación de un objeto A con un objeto B es que A debe tener forzosamente una instancia del objeto B, entonces se anota un número 1 debajo de la propiedad objeto B en el diagrama del objeto A. Ejemplo: en la Figura 2-6 el objeto ESTUDIANTE debe tener una instancia del objeto ESCUELA.
- ii. Si la relación es un objeto A puede tener varias instancias del objeto B, entonces se anotan las siglas MV debajo de la propiedad objeto del objeto A. Ejemplo: en la Figura 2-6 el objeto TUTOR puede apoyar a varios ESTUDIANTES, sin embargo un ESTUDIANTE sólo puede tener un TUTOR.
- iii. Por último, la relación muchos a muchos se ejemplifica con los objetos ESTUDIANTE Y MATERIA de la Figura 2-6.

2.7 Cálculo de Puntos de Función.

Puntos de Función es un modelo propuesto inicialmente por Albretch y está siendo utilizado en diferentes instituciones con varios usos, inclusive existe un grupo (The International Function Point Users Group) dedicado a mantener actualizada la forma de contabilizar utilizando Puntos de Función (PF). Además de que PF resulta fácil y rápido de aplicar para el cálculo de costos de un nuevo software, ayuda a delimitar qué se quiere del sistema y las restricciones bajo las cuales operará.

Por las anteriores razones Áncora utiliza PF para su primera aproximación de costos. En esta sección se explicará cómo contar los PF y en el capítulo respectivo al trabajo con Áncora se explicará la forma de reunirlos con los demás elementos del ARS y obtener un costo aproximado en tiempo, dinero o complejidad.

El modelo PF emplea cinco indicadores los cuales sirven para identificar a cada uno de los elementos del software, a los dos primeros se les conoce como funciones de datos y los otros tres son funciones de transacciones. Estos indicadores son:

1. Archivos Lógicos Internos (ALI). Es un grupo identificable de datos relacionados lógicamente o de información de control que pertenece al usuario, es mantenido dentro de las fronteras del sistema. La principal intención de un ALI es mantener los datos actualizados mediante uno o más procesos elementales de la aplicación a contabilizar.
2. Archivo de Interfaz Externa (AIE). Es un grupo identificable de datos relacionados lógicamente o de información de control que pertenece al usuario, es referido por la aplicación, pero mantenido dentro de las fronteras de otra aplicación. La principal intención de un AIE es mantener los datos referidos mediante uno o más procesos elementales de la aplicación a contabilizar. Un AIE puede ser un ALI de otra aplicación.
3. Entradas Externas (EE). Es un proceso elemental que procesa datos o información de control que viene de fuera de la frontera de la aplicación. El propósito principal de una EE es mantener uno o más ALI y/o alterar el comportamiento del sistema.
4. Salidas Externas (SE). Es un proceso elemental que envía datos o información de control fuera de la frontera de la aplicación. La intención principal de una SE es presentar información al usuario mediante el procesamiento lógico y recuperación de datos o información de control. El procesamiento lógico debe contener al menos una fórmula matemática, cálculo o crear datos derivados. Una EE puede también mantener una o más ALI y/o alterar el comportamiento del sistema.
5. Consultas Externas (CE). Es un proceso elemental que envía datos o información de control fuera de la frontera de la aplicación. La intención principal de una CE es presentar información al usuario la recuperación de datos o información de control. El procesamiento lógico no contiene

fórmulas matemáticas, ni cálculos y no crea datos derivados. No se actualizan ni un ALI, ni se altera el comportamiento del sistema.

Para realizar la contabilidad de puntos de función sin ajustar (T), se procede de la siguiente manera:

- i. A cada elemento en un software se le etiqueta como alguno de los cinco identificadores,
- ii. se le asigna su nivel de dificultad como se explica cómo en las siguientes secciones,
- iii. se multiplican por los pesos que se muestran en la Tabla 2-1,
- iv. se suman los valores de cada renglón colocando la SUMA en la última columna;
- v. luego se suman los valores hacia abajo colocando el resultado del valor T.

Tabla 2-1. Contabilidad de puntos de función sin ajustar.

Indicador	simple	mediano	complejo	SUMA
Archivos Lógicos Internos	7	10	15	
Archivos de Interfaz Externa	5	7	10	
Entradas Externas	3	4	6	
Salidas Externas	4	5	7	
Consultas Externas	3	4	6	
			T =	

Para determinar cuál será el peso que le corresponde a cada tipo de indicador se debe revisar las interacciones y características de cada elemento del sistema. En seguida se verá la forma de asignación de nivel de dificultad de cada tipo de indicador.

2.7.1 Asignación de pesos para ALI y AIE.

Se explica a continuación la forma de calcular el nivel de dificultad para los elementos del nuevo software que sean de algún tipo de indicador (ALI,AIE).

Lo primero que se debe hacer es identificar el número de Tipo de Dato Elemental (TDE) y Tipo de Registro Elemental (TRE) que contiene cada ALI o AIE, para hacerlo se aplican las siguientes reglas:

- 1) Para saber cuántos campos (TDE) tomar en cuenta para cada ALI o AIE.
 - a) Cuente un TDE para cada campo mantenido o recuperado de un ALI o AIE a través de un proceso elemental.
 - b) Cuando dos aplicaciones mantienen o refieren al mismo ALI o AIE y cada diferente proceso utiliza los TDE agrupados de diferente manera, deben contarse los TDE como se refieren. Ejemplo: una aplicación utiliza día, mes, año y otra lo usa como fecha; entonces hay dos TDE diferentes.
 - c) Contar un TDE para cada pieza de dato requerido por el usuario para establecer una relación con otros ALI o AIE. Ejemplo una llave externa.
- 2) Para el conteo de los TRE se aplican los siguientes criterios.
 - a) Cuente un TRE por cada subgrupo del ALI o AIE.

b) Si no hay subgrupos cuenta al ALI o AIE como un TRE.

Una vez que se tienen contados los TDE y TRE se aplica la Tabla 2-2 para determinar el nivel de complejidad del ALI o AIE.

Tabla 2-2. Asignación de nivel de dificultad para los Archivos Lógicos Internos o Archivos de Interfaz Externos.

	1 -19 TDE	20 - 50 TDE	51 ó + TDE
0 – 1 TRE	simple	simple	mediano
2 – 5 TRE	simple	mediano	complejo
6 ó + TRE	mediano	complejo	complejo

2.7.2 Asignación de pesos para EE, SE Y CE.

En esta sección se explica cómo asignar nivel de dificultad a cada Salida Externa o Consulta Externa que se ha definido en el nuevo software.

Sean TDE =Tipo de Dato Elemental y RA =Referencias a Archivos, entonces los pesos se calculan con la Tabla 2-3 ó 2-4 según se necesite.

Tabla 2-3 Asignación de nivel de dificultad para las Salidas Externas y Consultas Externas.

	1-5 TDE	6-19 TDE	20 ó + TDE
0 – 1 RA	simple	simple	mediano
2 – 3 RA	simple	mediano	complejo
4 ó + RA	mediano	complejo	complejo

Tabla 2-4 Asignación de nivel de dificultad para las Entradas Externas.

	1 - 4 TDE	5 -15 TDE	16 ó + TDE
0 – 1 RA	simple	simple	mediano
2 RA	Simple	mediano	complejo
3 ó + RA	Mediano	complejo	complejo

2.7.3 Ajuste de los Puntos de Función.

Los requerimientos de implantación de un nuevo software suelen influir grandemente en el costo de éste. Así, el modelo de PF lo prevé al usar catorce estimadores, que se listan abajo y definen en las subsecciones siguientes.

1. Comunicación de datos.
2. Procesamiento Distribuido de Datos.
3. Rendimiento.
4. Configuración Altamente Usada.
5. Promedio de Transacciones.
6. Entrada de Datos en Línea.
7. Eficiencia para el Usuario Final.

8. Actualización en Línea.
9. Procesamiento Complejo.
10. Reusabilidad.
11. Facilidad de Instalación.
12. Facilidad de Operación.
13. Varios Sitios.
14. Facilidad de Cambios.

A cada estimador se le asigna un grado de influencia, la cual puede ser: 0=sin influencia, 1=accidental, 2=moderado, 3=medio, 4=significativo, 5=esencial. Ya calificados los estimadores, se suman en la variable M.

2.7.3.1 Comunicación de Datos.

Describe el grado con el cual la aplicación se comunica directamente con el procesador. Para asignar el grado de influencia de este estimador se aplican los criterios listados en seguida:

- 0=La aplicación es puramente batch³ o para una PC aislada.
- 1=La aplicación es batch pero tiene entrada de datos o impresión remota.
- 2=La aplicación es batch pero tiene entrada de datos e impresión remota.
- 3=La aplicación incluye colección de datos en línea, el procesamiento batch se realiza en un front-end⁴ de teleproceso (TP) o sistema de consulta.
- 4=La aplicación es más que un front-end, pero soporta sólo un tipo de protocolo de comunicación de TP.
- 5=La aplicación es más que un front-end, y soporta más de un tipo de protocolo de comunicación de TP.

2.7.3.2 Procesamiento Distribuido de Datos.

Mide el grado con el que la aplicación transfiere datos entre componentes de la aplicación. La forma de asignar su grado de influencia como se muestra :

- 0=La aplicación no apoya en la transferencia de datos o procesado de funciones entre las componentes del sistema.
- 1=La aplicación prepara datos para que el usuario procese en otra componente del sistema como en una hoja de cálculo o un SMBD en PC.
- 2=Se preparan datos para transferir, entonces se transfieren y procesan sobre otra componente del sistema (no por el usuario final).
- 3=Los procesos o transferencia de datos se realizan en línea y en una sola dirección.
- 4=Los procesos o transferencia de datos se realizan en línea y en ambas direcciones.
- 5=Las funciones de procesamiento son realizadas dinámicamente sobre la mayor parte del sistema.

³ Proceso que utiliza datos de una bitácora de transacciones para la actualización de archivos.

⁴ Equipo de cómputo que sirve de representante para comunicarse con otros grupos de computadoras.

2.7.3.3 Rendimiento.

El rendimiento será crítico y tendrá influencia sobre cómo diseñar, desarrollar o implementar. La forma de asignar su grado de influencia se muestra en seguida:

0=No hay requerimientos de rendimiento especiales impuestos por el usuario.

1=Se establecieron y revisaron los requerimientos de rendimiento y diseño, pero no se requirieron acciones especiales.

2=El tiempo de respuesta o concurrencia es crítica durante las horas pico.

3=El tiempo de respuesta o concurrencia es crítica durante las horas de negocios. No hay un diseño especial para la CPU requerida. Los requerimientos de límites de procesamiento con los sistemas de interfaces están restringidos.

4=Además, los requerimientos de rendimiento puestos por el usuario son lo suficientemente estrictos para requerir el análisis de rendimiento de las tareas en la fase de diseño.

5=Además, las herramientas de análisis de rendimiento serán utilizadas en las fases de diseño, desarrollo y/o implantación para reconocer los requerimientos de rendimiento puestos por el usuario.

2.7.3.4 Configuración Altamente Usada.

El software será implementado en un entorno existente y fuertemente utilizado. La forma de asignar su grado de influencia se asigna como sigue:

0=No se han incluido restricciones operacionales implícita o explícitamente.

1=Existen restricciones operacionales, pero son menos restrictivas que en una aplicación típica. No hay que hacer esfuerzos especiales para cumplir con las restricciones.

2=Se incluyen algunas restricciones de seguridad o tiempo.

3=Se incluyen requerimientos específicos de proceso para una pieza específica de la aplicación.

4=Requerimientos de operación impuestas requieren restricciones especiales sobre la aplicación en el procesador central o un procesador dedicado.

5=Además, hay una restricción especial sobre la aplicación en los componentes distribuidos del sistema.

2.7.3.5 Promedio de Transacciones.

Un alto promedio de transacciones influenciará al diseño, desarrollo, implantación y soporte. En seguida se ve su grado de influencia:

0=No se anticipa un periodo pico de transacciones.

1=Se anticipa un periodo pico de transacciones (Ejemplo. Mes, quincena).

2=Se anticipa un periodo pico de transacciones semanal .

3=Se anticipa un periodo pico de transacciones diario.

4=Se impone en los requerimientos un periodo promedio de transacciones alto o los acuerdos en el nivel de servicios son suficientemente grandes para requerir tareas de análisis de rendimiento en la fase de diseño.

5=Además, las herramientas de análisis de rendimiento serán utilizadas en las fases de diseño, desarrollo y/o implantación para reconocer los requerimientos de rendimiento puestos por el usuario.

2.7.3.6 Entrada de Datos en Línea.

El software requerirá entradas interactivas. La forma de asignar su grado de influencia es la de la lista:

- 0=Todas las transacciones se procesan en modo batch.
- 1=Del 1 al 7 % de las transacciones se introducen de modo interactivo.
- 2=Del 8 al 15 % de las transacciones se introducen de modo interactivo.
- 3=Del 16 al 23 % de las transacciones se introducen de modo interactivo.
- 4=Del 24 al 30 % de las transacciones se introducen de modo interactivo.
- 5=Más del 30 % de las transacciones se introducen de modo interactivo.

2.7.3.7 Eficiencia para el Usuario Final.

Las funciones en línea proveídas tendrán que enfatizar un diseño para la eficiencia del usuario final. El diseño incluye la lista de abajo:

- Ayudas para la navegación del sistema.
- Menús.
- Ayuda en línea y documentos.
- Movimiento automático del cursor.
- Scrolling (manejo de la pantalla como documento en rollo de papel).
- Impresión remota vía transacciones en línea.
- Asignación de teclas de función.
- Activación de trabajos batch desde una transacción en línea.
- Selección del cursor desde la pantalla de datos.
- Uso intenso de video inverso, colores para resaltar, luminosidad, y otros indicadores.
- Copia en documento para el usuario de las transacciones en línea.
- Interfaz de ratón.
- Ventanas pop-up.
- Tan pocas ventanas como sea posible para lograr una función de negocio.
- Soporte bilingüe.
- Soporte multilingüe(más de dos y hasta seis lenguajes)

La forma de asignar su grado de influencia se ve en la siguiente lista:

- 0=Ninguna de las características de arriba.
- 1=De una a tres características de arriba.
- 2=De cuatro a cinco características de arriba.
- 3=Seis o más de las características de arriba, pero no hay requerimientos de eficiencia relacionados.
- 4=Seis o más de las características de arriba, existen requerimientos de eficiencia del usuario y requieren diseño de tareas para incluir factores humanos (Ejemplos. Minimizar tecleo, maximizar uso de omisiones, uso de plantillas).

5=Seis o más de las características de arriba, existen requerimientos de eficiencia del usuario - final y requieren de herramientas y procesos para demostrar que los objetivos han sido alcanzados..

2.7.3.8 Actualización en Línea.

Se necesitará la actualización de archivos maestros en forma interactiva. La forma de asignar su grado de influencia es como en la lista que se da enseguida:

0=No hay

1=Se incluye la actualización en línea de uno a tres archivos de control. El volumen de actualización es bajo y la recuperación es fácil.

2=Se incluye la actualización en línea de cuatro o más archivos de control. El volumen de actualización es bajo y la recuperación es fácil.

3=Se incluye la actualización en línea de la mayoría de los archivos lógicos internos.

4=Además, la protección contra pérdida de datos es esencial y tendrá que ser especialmente diseñada y programada en el sistema.

5=Además, se consideran dentro de los procesos de recuperación volúmenes altos. Se incluyen procesos de recuperación altamente automatizados con intervención mínima del operador.

2.7.3.9 Procesamiento Complejo.

Describe el grado en el cual el procesamiento lógico influencia el desarrollo de la aplicación. Algunas de las siguientes componentes podrían estar presentes en la aplicación:

- Control sensitivo (Ejemplo. Proceso de audición especial) proceso específico de seguridad.
- Procesamiento lógico extensivo.
- Procesamiento matemático extensivo.
- Una transacción incompleta provoca muchas excepciones resultantes del procesamiento al ser procesada nuevamente.
- El manejo de posibilidades múltiples de entrada o salida incluyen procesamiento complejo.

La forma de asignar su grado de influencia es como sigue:

0=Ninguna de las de arriba.

1=Alguna de las de arriba.

2=Dos de las de arriba.

3=Tres de las de arriba.

4=Cuatro de las de arriba.

5=Cinco de las de arriba.

2.7.3.10 Reusabilidad.

Describe el grado en el cual la aplicación y el código en la aplicación ha sido específicamente diseñado, desarrollado y soportado para que se pueda utilizar en otras aplicaciones. La forma de asignar su grado de influencia se muestra en seguida:

0=El código no es reusable.

- 1=El código reusable se utiliza en la aplicación.
- 2=Menos del 10% de la aplicación considera más de una necesidad del usuario.
- 3=El 10% o más porcentaje de la aplicación consideran más de una necesidad del usuario.
- 4=La aplicación fue específicamente empaquetada y/o documentada para facilitar el reuso, y la aplicación es personalizada por el usuario en el nivel de código fuente.
- 5=La aplicación fue específicamente empaquetada y/o documentada para facilitar el reuso, y la aplicación es personalizada por el usuario mediante parámetros de mantenimiento.

2.7.3.11 Facilidad de Instalación.

Describe el modo en que la conversión desde medios ambientes previos influenciarán el desarrollo de la aplicación.

La conversión e instalación como características de la aplicación deben ser probadas durante la fase de prueba. La forma de asignar su grado de influencia es:

- 0=No hay consideraciones especiales puestas por el usuario, ni se requirió un setup para la instalación.
- 1=No hay consideraciones especiales puestas por el usuario, pero se requirió un setup para la instalación.
- 2=La conversión e instalación fueron requeridas por el usuario, fueron proveídas y probadas guías para la conversión e instalación. El impacto de la conversión sobre el proyecto no se considera importante.
- 3=La conversión e instalación fueron requeridas por el usuario, fueron proveídas y probadas guías para la conversión e instalación. El impacto de la conversión sobre el proyecto es considerado importante.
- 4=Además de las dos anteriores, herramientas automáticas de conversión e instalación serán proveídas y probadas.
- 5=Además de las tres anteriores, la conversión e instalación automáticas serán proveídas y probadas.

2.7.3.12 Facilidad de Operación.

Describe el grado en el cual las aplicaciones atienden los aspectos operacionales, tales como: salvar y recuperar datos y recuperación de procesos. La facilidad de operación es una característica de la aplicación. Minimizando la necesidad de actividades manuales, tales como montaje de cintas, manejo de papel e intervención manual directa en el lugar. Las características pueden ser la de la lista siguiente:

- Se proveerán procedimientos de salvado y recuperación de datos y de procesos, pero se requiere la intervención del operador.
- Se proveerán procedimientos de salvado y recuperación de datos y de procesos, sin requerir la intervención del operador (cuenta doble).
- La aplicación minimiza el tener que montar cintas.
- La aplicación minimiza el tener que manejar el papel.

La forma de asignar su grado de influencia es como sigue:

0=No se establecieron requerimientos especiales de operación además de procedimientos normales de salvado de datos.

1=La aplicación considera una de las características sencillas.

2=La aplicación considera dos de las características sencillas o la doble.

3=La aplicación considera tres de las características sencillas o la doble y una sencilla.

4=La aplicación considera todas las cuatro características sencillas o la doble y dos sencillas.

5=No hay intervención de un operador del sistema.

2.7.3.13 Varios Sitios.

Describe el grado en el cual la aplicación será diseñada, desarrollada e implantada en múltiples localizaciones y organizaciones de usuarios. La forma de asignar su grado de influencia es:

0=Las especificaciones del usuario sólo consideran un sitio de instalación.

1=Las consideraciones de sitios múltiples se considerarán en el diseño y la aplicación correrá bajo instalaciones de hardware y software iguales.

2=Las consideraciones de sitios múltiples se considerarán en el diseño y la aplicación correrá bajo instalaciones similares de hardware y/o software.

3=Las consideraciones de sitios múltiples se considerarán en el diseño y la aplicación correrá bajo instalaciones de hardware y/o software diferentes.

4=La documentación y plan de soporte se proveerán y probarán para soportar la aplicación en sitios múltiples para aplicaciones del tipo descritas en 1 y 2.

5=La documentación y plan de soporte se proveerán y probarán para soportar la aplicación en sitios múltiples para aplicaciones del tipo descritas en 3.

2.7.3.14 Facilidad de Cambios.

Describe el grado en el cual la aplicación ha sido desarrollada para la modificación fácil del procesamiento lógico o las estructuras de datos. Las aplicaciones pueden tener alguna de las siguientes características:

- Consulta flexible y facilidad de reportes, mediante el manejo de requerimientos simples como el uso de los operadores lógicos y ú o sobre un ALI.
- Consulta flexible y facilidad de reportes, mediante el manejo de requerimientos simples como el uso de los operadores lógicos y ú o sobre dos ALI.
- Consulta flexible y facilidad de reportes, mediante el manejo de requerimientos simples como el uso combinado de los operadores lógicos y ú o sobre tres ALI.
- Datos de control del negocio se guardan en tablas que son mantenidas por el usuario con procesos interactivos en línea, pero los cambios se reflejan hasta el día siguiente del negocio.

- Datos de control del negocio se guardan en tablas que son mantenidas por el usuario con procesos interactivos en línea, los cambios se reflejan inmediatamente.

La forma de asignar su grado de influencia se muestra en seguida:

0=Ninguna de las de arriba.

1=Alguna de las de arriba.

2=Dos de las de arriba.

3=Tres de las de arriba.

4=Cuatro de las de arriba.

5=Cinco de las de arriba.

2.7.4 Contabilidad final en términos de Puntos de Función.

Finalmente la contabilidad definitiva de Puntos de Función está dado por los valores M y T que guardan una relación en la que la primera modifica a la segunda según la siguiente fórmula:

$$pf = T * (0.65 + 0.01 * M)$$

Debe notarse que hasta este punto no se le ha dado un significado práctico a los Puntos de Función, esto se hará en el siguiente capítulo.

2.8 Las Redes semánticas Naturales.

Este método es aplicado en México por psicólogos para definir conceptos ambiguos en una sociedad y medir varios comportamientos. El método fue iniciado en el área de Inteligencia Artificial por E. Feigebaum al realizar un modelo de procesamiento de información para simular el proceso elemental de aprendizaje del humano, este modelo fue reflejado en el Programa Elementary Perceiver and Memorizer (EPAM); posteriormente, fue modificado y ampliado por M. R. Quillian para mostrar un modelo de cómo la información semántica es almacenada dentro de la memoria de una persona. Adaptado en México por Figueroa, González y Solís para la psicometría, lo convirtieron en un modelo más natural y abierto. Para hacerlo más general sufre modificaciones nuevamente, estas son realizadas por Isabel Reyes Lagunes Facultad de Psicología de la UNAM, ella renombra los conceptos y acota el tipo de instrumentos.

2.8.1 Las Redes Semánticas Naturales en la definición de constructos.

Esta técnica sirve para conocer el significado que de diferentes conceptos pueden tener las personas. Funciona de la siguiente manera:

1. Se le entrega a la persona una tarjeta (ver Figura 2-7) con el concepto que se quiere investigar y se le pide que la defina con la mayor precisión posible con un mínimo de cinco palabras sueltas que considere están relacionados, sin utilizar artículos, preposiciones o conjunciones. Esta operación debe durar un máximo de 3 minutos por concepto a definir.

2. Una vez escritas las palabras definitorias, se le solicita al sujeto que asigne a cada palabra su importancia dentro del concepto dándole un peso de 1 a la de mayor importancia, 2 a la siguiente en importancia, y así sucesivamente hasta que termine de darle peso a todas las palabras. Esta actividad debe durar como máximo 1 minuto para cada concepto.

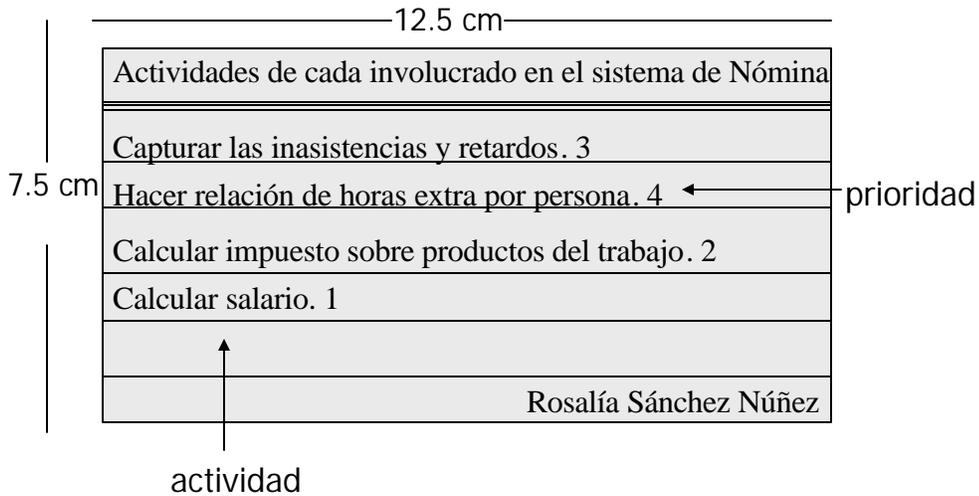


Figura 2-7. Tarjeta para la definición de constructos.

Después de recoger las tarjetas de los diferentes individuos se introducen en una hoja de cálculo como en el ejemplo de la Figura 2–8, en éste se utilizó el concepto *manzana* y después de los cálculos se nota que la palabra definitoria asociada a *manzana* con mayor peso semántico fue la de *fruta*.

Definitoria	Orden	1	2	3	4	5	6	7	8	9	10	P.S.
	Ponderación	(10)	(9)	(8)	(7)	(6)	(5)	(4)	(3)	(2)	(1)	
Árbol	Frecc.	4	1	2	3	5	4	5	7	9	10	
	V.P.	(40)	(9)	(16)	(21)	(30)	(20)	(20)	(21)	(18)	(10)	205
Fruta	Frecc.	9	8	7	5	6	5	5	1	2	2	
	V.P.	(90)	(72)	(56)	(35)	(36)	(25)	(20)	(3)	(4)	(2)	343
Pera	Frecc.	3	3	2	4	4	5	6	5	8	10	
	V.P.	(30)	(27)	(16)	(28)	(24)	(25)	(24)	(15)	(16)	(10)	215
Roja	Frecc.	10	8	8	5	5	5	4	3	2	0	
	V.P.	(100)	(72)	(64)	(35)	(30)	(25)	(16)	(9)	(4)	(0)	355

Figura 2-8 Formato de captura de la Red Semántica Natural para el concepto: Manzana. (V.P. = Valor Ponderado, P.S. = Peso Semántico)

Una vez calculada la red los valores a considerar son:

1. TR. Tamaño de la red, esto es el número de palabras diferentes definitorias del concepto.
2. PS. Peso semántico de cada una de las palabras definitorias. El PS indicará la prioridad de las palabras.

3. NR. Núcleo de la red, elementos que realmente representan el concepto y se obtiene graficando los pesos semánticos en forma descendiente y en el momento en que la curva adquiriera un carácter asintótico al eje X, las palabras definitorias restantes no formaran parte del NR. Esto le dará al Analista la lista de palabras (conceptos) que realmente afectan al sistema.
4. DSC. Distancia Semántica Cuantitativa, se obtiene para cada definitoria del Núcleo Semántico mediante una regla de tres, considerándose la definitoria del 100% a la de mayor PS. Con este índice el analista podrá asignar prioridades en los conceptos del sistema de software.

2.8.2 La Red Semántica Natural del Sistema de Software.

En el caso de los sistemas de software se recogerán, de la misma manera, las palabras definitorias de los siguientes conceptos:

1. Definir el sistema (nombre del sistema). Lista de sinónimos.
2. Elementos del sistema (nombre del sistema). Documentos de entrada al sistema (facturas, oficios, memos, etc.), objetos utilizados (impresora, maquina de escribir, fotocopiadora, foliadora, etc.), resultados obtenidos (reporte de totales, facturas, nomina, boleta de calificaciones, etc.).
3. La satisfacción por las actividades que realiza cada persona en el sistema (nombre del sistema)
4. Las actividades de cada participante dentro del sistema (nombre del sistema).

Los conceptos 1 a 3 pueden ser anónimos, pero el 4 debe llevar nombre de la persona para la posterior asignación de responsabilidades de los participantes. El sentido de la aplicación de este instrumento es obtener lo siguiente:

1. Identificar lo que los usuarios consideran que es su sistema.
2. Recolectar una lista de elementos del sistema y que algunos de ellos pueden no haber aparecido en los documentos oficiales ni en el momento de la observación.
3. Conocer las actividades formales e informales que los usuarios realizan dentro del sistema.
4. Asignar prioridades a los conceptos utilizados que brindarán un apoyo para delimitar las fronteras del sistema de software.
5. Saber que posición guardan los usuarios con respecto al sistema actual y poder motivar la elaboración del software de apoyo.

Ejemplo. Se construyó la Red Semántica Natural (RSN) del los elementos del sistema denominado Constructor de Curriculum Vitae (**CCV**) y se obtuvieron las definitorias de la Tabla 2-5 y la curva como se muestra en la Figura 2-9.

Tabla 2-5. Lista de definatorias sobre los elementos del sistema CCV

Definatorias
1. computadora
2. documentos comprobatorios
3. Experiencia profesional
4. computadora
5. Hoja
6. Escolaridad
7. impresora
8. Experiencia docente
9. Idiomas
10. Experiencia en investigación
11. Procesador de textos
12. Referencias
13. Manejo de maquinaria
14. Acta de nacimiento

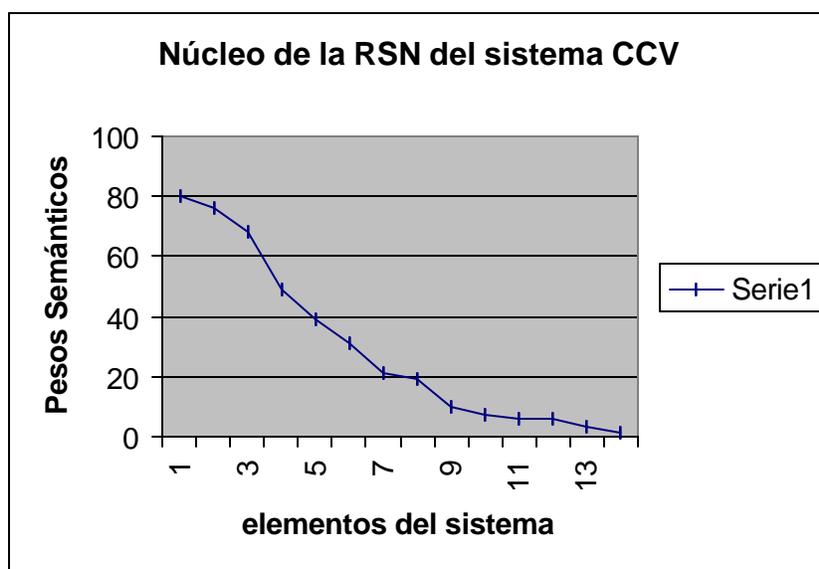


Figura 2-9. Curva de Pesos Semánticos para calcular el núcleo de la red de elementos del sistema CCV.

De acuerdo a la Figura 2-9 los elementos del 9 al 13 quedan fuera del núcleo de la red. Sin embargo, el elemento *experiencia en investigación* no debe quedar fuera de un sistema de elaboración de curriculum vitae. Respecto a los otros elementos fuera del núcleo de la red se ve que algunos son parte de otros elementos que sí están incluidos en la red, como el caso de *acta de nacimiento* es parte de *datos personales* y otros no van en un curriculum vitae, como es el caso de *referencias*.

Cabe hacer notar que los individuos entrevistados para construir la RSN del sistema CCV fueron tomados al azar en la ciudad de Xalapa, Veracruz y en su

mayoría fueron empleados de empresas de gobierno, sin experiencia en investigación.

2.9 Medición de actitudes.

Una definición de actitud social es “una organización duradera de creencias y cogniciones en general, dotada de una carga afectiva a favor o en contra de un objeto social definido, que predispone a una acción coherente con las cogniciones y afectos relativos a dicho objeto” .

Los elementos constitutivos de la actitud son componente cognitiva, componente afectivo y componente relativo a la conducta. Si se desea modificar una actitud, se puede empezar modificando la componente cognitiva dándole al usuario de un software nueva información sobre lo que éste, o alguno semejante, puede hacer por él.

Antes de comenzar a realizar actividades para modificar el conocimiento del usuario es necesario medir la actitud general ante la eventualidad de la construcción de un nuevo software y actuar de acuerdo a lo medido. Para esta medición se escogió el método de escalamiento de Likert que a continuación se explica.

Un instrumento que utilice la escala de Likert debe construirse como sigue:

- i. Cada inciso debe ser construido como una afirmación o juicio ante el cual se mide la reacción del sujeto.
- ii. Las afirmaciones califican al objeto de actitud que se está midiendo y deben expresar sólo una relación lógica.
- iii. La reacción ante cada afirmación debe ser medida en una escala de puntos que van del valor uno hasta un número non.
- iv. Estos puntos de la escala indican cuánto se está de acuerdo con la afirmación correspondiente. Ejemplo si se tiene una escala de cinco los puntos corresponden a 1=Muy en desacuerdo, 2=En desacuerdo, 3=Neutral, 4=De acuerdo y 5=Muy de acuerdo.

Una vez aplicado el instrumento debe ser calificado como sigue:

Para cada encuesta individual:

- i. Las afirmaciones a favor de elaborar el nuevo software calificarlas de valor superior a 1.
- ii. Las afirmaciones en contra de que se haga un nuevo software calificarlas del 1 a valor superior.
- iii. Para cada encuesta se suman las respuestas y se dividen entre el número de afirmaciones.

Para el grupo de encuestas, se observan todas las calificaciones de las encuestas y se obtienen:

- i. Moda. Valor más frecuente que apareció en las encuestas.

- ii. Mediana. Para obtenerlo, se acomodan en una lista L los valores de las diferentes encuestas de menor a mayor, luego se suman 1 más el número de encuestas (N) y se divide entre dos, lo que dará la posición de L donde estará el valor que será la mediana.
- iii. Media (M). El promedio de los valores obtenidos en cada encuesta. Esto es, la suma de todos los valores divididos entre el número de encuestas (N).
- iv. Desviación estándar (s). Se obtiene con la raíz cuadrada del siguiente valor: $((\sum(V_i - M)^2)/N)$. La s se interpreta como “cuánto se desvía, en promedio, de la media un conjunto de valores”.
- v. Varianza (s^2). Es la desviación estándar al cuadrado.

Una vez obtenidos todas las medidas se puede hacer la interpretación de cuál es la actitud hacia el nuevo software. Se ejemplifica con dos grupos de medidas que dan una idea de cómo hacer interpretaciones.

Se debe recordar que la variable a interpretar es la actitud hacia la elaboración de un nuevo software y que la escala de Likert que se utiliza en este ejemplo va desde 1(desacuerdo) hasta 7(acuerdo). Pasando por los valores 2(no totalmente en desacuerdo), 3(en desacuerdo con reservas), 4(ni de acuerdo, ni en desacuerdo), 5(de acuerdo con reservas) y 6(casi totalmente de acuerdo).

Ejemplo 1.

Supóngase que se obtuvieron las medidas de tendencia central que siguen:

Moda: 6

Mediana: 5.9

Media: 6.2

Desviación estándar: 0.7

Puntuación más alta observada: 7

Puntuación más baja observada: 4

Interpretación: Los usuarios están de acuerdo en que se elabore el nuevo software. La cantidad que más se repitió fue 6(casi totalmente de acuerdo). El 50% de los individuos están por encima de 5.9 el otro 50% por debajo de este valor. En promedio los individuos se sitúan en 6.2 (de acuerdo). Se desvían del 6.2 en un 0.7 unidades de la escala. Ninguna persona está en desacuerdo total de que se haga el software.

Ejemplo 2.

Supóngase que se obtuvieron las medidas de tendencia central que siguen:

Moda: 1

Mediana: 1.5

Media: 1.3

Desviación estándar: 0.4

Varianza: 0.16

Puntuación más alta observada: 4

Puntuación más baja observada: 1

Interpretación: La actitud hacia la elaboración de un nuevo software es de que no se debería hacer. Los datos se encuentran poco dispersos.

2.9.1 Construcción de la encuesta.

El elemento principal en la determinación de los requerimientos de software es el usuario. Por ello, si el usuario no está convencido de la necesidad del apoyo de la computadora en las tareas del sistema, el software no va a reflejar qué se quiere de él, ni será utilizado a pesar de haber sido terminado y tener todos los parámetros de calidad posibles.

Por lo anterior, en Áncora se propone la elaboración y aplicación de una encuesta de actitud. La elaboración estará basada en los sentimientos de satisfacción o insatisfacción y las actividades que el usuario escribió cuando se le entregó la tarjeta tipo tres y cuatro, respectivamente, de la RSN del sistema.

Originalmente se tienen dos listas de palabras definitorias de sentimientos y de actividades. Éstas se analizan y combinan en una serie de afirmaciones que serán calificadas en una escala de Likert de uno a cinco como ya se explicó. A continuación se muestra un ejemplo de elaboración de una encuesta de actitud.

Para el sistema Constructor de Curriculum Vitae (CCV) se obtuvieron las siguientes definitorias sobre satisfacción e insatisfacción del sistema actual (realizar un curriculum vitae con procesador de palabras o máquina de escribir) ordenadas por peso semántico: nerviosismo, flojera, molestia, estrés, orgullo, enojo, búsqueda, inseguridad, desesperación, confianza, motivación, curiosidad, duda, necesidad, frustración, impotencia, indeciso, insatisfacción, tedio y cansancio. La curva de la red semántica natural es la que se muestra en la Figura 2-10.

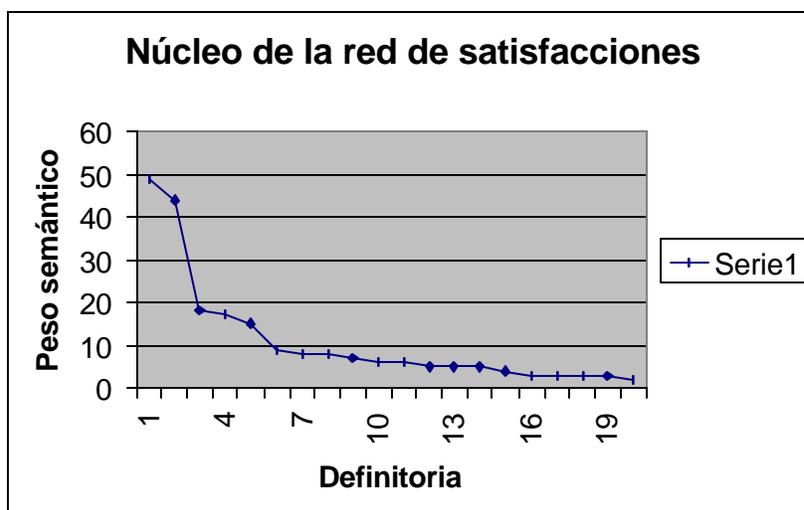


Figura 2-10. Red de satisfacciones e insatisfacciones sobre el sistema CCV

Ya que a partir de la palabra seis la curva se vuelve asintótica, las palabras que definen la satisfacción e insatisfacción en la creación de un curriculum vitae son las primeras cinco: flojera, molestia, estrés, orgullo y enojo. En este caso se decidió eliminar la palabra orgullo, porque se refiere a la obtención del producto final (el Curriculum Vitae) y no al proceso.

En el caso de las actividades que se realizan en CCV las definitorias en orden de peso semántico son: buscar, ordenar, registrar, revisar, integrar, fotocopiar, recordar, imprimir, empastar, solicitar, pelear y gastar.

Según la curva de la red semántica de las actividades de la Figura 2-11, el núcleo lo forman las primeras seis palabras: buscar, ordenar, registrar, revisar, integrar y fotocopiar.

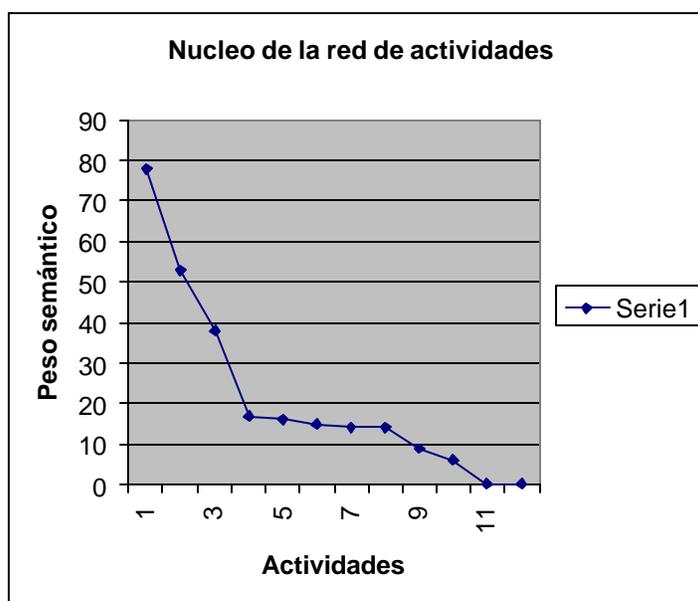


Figura 2-11. Red de actividades sobre el sistema CCV

Una vez obtenidas las actividades y los sentimientos de satisfacción e insatisfacción del CCV se procede a realizar la encuesta, en cuyas afirmaciones deben combinarse las actividades y los sentimientos. La siguiente lista muestra algunas de las afirmaciones que se incluyeron en la encuesta:

1. Utilizar medios computacionales para *ordenar* información que me permita construir mi curriculum bajaría mi *estrés*.
2. Pensar en *ordenar* información para construir mi curriculum me causa *molestia*.
3. El *buscar* mis datos en la computadora me reduce el *nerviosismo* de hacerlo en mis archivos no electrónicos.
4. Si tuviera todas mis constancias en la computadora, *fotocopiar* (meter el documento como imagen mediante el Scanner) la información para construir mi curriculum me daría menos *flojera*.

Las afirmaciones anteriores se califican poniendo una X en alguno de los cinco siguientes valores: Definitivamente sí, Probablemente sí, Indeciso, Probablemente no, Definitivamente no.

Una vez aplicada la encuesta, se interpreta como se explicó anteriormente, esta interpretación indicará a cliente y analista si es necesario incluir actividades que permitan que los usuarios conozcan las ventajas de tener un apoyo computacional. Además un análisis de cada afirmación permitirá observar en qué actividades están los problemas más fuertes.

2.10 Técnica de Grupo Nominal.

La Técnica de Grupo Nominal (TGN) fue creada en 1960 a partir de estudios sociopsicológicos. La TGN se desarrolla bajo una reunión estructurada de grupo para la toma de decisiones organizacionales y resolución de problemas.

Los procedimientos de que se compone la TGN son:

- i. Generación de ideas. Cada individuo en el grupo generará en forma silenciosa sus ideas y las escribirá.
- ii. Recolección de ideas. Mediante un recorrido de todos los participantes se escribirán las ideas a la vista de todos.
- iii. Discusión de ideas. Cada idea escrita se discute para clarificarla y evaluarla.
- iv. Votación sobre las ideas. Cada individuo votará por la prioridad individual de cada idea y se conjuntarán posteriormente para obtener su promedio.

Los Beneficios que se obtienen de la TGN son:

- i. Participación balanceada de todos los participantes.
- ii. Influencia balanceada sobre los individuos.
- iii. Producción de ideas más creativas que las de grupos interactivos.
- iv. Mayor satisfacción de los participantes sobre los resultados.
- v. Reducción del conformismo.
- vi. Anima a los participantes a confrontar los temas sobre una base de resolución de problemas en lugar de basarse en agresiones personales.
- vii. Conduce a una sensación de mayor acercamiento y complicidad.

Las limitaciones que puede tener la TGN son:

- i. Requiere de mayor preparación que una técnica espontánea.
- ii. Tiende a limitar los temas a tratar, no se puede cambiar a otro tema a la mitad de la reunión.
- iii. Necesita que todos los participantes estén de acuerdo en cooperar.

Aunque las limitaciones listadas pueden serlo en otros ámbitos de trabajo, en el caso del análisis de requerimientos de software resultan en ventajas.

Claramente la TGN necesita una preparación previa de lugar de reunión, materiales y temas a tratar.

2.11 La Reunión de Reflexión y Diseño.

Para llevar a cabo los diferentes modelos se utilizará la adaptación del método de Reunión de Reflexión y Diseño (RRD) usado para el modelado de sistemas sociales en México y que tiene como principios básicos los siguientes:

1. Debe proporcionarse una plataforma para que los procesos de reflexión y diseño se puedan dar en la mayor libertad posible, y en un plano espacio - tiempo tan amplio como lo consideren los participantes.
2. Asegurar una amplia participación de los involucrados en una situación problemática compleja, creando un ambiente democrático.
3. Propiciar un clima de confianza y respeto para que las personas que asisten como individuos y no como representantes de sus grupos de pertenencia, viertan sus opiniones aunque éstas sean antagónicas.
4. Motivar tanto el diseño de futuros deseables como el esfuerzo conjunto para su consecución.
5. Impulsar un aprendizaje social que aumente la habilidad para la auto – organización, la adaptación y la planeación.

El objetivo final de la RRD es la obtención de los diferentes modelos de qué se quiere del software, para lograrlo RRD propone las etapas de la Tabla 2-6.

Tabla 2-6. Etapas y puntos a tratar en la Reunión de Reflexión y Diseño.

ETAPA	PUNTOS A TRATAR
Exploración del ambiente.	Factores externos que han afectado al sistema. Posibles factores que podrían afectar al sistema.
Análisis interno del sistema.	Situación actual del sistema. Causas de la situación actual. Situación a futuro si no se hace el software para apoyar el sistema del cliente.
Diseño del software ideal.	Definir el software ideal para los próximos 5 años. Funcionamiento de las cosas en ese futuro ideal.
Identificación de obstáculos y oportunidades.	Identificar obstáculos para alcanzar el software ideal. Identificar oportunidades para lograr el software ideal.
Selección de líneas de acción.	Acciones posibles y necesarias para alcanzar el software ideal.
Diseño de líneas de acción.	Actividades necesarias para alcanzar las líneas de acción. Establecimiento de responsabilidades y acciones de cada involucrado.

2.11.1 El lugar para la RRD.

Para poder progresar en el establecimiento de lo que se quiere del software es importante que las reuniones se realicen en lugares alejados del sitio de trabajo, para que los involucrados no se distraigan con las actividades cotidianas que requieren de ellos. Sin embargo, en el caso de México debe ser dentro de la

ciudad donde se desarrollará el software; lo anterior se debe a que en las empresas mexicanas resulta muy difícil ausentarse las 24 horas del día.

2.11.2 Los asistentes a RRD.

Con respecto a las personas involucradas en el desarrollo del software resulta óptimo que haya al menos un representante de cada grupo de pertenencia de los que utilizarán el software, pero si no es posible, deben reunirse los involucrados de la empresa que tengan la posibilidad de ausentarse físicamente de la empresa, los analistas y un representante del grupo de diseño de software (si son del área de computación, pueden ser las mismas personas que fungen como analistas).

2.11.3 La duración de la RRD.

Aunque en otros países se recomienda una duración de cinco días, la RRD ha comprobado que en México resulta poco probable de cumplir un lapso de tiempo de reunión tan extenso, por lo que se deben utilizar un mínimo de 2 y un máximo de 3 días.

Para el caso de la revisión de los requerimientos de software sólo se necesitarán dos sesiones de 3 horas cada una. Lo anterior se debe a que el problema a resolver es más estructurado y el analista ya debe llevar propuestas. En la siguiente parte de este documento se detalla la forma en que Áncora propone llevar a cabo la RRD.

2.11.4 La conducción de la RRD.

Las reuniones de la RRD con los involucrados pueden ser de dos tipos:

1. La de grupos pequeños. Un máximo de 10 participantes, incluido el facilitador.
2. Las plenarias. Todos los involucrados.

Dependiendo de la cantidad de personas que usarán el software se realizan o no los dos tipos de juntas. Si es para pocos usuarios, todas las actividades de las reuniones se harán en un solo grupo.

Para las juntas de grupos pequeños debe haber un facilitador por grupo que debe comportarse como sigue:

1. No deben intervenir en las discusiones dando opiniones sobre el tema tratado.
2. Mantener un ambiente de confianza y respeto para todos los miembros del grupo y sus ideas.
3. Vigilar que participen de todos los integrantes del grupo.

2.11.5 Apoyos físicos para la RRD.

Para el logro de los puntos anteriores se tienen los siguientes apoyos:

1. El conjunto de Guiones y diálogos teóricos y actuales, elaborados previamente por el analista, así como el glosario que se lleva hasta el momento.
2. Un cartel pegado en una parte visible como el de la Figura 2-12.

3. Para los grupos pequeños una organización física del lugar de las reuniones en semicírculo con el facilitador y su rotafolio al frente. Debe evitarse el uso de mesas.
4. Para las sesiones plenarias un acomodo equivalente con semicírculos concéntricos o filas con el facilitador al frente.
5. La formulación de las reglas de conducción de las participaciones por parte de los involucrados, mismas que deberán respetarse durante todo el evento.
6. El uso del método TGN para recoger la prioridad de las escenas.
7. La elaboración de los modelos correspondientes al final de cada sesión.

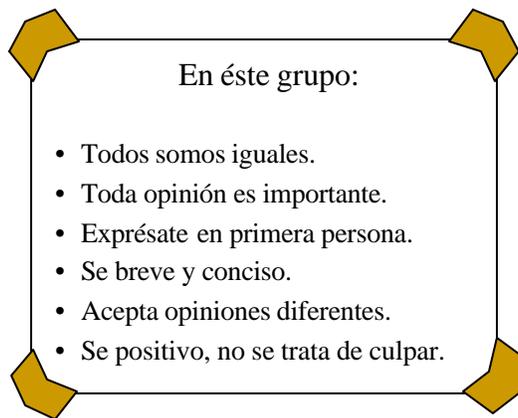


Figura 2-12. Cartel de apoyo para las reuniones de grupos pequeños.

2.11.6 Los resultados de la RRD.

Al término de la RRD se tendrán:

1. Un conjunto de Guiones y diálogos de grupos pequeños, en donde se establecen los papeles y responsabilidades de cada actor (involucrado).
2. Un conjunto de Guiones y diálogos de sesiones plenarias, en donde se establecen los papeles y responsabilidades de cada actor (involucrado).
3. Una serie de hojas de trabajo (las del rotafolio).
4. Un glosario homogéneo de términos.
5. Una definición del contexto donde se implantará el software.
6. El conjunto de interfaces que el nuevo software debe considerar.
7. Un manual de operación propuesto aceptado por los involucrados.
8. El costo tentativo del software.

2.11.7 El Calendario para la RRD.

Antes de empezar la RRD debe planearse un calendario de actividades, que podría verse como el de la Figura 2-13.

Debe notarse que es sólo una propuesta y que puede cambiarse al gusto y necesidades de la empresa para la que se realizará el software, pero respetando las etapas de la RRD. En el primer día se debe entregar, en la medida de lo posible, fotocopias de la documentación a cada involucrado. Para el siguiente día, el analista modificará guión de propuesta computacional y manual de operación según las correcciones obtenidas en la sesión del día 1.

Tema: {nombre del software}
 Propósito: Clasificación, jerarquización y validación de requerimientos del software.
 Lugar: {lugar dónde se llevará cabo la RRD} (fuera del lugar de trabajo de los involucrados)

{día 1} de {mes} del {año}

Hora	Tipo de reunión	Material requerido	Acciones
9:00 – 9:30	Reunión informal de los participantes	<ul style="list-style-type: none"> Hojas para anotar a participantes 	Anotar a las personas presentes
9:30 - 10:30	Sesión Plenaria Introdutoria.	<ul style="list-style-type: none"> Escrito con puntos a tratar a la vista de todos. 	Presentar el objetivo de la reunión y participantes.
10:30 - 11:30	Primera sesión en grupos pequeños: Clasificación de requerimientos y resolución de conflictos. Revisión de las escenas propuestas.	<ul style="list-style-type: none"> Guiones de situación teórica, actual y propuesta. Manual de operación. 	Leer, analizar y anotar correcciones sobre el material.
11:30 - 11:45	Receso.	<ul style="list-style-type: none"> Café y galletas 	El analista debe sacar la lista de correcciones
11:45 – 12:00	Primera sesión plenaria. Presentaciones.	<ul style="list-style-type: none"> Lista de las correcciones 	Leer la lista de correcciones y pedir consenso.
12:00 - 13:00	Segunda sesión en grupos pequeños: Restricciones bajo las cuáles deberá implantarse el nuevo software	<ul style="list-style-type: none"> Lista de modificadores de Puntos de Función (fotocopia para cada participante, lista y tablas correspondientes). Lista de quintetas y forma de comprobación. 	Anotar en la lista el grado de influencia de cada modificador.

{día 2} de {mes} del {año}

Hora	Tipo de reunión	Material requerido	Acciones
9:00 – 10:00	Tercera sesión en grupos pequeños: Asignación de prioridades de los requerimientos.	<ul style="list-style-type: none"> Guión de propuesta computacional. Manual de operación. Grupo de tarjetas para la asignación de prioridades de las escenas. 	Aplicar TGN.
10:00 – 10:15	Receso	<ul style="list-style-type: none"> Café y galletas 	El analista realizará la lista de jerarquización de escenas y anotará diferencias entre grupos.
10:15 – 10:30	Tercera sesión plenaria. Presentación de asignación de prioridades.	<ul style="list-style-type: none"> Lista de la jerarquización de prioridades. 	El analista lee las jerarquizaciones obtenidas en los grupos pequeños y recoge consenso.
10:30 – 11:30	Cuarta sesión en grupos pequeños: Validación de requerimientos	<ul style="list-style-type: none"> Guiones de situación actual y propuesta computacional. Manual de operación. Lista de jerarquización final. 	Se revisa la propuesta final para revisar si realmente corresponde a la solución de la problemática.
11:30 – 12:00	Cuarta sesión plenaria. Firma de acuerdos.	<ul style="list-style-type: none"> Todos los documentos anteriores. Oficio de firma de acuerdos. 	

Figura 2-13 Calendario de actividades para la RRD.

3 Áncora y el ciclo de vida.

Las etapas que forman el ciclo de vida de Áncora se muestran en la Figura 3-1 comenzando con la flecha gruesa de entrada a la etapa de Entendimiento del Dominio y Contexto de la Aplicación y termina con la flecha gruesa de salida en la etapa de Cierre. Las flechas dobles indican los artefactos que arroja cada etapa, la flecha doble la interacción con el almacén magnético de sistemas anteriores y por último las flechas simples indican el paso de una etapa a otra, debe notarse que hay retroalimentación de unas etapas con otras, aunque no en todas.

Las actividades y artefactos producidos en cada etapa son explicadas brevemente en seguida y de manera extensa a partir de la sección 3.1.

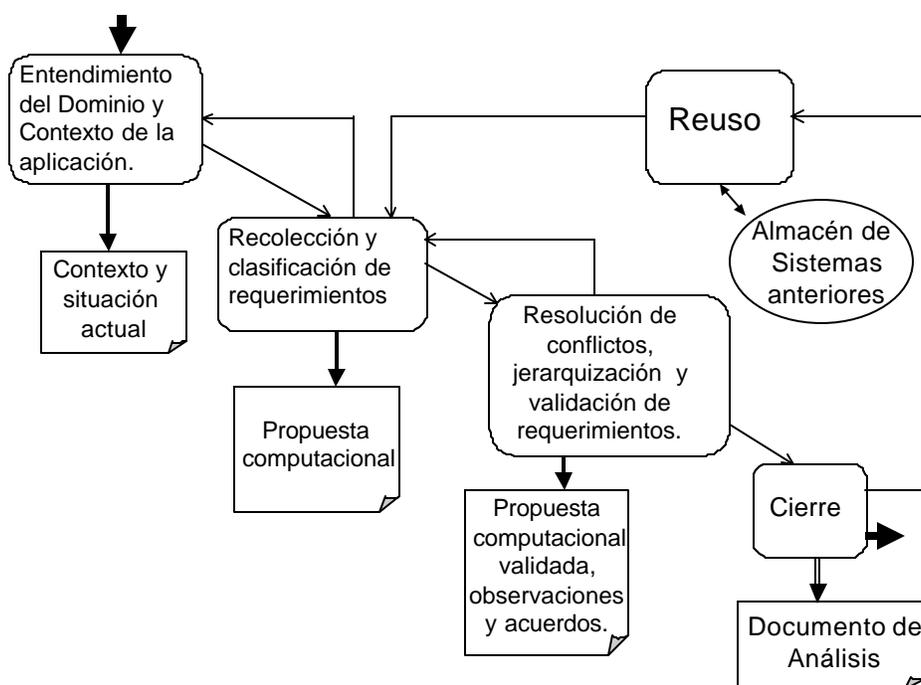


Figura 3-1 Ciclo de vida del ARS en Áncora.

Entendimiento del Dominio y Contexto de la Aplicación. En esta etapa se busca tener claros las metas de la empresa, el dominio de la aplicación, las actividades que se llevan a cabo en la situación actual del sistema para el cual se va a desarrollar el software de apoyo y la justificación del nuevo software. Para ello se llevarán a cabo las siguientes actividades que arrojan los artefactos correspondientes:

- *Leer material.* Con lo cual se podrá armar un primer glosario, esquema de la organización, lista tentativa de actividades que realizan los usuarios y guiones y diálogos de la situación teórica.

- *Entrevistas informales con el cliente.* Se obtendrán las metas de la empresa y se identificará la influencia del sistema en ellas.
- *Observar usuarios.* El analista podrá determinar cómo se realizan las tareas en el sistema actualmente y hará las anotaciones pertinentes.
- *Construir redes semánticas naturales del sistema.* Después de una reunión informal con los usuarios en la que se aplica cuatro veces la técnica de Red Semántica Natural (RSN) se tendrá un glosario más exacto y guiones y diálogos de la situación actual.
- *Construir y aplicar encuesta de actitud.* A partir de las RSN del sistema se construye una encuesta de actitud que permitirá saber qué sentimientos hay entre los usuarios con respecto a la construcción del software e identificar los principales problemas existentes que permitirán redactar la justificación del nuevo software.

El orden de aplicación de cada actividad no es secuencial, se realizarán según las necesidades, usos y costumbres de la empresa.

Recolección, clasificación de Requerimientos. Esta etapa tiene como objetivo crear la propuesta computacional del sistema compuesta por:

- La realización de la propuesta computacional, que incluye: Guión de propuesta computacional, manual preliminar de operación (prototipo rápido), modelo de datos (Objetos Semánticos), lista de responsabilidades, lista de beneficios esperados.
- Creación de la Bitácora de desarrollo del sistema, la cual consiste en una tabla de cuatro columnas; cada una contiene: una función del sistema, la forma de comprobación por parte del usuario, el tiempo propuesto y tiempo real de implantación.
- Aplicación de puntos de función. Esto permitirá establecer las restricciones (requerimientos no funcionales) del software, refinar la propuesta computacional y dar una medida de complejidad al sistema.

Para obtener todos los artefactos que conforman la propuesta computacional, la primera actividad es construir el Guión y de ahí se puede proseguir con otros artefactos que, muy posiblemente, retroalimentarán al Guión y éste a su vez a otros artefactos. Durante la creación de la propuesta computacional se realizarán entrevistas informales a cliente y usuarios.

Reuso de requerimientos. El objetivo de esta etapa es brindar al analista elementos de otros sistemas de software que ya han sido probados y cuyas especificaciones de requerimientos se han almacenado en una Base de Datos de Reuso.

- *Abstraer y guardar los guiones de sistemas terminados.* Esta es una operación de generalización en la que, mediante transformaciones respectivas de verbos y sustantivos a primitivas y nombres genéricos y la representación de escenas y condiciones de entrada en forma clausular, se obtendrá una representación abstracta de los guiones de la propuesta computacional.

- *Consultar y extraer elementos reutilizables.* Todos los artefactos de las especificaciones de requerimientos de sistemas semejantes que puedan reutilizarse, a saber: el guión o guiones de la propuesta computacional, manuales de operación, listas de restricciones y responsabilidades, bitácoras de desarrollo, cálculo de puntos de función y, si se uso el mismo paradigma de desarrollo elementos para la conexión con el análisis.

Resolución de conflictos, priorización y validación de requerimientos. Como su nombre lo indica en esta etapa se busca resolver los conflictos que puedan existir entre los diferentes requerimientos, asignar prioridad a cada requerimiento y obtener la validación por parte del usuario de los requerimientos. Las actividades son dos.

- *Preparación de la Reunión de Reflexión y Diseño (RRD).* Una vez obtenidos los documentos de la situación actual y de la propuesta computacional, estos se reproducen y empastan, entregando una copia a cada involucrado (clientes, usuarios, analistas, diseñadores) en el software. El motivo es el que antes de llevar a cabo una reunión los involucrados puedan analizar los documento poniendo sus anotaciones sobre el documento.
- *Reunión de Reflexión y Diseño.* Después de entregada la documentación se reúne a, por lo menos, un representante de cada grupo de interés y durante uno a dos días se lleva a cabo una reunión en donde todos los asistentes podrán verter sus opiniones y entre todos llegar a un contexto, además se aplicará la Técnica de Grupo Nominal para la jerarquización de requerimientos. Al finalizar la RRD se tendrán: Guiones, restricciones, prioridades, manual, responsabilidades y plan de prueba validados. Además de una lista de observaciones y acuerdos firmados de la RRD.

Cierre. En esta etapa se realizan una serie de actividades encaminadas a lograr una conexión con otras etapas del desarrollo del software. Las actividades son:

- Ratificar cálculo de Puntos de Función. Después de la RRD, tal vez, será necesario hacer una revisión de los Puntos de Función.
- Envío de oficios, correo electrónico y faxes para aclarar puntos pendientes.
- Conexión con el modelo de desarrollo escogido. Se realizarán una serie de traducciones que llevarán al planteamiento de los artefactos del análisis.

A partir de la siguiente sección se explicará la forma de llevar a cabo las diferentes actividades dentro de cada etapa y los artefactos que deben obtenerse al final de cada etapa.

3.1 Entendimiento del dominio y contexto de la aplicación.

Antes de comenzar a definir el nuevo software es necesario conocer el dominio de la aplicación y modelar la situación actual para entender la problemática. Para avanzar en estos puntos se deben realizar las siguientes actividades:

1. Búsqueda bibliográfica, construcción de un primer glosario y guión de la situación teórica. Para ello hay que conseguir todos los documentos que

hablen del sistema, antes de consultar a los involucrados dentro de la organización o empresa.

2. Conociendo la empresa y sus metas. Esto es, el entorno de la organización y hacia donde va ésta, lo cual resulta importante entender pues eso dará la pauta de la importancia del sistema software y permitirá definir cuales serán los recursos que realmente se le destinarán así como el enfoque de generalización y crecimiento que el software podría llegar a tener.
3. Generación del guión del estado actual. Además, se debe recolectar la forma de trabajo actual de la aplicación; esto se refiere a cuáles son las funciones, manuales o automáticas, que se realizan actualmente para el logro de las metas del sistema.
4. Identificación de los problemas. Una vez realizado el guión de la situación actual, con ayuda de la encuesta de actitud se pueden identificar tanto los problemas operacionales como los de actitud hacia el sistema actual y la eventualidad de la realización de un sistema de software de apoyo.

En las siguientes subsecciones se explicará cómo realizar estas actividades.

3.1.1 Construcción de un primer glosario y Guión teórico.

Debe recordarse que el cliente fue la persona, o grupo de personas, que contacto al analista para realizar el software; entonces, ellos están en la mejor disposición de que se realice el software. Al cliente debemos solicitarle que proporcione:

1. La documentación técnica referente al problema (Ejemplos: manuales de procedimientos, libros sobre el tema, páginas de Internet dónde se encuentre información).
2. Las formas asociadas con el sistema (Ejemplos: facturas, formas de inscripción, pantallas de captura, planos, tarjetas con datos recolectados en campo).

Con toda la información anterior se debe generar un glosario con los siguientes conceptos: Palabra, Significado en el dominio de la aplicación, formas donde aparece, referencia (manual, libro, persona, página de Internet, etc.) de donde obtuvo el significado.

Además, si es posible, se debe crear un Guión y sus diálogos del funcionamiento teórico del sistema, basándose únicamente en los manuales de procedimientos que el cliente proporcione.

3.1.2 Conociendo la empresa y sus metas.

Cuando se quiere hacer un nuevo software para una empresa en particular, se debe pensar en este software como una herramienta que apoyará las metas de la empresa. Por lo tanto se deben conocer las metas de la empresa y, de éstas, a cuales servirá el nuevo software.

Sin lo anterior el software resultaría un trabajo aislado sin conexión con el resto de la empresa, los resultados a obtener después de realizar esta actividad:

1. Metas principales de la empresa.

2. Esquema organizacional donde se describan las partes de la organización que requieren cambio.

El objetivo de este apartado es clarificar la situación del software frente a las necesidades organizacionales, por lo que se incluyen los primeros dos resultados, mismos que se obtienen de leer documentos oficiales, consultar al cliente y observar el lugar de trabajo.

3.1.3 El Guión de la situación actual.

El Guión de la situación actual se obtiene llevando a cabo las siguientes tareas:

1. Una observación silenciosa en el lugar de trabajo, esto es, sin intervenir u opinar sobre la forma de trabajar de los usuarios potenciales del nuevo software.
2. Aplicando la técnica de Redes Semánticas Naturales para descubrir y refinar los conceptos que se involucrarán en el software, a saber:
 - 2.1. Sinónimos del sistema. Ayudan a establecer las fronteras del sistema, esto es condiciones de entrada y salida.
 - 2.2. Elementos del sistema. Son los utensilios.
 - 2.3. Satisfacciones hacia el sistema. Problemática operacional.
 - 2.4. Actividades. Escenas y quintetas.

Al finalizar las dos tareas anteriores se debe tener los Guiones del estado actual, sus diálogos derivados de la observación y agregar en el glosario de términos que no hayan sido encontrados en la documentación formal. Sobre los Guiones deben subrayarse aquellas quintetas que sean problema.

En el ejemplo de la Figura 3-2 se muestran tres quintetas problemáticas, la primera representa una gran pérdida de tiempo y se presta a tener una agenda tachonada, problema que puede ser resuelto fácilmente usando medios computacionales; la segunda es provocada por el hecho de que el compromiso fue anotado en una hoja suelta que se puede perder, con la propuesta computacional puede ayudar el hecho de escribir todos los compromisos directamente sobre la computadora, pero no siempre será así y la tercera quinteta es un problema que no se resolverá con la propuesta computacional, ya que involucra poder comunicarse con la persona dueña de la Agenda de Compromisos.

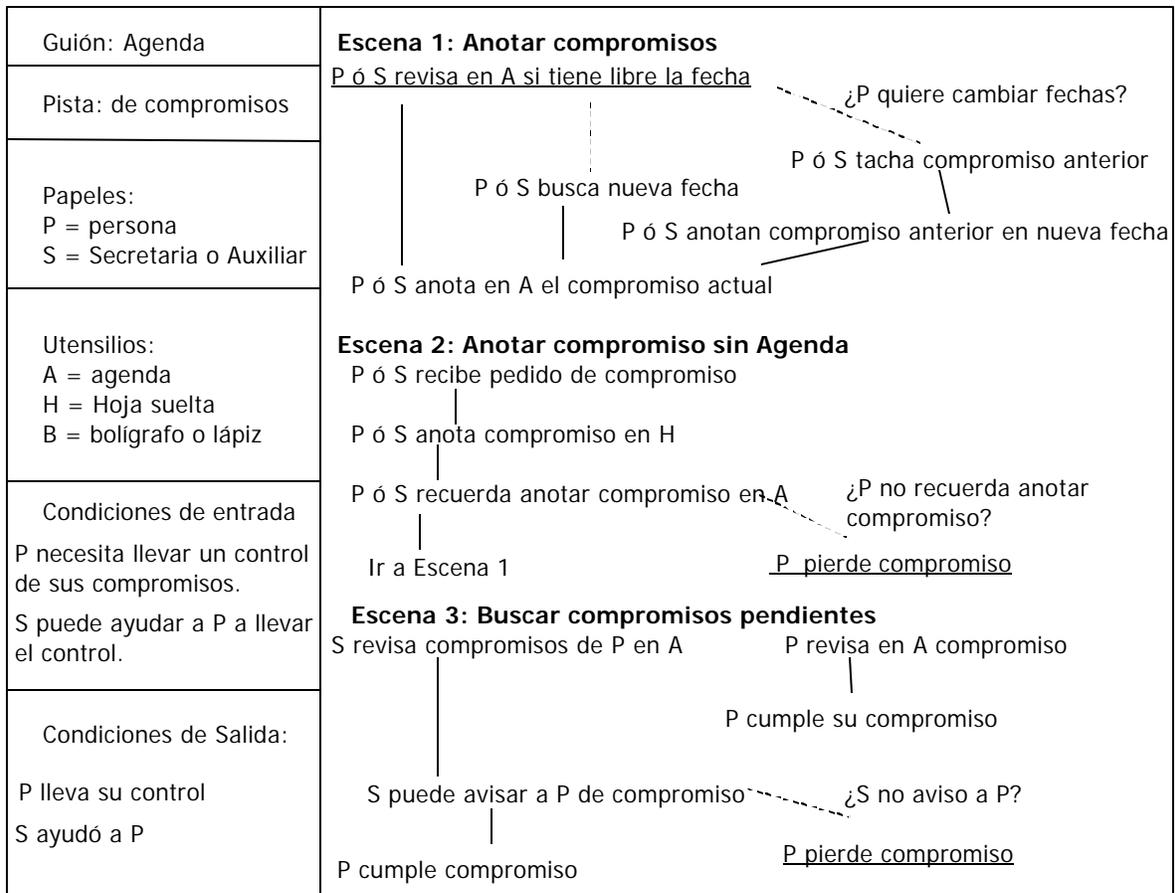


Figura 3-2. Guión del estado actual del sistema de Agenda de Compromisos.

3.1.4 Identificación de la problemática.

A pesar de que en la creación del guión de la situación actual ya fueron identificados algunos problemas operacionales, todavía falta corroborarlos con los puntos de vista de los usuarios.

También es necesario medir la actitud de los involucrados ante la eventualidad de un nuevo software. Para que el analista pueda conocer las actitudes ante el nuevo software, se debe utilizar una herramienta de medición, en el caso de Áncora se realiza una encuesta como se presenta en el capítulo anterior.

Además, dependiendo de los resultados, se agregan o no actividades de modificación de la conducta dentro de la RRD. Dichas actividades consistirían en mostrar sistemas de software equivalentes para indicar sus beneficios, puntualizando aquellas actividades que se verán beneficiadas y las habilidades que los usuarios tendrán que adquirir.

Al terminar esta actividad se debe tener un documento de justificación de la elaboración del nuevo software señalando cuáles son los problemas principales y cómo afectan a las metas de la empresa.

3.2 Recolección y clasificación de Requerimientos.

La meta de esta etapa es realizar una propuesta computacional, para lo cual se realizarán varios modelos que son:

1. Guión de la propuesta computacional en el cual deben quedar aclaradas las fronteras del sistema (condiciones de entrada y de salida), cómo se clasificarán los requerimientos (pistas y escenas), qué requerimientos cubrirán algunas de las necesidades de los usuarios (las escenas), cómo se cubrirán esos requerimientos (las quintetas), con qué recursos físicos se cubrirán (utensilios) y sobre todo quiénes harán las tareas (papeles).
2. Prototipo rápido.
3. Modelo de datos.

Además hay dos puntos más que en Áncora se incluyen:

4. Bitácora de desarrollo del software.
5. Cálculo de Puntos de Función.

3.2.1 Guión de la propuesta computacional.

A partir del guión de la situación actual y del guión teórico, de este último si lo hay, se construirá el guión de la propuesta computacional. Por supuesto muchos elementos cambiarán y estos cambios deben considerarse cuidadosamente para que al presentar la propuesta los usuarios no se sientan amenazados o angustiados con las nuevas responsabilidades y habilidades que tienen que adquirir.

En la propuesta computacional sólo se presentarán aquellas escenas que pueden realizarse con ayuda de la computadora, dejando a un lado la mayoría de actividades manuales observadas y anotadas en los guiones anteriores, deben incluirse sólo aquellas actividades manuales que se necesiten para la consecución de las automáticas.

A continuación se explica cómo ir construyendo el guión de la propuesta computacional:

1. Título del Guión. Puede conservarse el mismo nombre del sistema actual, pero darle además un nombre pequeño, mnemónico y atractivo para hacerlo familiar al usuario.
2. Pistas. Además de su nombre largo, en el que quede claro el conjunto de actividades que se realizarán, también deben llevar un nombre llamativo y fácil de recordar.
3. Utensilios. Los de los guiones anteriores que sean susceptibles a ser automatizados (que se convertirán en archivos) y aquellos que se necesitan para el nuevo software (Ejemplos: impresora, scanner, formatos).
4. Papeles. Este es el punto más delicado de la propuesta computacional, pues las tareas que los papeles realizaban cambiarán y hasta los mismos papeles podrían modificarse. Para aclarar este punto se da un ejemplo: supóngase que se está realizando un sistema de auditoría de gastos a los municipios de un estado. En el guión de la situación actual

se identificó un papel *receptor de documentos* que era la persona que recibía los documentos comprobatorios y elaboraba un oficio de recepción de los mismos. En la propuesta computacional este papel podría cambiar a *capturista de documentos*. Este pequeño cambio puede llevar a confusiones que hagan que el sistema sea tomado con reservas. Para evitar problemas debe realizarse una tabla de cambios de nombres de papel o no cambiar los nombres de papeles.

5. Condiciones de Entrada. Pueden ser las mismas que en el guión de la situación actual, aunque tal vez haya variaciones si se decide eliminar algunas escenas con actividades puramente manuales.
6. Condiciones de salida. Será una lista de quintetas con los resultados que el software brindará.
7. Escenas. Solamente son aquellas en las que existen procesos donde la computadora tiene que apoyar a las actividades de los usuarios.
8. Quintetas. Servirán para expresar con más detalle lo que se realizará en la escena a la que pertenecen. Algunas serán heredadas de los guiones actual y/o teórico pero serán las menos.

3.2.2 Construyendo un primer prototipo.

Para revisar varios aspectos que queden ocultos con otras herramientas de modelado, se propone construir uno o varios prototipos del nuevo software.

Así, uno de los prototipos puede ser el Manual de Operación, el cual se explica como crearlo en esta sección. El objetivo principal de incluir el manual de operación en la etapa de análisis se debe a que aclara de manera esquemática, al usuario y al analista, la forma en que todos los elementos van a interactuar. Sirve para mostrar aquellos puntos que estaban "sobrentendidos". En general, los "sobrentendidos" resultan en confusiones y conflictos posteriores. Es claro que, el manual de operación definitivo se completará después que se termine la programación.

3.2.2.1 Contenido General de un Manual de Operación.

El manual de operación sirve como el primer prototipo del sistema, y debe incluir:

1. Una introducción en que se muestre una visión general del sistema y que contenga los siguientes aspectos:
 - a. Una descripción funcional (general) sobre qué puede hacer el sistema.
 - b. Una descripción de cómo instalar el sistema. Puede hacerse de manera tentativa.
2. El índice general.
3. La explicación de cada funcionalidad.
4. Una guía que explique cómo reaccionar ante situaciones surgidas mientras el sistema se encuentra en uso. Es obvio que esta parte no puede hacerse completa en el análisis del sistema, pero puede iniciarse.

Debe aclararse que los puntos anteriores se refieren sólo al contenido del manual de operación. La forma de desarrollarlo será integrando los elementos de tal forma

que resulten amenos al usuario, para lo cual a continuación se listan algunas recomendaciones:

1. Debe ser realista acerca de las capacidades del sistema, sin destacarse en demasía las características novedosas o "muy poderosas".
2. Deben utilizarse formas activas en lugar de pasivas. Es preferible "verá un cursor en la esquina izquierda superior ...", en lugar de "un cursor será visto ...".
3. Deben evitarse las frases largas que incluyan varios hechos distintos.
4. La referencia a otras secciones debe hacerse con el nombre y número de éstas.
5. Si puede explicar su idea con cinco palabras, no use diez.
6. Los párrafos cortos en un manual son mejores.
7. Los títulos y subtítulos deben aparecer claramente.
8. Se debe cuidar la ortografía y las construcciones gramaticales sin abusar de los infinitivos (Ejemplos: mover, capturar, imprimir), participios, pasados o gerundios (Ejemplos: capturando, recibió, moviendo).

Para construir el prototipo con Áncora, cada escena es una función general y se debe explicar de manera sencilla con uno o dos párrafos.

Para detallar cada función general se desglosa en las quintetas se hará dibujando los utensilios que maneja, los cuales vienen en cada quinteta.

La explicación de cada quinteta puede hacerse agregando su diálogo o una breve descripción.

Para cada utensilio que incluya datos (Ejemplo: factura, boleta de calificaciones) se deberán agregar los posibles valores de éstos y si hay un dato compuesto se debe agregar su fórmula (ejemplo: fecha de compromiso = día + mes + año).

Todos los manuales de operación tienen características semejantes, así que se puede tener un "presentación general" (en el Anexo B se presenta un esquema de manual del usuario) en la cuál se le van agregando las escenas, quintetas y diálogos particulares. El manual preliminar propuesto por el analista debe estar listo antes de la RRD.

3.2.3 El Modelo de Datos.

Aunque en la etapa del análisis de requerimientos no se puede realizar un modelo de datos completo, si es conveniente esbozarlo, sobre todo para poder hacer un cálculo aproximado del costo del software. Para ello Áncora utiliza dos opciones de modelado Entidad - Relación (E-R) y Objetos Semánticos (OS).

Dependiendo del paradigma de desarrollo de software que se quiera seguir, se puede optar por E-R u OS, es decir, si se desea modelara la base de datos relacional es mejor utilizar E-R, pero si su base de datos será orientada a objetos utilizar OS será la mejor opción.

En esta sección se explica como formar un primer modelo del modelo de datos ya sea utilizando E-R u OS.

3.2.3.1 Modelo de datos con E-R.

En Áncora las entidades del modelo E-R son los papeles, utensilios automatizables (archivos, catálogos) y reportes generados por el software que representa el guión.

En el ejemplo de la Figura 3-3 se muestran las varias entidades del sistema de Agenda de Compromisos. Por ejemplo existe la entidad Agente o Secretaria (una persona) que está relaciona, mediante la relación binaria modifica con Agenda, misma que la puede llevar varias (m) personas, además cada uno de estas m personas pueden introducir n mensajes a la Agenda.

Para facilitar la lectura de las relaciones, en el caso que se sepan los valores n ó m de la cardinalidad de la relación se pueden poner. Ejemplo la relación entre año y mes, es claramente 1:12.

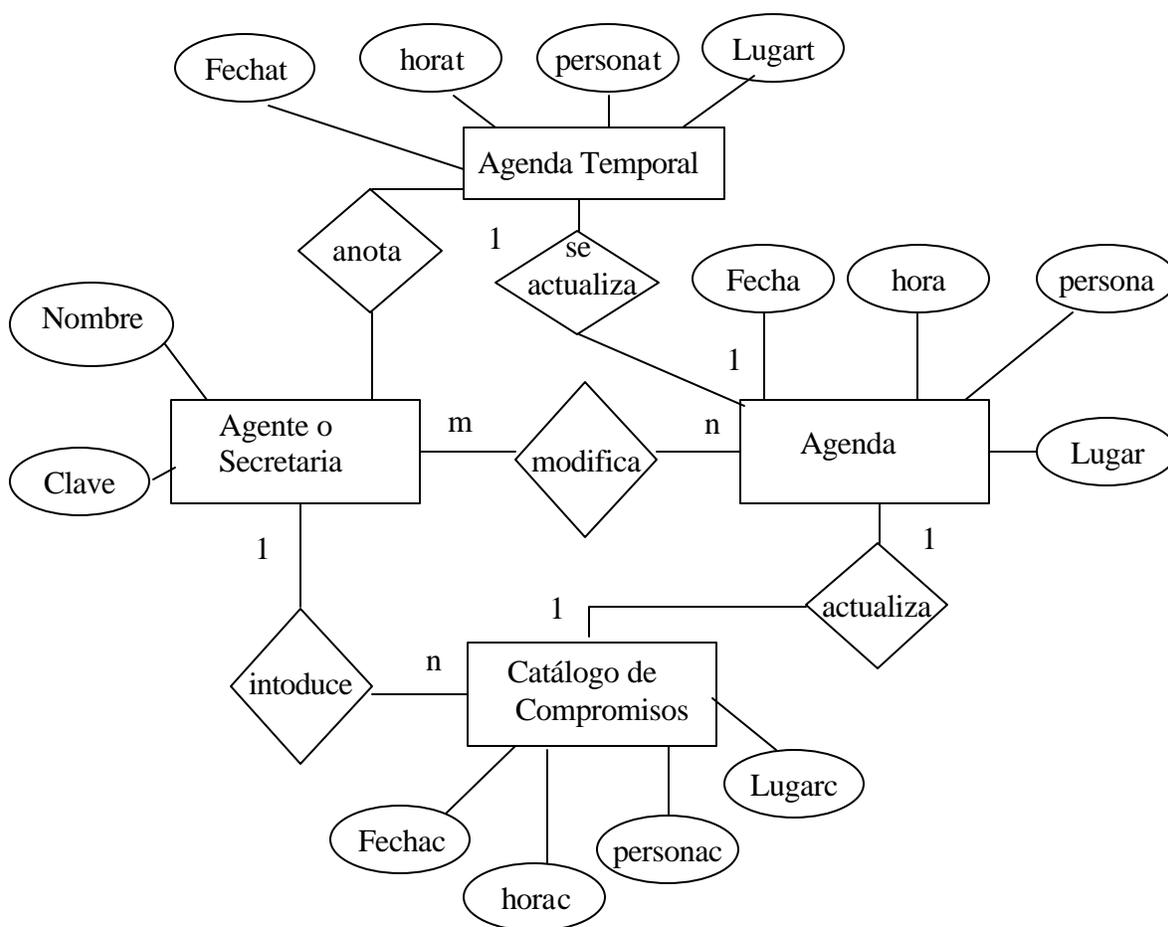


Figura 3-3. Diagrama Entidad-Relación correspondiente al Guión de la propuesta computacional del sistema de Agenda de Compromisos.

3.2.3.2 El modelo de datos con Objetos Semánticos.

Alternativamente al modelo E-R, resulta recomendable utilizar el modelo de Objetos Semánticos (OS) si se piensa continuar con el desarrollo utilizando el paradigma OO. En esta subsección se muestra como pasar de los Guiones de Áncora al MOS.

Tanto en los Guiones como en el prototipo rápido ya se han determinado los resultados que el software debe brindar, así que siguiendo el modelado top-down de los objetos semánticos se pueden formar los objetos semánticos para el modelo de datos de la siguiente manera:

1. Se listan los reportes y pantallas que aparecen en el guión y el Manual preliminar del Usuario.
2. Para cada reporte y pantalla se crea un objeto semántico cuyas propiedades pueden ser valores simple u otros objetos. Debe cuidarse que los objetos no hayan sido creados previamente, pues no tendría sentido repetirlos.
3. Las propiedades para cada objeto semántico se obtienen del Manual Preliminar del Usuario y de los objetos semánticos definidos previamente.

Cabe aclarar que al crear al modelo OS puede presentarse la necesidad de definir más detalles en el Manual Preliminar del Usuario, lo cual resulta provechoso para entregar un prototipo más completo.

Un ejemplo de MOS sería el de la Figura 3-4 referente a la Agenda de Compromisos, comprende sólo dos objetos, pues Catálogo de Compromisos y Agenda temporal tienen las mismas propiedades.

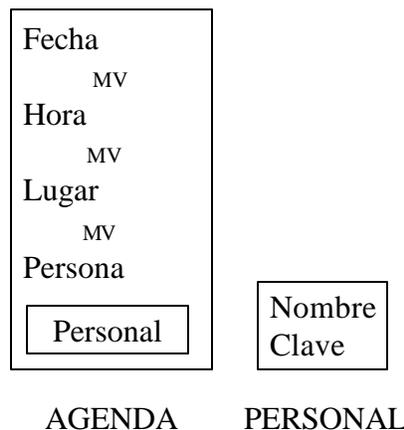


Figura 1-4. Modelo de Objetos Semánticos para el Software Agenda de Compromisos.

3.2.4 Bitácora de desarrollo del Software.

Como una guía para el desarrollo del sistema de software, se elaborará una bitácora en forma de tabla que contiene una entrada por cada quinteta diferente, a la cual se le agregará una columna con una serie de operaciones a realizar, otra que indica el tiempo de elaboración propuesto y otra con el tiempo de elaboración

real (mismo que se llenará al terminar el software). La última columna servirá para ir guardando la historia de tiempos de elaboración y poder así realizar un mejor cálculo cada vez.

La columna con las operaciones corresponde a un refinamiento de la quinteta, donde se agrega detalle de la operación en términos funcionales, dejando la implementación para después.

Esta bitácora es, en sí misma, una guía para el diseño del sistema y para realizar pruebas de sistema, aún cuando no especifica la manera de realizar tales etapas. En la Tabla 3-2 se muestra un ejemplo de bitácora de pruebas correspondiente al ejemplo de agenda de Compromisos.

Tabla 3-2. Bitácora de pruebas funcionales

Requerimiento (quinteta)	Forma de Comprobación	Tiempo propuesto
P anota en K compromiso permanente	P puede ver en K que aparece el compromiso y cada vez que se genere la A anual el compromiso permanente anotado en K estará incluido en A.	2 días
P genera A cada principio de año	P da la orden al software Agenda y éste genera un archivo con los días del año. Cada día tendrá 10 entradas cada una corresponderá a una hora. En A aparecerán todos los compromisos anotados en K.	2 días
P imprime A con II	En la impresora debe empezar a imprimirse la Agenda de la persona con los compromisos ya anotados.	2 días
P visualiza fecha en A	Al oprimir el botón de visualizar una fecha, la máquina sugerirá la actual y esperará a que el usuario escriba la fecha deseada. El software mostrará la fecha deseada con una hora específica (si se tecléo) o todas las horas correspondientes a ese día.	2 días
P pide re - acomodar compromisos	El software permitirá que el compromiso existente en la Agenda pueda moverse a otro momento y lo mostrará donde quedó.	2 días
P anota compromiso en H o At	Si el compromiso fue anotado en At, éste tendrá una nueva entrada misma que se pasará a la Agenda en otro momento.	1 día
P actualiza A con At	El software pasará todos los elementos de At a A usando un proceso batch.	2 días
P revisa compromisos en A	La máquina sugerirá la fecha actual y esperará a que el usuario escriba la fecha deseada. Después desplegará los compromisos que existen para ese día.	2 días
Tiempo total de desarrollo y prueba del sistema		15 días

3.2.4.1 Revisión de la bitácora de desarrollo.

Una vez que se tiene la bitácora de desarrollo, conviene aplicar dos verificaciones que ayudarán en la mejor definición del software y especialmente al llegar a las pruebas:

1. Revisar consistencia entre quintetas relacionadas. Por ejemplo, la segunda quinteta de la Tabla 3-2 pone un límite de diez compromisos por día, pero en la primera, donde se prepara el catálogo de compromisos fijos, no se indica, creando un problema potencial que aparecerá en las pruebas o en la ejecución.
2. Revisar los verbos dentro del refinamiento de cada quinteta, de modo que correspondan a acciones completas; por ejemplo, la quinteta tres pide observar que se **comience** a imprimir, cuando debería indicar que se **termine** de imprimir.

La Tabla 3-3 muestra algunas quintetas de la misma bitácora después de una revisión (solo las primeras columnas).

Tabla 3-3 Quintetas revisada

Requerimiento (quinteta)	Refinamiento de la quinteta (operaciones a realizar)
P anota en K compromiso permanente	P registra un compromiso en K y recibe confirmación cuando se termina; a lo más puede registrar diez compromisos por fecha.
P genera A cada principio de año	P da la orden al software Agenda y éste genera un archivo con los días del año. Cada día tendrá 10 entradas cada una corresponderá a una hora. En A aparecerán todos los compromisos anotados en K.
P imprime A con II	En la impresora debe terminar de imprimirse la Agenda de la persona con los compromisos ya anotados.
P visualiza fecha en A	Se solicitará visualizar una fecha y una hora y el software mostrará la fecha deseada con una hora específica o todas las horas correspondientes a ese día si no se especificó; si no se indica fecha tomará la del día corriente.

3.2.5 Primera estimación del costo del nuevo software.

En el desarrollo de un nuevo software hay varios costos que considerar, entre ellos están: el hardware necesario para que corra el software, el del software con el que se desarrollará el nuevo sistema de software y el costo por desarrollo de éste, es para este último que se utiliza una adaptación del método de Puntos de Función descrito en el capítulo anterior y que se aplica como sigue.

En el caso de los archivos, su grado de dificultad se hace patente cuando realizamos el diagrama E-R y OS (ver Figuras 3-3), del diagrama se deducen a

grandes rasgos los archivos que contendrá el sistema y se le asigna su tipo de indicador (función de datos) y grado de dificultad como en el ejemplo de la Tabla 3-4. Debe notarse que a pesar de que la relación modifica es m:n no se considerará como un archivo, pues al usuario no le interesa saber quién anotó el compromiso, sólo desea controlar la entrada al sistema a través de clave.

Tabla 3-4. Identificación del grado de dificultad de Archivos.

Archivo	Indicador	TRE	TDE	Dificultad
Agente o Secretaria	ALI	1	2	Simple
Agenda	ALI	1	4	Simple
Agenda temporal	ALI	1	4	Simple
Catálogo de compromisos fijos	ALI	1	4	Simple

Para determinar la dificultad de cada función se debe hacer una Tabla de Dificultades (ver ejemplo en Tabla 3-5) con la lista de funciones y su forma de comprobación; esta última permitirá asignar el tipo de indicador (función transaccional) y la dificultad a cada función, misma que se agrega como una cuarta columna de la Tabla de Dificultades.

Tabla 3-5. Dificultad de los requerimientos del software Agenda de Compromisos.

Requerimiento (quinteta)	Tipo de indicador	RA	TDE	Dificultad
P anota en K compromiso permanente	EE	1	4	Simple
P genera A cada principio de año	EE	2	8	Mediano
P imprime A con II	CE	1	4	Simple
P visualiza fecha en A	CE	1	4	Simple
P pide reacomodar compromisos	SE	1	4	Simple
P anota compromiso en H o At	EE	1	4	Simple
P actualiza A con At	EE	2	8	Mediano
P revisa compromisos en A	CE	1	4	Simple

A continuación se deben realizar la contabilidad de Puntos de Función sin ajustar basados en la 3-6.

Tabla 3-6. Puntos de Función sin ajustar.

Indicador	simple	mediano	Complejo	SUMA
Archivos Lógicos Internos	4 * 7	0 * 10	0 * 15	28
Archivos de Interfaz Externa	0 * 5	0 * 7	0 * 10	0
Entradas Externas	2 * 3	2 * 4	0 * 6	14
Salidas Externas	1 * 4	0 * 5	0 * 7	4
Consultas Externas	3 * 3	0 * 4	0 * 6	9
			T =	55

Ahora deben aplicarse los modificadores. Para ello se hace una tabla con varios de los elementos de los Guiones de la propuesta computacional, éstos son:

utensilios, condiciones de entrada y de salida; y las restricciones que sobre ellas hayan marcado los involucrados. En la Tabla 3-7 se muestra el ejemplo que corresponde al software Agenda de Compromisos.

Tabla 3-7. Relación de elementos del software y sus restricciones.

Elementos del Software	Restricciones
Archivo Agenda	Compartido de lectura y escritura sobre ambiente monousuario y multitarea.
Archivo Agenda Temporal	Compartido de lectura y escritura sobre ambiente monousuario y multitarea.
Computadora	Personal con sistema Windows 95 de Microsoft, 32 Mb de RAM, 1.2 Gb disco duro.
Hoja cualquiera	Ninguna
Catálogo de compromisos permanentes	Compartido de lectura y escritura sobre ambiente monousuario y multitarea.
Impresora	LasserJet4 Hp
P necesita controlar sus compromisos	Sólo se permite la entrada con contraseña.
S puede ayudar a P	Sólo se permite la entrada con contraseña.
P lleva su control	Ninguna
S ayudó a P	Ninguna

Después de haber marcado las restricciones y formas de comprobación se está en posibilidades de asignar el grado de influencia a cada modificador de Puntos de Función como se ve en la Tabla 3-8.

Tabla 3-8. Modificadores del sistema de Agenda de Compromiso.

Modificador	Grado de influencia
Comunicación de datos.	0
Procesamiento Distribuido de Datos.	0
Rendimiento.	0
Configuración Altamente Usada.	1
Promedio de Transacciones.	1
Entrada de Datos en Línea.	5
Eficiencia para el Usuario Final.	5
Actualización en Línea.	1
Procesamiento Complejo.	0
Reusabilidad.	0
Facilidad de Instalación.	1
Facilidad de Operación.	1
Varios Sitios.	0
Facilidad de Cambios	0
M=	15

El paso final será aplicar la fórmula como sigue:

$$pf = T * (0.65 + 0.01 * M) = 55 * (0.65 + 0.01 * 15) = 44$$

Se han obtenido 44 puntos de función; puestos así no tiene mucho sentido, pero, se puede jugar con esto puntos de función y ponerlos en términos de moneda y/o tiempo como sigue.

Ejemplo. Si se quieren poner los puntos de función en términos de pesos, se hace lo siguiente: los puntos de función deben multiplicarse por 100 Dólares (USA) y eso dará el Costo de desarrollo. Faltaría adicionar las ganancias, que se propone sea una cantidad igual a la del costo de desarrollo del software.

3.3 Solución de Conflictos, Jerarquía y Validación de requerimientos de software.

Estas etapas se cubren en la RRD que se llevará a cabo y donde se espera:

- i. la clasificación de los requerimientos marcada por las escenas de la propuesta computacional se corrobore o corrija,
- ii. establecer prioridades de requerimientos (las escenas),
- iii. aceptar o modificar la asignación de responsabilidades de las partes involucradas,
- iv. resolver conflictos y
- v. validar la propuesta computacional.

Sin embargo, es probable que bajo la presión de llevar una RRD exitosa, al analista se le hayan escapado algunos puntos, por ello es recomendable que se sienta con los resultados de la RRD y los revise junto con el cliente.

3.3.1 La Técnica de Grupo Nominal en la RRD.

La Técnica de Grupo Nominal (TGN), en Áncora, se desarrolla para una reunión estructurada de grupo para la jerarquización de requerimientos, llevándose a cabo, en la Reunión de Reflexión y Diseño, como sigue:

1. Se entregan n tarjetas (n se sustituye por el número de escenas que se pusieron en el Guión de la propuesta computacional) a cada participante para que registren en cada una de ellas descripción de algún requerimiento para el nuevo software.
2. Se solicita marcar con el número n aquella tarjeta que contiene el requerimiento que cada uno considere más importante.
3. La tarea menos importante se califica con el número 1.
4. Las otras n - 2 tarjetas se califican en orden de importancia de manera decreciente.
5. Al finalizar el llenado de las tarjetas los participantes pasan a escribir en una hoja de rotafolio su asignación de valores. Al finalizar el llenado de la hoja, se suman los valores asignados por todo el grupo de personas y se asignan las prioridades; siendo la de mayor prioridad la que haya obtenido mayor puntuación.

3.3.2 Reunión de Reflexión y Diseño con los involucrados.

Antes de que se lleve a cabo la RRD se deben preparar los siguientes puntos:

1. Un calendario de actividades que puede ser como el propuesto en la sección de la RRD.
2. Si existen problemas de actitud de parte de los usuarios deben agregarse actividades de motivación consistentes en enseñar sistemas equivalentes y las ventajas que el nuevo software traerían a la empresa y sus empleados.
3. Preparar a los facilitadores haciendo hincapié en los siguientes puntos:
 - 3.1. Un facilitador no debe intervenir en la esencia de la discusión.
 - 3.2. Proporcionar y mantener una atmósfera de confianza y respeto en el grupo.
 - 3.3. Motivar la participación de todos los miembros del grupo.
4. Tener los documentos y materiales de apoyo, mostrados en la Tabla 3-9, para llevar una RRD ordenada y efectiva:

Tabla 3-9. Documentos y materiales necesarios para la RRD.

Documento	Contenido
1. Contexto y situación actual.	Metas de la empresa. Organigrama de la empresa. Guión y diálogos de la situación teórica y actual. Justificación del nuevo software. Glosario.
2. Propuesta computacional.	Guión y diálogos propuestos. Manual de operación. Modelo Entidad – Relación. Beneficios del nuevo software
3. Restricciones y responsabilidades.	Identificadores. Plan de prueba funcional. Restricciones. Costo en términos de puntos de función. Tabla de responsabilidades.
4. Otros materiales.	Calendario y plan de actividades de la RRD. Tarjetas para la asignación de responsabilidades. Cartel de reglas de comportamiento. Cartel de puntos a tratar. Cartel de la lista de escenas para la asignación de prioridades. Hojas de rotafolio, plumones y rotafolio. Fotocopia de los documentos 1 a 3 para cada involucrado. Oficio de acuerdos firmado por los involucrados que asistieron a la RRD. Lápices y bolígrafos. Café, té, galletas, azúcar, crema, agua, cafeteras, vasos térmicos, cucharillas, servilletas, etc.

Al finalizar la RRD se deben tener todos los documentos de la Tabla 3-9, modificados y aprobados, además de hojas de rotafolio y acuerdos a los que se llegaron que se deben considerar en el software.

3.4 El reuso en Áncora.

El reuso da una manera de realizar un rápido análisis de requerimientos, que además garantiza una calidad estándar de los mismos, ya que los elementos a reusar han sido previamente probados. Siempre se debe tener en mente que hay varios elementos que se pueden reusar:

- i. Guiones para sistemas semejantes.
- ii. Pistas para subsistemas semejantes.
- iii. Escenas para módulos semejantes.
- iv. Quintetas para funciones u operaciones iguales.
- v. Prototipo rápido (manual de operación).
- vi. Costos en tiempo y dinero.

En el caso que el analista realice varios sistemas para una misma empresa se pueden reusar las definiciones de contexto organizacional y del hardware y software existente.

Antes de proseguir es necesario aclarar qué se entiende por semejante e igual en los elementos mencionados arriba.

- i. Una función f1 es igual a otra función f2, si y sólo si, el verbo que describe las quintetas de f1 y f2 son sinónimos y las quintetas tienen la misma cantidad de elementos.
- ii. Un módulo M1 es semejante a otro módulo M2, si y sólo si, los verbos utilizados para describir las escenas de M1 y M2 son mapeados a la misma primitiva.
- iii. Un subsistema S1 es semejante a otro subsistema S2, si y sólo si, los verbos utilizados para describir las pistas que representan a S1 y S2 son mapeados a la misma primitiva y las condiciones de entrada y salida pueden ser mapeadas a las mismas primitivas.
- iv. Una Sistema de Software SS1 es semejante a otro Sistema de software SS2, si y sólo si, los verbos utilizados para describir sus pistas son mapeados a las mismas primitiva y las condiciones de entrada y salida pueden ser mapeadas a las mismas primitivas.

También se define:

- i. Primitiva. Verbo que representa una acción generalizada; esto es, una primitiva resume varios verbos, en la Tabla 3-10 se pueden ver varias primitivas útiles en la elaboración de sistemas de software, pero el analista puede ir creando sus propias primitivas.
- ii. Sustantivos genéricos. Se trata de los papeles o utensilios que aparecen en un Guión, como los que se muestran en la Tabla 3-11.

Un ejemplo del uso de primitivas y sustantivos genéricos sería el siguiente:

Suponga la existencia de dos sistemas de software, uno de nómina que debe tener un catálogo de pagos el cual debe modificarse periódicamente según las prestaciones de la empresa, otro de control escolar que tiene un catálogo de materias que debe irse renovando dependiendo de los planes

de estudio o de los profesores disponibles. Debe notarse que se utilizaron dos verbos diferentes (modificar y renovar) para significar la misma acción general, indicador o primitiva: ACTUALIZAR y se mencionaron dos catálogos (pagos y materias) que vienen a ser el sustantivo genérico ALI.

Tabla 3-10. Primitivas y sus verbos.

Primitiva	Verbos a mapear
ACTUALIZAR	Actualizar, renovar, modificar, modernizar, remozar, anotar, controlar.
REUNIR	Unir, ligar, importar, incluir, anexar.
INGRESAR	Ingresar, capturar, leer, percibir, digitalizar.
CREAR	Procesar, calcular, crear, generar, construir.
CONSULTAR	Consultar, desplegar, emitir, imprimir.
SELECCIONAR	Seleccionar, elegir

Tabla 3-11. Los sustantivos genéricos

Jefe	Director, dueño, gerente, encargado, jefe
Auxiliar	Secretaria, capturista, empleado, cajero, auxiliar
Penumbra	Cliente, comprador, tarjeta - aviente, votante, usuario
ALI ⁵ (Archivo Lógico Interno)	Archivo maestro, catálogo, archivo temporal, archivo de transacciones internas.
AIE (Archivo de Interfaz Externo)	Archivo de transacciones externas, lecturas de datos de control.
FORMA	Hoja, factura, boleta, recibo.

Para poder reusar primero hay que guardar los elementos de reuso de una forma que sirva en la siguiente sección se explica cómo Áncora propone guardar.

3.4.1 Guardando en Áncora.

Una vez que se ha terminado la Especificación de Requerimientos de Software es posible guardar todos aquellos elementos que puedan llegar a servir en otros sistemas, en Áncora se propone:

- 1) Convertir el Guión del sistema en un Guión abstracto, para ello se procede como sigue:
 - a) Papel → nombre genérico
 - b) Útil → ali_j , aie_j , $impresora_k$, (nombres genéricos)
 - c) Escenas → nombre genérico y verbo mapeado
 - d) Quinteta → $verbo_mapeado(p_y, u_z, u_x, v_t_p)$.
 - e) Condiciones_entrada → $verbo_mapeado(p_i, u_j)$
- 2) Guardar el sistema bajo cuatro encabezados: papeles, escenas, condiciones de entrada y utensilios.

Ejemplo: el sistema Agenda de Compromisos se guardará con los encabezados siguientes: Para papeles (Jefe, Auxiliar), para utensilios (ALI1, ALI2, computadora1, forma1, ALI3, impresora1), condiciones de entrada serían

⁵ Tanto ALI como AIE son nombres tomados del modelo de Puntos de Función.

(Actualizar(jefe, ALI1) y Actualizar(auxiliar, ALI1)), en el caso de las escenas sería como en la Figura 3-5.

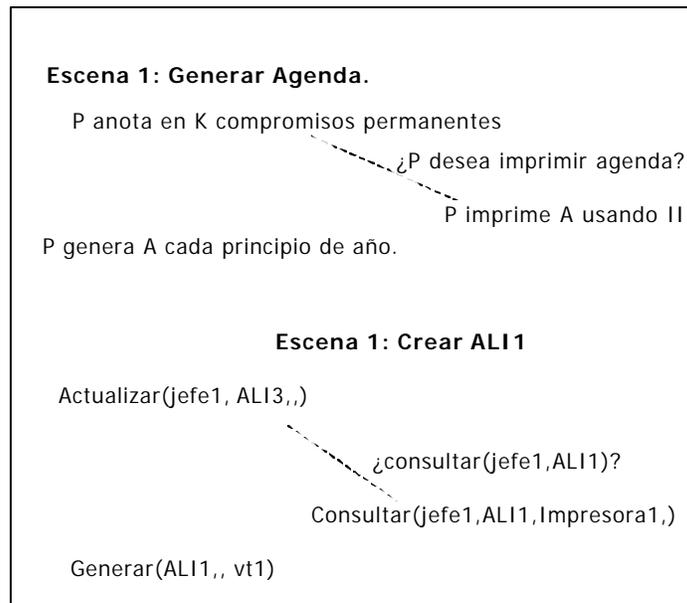


Figura 3-5. Mapeo de la Escena 1 del Guión Agenda de Compromisos

3.4.2 Reusando con Áncora.

Resulta común que en el momento en que se está realizando un nuevo software, el analista piense “esto de parece al sistema de software X” y que cuando llegue a su escritorio saque la documentación del software X y la revise buscando los puntos comunes.

Si la forma en que guardamos la documentación de sistemas anteriores resulta sistemática, se podrán reusar varios elementos, como los que se explicaron en la sección anterior.

Así, si se desea conocer el contenido de todo un sistema semejante, se buscarán los encabezados de sistemas anteriores por papeles, utensilios, escenas sueltas o condiciones de entrada. Si dos de los cuatro posibles encabezados coinciden, entonces se revisan los sistemas coincidentes y se ve si sirven. Con lo anterior se podrán entonces reusar los elementos de los sistemas que coincidan.

Ejemplo: Considere que se requiere un software que ayude al encargado y/o auxiliar a llevar el control por computadora de las existencias de los productos de una ferretería y sea capaz de indicar cuándo y en qué cantidad pedir nuevos productos del catálogo de productos para mantener un inventario adecuado, mismo que debe imprimirse mensualmente. Para llevar a cabo la actualización de los productos, se tiene una lista acumulada de los productos vendidos, misma que se actualiza con las facturas hechos al vender un producto.

Si se cambia a primitivas las partes subrayadas se tendrá (Actualizar(jefe, ALI) y Actualizar(auxiliar, ALI)) como condiciones de entrada y como utensilios (inventario de productos, catálogo de productos, computadora, impresora, lista acumulada de los productos vendidos, las facturas) o sea (ALI1, ALI2, computadora1, impresora1, ALI3, forma1). Como se puede notar ya se tienen dos encabezados que coinciden con el sistema de Agenda de Compromisos, por lo cual se puede pensar en reusarlo.

3.5 Cierre de Áncora.

Son varios los pequeños o grandes detalles que todavía quedan por cubrir para dar por terminado el Análisis de Requerimientos de Software (ARS). Son dos las actividades que esta etapa considera:

1. Cierre de detalles.
2. Conexión con el diseño.

3.5.1 Cierre de detalles.

Pensar en un Análisis de Requerimientos de Software (ARS) completo es utópico por varias razones:

1. El usuario puede pensar en nuevos requerimientos al ver el avance de los modelos.
2. La necesidad de terminar rápidamente con la etapa de ARS para empezar con las siguientes.
3. Falta de dinero para cubrir todas las necesidades.

Por lo anterior, cada que se lleva a cabo la RRD es necesario decidir un cierre temporal de los modelos del ARS. Para ello se deben redondear los puntos pendientes de la RRD mediante la comunicación vía oficios o Internet con usuarios.

Tal vez sea necesario reabrir la etapa de ARS, pero en Áncora esto no es problema

3.5.2 Conexión con el diseño.

Áncora brinda algunas pautas para trasladar el análisis de requerimientos a tres paradigmas diferentes utilizando una de sus metodologías representantes:

1. Funciones (Metodología Estructurada Moderna de Yourdon).
2. Objetos (la basada en UML),
3. Estados (Jackson System Development Method)

3.5.2.1 Metodología Estructurada Moderna de Yourdon.

En el análisis de la Metodología Estructurada Moderna de Yourdon (MEMY) existen varios modelos: Esencial, Ambiental y de Implantación del Usuario. Como recordatorio de lo que son los modelos de la metodología de Yourdon se tiene la lista siguiente:

1. Modelo Esencial (ME). Es un modelo de lo que el sistema debe hacer para satisfacer los requerimientos del usuario.

2. Modelo Ambiental (MA). Debe definir los elementos que son del sistema y los que no lo son.
3. Modelo de Implantación del Usuario (MIU). Deben definirse las interfaces del sistema con el medio ambiente que lo rodea y se realiza de manera conjunta de le analista con el diseñador.

En esta metodología una de las herramientas más utilizadas son los Diagramas de Flujo de Datos que consisten en círculos que representan funciones, rectángulos para representar elementos terminales y dos rayas paralelas⁶ con una frase nominal en medio de ellas para los almacenamientos. En la Tabla 3-12 se muestra la relación de los DFD de MEMY con los Guiones de Áncora y en la 3-6 aparece el DFD correspondiente a la escena Anotar Compromiso de la propuesta computacional de la Figura 2-4 y que corresponde a una parte del Modelo Esencial de MEMY.

Los diferentes modelos de MEMY sugieren productos a obtener a través de una serie de pasos. Una lista más de los productos de MEMY y los de Áncora se muestra en la Tabla 3-13. Al terminar de utilizar las Tablas 3-12 y 3-13 se tendrá la traducción al MEMY y elementos de diseño que permitirán continuar con el enfoque estructurado para el desarrollo del sistema.

Tabla 3-12. Conexión de los Guiones y los DFD.

Elementos de los Guiones de Áncora	Elementos de MEMY
Nombre de Guión	Nombre del sistema de Software
Nombre de Pista	Nombre de subsistema de software
Papeles	Símbolos terminales del DFD
Utensilios automatizables	Símbolos de almacén del DFD
Utensilios físicos (Ejemplos: Lápiz, engrapadora, etc.)	No están considerados.
Utensilios que sean formatos de datos	Símbolo terminal
Condiciones de entrada	Estímulos de fuera sistema
Condiciones de salida	Símbolos terminales o de almacén
Escenas	Símbolo de función del modelo esencial.
Acciones de las quintetas	Símbolo de función en el modelo de implementación del usuario.

⁶ Puede variar a un rectángulo sin un lado o un rectángulo con puntas redondeadas.

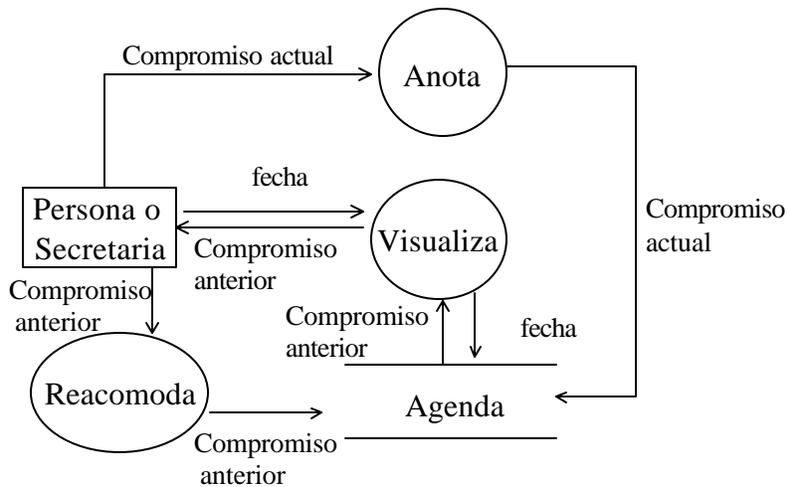


Figura 3-6. DFD de la Escena Anotar Compromisos

Tabla 3-13. Relación de los elementos de Áncora y del MIU de MEMY.

Modelo de MEMY	Metas del MEMY	Apoyos de Áncora
ME	DFD de temas separados y tamaños medianos.	Aplicar Tabla 3-12 sobre el Guión de la propuesta computacional para cada Escena.
MA	Enunciado del propósito del sistema.	Justificación y beneficios del software.
MA	Diagramas de contexto. DFD que incluye personas, datos, sistemas que entran o salen e interactúan con el sistema.	Realizar DFD que utilice las Condiciones de Entrada y Salida, los papeles y utensilios del Guión de la propuesta computacional.
MA	Lista de eventos.	Condiciones de entrada.
MIU	Determinar metas de automatización.	Guión de propuesta computacional.
MIU	Determinar Interfaz del usuario	Manual de operación.
MIU	Identificar actividades manuales para el soporte del sistema.	Manual de Operación y diálogos de quintetas.
MIU	Especificar restricciones operacionales.	Modificadores de Puntos de Función.

3.5.2.2 Metodologías basadas en UML.

UML (del inglés Unified Modeling Language) es un lenguaje de modelado que es utilizado para especificar, visualizar, construir y documentar artefactos de un sistema de software. Con UML se puede, además de otros aspectos, expresar el comportamiento dinámico (modelo dinámico) e información de la estructura estática (modelo estático) de un sistema.

En este apartado sólo mostraremos la equivalencia de Áncora con los siguientes diagramas:

- i. Casos de Uso.
- ii. Diagramas de secuencia
- iii. Diagramas de clases.
- iv. Diagramas de actividades.

Para empezar a realizar los diagramas se aplica la Tabla 3-14 en la cual se identifican los elementos equivalentes entre Áncora y UML.

Tabla 3-14. Equivalencias entre Áncora y UML

Elementos de Áncora	Modelo Dinámico de UML	Modelo Estático de UML
Escena	Caso de uso	
Papel	Actor	Clase
Utensilio automatizable	Clase entidad	Clase
Verbo de una quinteta	Clases de control	Operación de la clase Asociación del papel con el utensilio
Utensilio de las condiciones de entrada y salida	Clase interfaz o frontera	

Tabla 3-15. Vistas y Diagramas de UML

Área	Vista	Diagramas	Conceptos principales
Estructural	Estática	Clases	Clase, asociación, interfaz, generalización, dependencia, realización.
	Casos de Uso	Casos de Uso	Caso de uso, actor, asociación, extensión, inclusión.
	Implantación	Componentes	Componente, interfaz, dependencia, realización.
	Despliegue	Despliegue	Nodo, componente, dependencia, realización.
Dinámica	Máquina de estados	Gráfico de estados	Estado, evento, transición, acción.
	Actividades	Actividades	Estado, actividad, transición terminal, bifurcación, reunión.
	Interacción	Secuencia	Interacción, objeto, mensaje, activación.
Colaboración		Colaboración, interacción, papel de colaboración, mensaje.	
Modelo Administrativo	Modelo Administrativo	Clases	Paquete, subsistema, modelo.
Extensibilidad	Todos	Todos	Restricciones, estereotipo, valores etiquetados.

De los Guiones de Áncora a los Casos de Uso.

Pasar de los Guiones a los Diagramas de Casos de Uso es tarea sencilla y se procede como sigue:

1. Cada escena es un Caso de Uso.
2. Cada papel es un actor.
3. Otros actores son: los AIE

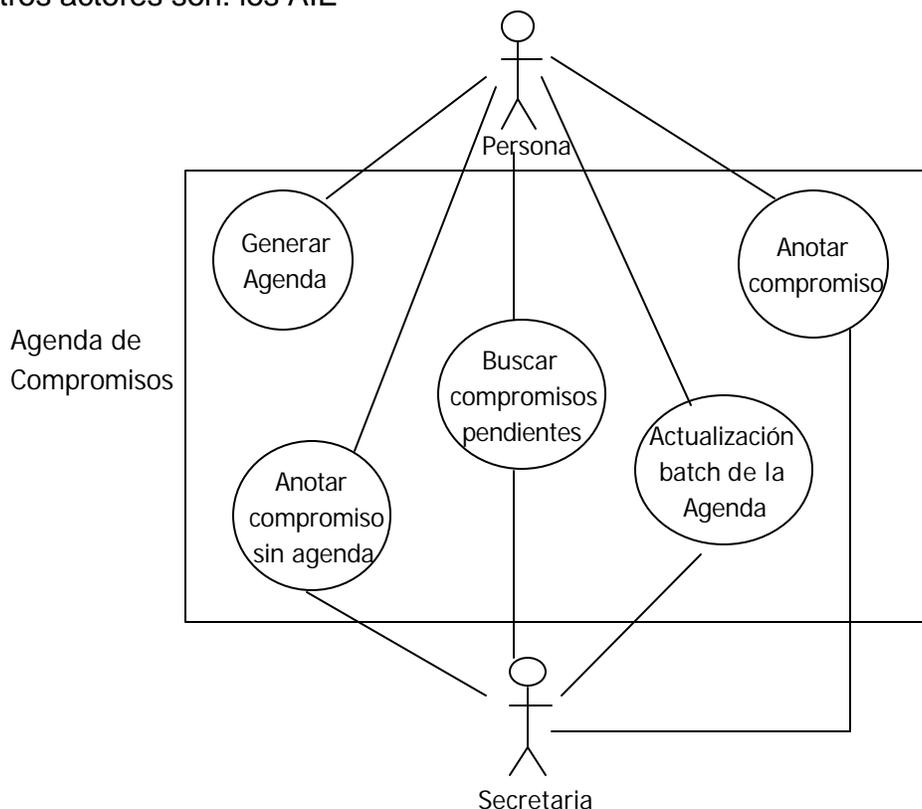


Figura 3-7. Casos de Uso del Sistema Agenda de Compromisos.

Dentro de la vista de Casos de Uso, Objectory propone también los diagramas de relación de casos de uso que sirven para mostrar cómo cada caso interactúa con los demás y para detallar los casos de uso particulares. Los tipos de relación que existen entre los casos de uso son de tres tipos:

- a) Asociación. Camino de comunicación entre los actores y Casos de Uso. Como ya se vió en la Figura 3-7 se representa con una raya continua sin flecha.
- b) Include. Relación entre Casos de Uso que incluye comportamiento adicional al Caso de Uso base de manera explícita. Se representa con una flecha de línea punteada y dirección hacia afuera del Caso de Uso base.
- c) Extend. Inserción de comportamiento adicional en un Caso de Uso base. Se representa con una flecha de línea punteada y dirección hacia el Caso de Uso base.
- d) Generalización de Caso de Uso. Relación entre un Caso de Uso general y uno específico al que hereda y al que se le agregan características. La flecha que se utiliza para la representación es con raya continua, la punta de la flecha

como un triángulo sin relleno y apuntando hacia el Caso de Uso padre o base. Un ejemplo de estos diagramas se muestra en la Figura 3-8.

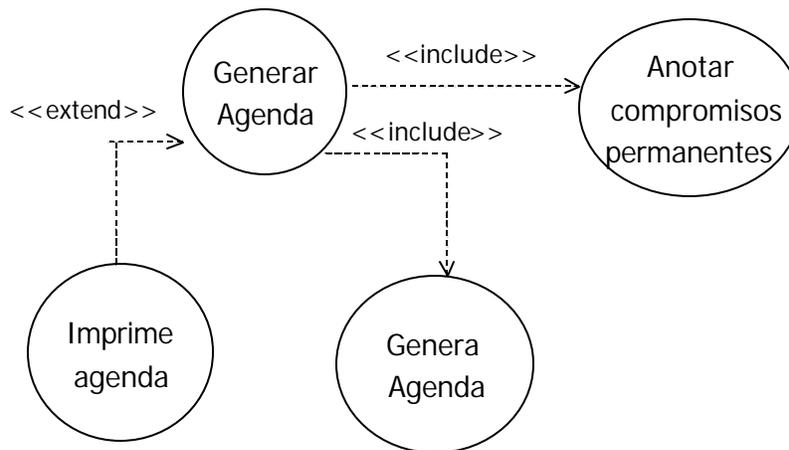


Figura 3-8. Diagrama de Relación de Casos de Uso para el Caso de Uso Generar Agenda.

Del plan de pruebas a los Diagramas de Secuencia de UML.

Un diagrama de secuencia de UML es la representación de una situación particular de una actividad de un sistema de software. En el diagrama aparecen las clases que en esa situación intervienen y los mensajes que entre ellas comparten, la situación de la actividad puede ser o no exitosa.

En Áncora se realiza una bitácora de pruebas de funcionalidad exitosa, entonces la forma de pasar de la bitácora de Áncora a un diagrama de secuencia de UML es casi inmediata, ya que en la bitácora aparecen los papeles y útiles necesarios para llevar a cabo la funcionalidad, y éstas son las clases que en el diagrama de secuencias se requieren. En la bitácora aparece también la forma en que deben comunicarse el software y los usuarios, de esa manera pasan a ser los mensajes entre clases.

Un ejemplo se muestra en la Figura 3-9, donde se ve el paso de mensajes entre las clases para obtener una generación de Agenda exitosa.

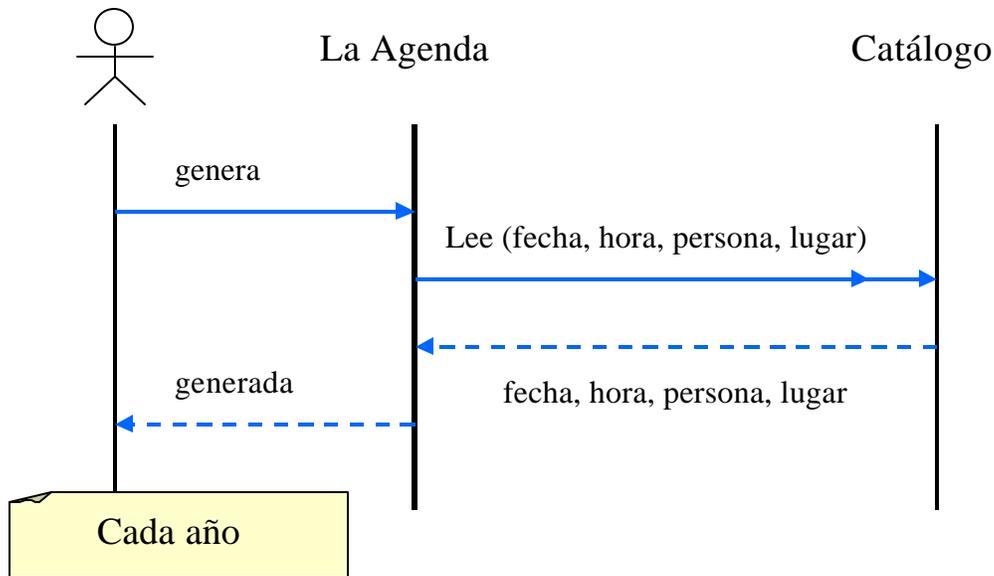


Figura 3-9. Diagrama de Secuencia exitoso del Caso de Uso Genera Agenda.

Del modelo E-R construido en Áncora al diagrama de clases de la vista estructural de UML.

La vista estática de Objectory capturan la estructura de objetos que unifica la estructura de datos y las características de comportamiento. También, la vista estática, describe el comportamiento de las unidades como herramientas de modelado discreto, pero no contienen los detalles de su comportamiento.

El tipo de diagramas de la vista estática empieza con la definición de las clases que consta de un rectángulo con el nombre de la clase, sus atributos y operaciones.

Uno de los problemas es saber cuáles clases se definirán. La conexión con Áncora simplifica el problema y propone la siguiente forma para la definición de clases.

- i. Cada entidad del modelo E-R es una clase, sus atributos ya están definidos en el propio E-R y sus operaciones son aquellas definidas por las quintetas.
- ii. Las relaciones del modelo E-R cuya cardinalidad sea n:m también son clases.

En la Figura 3-10 se muestra la lista de clases del software Agenda Electrónica.

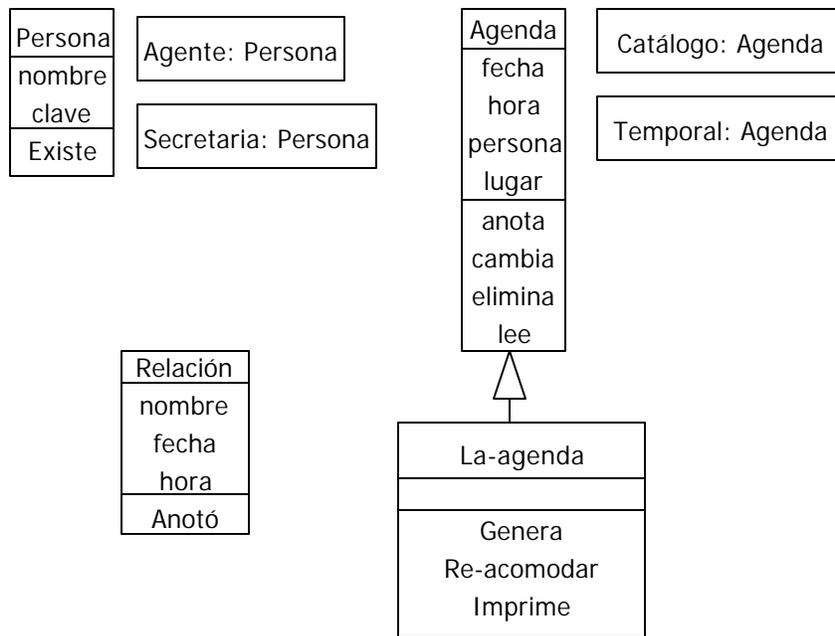


Figura 3-10. Clases e instancias para el software Agenda de Compromisos.

Puede notarse que en la Figura 3-10 también aparecen instancias de clase, éstas son definidas al notar que éstas tienen los mismos atributos y operaciones. Otro detalle es la herencia de la clase Agenda a la clase La-agenda que es expresada así después de ver que sólo la agenda principal tiene las operaciones genera, re-acomodar e imprime.

De los Guiones al Diagrama de Actividades.

Los diagramas de actividades de UML son una representación de los flujos de trabajo dentro del software. Una forma de diagrama de actividades es la de Swimlanes, donde se muestran las actividades de acuerdo a las responsabilidades de los actores. Las partes que conforman un Diagrama de actividades de UML se corresponden a los elementos de un Guión como sigue:

- i. Un actividad (representado como un rectángulo con puntas redondeadas) es una quinteta.
- ii. Los actores son papeles.
- iii. Un diamante es una condición.
- iv. Un fork sirve para representar las actividades paralelas. En los guiones se indica poniendo número de escena repetida un punto y el número de actividad paralela, es decir la notación de subsección en un texto. Ejemplos: 1.2, 5.3, etc.
- v. Un join indica la reunión de actividades paralelas desembocadas en una. En los guiones esto se indica con rayas.

En la Figura 3-11 se muestra el diagrama de actividades swimlane que corresponde a parte del software agenda de compromisos.

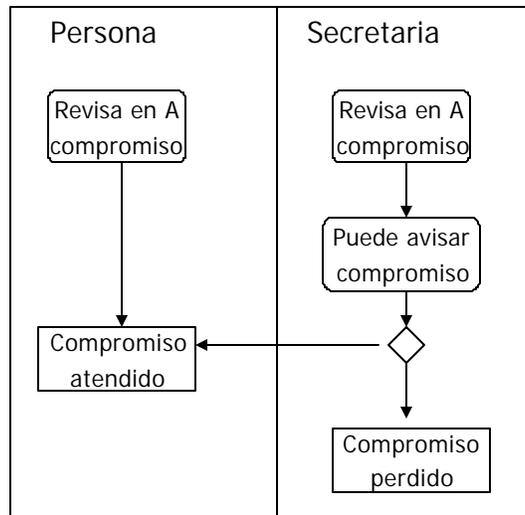


Figura 3-11. Diagrama de Actividades Swimlane del Caso de Uso Buscar Compromisos Pendientes del software Agenda de Actividades.

3.5.2.3 Metodología para el Desarrollo de Sistemas de Jackson (DSJ).

El DSJ se recomienda para sistemas basados en eventos, consta de los pasos de Entidad –Acción, Estructura de Entidad, Modelo Inicial, Funciones, Temporización del Sistema e Implementación. Sólo los tres primeros pasos conciernen al análisis de los sistemas y son los que aquí se explicarán.

Paso Entidad-Acción.

En este paso se identifican las entidades del mundo real y las acciones que suceden en éste. Para lograrlo a partir de los modelos de Áncora basta con nombrar a papeles y utensilios automatizables que correspondan al mundo real como entidades. Los verbos que acompañan a las entidades en las quintetas serán las acciones.

En el caso del ejemplo de la Agenda de Compromisos las entidades serían: secretaria y/o persona dueña de la agenda y agenda de compromisos.

A la lista de entidades y acciones seleccionadas pueden agregársele nuevas si más adelante en el desarrollo se considera necesario.

Paso Estructura de Entidad.

En este paso se utilizan los diagramas de estructura de Jackson para ordenar en el tiempo las acciones que afectan a cada entidad. A partir de Áncora se van poniendo las acciones de cada entidad desde la primera escena en que aparezcan. Como ejemplo se muestra en la Figura 3-12 el diagrama de estructura de la Agenda.

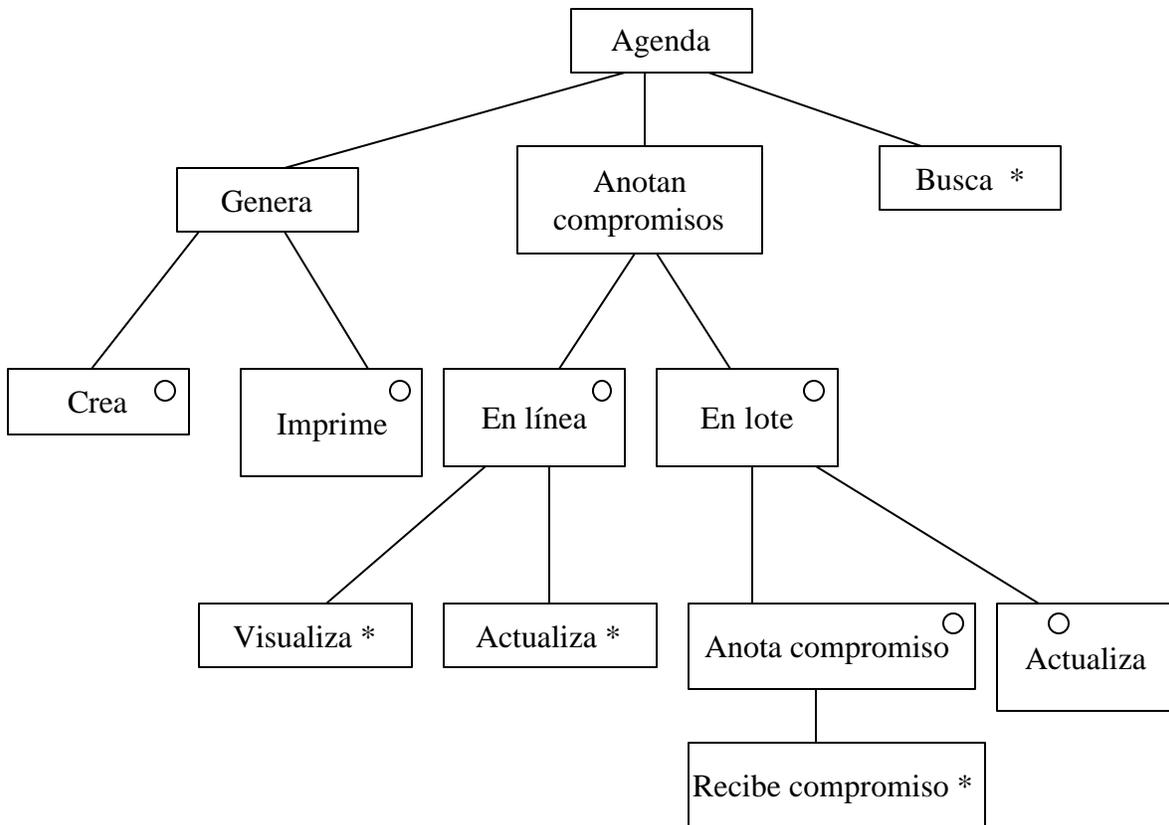


Figura 3-11 Ejemplo de Diagrama de Estructura

Paso del modelo inicial.

En este paso se comienza a definir un modelo del mundo real del sistema mediante los diagramas de especificación del sistema (ver Figura 3-13). Los diagramas tienen dos tipos de conexión entre procesos, la de flujo de datos que se utiliza cuando un proceso genera datos y los envía a otro proceso. El segundo tipo de conexión por vector de estados se utiliza cuando un proceso accede al vector de estado de otro proceso. Un ejemplo de estos diagramas se ve en la Figura 3-13 donde se muestra el proceso anotar compromiso del mundo real unido al proceso actualiza del sistema.

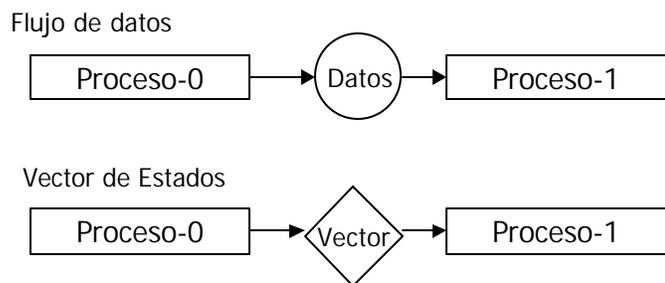


Figura 3-13. Diagramas de Especificación del Sistema.



Figura 3-14. Ejemplo de Diagrama de Especificación del Sistema del proceso real Anotar Compromiso.

4 Glosario.

Término	Significado
Cliente	Persona o grupo de personas que encargan la realización de un nuevo software.
Guión	Estructura que busca representar una situación estereotipada del mundo real.
Diálogo	Detalle escrito en forma de diálogo teatral de alguna quinteta ambigua o que involucra acciones manuales.
Interesado o Involucrado	Se refiere a todas las personas que de alguna manera tienen que ver con la definición de los requerimientos del sistema de software: clientes, usuarios, proveedores de información y expertos en computación.
Quinteta	Representación estructurada de una acción del sistema.
Sistema.	Combinación de procedimientos destinados a producir ciertos resultados. Ocurre en la empresa cliente.
Software o sistema de software	Herramienta automatizada que sirve de apoyo para la consecución de uno o varios procedimientos de un sistema.
Usuario	Persona o grupo de personas que utilizarán el software para actualizarlo y/o consultarlo

Apéndice A. Ejemplos de Guiones.

En este anexo se dan ejemplos de los guiones de sistemas de software realizados con la metodología Áncora. La forma de presentarlos es dando una explicación textual de qué se desea del software y los guiones de situación actual del sistema (si existe) y de propuesta computacional.

A.1 Sistema Conserva.

Los Centros de Datos dependientes de Conservación Internacional tienen la necesidad de crear y mantener bases de datos de diferentes grupos de especies, de manera que, eventualmente, puedan combinarse. Cada centro tiene características especiales, que deben reflejarse en cierta variación de las bases de datos. Anteriormente, se emplearon diversos programas que realizaban parte de la captura, pero que resultaron inadecuados, por lo que se buscó contar con un sistema flexible, capaz de adecuarse a los diferentes centros.

El sistema **Conserva** brindará una solución al problema mencionado, permitiendo la definición, captura y explotación de datos para la conservación de especies.

A.1.1 Sistema Actual.

A continuación se muestra cómo es el sistema actualmente, o mejor dicho cómo se trabaja actualmente en los diferentes centros de apoyo a la conservación de especies. Se hace el guión correspondiente, el cual se ilustra en la Figura A1.

<p><u>Guión:</u> CONSERVA ACTUAL</p> <p><u>Papeles:</u> Investigador Capturista Programador</p> <p><u>Utensilios:</u> Observaciones de campo Datos de las especies</p> <p><u>Condiciones de Entrada:</u> Investigador requiere guardar y categorizar datos Programador puede hacer sistema Investigador tiene dinero para pagar al programador Capturista tiene datos por meter</p> <p><u>Condiciones de Salida:</u> Programador hizo sistema Investigador obtuvo resultados Capturista introdujo datos</p>	<p>Escena: Definir necesidades Investigador pide sistema al programador</p> <p>Programador realiza sistema pedido</p>
	<p>Escena: Introducir datos <u>Investigador entrega sólo los datos que soporta el sistema fijo</u></p> <p>Capturista introduce datos</p>
	<p>Escena: Obtener datos Investigador entra al sistema</p> <p>Investigador consulta datos</p> <p><u>Investigador saca solo los resultados que permite el programa</u></p>

Figura A1 Guión correspondiente a la situación actual

A.1.2 Problemas principales.

Cada sistema realizado en cada centro cubre sólo las necesidades particulares del centro. En cuanto se necesita una modificación hay que volver a contratar programadores, que con frecuencia comienzan desde el principio o les cuesta mucho trabajo entender los programas viejos.

El formato de los archivos de datos de los diferentes sistemas es al gusto del programador y con frecuencia son incompatibles con los de otros centros.

Los formatos de captura resultan inflexibles, ya que si se quiere agregar un dato nuevo, esto no se logra.

A.1.3 Estrategia computacional

Dados los problemas descritos, se decidió realizar un paquete flexible que permita al usuario ir definiendo sus propios datos y metas. La forma de trabajar con un nuevo sistema computacional se muestra en el guión de la Figura A2.

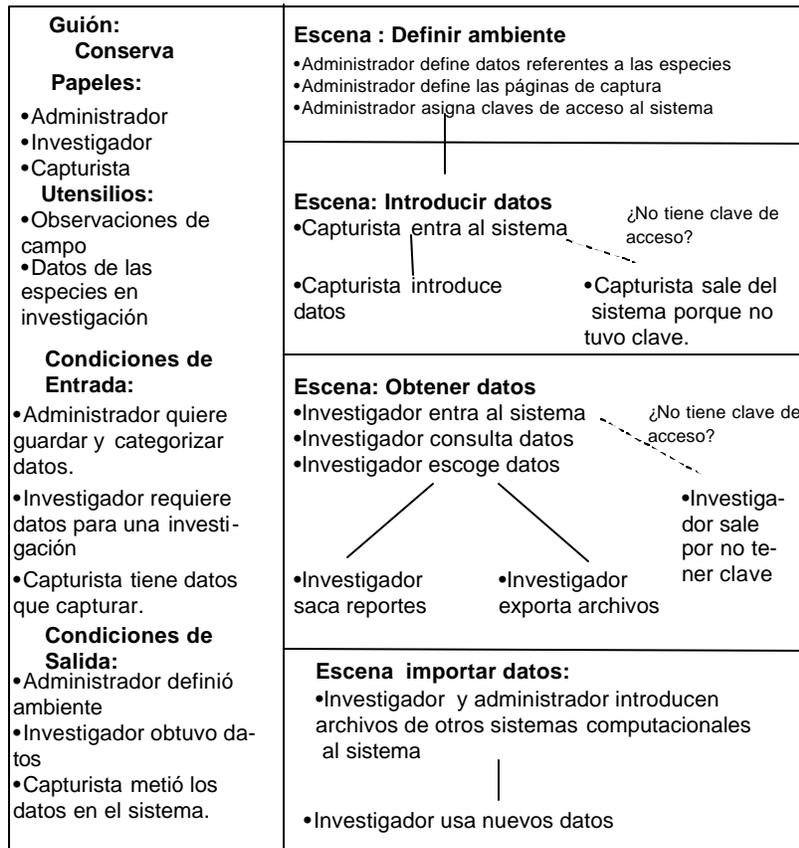


Figura A2 Estrategia Computacional para el sistema Conserva

A.2 Sistema Metropolitano.

El Canal 11 del IPN de la televisión mexicana cuenta con una serie de programas de concursos, donde se pretende que los participantes, en su mayoría alumnos de vocacionales, ejerciten sus habilidades intelectuales además de entretenerse un rato. Para mantener el interés se proponen aprovechar la capacidad de las computadoras a través de juegos. En meses recientes acudieron al CIC del IPN solicitando colaboración para realizar algunos juegos. A raíz de ello y siguiendo una idea propuesta por Luis Velo y Ernesto Nieto, del Cana 11, se comenzó a trabajar sobre un juego cuyo escenario sería el Metro capitalino y que consistiría en preguntas relacionadas con las estaciones y sus zonas aledañas, empleando recursos multimedia para ambientarlo.

A.2.1 Propuesta computacional.

Los objetivos pueden escribirse como:

1. Desarrollar un juego que pueda emplearse en programas televisados de concurso, donde dos equipos competirán para llegar primero a su destino y el avance se determinará por sus respuestas a preguntas relacionadas con el Sistema de Transporte Colectivo (Metro).
2. Difundir la cultura que se encuentra en torno al metro capitalino.

<p>Guión: El Metropolitano Pista: General Papeles: Diseñador Encargado</p> <p>Utensilios: Lista de Estaciones del metro. Listas de preguntas. Libros. Revistas. Imágenes</p> <p>Condiciones de Entrada: Diseñador y Encargado son personal del Canal 11. Se quiere hacer un Jugo por computadora para TV.</p> <p>Condiciones de Salida: Hay una Base de Datos de imágenes y preguntas asociadas grande Se puede jugar.</p>	<p>Escena 1: Crear imágenes y datos. Diseñador genera imágenes en equipo especial Encargado asocia imágenes y preguntas.</p>
	<p>Escena 2: Actualizar Base de Datos Encargado integra nuevas preguntas</p>
	<p>Escena 3: Iniciar Base de Datos. Encargado escoge preguntas Encargado carga juego</p>
	<p>Escena 4: Realiza Juego. Equipos juegan juego</p>

Figura A3. Guión General de la propuesta computacional para el Software “El Metropolitano”.

A continuación se muestra el guión propuesto para el software del juego “El Metropolitano”, que va de acuerdo a las entrevistas con los usuarios y corresponde a la Escena 4 de la pista “General”.

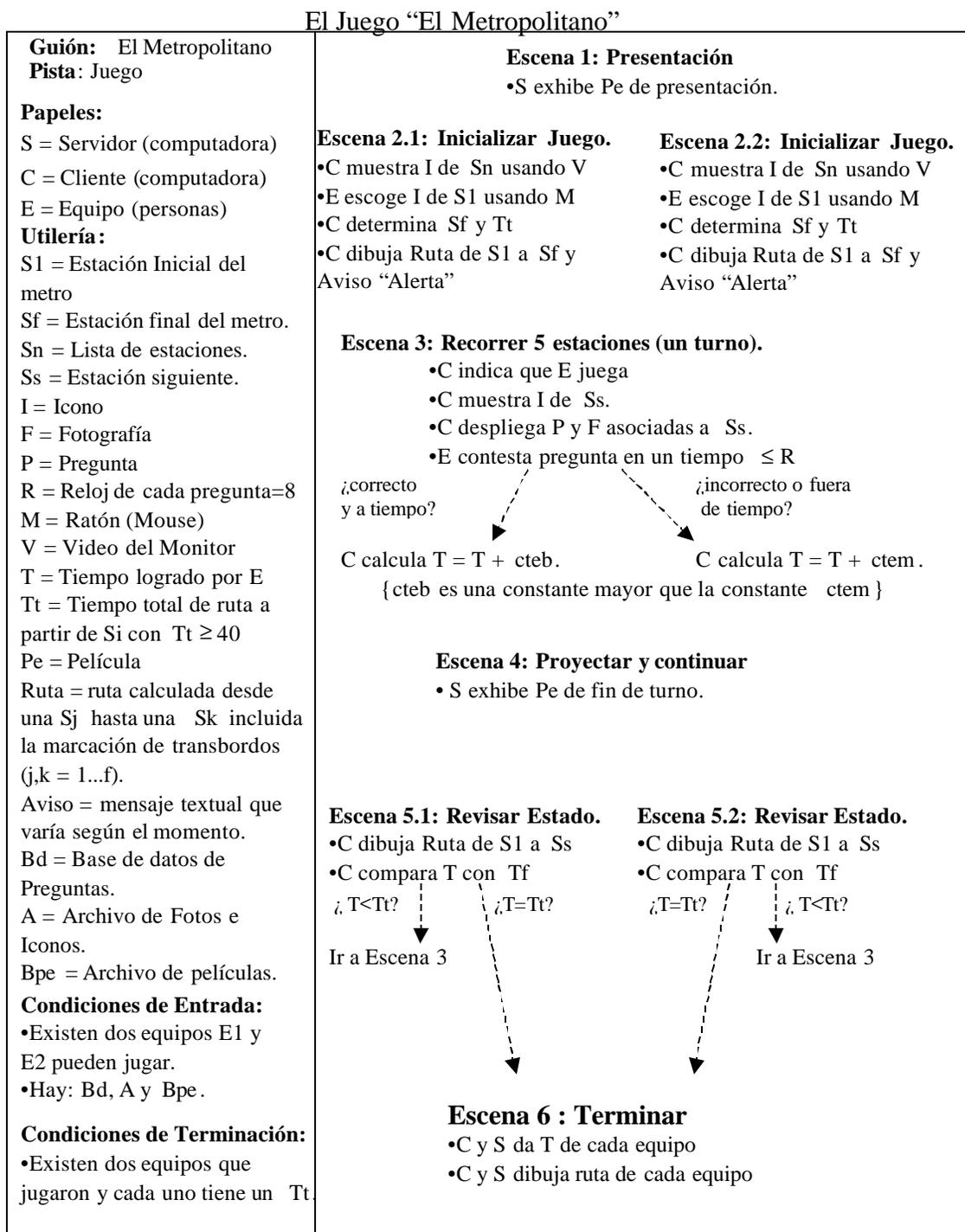


Figura A4. Pista correspondiente al juego de la propuesta computacional.

A.3 Sistema para el Control y Seguimiento de las Auditorías.

El sistema para el control y seguimiento de las auditorías, permitirá al departamento de Auditoría Financiera de la Contaduría Mayor de Hacienda del Estado de Veracruz, sistematizar el control y seguimiento de las auditorías que se realizan a cada uno de los municipios del estado. Además contará con la facilidad de poder llevar un control de la documentación que el ayuntamiento ha remitido o adeuda a esta institución.

Este sistema podrá ser empleado tanto por el jefe del departamento de auditoría financiera así como por los auditores y capturistas, por lo que tiene disponibles diferentes niveles de acceso de acuerdo a las funciones que cada empleado realiza.

Debido a la importancia de la información y a su confidencialidad, el sistema contará con un proceso para la asignación de claves de acceso, el cual solo podrá ser operado por el jefe del departamento, para asegurar que sólo tendrán acceso al sistema determinado personal.

Debido al número de usuarios, el sistema podrá ser utilizado en red, lo que permitirá atender a más de un usuario al mismo tiempo.

Tomando en cuenta que los auditores requieren en algunas ocasiones asistir físicamente a los municipios con la finalidad de realizar ahí mismo las auditorías, el sistema estará provisto de un proceso de preparación de información de determinado municipio, para que ésta pueda ser transferible a un equipo de computo portátil, lo que permitirá al auditor llevar consigo una parte del SICSAF.

Resumiendo entonces el Sistema para el Control y Seguimiento de Auditoría Financiera contará con las características que a continuación se listan:

1. Llevará un control del estado actual de las auditorías.
2. Todos los empleados del departamento de auditoría financiera podrán tener acceso a él.
3. Será totalmente seguro al contar con claves de acceso para cada uno de los usuarios. Y se definirá el nivel de acceso que tendrán cada uno de los usuarios de acuerdo a las funciones que realizan.
4. El sistema tendrá una parte portable, lo que le permitirá al auditor llevar la información que requiera al momento de hacer alguna auditoría directamente en el municipio.
5. Podrá generar reportes preestablecidos en el momento que se requieran.

A.3.1 Situación Actual.

<p><u>Guión:</u> Sistema para el control de auditorías.</p> <p><u>Papeles:</u> JA= Jefe de auditoría. AF = Auditoría Financiera. CM = Contador Mayor. AE = Auditores Externos. Ai= Auditor. MPj= Municipio. CT= Capturistas. CV= Comision de Vigilancia.</p> <p><u>Utensilios:</u> CPD = Cuenta Pública Documentada. PO= Pliego de Observaciones. WORD = Procesador de textos. OD= Oficio de Descargo. FC= Formato de Cédula. Il= Impresora. OF= Oficio de felicitación. RC= Recordatorio. AM= Amonestacion. RA = Reportes de Auditorías.</p> <p><u>Condiciones de entrada:</u> AF recibe JA del MP</p> <p><u>Condiciones de salida:</u> AF envía PO al MP.</p>	<p>Escena 1: Obligación. MPj debe enviar mensualmente CPD a AF.</p>
	<p>Escena 2: Recibir Cuenta Pública. MP entrega CPD a AF. JA documenta CPD.</p>
	<p>Escena 3: Asignación JA asigna por antigüedad CPD a Ai JA registra CPD asignada. Ai llena FC</p>
	<p>Escena 4: Análisis de la Cuenta Pública Documentada.</p> <p>¿Tiene irregularidades? — AF envía MPj OF</p> <p>Ai analiza CPD Ai emite en borrador PO AF entrega CPD a MPj</p>
	<p>Escena 5: Revisión del Pliego de Observaciones. CT captura PO en WORD. CT entrega a JA PO en disquete. JA verifica, da formato a PO. JA genera PO con Il</p>
	<p>Escena 6: Evío a firma. JA firma PO JA envía PO a CM CM firma PO. CM devuelve a JA PO.</p>
	<p>Escena 7: Se notifica al Municipio. AF informa a MPj que venga por el PO. MPj viene por PO a AF.</p>
	<p>Escena 8: Control de avisos. MPj envía a AF OD Ir a Escena 4</p>
	<p>Escena 9: Emisión de reportes. CM solicita AF RA AF construye RA AF entrega RA a CM</p>

Figura A5. Guión de la situación actual del Sistema de Control de Auditorías

A.3.2 Propuesta Computacional.

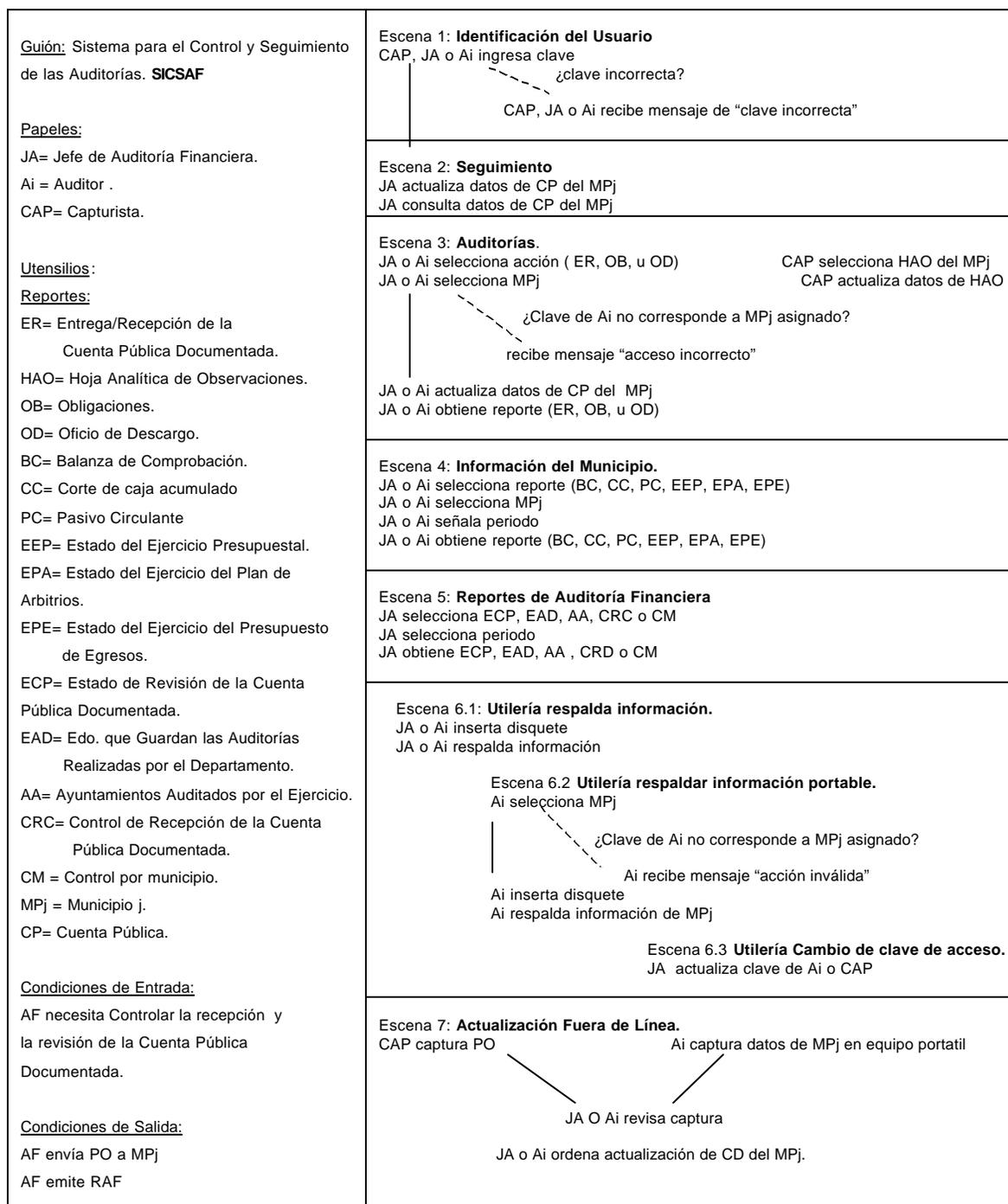


Figura A6. Propuesta Computacional del SICSAF.

A.4 Sistema de Creación de Curriculum Vitae.

Lo que el Sistema de Creación de Currículum Vitae (CCV) hace es ayudar a llevar un control de los documentos que son utilizados para elaborar un Currículum. Además contará con clave de acceso para cada usuario, así su documentación contará con seguridad y cada vez que necesite realizar un currículum será más fácil.

El sistema CCV informará de los documentos con los que ya cuenta o tendrá la opción de introducirlos por vez primera.

Cuenta con formatos de solicitudes, sólo bastará con llenar a quien va dirigido y el nombre del solicitante, imprimir y enviarlos a quien corresponda.

A.4.1 Situación Actual

<p>Guion: Elaboración de Currículum</p>	<p>ESCENA 1: Recopilación de Documentos P busca D ¿P no tiene todos los D completos? P identifica D faltantes P reúne D en F</p>
<p>Papeles: P= Persona</p>	<p>ESCENA 2: Solicitud de faltantes P tiene faltantes ¿D faltante corresponde a DP, DE, DEL, DED? P Acude a DEPC P fotocopia D obtenidos</p>
<p>Utensilios: I= Información H= Hojas ME= Máquina eléctrica D= documentos C= Currícula F =Folder DP =Datos Personales DE =Datos de escolaridad DEL =Datos de experiencia laboral DED= Datos de experiencia docente DEPC= Dependencia correspondiente</p>	<p>ESCENA 3: .Ordena Documentos P ordena D</p> <p>ESCENA 4: Elaboración de Curriculum P necesita mecanografiar I P Utiliza H en ME para mecanografiar C P escribe I de D P Retira H</p>
<p>Condiciones de entrada: P necesita elaborar Curriculum</p>	<p>ESCENA 5: Ordenar Currícula P ordena curriculum P fotocopia curriculum P engargola curriculum</p>
<p>Condiciones de salida: P elaboró curriculum</p>	

Figura A7. Guión para la situación actual

A.4.2 Propuesta Computacional

<p>Guión: Creación de Curriculum Vitae</p>	<p>ESCENA 1: Acceso al sistema P enciende C P proporciona CV</p> <p>¿P dio CV errónea?</p> <p>P da orden para hacer CU P sale de sistema</p>
<p>Papeles: P = Persona</p>	<p>ESCENA 2: Recopilación de Documentos P busca D en A</p> <p>¿P tiene todos los D completos?</p> <p>P va a escena 4</p>
<p>Utensilios: T = Teclado I = Impresora CV = clave de cada persona D = Documentos DT = Datos CU = Curriculum S = Scanner F = Formato C = Computadora DEPC = Depcia. correspondiente DC = Documentos comprobatorios A = Archivo TC = Tipo de currículum: ED = Currículum para el sistema Educativo GE = Currículum para el Gobierno estatal GF = Currículum para el Gobierno federal P = Currículum para el sector Privado</p>	<p>ESCENA 3: Solicitud de faltantes P identifica D faltantes P pide faltantes en las DEPC</p> <p>ESCENA 4: Elección del tipo de currículum P en A elige TC (de las plantillas del sistema)</p> <p>¿P decide otro TC ?</p> <p>P genera nueva plantilla</p> <p>ESCENA 5: Introducir datos 5.1 P introduce D utilizando S 5.2 P introduce DT usando T</p> <p>P guarda DT en A para llevar control</p> <p>¿P no desea imprimir?</p> <p>P sale de sistema</p>
<p>Condiciones de entrada: P necesita elaborar Currículum</p>	
<p>Condiciones de salida: P obtuvo currículum</p>	<p>ESCENA 6: Obtención de Currículum P obtiene impresión de CU</p>

Figura A8. Guión para la propuesta del sistema CCV

A.5 Sistema de Control Escolar.

Como toda institución educativa, en el Instituto de Salud Pública se realizan trámites administrativos, relacionados con la escolaridad de los alumnos, lo que genera un gran número de documentos entre ellos, listas, boletas de calificaciones, actas de examen, etcétera.

Aunque dentro del Instituto existen sistemas para manejar algunos aspectos muy específicos de control escolar como la selección de aspirantes y la evaluación que los alumnos realizan a los maestros al finalizar cada periodo escolar, un sistema no esta relacionado con el otro y además muchos trámites y procedimientos de este control se realizan manualmente. Lo que genera que una misma información sea capturada dos veces, dando lugar a errores y duplicaciones .

Este sistema estará integrado por los siguientes módulos:

1. Selección de aspirantes
2. Inscripciones
3. Control de pagos
4. Emisión de boletas
5. Control de maestros
6. Emisión de documentos finales.

Cada uno de estos módulos, hasta el momento funcionaban en forma independiente, en algunos casos en forma automatizada y en otros de manera manual. Los beneficios que se esperan al desarrollar este sistema son:

1. Integrar el sistema de selección de aspirantes y el de cuestionarios de opinión en uno sólo, así como desarrollar los otros módulos.
2. Evitar doble captura de información, permitiendo de esta forma duplicaciones y errores.
3. El software sólo permitirá la entrada a personal autorizado.
4. El software permitirá la calificación automática de los exámenes de admisión, con la finalidad de agilizar el proceso de evaluación.
5. El software permitirá elegir las características de evaluación y asignar los puntajes para la calificación final del examen de selección.
6. En el módulo de inscripciones se evitará la duplicación de la información, al incorporar la información automática de los aspirantes seleccionados del modulo de selección de aspirantes.
7. El software permitirá la elaboración automática de documentos oficiales, oficios, reportes, etc.
8. En el módulo de emisión de boletas se evitará la doble captura de calificaciones.
9. El software contará con un modulo que permitirá el respaldo y recuperación de la información con la finalidad de evitar alguna pérdida irrecuperable de la misma.

A.5.1 Situación Actual.

Debido a la extensión de este sistema sólo se presenta una pista, es decir un subsistema.

<p>Guión: Control Escolar</p> <p>Pista: selección de aspirantes</p> <p>Papeles: Ai =Aspirantes Si = Secretaria PEVi = Profesores evaluadores PENi = Profesores entrevistadores Ad = Analista de datos CD = Coordinador Docente D = Director</p> <p>Útiles: T = Tarjeta de resumen de datos L = Lapiz C = Computadora I = Impresora SI = Solicitud de Ingreso LA = Lista de Aspirantes HE = Hoja de Entrevista LRF = Lista de resultados finales LAS = Lista de Aspirantes seleccionados CA = Carta de Aceptación</p> <p>Condiciones de entrada: Ai quiere ingresar al posgrado</p> <p>Condiciones de salida: Ai recibe resultado de ingreso o rechazo</p>	<p>Escena 1: Solicitud de ingreso Ai llena SI Ai entrega SI a S S asigna generación y número de control a cada SI S llena T</p> <p>Escena 2: Vaciado de datos: S captura T con C S imprime LA con I</p> <p>Escena 3: Aplicación de Exámenes Ai presenta diversos tipos de exámenes</p> <p>Escena 4: Calificación de exámenes PEVi califican los exámenes de los Ai S captura los resultados con C S imprime en I una HE</p> <p>Escena 5: Realización de Entrevistas Ai acude a las Entrevistas con los PENi Los PENi leen la SI y la HE Los PENi interrogan a los Ai sobre diversos aspectos Ai sale de la entrevista Los PENi completan la HE y emiten dictamen, y también la firman Los PENi entregan la documentación a S</p> <p>Escena 6: Procesamiento y análisis de la información S captura en C los dictámenes de las HE Ad elabora puntajes finales con C S elabora LRF CD y D discuten LRF CD y D emiten decisión final CD entrega decisión final a S S elabora LAS</p> <p>Escena 7: Entrega de resultados Ai llama por teléfono a S S informa resultado A Ai de aceptación o rechazo S elabora CA S manda por correo CA</p>
--	--

Figura A9. Situación actual de la selección de aspirantes.

A.5.2 Propuesta Computacional.

El subsistema de Selección de Aspirantes varía bastante de la situación actual ya que sólo se consideran aquellas acciones donde la computadora interviene.

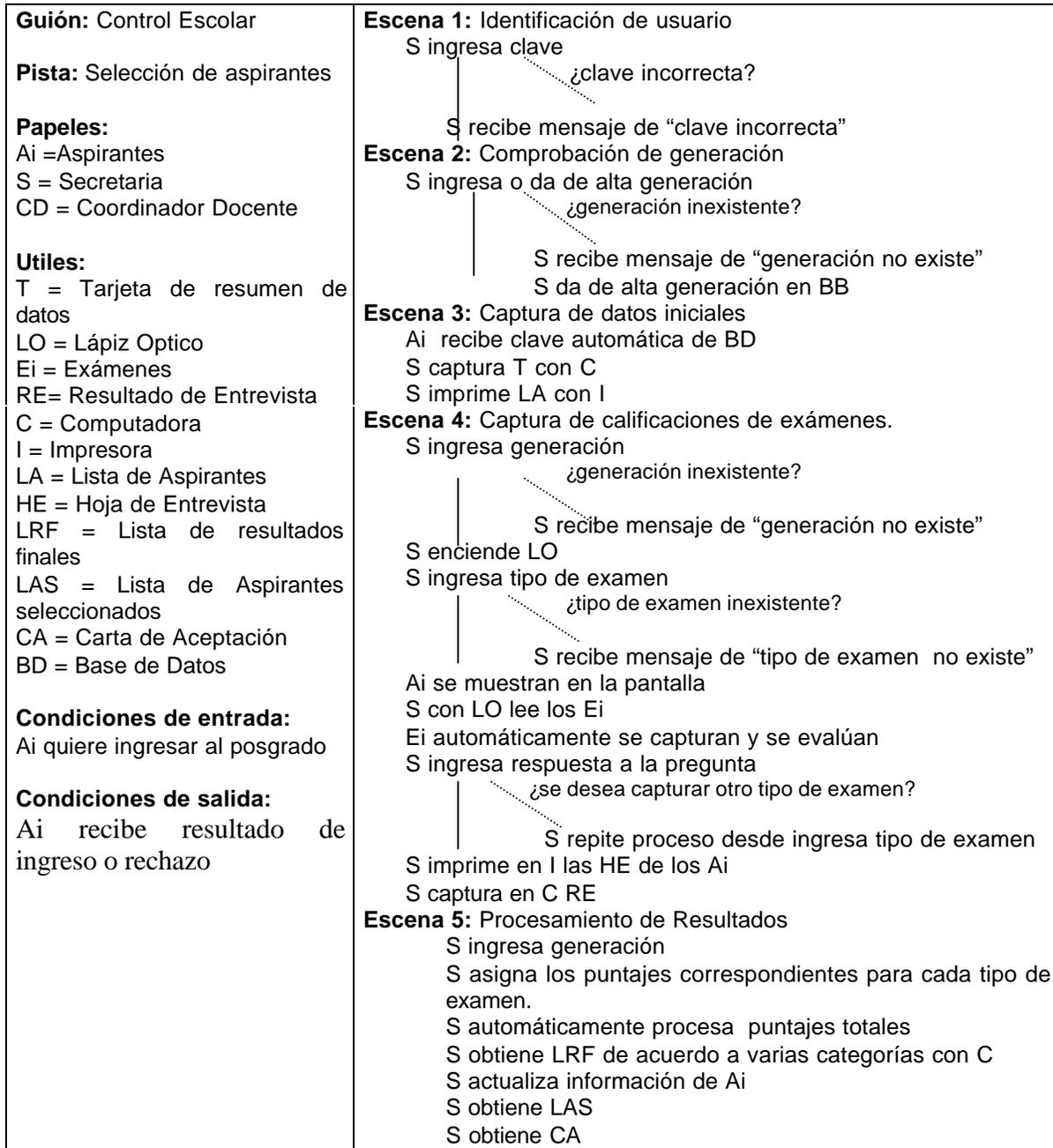


Figura A10. Guión Propuesto para subsistema Selección de Aspirantes del sistema Control Escolar.

Apéndice B. Formas para la documentación del Análisis de Requerimientos de Software.

Resulta de gran ayuda para el analista de software contar con una serie de formatos que le vayan indicando cómo presentar los resultados parciales del Análisis de Requerimientos de Software (ARS) y cómo avanzar en él. Áncora brinda una serie de formatos que se irán presentando en este anexo.

B.1 Formatos para el conocimiento del dominio y contexto de la aplicación.

En Áncora se manejan formatos para investigar el dominio y para presentar lo investigado.

B.1.1 Tarjeta para las Redes Semánticas Naturales.

Recuerdese que ésta tarjeta se utiliza para reunir la siguiente información sobre el sistema: Sinónimos del sistema, elementos del sistema, satisfacción actual, actividades de cada involucrado. Se presenta un ejemplo en la Figura B1.

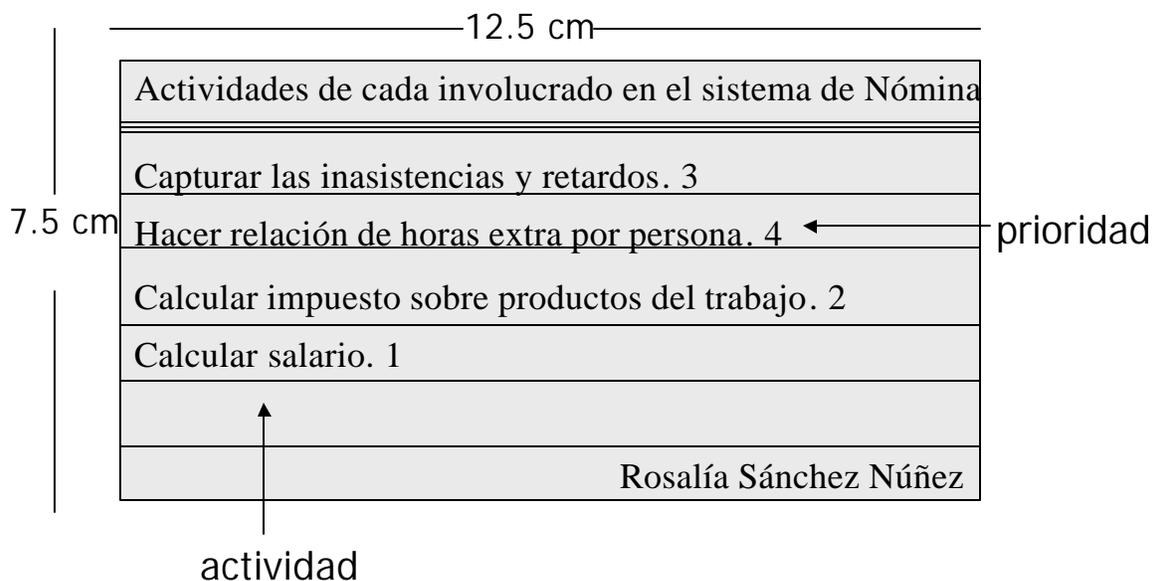


Figura B1. Tarjeta de cartulina utilizada en la colección de elementos del sistema.

B.1.2 Encuesta de Actitud.

Debe recordarse que la encuesta de actitud está formada de afirmaciones compuestas por un sentimiento afirmativo o negativo al sistema actual y una de las actividades que se realiza en éste. El objetivo principal de la encuesta es saber la actitud que tienen los usuarios hacia la eventualidad de la elaboración del nuevo software, así que las afirmaciones deben mencionar el hecho. Un patrón de afirmaciones puede ser como en la Figura B2.

Que se elabore un software de apoyo al {nombre del sistema} ayudaría a {nombre de una actividad dentro del sistema} y me daría menos {nombre de un sentimiento negativo hacia el sistema}.

Figura B2. Afirmación típica de la encuesta de actitud.

Ahora bien, la forma de evaluar cada afirmación debe ser un número non entre tres y siete valores una posible escala, de cinco valores, se muestra en la Figura B3.

- Definitivamente SÍ.
- Probablemente SÍ
- Indeciso
- Probablemente NO
- Definitivamente NO

Figura B3. Una escala de Likert para la calificación de afirmaciones.

B.1.3 Glosario del sistema.

Concepto	Significado	Formas donde aparece	Lista de posibles valores	Ref.
Archivo	Conjunto de registros en una base de datos			1
Computadora	Máquina de propósito general que procesa datos de acuerdo con el conjunto de instrucciones que están almacenadas internamente, bien sea temporal o permanentemente.			2
Exámenes	Instrumento de medición de conocimientos o aptitudes			
Impresora	Dispositivo usado para imprimir en papel la información de un computador.			
Lápiz Optico	Dispositivo óptico manual que se usa para leer tipos mecanografiados, impresos, OCR y códigos de barra.			2
Laptop	Computadora portátil que tiene una pantalla plana y usualmente pesa menos de 6 Kg. Utiliza corriente alterna y/o baterías.			2
Matrícula	Clave de identificación única para cada alumno.	Pago de arancel, listas de asistencia, acta de examen	Numérico	
Menú	Lista de opciones que el usuario observa para elegir lo que desea hacer con el programa que está usando			1
Red de computadoras	Conexión para intercambio de datos de sistemas de computadoras situadas en diferentes lugares.			1
Tarjeta de resumen de datos	Es una tarjeta de papel en donde el alumno anota			

Lista de Referencias.

1.	"Computación. Fundamentos, aplicaciones y programación", Gary G. Bitter, Addison-Wesley, 1987
2.	"Diccionario de Computación", Alan Freedman, McGraw-Hill, 1994

B.2 Recolección y clasificación de requerimientos.

Esta etapa del ARS trata del establecimiento inicial de los modelos de requerimientos, entre ellos está el prototipo rápido del cual Áncora propone el siguiente.

B.2.1 El Prototipo Rápido.

En Áncora se considera un prototipo rápido que es el Manual de Operación del usuario, una forma de hacer este manual se propone aquí con el relleno del patrón que sigue:

Manual de Operación del Software {nombre del software}

Bienvenidos a {nombre del software}, al utilizar este producto usted verá a la computadora como un ayudante eficiente de sus tareas.

Lo que {nombre del software} hace es ayudarle a llevar {objetivo general del software}.

Instalación.

Para el correcto funcionamiento de {nombre del software} será necesario contar con el siguiente equipo:

Equipo principal:

CPU	
Monitor	
Disco duro	con x Mb disponibles
RAM	
Sistema Operativo	
Impresora	
Teclado	
Ratón	

Equipo auxiliar:

CPU	
Monitor	
Disco duro con x Mb disponibles	
RAM	
Sistema Operativo	
Teclado	

Instalación.

Para que {nombre del software} quede listo para ser utilizado, debe realizar su instalación, la cual se realiza de la siguiente forma:

1. Introduzca el CD en donde corresponde y averiguar la letra del dispositivo de CD (generalmente es D: ó E:).
2. A partir de que cierre la compuerta del dispositivo de CD, la máquina le irá indicando que hacer. En caso de que no sea así, proceda como sigue.
 - a. Oprima el icono de “Mi PC”.
 - b. Oprima con el ratón el icono del dispositivo de CD y escoja el archivo instalar.

Índice general.

Pista 1.

Escena 1.1.
Escena 1.2.

...

Escena 1.i.

Pista 2.

Escena 2.1.
Escena 2.2.

...

Escena 2.j.

...

Pista k.

...

Explicación de cada funcionalidad

Pista 1.

El subsistema de {nombre del subsistema} tiene como objetivo brindarle {objetivo del subsistema} y para ello se ha dividido en las partes que se irán detallando en seguida.

Escena 1.1.

Mediante el grupo de operaciones comprendidas en {Escena 1.1} usted podrá {objetivo de la escena 1.1}, las operaciones se describen a continuación:

Quinteta 1.1.1.

La operación {quinteta 1.1.1} consiste en permitir que {objetivo de la quinteta 1.1.1}, para ello es necesario que conozca los elementos que conforman la {utileria de la quinteta 1.1.1} que se muestra en la Figura {número de pista. número progresivo}, cuya descripción se da en seguida:

1. Campo 1. Se refiere a Los valores que puede llegar a tomar son:
2. Campo 2. Es la Los valores que contendrá van de ... a

{poner dibujo de la utilería de la quinteta 1.1.1 (Ej. pantalla o reporte)}

Figura {número de pista-número progresivo}

Además debe tener en cuenta que para que {quinteta 1.1} llegue a buen término se debe hacer lo siguiente:

{incluir diálogo de quinteta 1.1}

Guía de situaciones que pueden surgir con el uso de {nombre del software}.

Cuando se está utilizando {nombre del software}, pueden surgir las siguientes situaciones:

1. Falta de energía eléctrica. Cuando esto sucede algunos de los últimos datos que se habían introducido estarán desactualizados, por ello es necesario que cuando la energía se restablezca usted revise los últimos datos que introdujo.

B.3 Jerarquización, validación y resolución de conflictos de requerimientos.

La mayor parte de las actividades de estas etapas se llevan a cabo en la RRD que requiere de varios elementos para llevarse a cabo y que se explican en esta sección.

B.3.1 La Reunión de Reflexión y Diseño.

Como ya se explicó en el texto la RRD es una junta de todos los tipos de involucrados en el nuevo software y para ello se prepararon los documentos y carteles de la Tabla B1. En este anexo se muestran algunos de ellos.

Tabla B1. Lista de Documentos que deben tenerse para la RRD.

Documento	Contenido
1. Contexto y situación actual.	Metas de la empresa. Organigrama de la empresa. Guión y diálogos de la situación teórica y actual. Justificación del nuevo software. Glosario.
2. Propuesta computacional.	Guión y diálogos propuestos. Manual de operación. Modelo Entidad – Relación. Beneficios del nuevo software
3. Restricciones y responsabilidades.	Identificadores. Plan de prueba funcional. Restricciones. Costo en términos de puntos de función. Tabla de responsabilidades.
4. Otros materiales.	Calendario y plan de actividades de la RRD. Tarjetas para la asignación de responsabilidades. Cartel de reglas de comportamiento. Cartel de puntos a tratar. Cartel de la lista de escenas para la asignación de prioridades. Hojas de rotafolio, plumones y rotafolio. Fotocopia de los documentos 1 a 3 para cada involucrado. Oficio de acuerdos firmado por los involucrados que asistieron a la RRD. Lápices y bolígrafos. Café, té, galletas, azúcar, crema, agua, cafeteras, vasos térmicos, cucharillas, servilletas, etc.

Preguntas a resolver

- ¿La situación actual está bien entendida?
- ¿Lo que se necesita, es lo que se está proponiendo?
- ¿Lo que se propone está bien clasificado?
- ¿Cuál es la prioridad de cada parte?
- ¿Están bien definidas las restricciones?
- Las responsabilidades, ¿son correctas?

Figura B4. Cartel de puntos a resolver en la RRD.

En éste grupo:

- Todos somos iguales.
- Toda opinión es importante.
- Exprésate en primera persona.
- Se breve y conciso.
- Acepta opiniones diferentes.
- Se positivo, no se trata de culpar.

Figura B5. Cartel de reglas de comportamiento en las reuniones de grupos pequeños.

Tema: {nombre del software}

Propósito: Clasificación, jerarquización y validación de requerimientos del software.

Fecha: {día 1} y {día 2} de {mes} del {año}

Lugar: {lugar dónde se llevará cabo la RRD} (fuera del lugar de trabajo de los usuarios y de los analistas)

{día 1} de {mes} del {año}

Hora	Tipo de reunión	Material requerido	Acciones
9:00 – 9:30	Reunión informal de los participantes	<ul style="list-style-type: none"> Hojas para anotar a participantes 	Anotar a las personas presentes
9:30 - 10:30	Sesión Plenaria Introdutoria.	<ul style="list-style-type: none"> Escrito con puntos a tratar a la vista de todos. 	Presentar el objetivo de la reunión y participantes.
10:30 - 11:30	Primera sesión en grupos pequeños: Clasificación de requerimientos y resolución de conflictos. Revisión de las escenas propuestas.	<ul style="list-style-type: none"> Documento de Contexto y situación actual. 	Leer, analizar y anotar correcciones sobre el material.
11:30 - 11:45	Receso.	<ul style="list-style-type: none"> Café y galletas 	El analista debe sacar la lista de correcciones
11:45 – 12:00	Primera sesión plenaria. Presentaciones.	<ul style="list-style-type: none"> Lista de las correcciones sobre la situación actual. 	Leer la lista de correcciones y pedir aclaraciones.
12:00 - 13:00	Segunda sesión en grupos pequeños: Restricciones bajo las cuáles deberá implantarse el nuevo software	<ul style="list-style-type: none"> Documento de Propuesta Computacional. 	Leer, analizar y anotar correcciones sobre el material. Debe incluirse en la revisión si las funciones están bien clasificadas (las escenas contienen las quintetas adecuadas).
13:00 – 13:30	Segunda sesión plenaria. Presentaciones.	<ul style="list-style-type: none"> Lista de las correcciones 	Leer la lista de correcciones y pedir aclaraciones.

{día 2} de {mes} del {año}

Hora	Tipo de reunión	Material requerido	Acciones
9:00 – 10:00	Tercera sesión en grupos pequeños: Resumen de la sesión anterior Asignación de prioridades de los requerimientos.	<ul style="list-style-type: none"> Lista de observaciones con sus aclaraciones de la situación actual y la propuesta computacional. Guión de propuesta computacional. Grupo de tarjetas para la asignación de prioridades de las escenas. Cartel con lista de escenas con espacio contiguo para anotar valores de cada participante y su suma. 	Entregar fotocopia de las observaciones y sus aclaraciones que se considerarán para el documento de análisis de requerimientos. Aplicar Técnica de Grupo Nominal.
10:00 – 10:30	Tercera sesión plenaria. Presentación de asignación de prioridades.	<ul style="list-style-type: none"> Lista de la jerarquización de prioridades. 	El analista lee las jerarquizaciones obtenidas en los grupos pequeños y recoge consenso.
10:00 – 10:15	Receso	<ul style="list-style-type: none"> Café y galletas 	
10:30 – 11:30	Cuarta sesión en grupos pequeños: Validación de requerimientos Verificación de las restricciones Verificación de responsabilidades	<ul style="list-style-type: none"> Documento de Restricciones y responsabilidades. Carteles de restricciones y de responsabilidades. 	Se revisa las listas presentadas por el analista y se hacen anotaciones sobre el material.
11:30 – 12:00	Cuarta sesión plenaria. Firma de acuerdos.	<ul style="list-style-type: none"> Todos los documentos anteriores. Oficio de firma de acuerdos. 	

Figura B6. Calendario de la RRD

Restricciones bajo las cuales correrá el “Sistema para el control y seguimiento de las auditorías”

1. El sistema correrá en la red de la Legislatura.
2. Algunos datos se cargarán fuera de línea y luego se actualizará con ellos los archivos maestros. Tal es el caso de los datos introducidos por capturistas o en el municipio mediante una computadora portátil.
3. El acceso a los datos se hará de forma restringida mediante claves de acceso.
4. Habrá conexión con algunos archivos del sistema SICOM.
5. No existen restricciones de rendimiento: tiempo de respuesta, concurrencia, consideraciones de diseño o implantación.
6. El software será diseñado e implantado con las herramientas de desarrollo existentes en la legislatura.
7. No se anticipa un periodo pico de transacciones.
8. Más del 30% de los movimientos se hará de forma interactiva.
9. Habrá manejo de menús, movimiento automático del cursor, manejo de luminosidad en los campos, interfaz de ratón, pocas ventanas abiertas al mismo tiempo.
10. Se hará una rutina especial para salvado de datos que será operada por el usuario.

Figura B7. Ejemplo sobre la lista de restricciones que debe llevarse a la RRD.

Escena	Prioridades asignadas	Suma
1. Identificación del usuario		
2. Seguimiento		
3. Auditorías		
4. Información del Municipio		
5. Reportes de Auditoría Financiera		
6.1 Utilerías de respaldo de información		
6.2 Utilería de respaldo de información portable		
6.3 Utilería de cambio de claves de acceso		
7. Actualización fuera de línea		

Figura B8. Ejemplo de Cartel para la jerarquización de requerimientos en el sistema Control y Seguimiento de Auditorías.

Integrante del Software	Responsabilidad del integrante
Software	<ol style="list-style-type: none"> 1. El SICSAF brindará un mecanismo de seguridad de acceso a los datos. 2. El SICSAF permitirá revisar la bitácora del control de auditorías. 3. El sistema de software se encargará de brindar los mecanismos para ingresar datos referentes a las auditorías de los municipios. 4. SICSAF permitirá acceder la información referente a la cuenta pública de los ayuntamientos utilizando información propia y del SICOM. 5. SICSAF permitirá generar los reportes que han sido definidos por el usuario para este software. 6. SICSAF permitirá que la información se resguarde en discos flexibles y de ser necesario se pueda recuperar de éstos. Además permitirá realizar copia y recuperación de información de auditorías de campo. Para lograr un acceso a los datos adecuado se brinda la posibilidad de asignación y cambio de claves. 7. Utilizando SICSAF se podrá actualizar los archivos maestros de datos que fueron capturados fuera de línea en el formato especificado.
Hardware	Mediante medios magnéticos se almacenarán los datos que el SICSAF genera y que fueron introducidos por el usuario.
Usuarios	<p>Tendrán la responsabilidad sobre la introducción, veracidad y oportunidad de los datos.</p> <p>Dependerá de ellos el momento de salvado y recuperación de los archivos que utiliza SICSAF.</p> <p>La asignación de acceso a los datos será responsabilidad del Jefe de Auditoría Financiera.</p>

Figura B9. Ejemplo de asignación de responsabilidades para el “Sistema de Control y Seguimiento de Auditoría Financiera” (SICSAF).