

Universidad Católica de Valparaíso
Facultad de Ingeniería
Escuela de Ingeniería Industrial

**Modelamiento Basado en Entidades Dinámicas:
Una Estrategia de Integración de Modelos
Conceptuales de Sistemas de Información**

por

Rodrigo Alfaro Arancibia

Memoria para optar al título de
Ingeniero Civil Industrial

Profesor Guía Guillermo Bustos Reinoso

Julio, 2000

Índice

Resumen.....	8
1 Introducción	9
2 Modelamiento conceptual de sistemas	11
2.1 Fundamentos del modelamiento conceptual de sistemas.....	11
2.2 Herramientas para modelar sistemas	13
2.2.1 Dimensión estática	15
2.2.1.1 Diagrama Entidad Relacionamiento	15
2.2.1.2 Diagrama de Clases	17
2.2.1.3 Diccionario de Datos.....	18
2.2.2 Dimensión funcional	20
2.2.2.1 Diagrama de Flujo de Datos.....	20
2.2.2.2 Diagrama de Flujo de Datos con Flujos de Control	21
2.2.2.3 Especificación de Procesos.....	23
2.2.2.4 Diagrama de Casos de Uso.....	24
2.2.2.5 Diagrama de Actividades	25
2.2.3 Dimensión dinámica.....	26
2.2.3.1 Diagrama de Transición de Estados	26
2.2.3.2 Red de Petri.....	27
2.2.3.3 Diagrama de Estados.....	28
2.2.3.4 Diagrama de Interacción	29
2.2.3.4.1 Diagrama de Secuencia	30
2.2.3.4.2 Diagrama de Colaboración	30
3 Principales estrategias de integración de modelos	33
3.1 Fundamentos	33
3.2 Análisis Estructurado Moderno.....	36
3.3 Estrategia Genérica	38
3.4 Orientación a Objetos	40
3.5 Consideraciones de cada estrategia	43
4 Estrategia propuesta.....	45
4.1 Fundamentos	45
4.2 Modelamiento de la dimensión dinámica	48
4.3 Modelamiento de la dimensión estática.....	59
4.4 Modelamiento de la dimensión funcional.....	60
4.5 Relación dinámica - estática.....	62
4.6 Relación dinámica – funcional	71
4.7 Relación estática - funcional	73
4.8 Reglas de balanceo entre los modelos utilizados	75
5 Consideraciones finales	81
5.1 Consideraciones respecto a la estrategia.....	81
5.2 Consideraciones respecto al modelamiento	81

6 Conclusiones	85
6.1 Conclusiones respecto de la estrategia.....	85
6.2 Extensiones	85
6.3 Futuras investigaciones.....	86
Anexo 1: Redes de Petri	88
1 Elementos básicos	88
2 Reglas de funcionamiento.....	89
3 Relaciones causales entre eventos	90
3.1 Rama restauradora de entrada en una alteración.....	90
3.2 Rama restauradora de salida en una alteración	91
4 Jerarquización en las Redes de Petri: Redes Canal/Actividad.....	91
Anexo 2: Caso de estudio	94
1 Descripción del problema	94
2 Modelamiento de la perspectiva dinámica, RP	95
3 Modelamiento de la perspectiva funcional	99
3.1 Diagrama de Flujo de Datos	101
3.2 Especificación de Procesos.....	106
4 Modelamiento de la perspectiva estática	112
4.1 Diagrama Entidad Relacionamiento.....	112
4.2 Diccionario de Datos.....	113
Bibliografía	125

Lista de Abreviaturas y Siglas

AEM	=	Análisis Estructurado Moderno
AOO	=	Análisis Orientado a Objeto
CV	=	Ciclo de Vida
DA	=	Diagrama de Actividades
DC	=	Diagrama de Clases
DCO	=	Diagrama de Colaboración
DCU	=	Diagrama de Casos de Uso (<i>Use Cases</i>)
DD	=	Diccionario de Datos
DER	=	Diagrama Entidad - Relacionamiento
DE	=	Diagrama de Estados
DFD	=	Diagrama de Flujo de Datos
DFD c/FC	=	Diagrama de Flujo de Datos con Flujos de Control
DI	=	Diagrama de Interacción
DS	=	Diagrama de Secuencias
DTE	=	Diagrama de Transición de Estados
EG	=	Estrategia Genérica
EP	=	Especificación de Procesos
MER	=	Modelo Entidad - Relacionamiento
MOBED	=	Modelamiento Basado en Entidades Dinámicas
MOO	=	Modelamiento Orientado a Objeto
OO	=	Orientación a Objeto
pág.	=	Página
RP	=	Red de Petri
UML	=	<i>Unified Modeling Language</i>

Lista de Figuras

Figura 2.1:	Diagrama Entidad Relacionamiento	17
Figura 2.2:	Diagrama de Clases	18
Figura 2.3:	Diagrama de Flujo de Datos.....	21
Figura 2.4:	Diagrama de Flujo de Datos con Flujos de Control	22
Figura 2.5:	Diagrama de Casos de Uso	24
Figura 2.6:	Diagrama de Actividades.....	25
Figura 2.7:	Diagrama de Transición de Estados variante Mealy	27
Figura 2.8:	Red de Petri.....	28
Figura 2.9:	Diagrama de Estados	29
Figura 2.10:	Diagrama de Secuencia.....	30
Figura 2.11:	Diagrama de Colaboración.....	31
Figura 2.12:	Mapa dimensión – nivel de abstracción.....	32
Figura 3.1:	Etapas del desarrollo de Sistemas.....	33
Figura 4.1:	Entidad dinámica proyectada a cada dimensión.....	47
Figura 4.2:	Ejemplo de ciclo de vida una entidad dinámica.....	50
Figura 4.3:	Ejemplo de ciclo de vida una entidad dinámica con <i>swimlanes</i>	51
Figura 4.4:	Ejemplo de ciclo de vida una entidad dinámica activa.....	52
Figura 4.5:	Estados esenciales de una entidad dinámica.	53
Figura 4.6:	Ciclo de vida una entidad dinámica con actividades.....	54
Figura 4.7:	Ejemplo de una actividad con cohesión coincidental.....	55
Figura 4.8:	Ejemplo de una actividad con cohesión lógica.	56
Figura 4.9:	Ejemplo de una actividad cohesionada secuencialmente.	56
Figura 4.10:	Ejemplo de una actividad con cohesión temporal.	57
Figura 4.11:	Ejemplo de una actividad con cohesión comunicacional.	57
Figura 4.12:	Ejemplo de una actividad cohesionada funcionalmente.	58
Figura 4.13:	Espectro de cohesión y multiplicidad.....	59
Figura 4.14:	Ejemplo de DER con la jerarquía de una entidad dinámica.	60
Figura 4.15:	Ejemplo de Diagrama de Flujo de Datos.....	61
Figura 4.16:	Entidad dinámica en las dimensiones estática y dinámica.....	62
Figura 4.17:	Representación de la entidad dinámica en el DER.....	63
Figura 4.18:	Comportamiento de las instancias de la entidad dinámica.	64
Figura 4.19:	Modelo genérico de comportamiento de la entidad dinámica.	65
Figura 4.20:	Modelo genérico de comportamiento v/s representación estática de la entidad.....	66
Figura 4.21:	Criterio de clasificación según los estados de las instancias.	67
Figura 4.22:	Criterio de clasificación de estados v/s representación estática de la entidad.....	68
Figura 4.23:	Resumen de criterios de clasificación v/s representación estática de la entidad dinámica.....	69
Figura 4.24:	Representación de la entidad dinámica en el DER incluyendo relaciones con otras entidades.....	70
Figura 4.25:	Entidad dinámica en las dimensiones funcional y dinámica.....	71
Figura 4.26:	Relación dinámica-funcional.....	72

Figura 4.27:	Entidad dinámica en las dimensiones estática y funcional.	73
Figura 4.28:	Relación funcional-estática.	74
Figura 4.29:	Mapa dimensión – nivel de abstracción de la estrategia propuesta.....	75
Figura 4.30:	Relación tridimensional entre los modelos utilizados.	76
Figura 5.1:	Evento que produce cambios de estados en dos entidades.	82
Figura 5.2:	Solución al evento que produce cambios de estados en dos entidades..	83
Figura 5.3:	Salidas no representadas en una actividad.	84
Figura A1.1:	Ejemplo de una Red de Petri.	88
Figura A1.2:	Marcación en las redes de Petri.	89
Figura A1.3:	Problemas con la habilitación de “cruza semáforo”.....	90
Figura A1.4:	Ramas restauradoras.	90
Figura A1.5:	Rama restauradora de salida.....	91
Figura A1.6:	Cómo formar una Red de Petri Canal/Actividad.....	92
Figura A1.7:	Ejemplo de una Red de Petri Canal/Actividad.....	93
Figura A1.8:	Red de Petri Abstracta Canal/Actividad.	93
Figura A2.1:	Ciclo de vida de la entidad Proyecto, parte 1.....	96
Figura A2.2:	Ciclo de vida de la entidad Proyecto, parte 2.....	97
Figura A2.3:	Ciclo de vida de la entidad Cuenta Presupuestaria.....	98
Figura A2.4:	Diagrama de Flujo de Datos asociado al ciclo de vida.....	99
Figura A2.5:	Diagrama de Flujo de Datos extendido.....	100
Figura A2.6:	Diagrama de Contexto o de nivel cero.....	101
Figura A2.7:	Diagrama cero o de nivel uno.	101
Figura A2.8:	Diagrama uno.	102
Figura A2.9:	Diagrama dos.....	103
Figura A2.10:	Diagrama tres.	104
Figura A2.11:	Diagrama cuatro.	105
Figura A2.12:	Modelo Entidad-Relacionamiento.....	112

Lista de Tablas

Tabla 2.1 – Modelos y dimensiones que describen.....	13
Tabla 2.2 – Modelos y nivel de abstracción.	14
Tabla 2.3 – Simbología utilizada en un diccionario de datos.	19
Tabla 3.1 – Elementos, relaciones y descripción en las estrategias de integración.	35
Tabla 3.2 – Balanceo entre modelos utilizados en AEM.	36
Tabla 3.3 – Balanceo entre modelos utilizados en EG.....	38
Tabla 3.4 – Balanceo entre modelos utilizados en UML.	40
Tabla 4.1 – Balanceo entre modelos utilizados en la Estrategia propuesta.....	77

Resumen

Un requisito importante para desarrollar sistemas de información de calidad es mantener consistencia en cada modelo y mantener consistencia entre los distintos modelos utilizados para representarlo.

Una estrategia de integración de modelos de sistemas de información describe los principios y reglas que se deben cumplir para mantener consistencia entre los modelos que representen un sistema. A diferencia de una metodología, una estrategia no propone un orden ni un método para elaborar los modelos.

Para lograr consistencia entre los modelos se pueden adoptar distintas metodologías y estrategias de integración, tales como: Análisis Estructurado Moderno, Estrategia Genérica de Integración u Orientación a Objeto a través de UML.

Este documento presenta los fundamentos, las características y las reglas de balanceo y consistencia de una Estrategia de Integración de Modelos Conceptuales de Sistemas de Información que tiene como base de construcción a la entidad dinámica, la cual posee atributos y comportamiento. Se establecen relaciones entre todas las perspectivas, es decir, relaciones Estática-Funcionales, Funcional-Dinámicas y Dinámica-Estáticas, utilizando modelos existentes y relacionándolos de una nueva forma.

Para lograr esto, se describen aspectos esenciales del modelamiento conceptual de sistemas de información, principios y herramientas para modelar conceptualmente un sistema y las principales estrategias y metodologías de integración que se utilizan actualmente.

El objetivo de esta investigación es realizar el primer planteamiento formal de esta estrategia, fundamentando las relaciones de integración entre modelos conceptuales de sistemas de información y sentar las bases para futuras investigaciones y extensiones.

Palabras-claves: *Sistemas de Información*
Modelamiento conceptual
Análisis de Sistemas
Integración de modelos
Entidad dinámica.

1 Introducción

El modelamiento es una actividad crítica en el desarrollo de sistemas de calidad. Los modelos pueden ser descriptivos o prescriptivos, ya sea que se construyan para entender mejor un sistema existente o para representar un sistema a desarrollar, además sirven para apoyar futuras decisiones durante las etapas posteriores en el desarrollo de sistemas.

Un requisito importante para desarrollar sistemas de información de calidad es la consistencia interna y entre los modelos utilizados para representar al sistema. Una manera de asegurar consistencia es seguir una metodología establecida o enmarcarse en una estrategia de integración.

En la actualidad, existen variadas herramientas, modelos, estrategias, enfoques y metodologías para modelar sistemas de información. La aparición de enfoques nuevos, sin una fundamentación y una conceptualización claras, puede provocar que analistas y desarrolladores de sistemas tomen una decisión errónea a la hora de escoger alguna de ellas.

Proponer una estrategia de integración de modelos conceptuales de sistemas de información requiere de un análisis previo de los modelos a utilizar, de sus alternativas y de la forma en que se relacionan unos con otros. Además, se deben conocer los principios en que se basan otras estrategias de integración, además de sus ventajas y desventajas, y en qué contexto es más útil utilizar una u otra.

El presente documento pretende entregar una visión y descripción objetiva de herramientas, modelos, enfoques y estrategias más usadas o conocidas, para finalmente entregar otra estrategia alternativa de integración de modelos de sistemas de información, que utiliza las herramientas existentes de una forma distinta.

El principal objetivo de esta investigación ha sido proponer una estrategia, incorporándola en el contexto actual y aplicarla a un caso práctico, para que en el futuro sea utilizada por los analistas en el análisis y modelamiento de sistemas, y fundamentarla para que además sienta las bases de futuras investigaciones en torno al tema y extensiones de mismo.

El contenido de este documento se divide en cuatro partes, que son: Modelamiento conceptual de sistemas, Principales estrategias de integración de modelos de sistemas de información, Estrategia propuesta y Consideraciones finales. Posteriormente, se incluyen dos anexos, el primero profundiza la descripción de la Red de Petri elemental y su variante Canal/Actividad, el segundo es un caso práctico de control de proyectos públicos modelado mediante la estrategia propuesta.

La primera sección intenta enmarcar al lector en el modelamiento conceptual de sistemas. Se hace referencia a los principios en que se basa el modelamiento, los enfoques, los supuestos y los objetivos que persigue. Se describen los principales modelos utilizados para modelar sistemas de información, clasificados según dos criterios. El primero es según la

dimensión del modelamiento que apoyan, éstas son: Estática, Dinámica y Funcional. El segundo criterio es según el nivel de abstracción con que se utilizan en cada dimensión, es decir, para relacionar los elementos de una dimensión o para describirlos. Con estas clasificaciones se establecen criterios objetivos con los cuales se podrán clasificar nuevos modelos o perfeccionar alguno existente.

La segunda sección fundamenta la necesidad de integrar los modelos y describe algunas estrategias de integración, éstas son: Análisis Estructurado Moderno, Estrategia Genérica de Integración y Análisis Orientado a Objeto a través de UML. Se nombran sus principios, los modelos que se usan en cada una, sus reglas de balanceo y algunas consideraciones respecto de cada una de ellas.

En la tercera sección de este documento se describe la estrategia propuesta, estableciendo sus principios y fundamentos, posteriormente se describen los modelos que se deben usar para modelar el sistema desde cada dimensión y, finalmente, se definen las reglas de balanceo que deben considerarse para construir los modelos de las distintas perspectivas.

Posteriormente, se realizan las consideraciones finales, con lo que se busca establecer algunas ventajas y desventajas descubiertas a la luz de la aplicación de esta estrategia a algunos casos prácticos.

En el Anexo 1 se explican con más detalle las reglas de funcionamiento de la Red de Petri elemental y de la Red de Petri en su variante Canal/Actividad, modelo que es utilizado en la estrategia propuesta.

En el Anexo 2, se aplica la estrategia propuesta a un caso práctico, para esto se describe el problema en que se aplicará y posteriormente se presentan los modelos desde cada perspectiva.

En resumen, la estrategia propuesta en este informe, es presentada y definida conceptualmente, fundamentada y respaldada con teorías conocidas y aplicada a un caso práctico de control de proyectos. Con esta metodología se lograron refinar los conceptos que ella involucra y se encontraron algunos conflictos con que se pueden encontrar los analistas al modelar sistemas de información.

2 Modelamiento conceptual de sistemas

2.1 Fundamentos del modelamiento conceptual de sistemas

Los modelos se crean para lograr un mejor entendimiento de un ente a representar; si el modelo busca representar un ente que existe, será descriptivo y si el ente no existe, será un modelo prescriptivo. Por otra parte, cuando el ente es algo tangible, se pueden construir modelos idénticos al real en menor escala. Pero cuando el ente es un Sistema de Información, el analista debe ser capaz de modelar los datos que el sistema almacena, transforma y procesa, las funciones que realizan la transformación, y el comportamiento que tiene el sistema mientras realiza los procesos y transformaciones.

En fases iniciales del desarrollo de sistemas, específicamente en la fase de Análisis, se deben crear los modelos conceptuales del sistema. Según [Høydalsvik&Sindre93], estos modelos deben describir el problema y los requerimientos de los usuarios, considerando que en una etapa posterior de diseño se construirán los modelos que entreguen la solución que satisfaga los requerimientos de los usuarios y el problema planteado. Durante el análisis, el analista debe poner especial atención en el dominio del conocimiento, objetivos, requerimientos y entorno del sistema, es decir, debe “orientarse al problema” y no limitarse a modelar la solución al problema. De esta forma se facilita el modelamiento y la validación por parte de los usuarios.

Cuando los analistas se “orientan al problema”, la brecha entre el sistema percibido y el problema representado se hace más pequeña. Otra forma de modelar un sistema es “orientarse al objetivo”, donde el conocimiento se estructura de una forma que tiene más relación con la siguiente etapa de diseño, que con el entendimiento y dominio del problema que el analista y los usuarios obtengan. Las principales críticas a la “orientación al objetivo” expuestas en [Høydalsvik&Sindre93], entre otras, son:

- Un análisis “orientado al objetivo” pierde los potenciales descubrimientos de las necesidades y cambios organizacionales en el sistema, lo cual es una fuente de insatisfacción común.
- Orientarse al objetivo implica tomar decisiones de diseño, las cuales no deberían hacerse antes del entendimiento completo del problema.

Los modelos creados hacen uso de notación gráfica que representa principalmente la información, los procesos y el comportamiento del sistema. Esto puede complementarse con texto, ya sea en lenguaje cotidiano o especializado.

Según [Pressman98], pág. 190, los principales papeles de los modelos son:

- Ayudar al analista a entender la información, la función y el comportamiento del sistema, haciendo por tanto más fácil y sistemática la tarea de analizar requerimientos.
- Convertirse en el punto de comparación entre lo logrado y lo planificado.
- Fundamentar el diseño, proporcionando al diseñador una representación lógica, o esencial, de la implementación.

Entonces, los modelos deben centrarse en “qué” hará el sistema y no en “cómo” lo debe hacer, utilizando el supuesto de “tecnología perfecta”¹, sin tener en cuenta las restricciones de implementación. Con esto se obtiene el diseño lógico o conceptual de sistemas de información, en el que se define qué hará el sistema. En una etapa posterior se construye el diseño físico, en el que se determina cómo lo hará, incorporando las restricciones y requisitos de implementación.

Para comenzar la descripción del sistema, se analizan los distintos requerimientos de información y los eventos relevantes que se deben registrar para entregar información útil a la toma de decisiones. Con esto se definen los objetivos del sistema en un contexto determinado.

Para modelar sistemas de información desde las perspectivas internas, se utilizan tres puntos de vista o dimensiones: la Estática, la Dinámica y la Funcional. Estas perspectivas son casi ortogonales, es decir, los elementos modelados en una perspectiva no debieran representarse en las otras. Sin embargo, existen elementos comunes en las diferentes intersecciones, por lo que deben modelarse consistentemente, ya que un cambio que se produzca en el sistema influye en las tres perspectivas.

El orden que se escoja para modelar depende de diversos factores, tales como la experiencia del modelador, la estrategia de integración o la metodología, las que están muy ligadas también al contexto de sistema que se desea modelar.

Un elemento del sistema puede tener una representación desde una, dos o tres perspectivas, es decir, todos los elementos deben ser representados por lo menos desde una perspectiva.

Por otra parte, los modelos utilizados para representar los elementos y sus relaciones en el sistema pueden ser puros, es decir, que describan sólo una dimensión, o pueden ser mixtos, combinando dimensiones.

¹ Tecnología Perfecta: Supuesto que existe la tecnología con la capacidad y rapidez ilimitadas para implementar cualquier diseño conceptual.

2.2 Herramientas para modelar sistemas

Se pueden definir dos formas de clasificar las herramientas y diagramas. La primera forma es según la dimensión que modelan y la segunda es clasificar los modelos según el nivel de abstracción con que éstos representan al sistema.

La primera forma, según la dimensión que modelan, clasifica los modelos de acuerdo a si representan elementos y/o relaciones de una o más perspectivas, ya sea la Estática, Funcional o Dinámica. De esta manera se pueden encontrar los llamados “modelos puros”, que son los modelos que representan elementos o relaciones de una sola dimensión, por ejemplo: el Diagrama Entidad Relacionamento describe las relaciones (relacionamientos) entre elementos de la perspectiva estática (entidades). Otro tipo de modelos que se pueden encontrar con esta clasificación son los “modelos mixtos”, estos son modelos que representan elementos y/o relaciones de más de una dimensión, por ejemplo: el Diagrama de Actividades describe la relación dinámica (paralelismo, secuencia, sincronización) entre elementos de la dimensión funcional (procesos o actividades); la Red de Petri es un modelo dinámico, pero en su variante Canal/Actividad, representa también aspectos funcionales.

La tabla 2.1, muestra los modelos más utilizados clasificados de esta forma, es decir, se muestran los modelos en las filas, y en las columnas, la dimensión que éstos describen. En esta tabla se puede observar cuáles modelos son “puros” y cuales son “mixtos”, de acuerdo a las definiciones entregadas en el párrafo anterior.

Tabla 2.1 – Modelos y dimensiones que describen.

Modelo	Abreviatura	Dimensión		
		Dinámica	Estática	Funcional
Diagrama Entidad Relacionamento	DER		3	
Diagrama de Clases	DC		3	3
Diccionario de Datos	DD		3	
Diagrama de Casos de Uso	DCU	3		3
Diagrama de Flujo de Datos	DFD		3	3
Especificación de Procesos	EP		3	3
DFD con Flujos de Control	DFD c/FC	3	3	3
Diagrama de Actividades	DA	3		3
Diagrama de Transición de Estados	DTE	3		
Redes de Petri	RP	3		3
Diagrama de Estados	DE	3		
Diagramas de Interacción	DI	3		

La segunda forma de clasificar los modelos es según el nivel de abstracción que éstos tienen, ya sea que representen las relaciones entre los elementos o los describan. Entre los elementos de cada dimensión se pueden encontrar: entidad, objeto, proceso, estado. Por otra parte, entre las relaciones de cada dimensión se pueden encontrar: relacionamiento, flujo de datos, evento.

Aunque el nivel de abstracción está dado también por la forma en que se integren los modelos, la tabla 2.2 muestra una clasificación de los modelos más utilizados, es decir, si muestran relaciones o descripciones de los elementos de una o más dimensiones, por ejemplo: el DER muestra las relaciones (relacionamientos) entre elementos (entidades) de la dimensión estática y el DD describe estos elementos; el DFD representa relaciones (flujos de datos) entre elementos (procesos) de la dimensión funcional, pero también se utiliza para describir los procesos a través de subprocesos, o procesos de menor nivel, por esta razón aparece en ambas categorías, lo mismo ocurre con el DFD c/FC.

Tabla 2.2 – Modelos y nivel de abstracción.

Modelo	Abreviatura	Nivel de abstracción	
		Relaciones entre elementos	Descripción de los elementos
Diagrama Entidad Relacionamiento	DER	3	
Diagrama de Clases	DC	3	3
Diccionario de Datos	DD		3
Diagrama de Casos de Uso	DCU	3	
Diagrama de Flujo de Datos	DFD	3	3
Especificación de Procesos	EP		3
DFD con Flujos de Control	DFD c/FC	3	3
Diagrama de Actividades	DA	3	
Diagrama de Transición de Estados	DTE	3	3
Redes de Petri	RP	3	3
Diagrama de Estados	DE	3	3
Diagramas de Interacción	DI	3	

Existen muchas herramientas y modelos para representar los distintos elementos y relaciones en los sistemas. A continuación se describen, clasificados según la dimensión que representan, los modelos considerados más importantes y consolidados.

2.2.1 Dimensión estática

El modelamiento estático define los datos y la información que el sistema debe memorizar o almacenar. Desde esta perspectiva interesa encontrar y modelar los datos que son importantes de “recordar” por el sistema, lo que permanece en el tiempo. Es una perspectiva pasiva y permanente, donde importa la estructura de datos y sus relaciones. Para modelar esta perspectiva se utilizan principalmente dos modelos: el Diagrama Entidad Relacionamiento y el Diagrama de Clases. Ambos modelos pueden complementarse con un Diccionario de Datos.

2.2.1.1 Diagrama Entidad Relacionamiento (DER)

El DER describe cómo se relacionan los elementos estáticos. Este modelo se compone de tres piezas claves: las entidades, los atributos que describen a las entidades, y los relacionamientos que conectan las entidades entre sí, los que se explican a continuación:

- **Entidad:** Es la representación de cualquier composición de información que deba almacenar el sistema, generalmente tiene un gran número de propiedades o atributos diferentes. Una entidad, a diferencia de un objeto, sólo encapsula datos. En el DER se representan como cajas.
- **Atributo:** Los atributos describen las propiedades de una entidad, las que pueden representarse en el DER o bien describirse en el Diccionario de Datos. Los atributos pueden representar la ocurrencia de una entidad, describir la ocurrencia, o referirse a otra ocurrencia del modelo. Además, un tipo especial de atributo es el identificador, el cual siempre debe ser único y se utiliza para referenciar a una instancia específica. La cantidad, el tipo y el nombre de los atributos dependen del contexto del sistema modelado.
- **Relacionamiento:** Las entidades se conectan entre sí a través de los relacionamientos. Estos representan las conexiones entre entidades que son relevantes para el sistema. Cada trío “entidad – relacionamiento – entidad” tiene asociada una cardinalidad, ésta representa el número mínimo y máximo de ocurrencias de una entidad (instancia) que se pueden relacionar con ocurrencias de la otra entidad. Generalmente la cardinalidad se expresa como (0,n), (1,1) o (1,n). Se representan como rombos en el DER.

En el Diagrama Entidad Relacionamiento se pueden definir Jerarquías de Herencia o Tipos, lo que se realiza a través de relacionamientos de subconjunto, donde el conjunto del cual son tomados los subconjuntos es denominado padre o súperentidad y los subconjuntos son denominados hijos o subentidades. Todas las propiedades (o atributos) y relacionamientos del padre son válidos para sus hijos. Un tipo de entidad puede estar involucrado en más de una jerarquía de herencia.

Además, una forma de obtener una abstracción mayor es jerarquizando el DER, ya que se pueden agrupar algunos componentes en *clusters*. En [Teorey89], se propone el *clustering*

como una manera de simplificar el modelo y lograr una mejor comprensión de éste por parte del usuario final, pero sin bajar el nivel de detalle para satisfacer las necesidades de los analistas. La agrupación se basa en la cohesión de los elementos, donde la cohesión se define como la fuerza interna de los relacionamientos. Un *cluster* será mejor en la medida que tenga mayor cohesión.

En [Teorey89] y en [Feldman&Miller89], se definen grados de cohesión, los cuales son: dominancia, abstracción, restricción, relacionamiento simple, relacionamiento múltiple, sin relacionamiento. Éstos se explican a continuación.

- **Dominancia:** Una entidad relacionada con entidades a través de relacionamientos con cardinalidades máximas, del tipo 1:n (uno es a ene), tiene máxima cohesión debido a la dependencia de existencia en una sola entidad.
- **Abstracción:** Esta cohesión no es tan fuerte como la dependencia de existencia o la dominancia, es la relación entre todas las subentidades bajo una misma entidad del tope de una jerarquía de herencia.
- **Restricción:** Moderadamente fuerte, entidades relacionadas mediante restricciones adicionales sobre los relacionamientos.
- **Relacionamiento simple:** Hay un espectro de cohesión dentro de las relaciones no restrictivas, con relaciones unitarias o de autorelacionamiento que tienen la cohesión más alta, luego las binarias uno-a-uno, posteriormente las binarias uno-a-muchos, y finalmente las binarias muchos-a-muchos.
- **Relacionamiento múltiple:** Las relaciones del grado ternarias y más altas tienen la relación de cohesión más baja, debido a la falta de dominancia de una entidad en cada relación.
- **Sin relacionamiento:** Es el resultado de cohesionar entidades bajo un criterio no estático, por ejemplo, entidades que pertenecen al mismo área funcional.

En la figura 2.1 se presenta un DER, en ella se indican las figuras con las que se representan las entidades, relacionamientos, cardinalidad, atributos e identificadores.

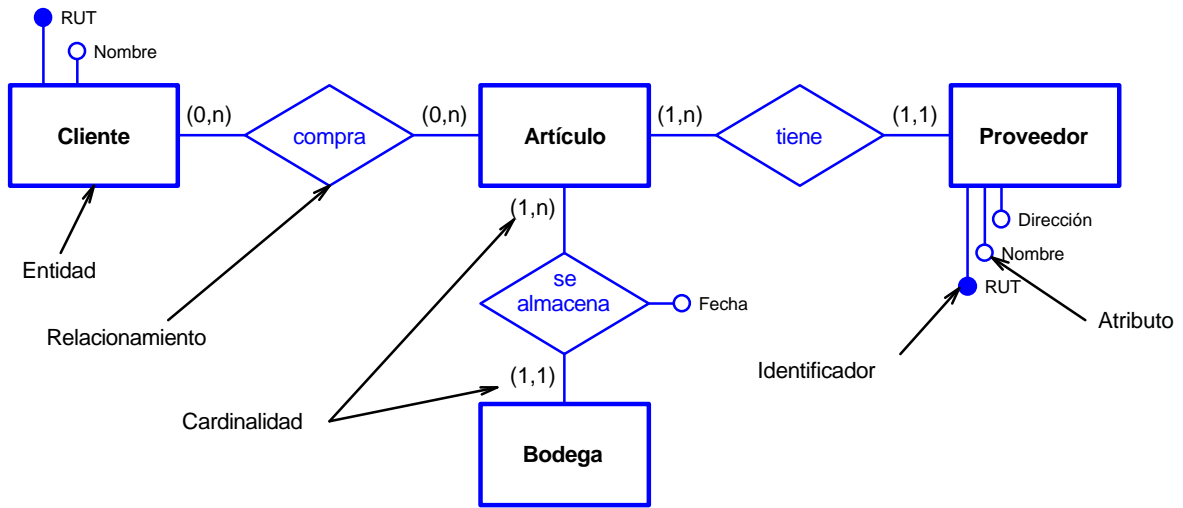


Figura 2.1: Diagrama Entidad Relación²

2.2.1.2 Diagrama de Clases (DC)

El Diagrama de Clases pertenece al enfoque de Orientación a Objeto, describe las variedades de relaciones estáticas que existen entre los tipos de objetos en el sistema. Principalmente hay dos tipos de relaciones: Asociaciones, las que son similares a los relacionamientos del DER, y los Subtipos, similar a la herencia. El Diagrama de Clases muestra los atributos y las operaciones de una clase y las restricciones que son aplicadas a los objetos conectados. Como una operación de una clase puede ser considerada un elemento funcional, a este modelo se le puede clasificar como “mixto”, ya que representa elementos de dos dimensiones.

Tal como se muestra en la figura 2.2, el Diagrama de Clases está compuesto principalmente por:

- **Objetos:** Existen muchas descripciones de objeto, ya que en general los objetos no están definidos, sino más bien, están descritos. Una descripción de Grady Booch que incluye a otras, es la citada por [Yourdon95], la cual describe un objeto como: “Una cosa tangible y/o visible, algo que puede aprender intelectualmente, algo hacia lo que se dirige el pensamiento o la acción. Un objeto tiene estados, comportamiento e identidad; la estructura y comportamiento de objetos similares se define como clase; los términos instancia y objeto son equivalentes”. Por otra parte, según [deChampeaux94] un objeto “Es identificable, tiene características que alcanzan un estado local, tiene operaciones que pueden cambiar el estado del sistema local, además de inducir operaciones en sus pares”.

² Para mayor información, véase [Batini94], [Pressman98].

- **Asociaciones:** Las asociaciones representan relaciones entre instancias de clases. Desde una perspectiva lógica o conceptual, las asociaciones representan relaciones entre clases de objetos. Cada asociación tiene dos roles, cada rol es una dirección en la asociación y pueden nombrarse con una etiqueta. Un rol puede tener multiplicidad, que es una indicación de cuántos objetos pueden participar en la relación (similar a la cardinalidad del DER).
- **Atributos:** Dependiendo del nivel de detalle en el diagrama, la notación para un atributo puede mostrar los nombres de los atributos, tipo y valor inicial. Los valores de un atributo pueden variar continuamente durante la vida del sistema o pueden que por lo general no varíen, llamando a estos últimos atributos sólo de lectura.
- **Operaciones:** Una operación define un servicio ofrecido por un objeto junto con la información que puede suministrar cuando es invocado, dicho de otra manera, son los procesos que realiza una clase. Así como hay atributos sólo de lectura, también pueden haber operaciones privadas, es decir, operaciones que sólo puedan activar instancias de la clase a la que pertenecen.

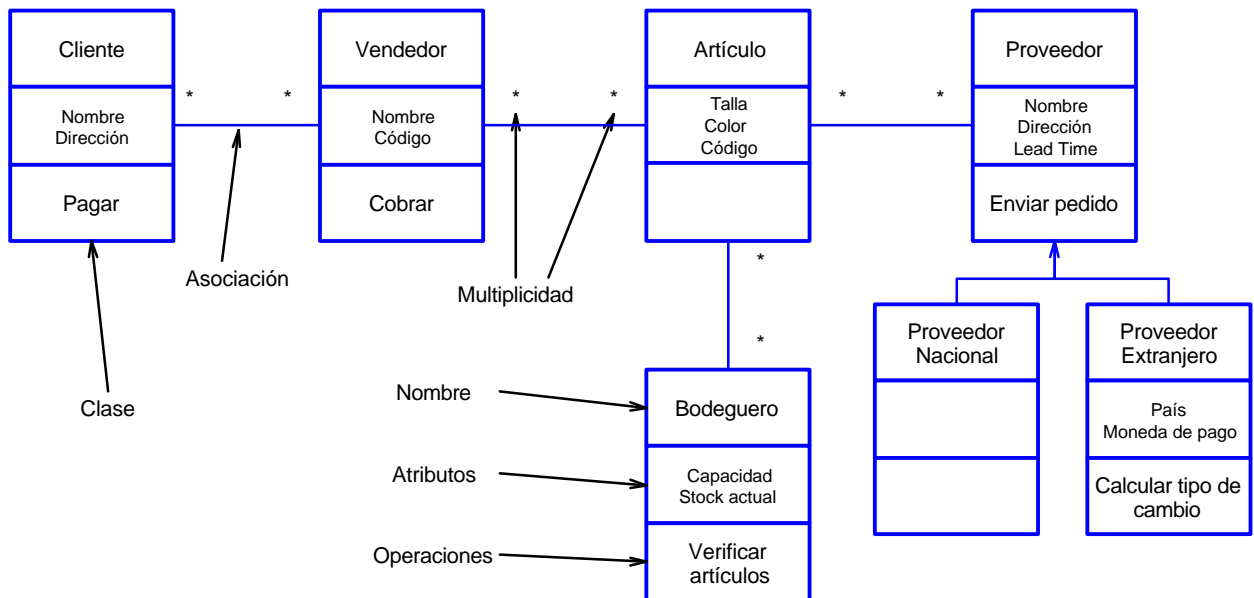


Figura 2.2: Diagrama de Clases³

2.2.1.3 Diccionario de Datos (DD)

El Diccionario de datos es un depósito que contiene definiciones y descripciones de todos los datos consumidos y producidos en el sistema. Según Yourdon, citado por

³ Para mayor información, véase [deChampeaux94], [Fowler&Scott97], [Yourdon95].

[Pressman98], pág. 222, un diccionario de datos es: “Un listado organizado de todos los elementos de datos que son pertinentes para el sistema, con definiciones precisas y rigurosas que permiten que el usuario y el analista del sistema tengan una misma comprensión de las entradas, salidas, de las componentes de los almacenes, y de los cálculos intermedios”. A esto se puede agregar que es de mucha utilidad en la mantención y redefinición del sistema, sobretodo cuando los analistas que mantienen el sistema no son los mismos que los que lo analizaron y construyeron inicialmente. Asimismo, permite mantener consistencia entre los distintos modelos utilizados. El diccionario generalmente está compuesto de:

- **Nombre:** El nombre principal de una entidad, clase, atributo, flujo de datos o depósito.
- **Alias:** Otros nombres usados para el nombre.
- **Dónde se usa/ cómo se usa:** En qué procesos se usa el elemento descrito. Además, se puede especificar en qué procesos se crea, se lee, se actualiza y se destruye un dato o flujo de datos.
- **Descripción del contenido:** El contenido es representado mediante una notación. La notación utilizada para describir los elementos en un diccionario de datos está dada en la tabla 2.3.
- **Información adicional:** Otra información opcional referente al tipo, valor inicial, restricciones, etc.

Tabla 2.3 – Simbología utilizada en un diccionario de datos.

Símbolo	Significado
=	Está compuesto de
+	Y (conjunción)
()	Optativo
{ }	Iteración o repetición
[]	Selección de alternativas
	Separador de alternativas
@	Identificador
* *	Comentario

Fuente: [Bustos99]

Ejemplo:

Proveedor = @RUT + Nombre + Dirección

Cliente = @RUT + Nombre

Dirección = Calle + Número + (Comuna) + Ciudad + País

País = * lugar de origen = [Chile|Argentina] *

Suponiendo que los proveedores pueden ser sólo de Chile o Argentina

2.2.2 Dimensión funcional

Desde la perspectiva funcional interesa modelar las transformaciones que realiza el sistema, o dicho de otra forma, la función de conversión del sistema. El sistema recibirá datos o paquetes de datos y los procesará para entregar información. En esta dimensión también se modelan las interacciones con entidades externas al sistema, lo que a su vez define los límites del sistema.

Para modelar esta perspectiva se utilizan distintos modelos, la utilización de alguno de ellos depende de la estrategia que haya elegido el analista para modelar el sistema. Estos modelos son: Diagrama de Flujo de Datos, Diagrama de Flujo de Datos con Flujos de Control, Especificación de Procesos, Diagrama de Casos de Uso (*Use Cases*) y Diagrama de Actividades, los que se describen continuación.

2.2.2.1 Diagrama de Flujo de Datos (DFD)

Las transformaciones que realiza el sistema a los datos e información para entregar las salidas requeridas son representadas con el DFD, también conocido como Grafo de Flujo de Datos o Diagrama de Burbujas. El diagrama describe cómo se relacionan los procesos a través de los flujos de datos, no indica la secuencia de procesamiento, sólo las transformaciones que se realizan a un dato o flujo de datos al atravesar al sistema. El diagrama tampoco indica si los procesos solicitan las entradas o éstas fluyen solas, ni si los procesos generan espontáneamente sus salidas o lo hacen a pedido.

Se puede utilizar el Diagrama de Flujo de Datos para representar un sistema a cualquier nivel de abstracción, tanto en forma lógica como física, dividiéndose en niveles para entregar más detalle. El diagrama más general o global es el Diagrama de Contexto o de nivel cero, el que representa el sistema completo y sus relaciones a través de flujos de entrada y salida con agentes externos o Terminadores. Posteriormente, en la primera partición o de nivel uno se obtiene el Diagrama cero, y así sucesivamente describiendo procesos con procesos de nivel menor, se llega a niveles más detallados, hasta que los procesos son descritos por medio de Especificación de Procesos.

El Diagrama de Flujo de Datos tradicional tiene cuatro componentes: Terminadores, Procesos, Flujos de Datos y Depósitos de Datos. La figura 2.3 muestra cómo se relacionan estos componentes en el Diagrama.

- **Terminador:** Es una entidad externa, también llamada Agente, produce o consume datos o información, está fuera de los límites del sistema y, por lo tanto, no es controlable. Se representa a través de cajas.
- **Proceso:** Es una función de conversión, un transformador de información que reside dentro de los límites del sistema. Generalmente se le nombra con un verbo más un sustantivo. La función puede ser una simple comparación o un sofisticado algoritmo. Cada proceso tiene a lo menos un flujo de entrada y uno de salida. Cada proceso debe

desagregarse en un nivel inferior o en una especificación de procesos. Se representan a través de circunferencias.

- **Flujo de datos:** Es un grupo de datos o información en cualquiera de sus formas, tales como: texto, video, fotos, audio; que generan o reciben los procesos. Los flujos de datos pueden ser de entrada, de salida o de diálogo. Se representan a través de flechas.
- **Depósito:** Es un almacén, en el que se guardan datos en forma ordenada para uso de uno o varios procesos del sistema que operan en distintos momentos del tiempo. Se representa a través de rectángulos con un costado abierto.

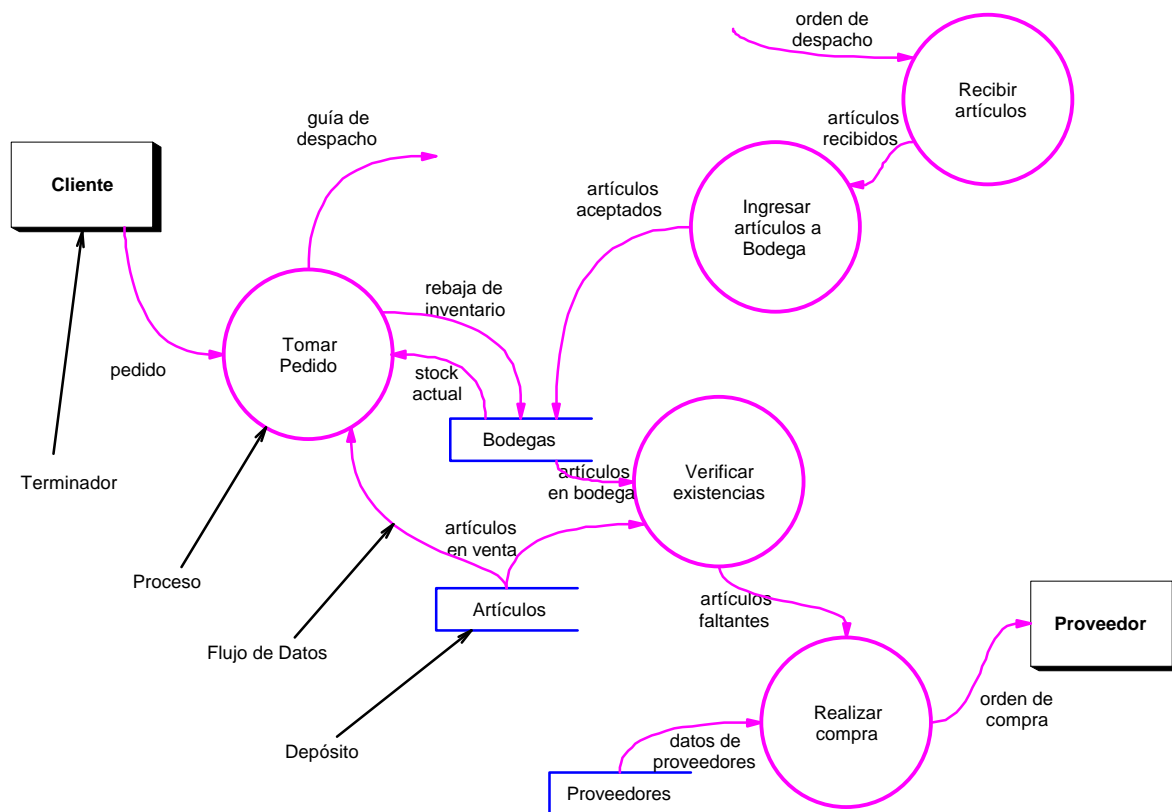


Figura 2.3: Diagrama de Flujo de Datos⁴

2.2.2.2 Diagrama de Flujo de Datos con Flujos de Control (DFD c/FC)

Cuando se requiere representar procesos con una dimensión temporal, se incorporan al DFD tradicional los flujos de control. Estos flujos son recibidos y emitidos por un proceso de control, el que activa o desactiva procesos a medida que va recibiendo señales específicas.

⁴ Para mayor información, véase [Acevedo90], [Pressman98], [Yourdon94].

Este modelo incorpora dos componentes adicionales al DFD tradicional, éstos son los flujos de control y los procesos de control.

- **Flujos de control:** Es una señal que llega o sale de un proceso de control. Los flujos de salida activan o habilitan otro proceso, ya sea tradicional o de control. Los flujos de entrada informan que un proceso ha concluido o que existe una situación de excepción. Un flujo de control se representa como un flujo de datos normal, pero con línea punteada.
- **Proceso de control:** Es un proceso al cual llegan los flujos de control y que con estas señales recibidas decide activar o desactivar algún o algunos procesos tradicionales para darles una secuencia. Un proceso de control da una primera vista de la dimensión dinámica. Posteriormente se puede describir su comportamiento a través de un Diagrama de Transición de Estados. Se representa como un proceso común, pero con línea punteada.

La figura 2.4 muestra cómo se incorpora el Proceso de Control al Diagrama de Flujo de Datos tradicional.

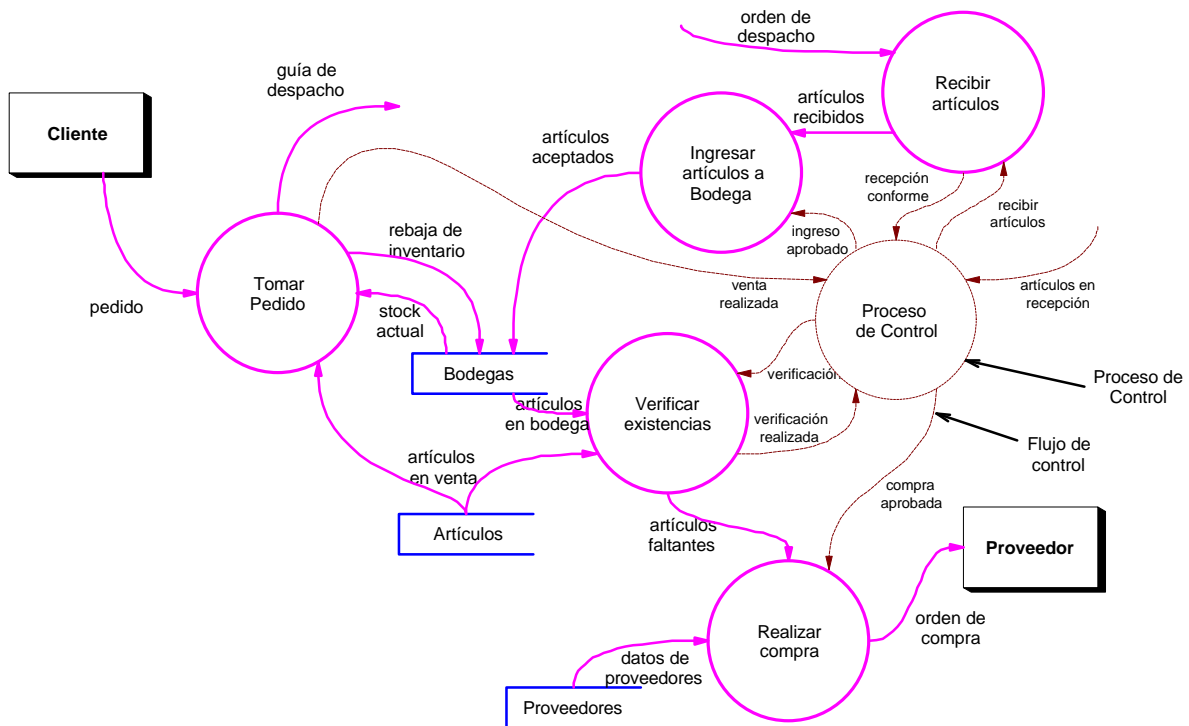


Figura 2.4: Diagrama de Flujo de Datos con Flujos de Control⁵

⁵ Para mayor información, véase [Acevedo90], [Pressman98], [Yourdon94].

2.2.2.3 Especificación de Procesos (EP)

Para describir los procesos elementales, es decir, los procesos que no se desean descomponer en otros procesos, se utiliza la Especificación de procesos, también conocida como Mini-especificaciones o Descripción de procedimientos.

La Especificación de Procesos define cómo deben transformarse las entradas en salidas en cada proceso elemental, cada proceso puede ser descrito de manera distinta. Esta descripción se puede hacer en forma narrativa, en español estructurado (pseudoalgoritmo), ecuaciones matemáticas, pre y post-condiciones, tablas, diagramas o gráficos. La Especificación de Procesos no debe imponer decisiones de implementación.

Ejemplo de especificación de procesos con pre y post-condiciones:

Proceso: Crear cuenta presupuestaria

PRE-CONDICIÓN

Cuenta presupuestaria solicitada y Asignación aprobada

POST-CONDICIÓN

Cuenta presupuestaria creada

Proceso: Licitar proyecto

PRE-CONDICIÓN 1

Recepción y conformidad de propuestas priorizadas

POST-CONDICIÓN 1

Proyecto licitado

PRE-CONDICIÓN 2

Recepción y rechazo de propuestas priorizadas

POST-CONDICIÓN 2

Proyecto con licitación rechazada y Publicar bases de licitación

2.2.2.4 Diagrama de Casos de Uso (DCU)

Un caso de uso es una representación de los procesos del sistema que son percibidos por los clientes, es decir, de los procesos visibles desde el exterior. En esencia, un caso de uso es una interacción típica entre el usuario y el sistema. Según [Fowler&Scott97], los casos de uso capturan alguna función visible del usuario y alcanzan una meta para él. [Jacobson95] introduce un diagrama para visualizar los casos de uso.

El diagrama está compuesto por los elementos de la figura 2.5.

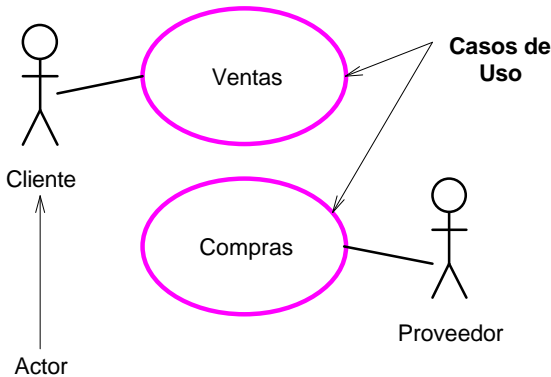


Figura 2.5: Diagrama de Casos de Uso ⁶

- **Actores:** Un actor es un rol que un usuario juega con respecto al sistema. Probablemente hay muchas personas que juegan un mismo rol, pero en lo que al sistema concierne, todos son representados por un actor que juega ese rol. Un usuario puede tener muchos roles en una organización. Cuando se definen los actores, es importante pensar en roles y no en personas o cargos de trabajo.

Los actores están fuera del caso de uso, un actor puede interactuar con varios casos de uso y un caso de uso puede interactuar con más de un actor. Las relaciones que pudieran existir entre los actores no deben representarse, y si esas relaciones son importantes para el sistema, los actores deben incorporarse en él.

Los actores no necesitan ser humanos, un actor puede ser un sistema externo que necesita información del sistema en cuestión. Qué sistemas externos incluir como actores depende del modelador y del contexto, según [Fowler&Scott97], se deben mostrar actores sólo cuando sean ellos quienes necesitan del caso de uso.

- **Usos y extensiones:** Una buena forma para identificar casos de uso es a través de los eventos externos. Pensar en todos los eventos que vienen del mundo externo y a los que el sistema debe reaccionar. Una extensión es un caso de uso que es similar a otro, pero hace algo más. Una relación de uso es un caso de uso que tiene parte del comportamiento que es común a otros casos de uso y que no se desea describir repetidamente. Ambas implican descomponer los factores que no son del

⁶ Para mayor información, véase [Fowler&Scott97], [Jacobson95].

comportamiento común de varios casos de uso a un caso de uso que será uso, o extensión de uno o varios casos de uso.

2.2.2.5 Diagrama de Actividades (DA)

El Diagrama de Actividades combina ideas de distintas técnicas: El Diagrama de Eventos de J. Odell, Técnicas de Modelamiento de Estados y Redes de Petri. Estos diagramas son particularmente usados en conjunto con los Flujos de Trabajo (*flowchart*) y se utilizan para describir el comportamiento que tiene un grupo de procesos paralelos o concurrentes. Las actividades se pueden ubicar en áreas definidas por *Swimlanes*, que dividen a través de líneas verticales las funciones en donde se realizan las actividades. Debido a que estos diagramas representan cómo se relacionan los procesos (funcionales) en el tiempo (dinámico), se clasifican como mixtos.

En un diagrama lógico o conceptual, una actividad es una tarea que necesita realizarse, ya sea manual o automatizada. Cada actividad puede ser seguida de otra actividad. Es una secuencia simple o de varias actividades con un disparador o *trigger*, que puede ser una expresión lógica “*true*” o “*false*” tal como en un Diagrama de Estados. Además, con el *trigger* se puede lograr una sincronización de las actividades. Como se muestra en la figura 2.6, en el diagrama se pueden representar actividades que se realizan en paralelo.

El Diagrama de Actividades permite elegir el orden en que se realizan las actividades o procesos. En otras palabras, muestra los estados y la secuencia de reglas que se deben seguir. Esta es la diferencia más importante con el flujo de trabajo, el cual normalmente se limita a procesos secuenciales, el diagrama de actividades puede tomar actividades en paralelo.

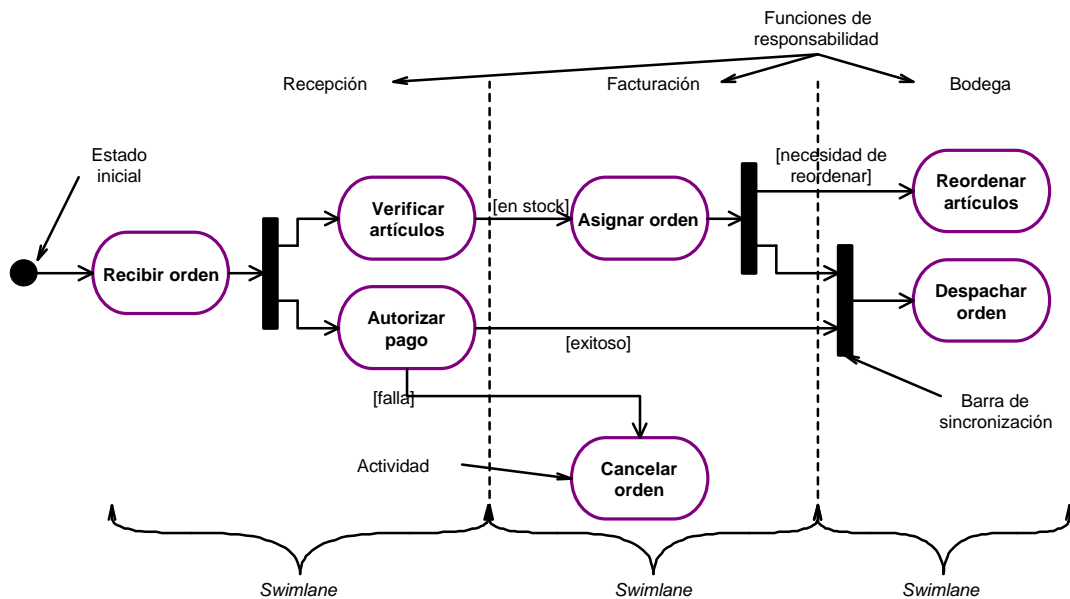


Figura 2.6: Diagrama de Actividades⁷

⁷ Para mayor información, véase [Fowler&Scott97].

2.2.3 Dimensión dinámica

Finalmente desde la perspectiva dinámica lo que interesa es el comportamiento del sistema, interesan los eventos que producirán cambios en el estado del sistema. Es una perspectiva activa y temporal con relaciones dinámicas que se crean y se destruyen durante la vida del sistema, que pasa por estados y transiciones.

Para modelar esta perspectiva se utilizan principalmente los siguientes modelos: Diagramas de Transición de Estados, Diagramas de Estado, Redes de Petri y Diagramas de Interacción.

2.2.3.1 Diagrama de Transición de Estados (DTE)

El Diagrama de Transición de Estados describe el comportamiento del sistema basándose en una máquina hipotética, la Máquina de Estados Finitos. La máquina responde a un *input*, generando un *output* y cambiando internamente de estado, posee un conjunto finito de estados y en cada instante de tiempo se encuentra en un único estado. El *output* y el próximo estado dependen del estado actual y del *input*.

El Diagrama de Transición de Estados se puede particionar, utilizando una Jerarquización de Estados, tal como la jerarquización de procesos en un DFD.

Existen dos variantes del DTE, si la acción se representa en la transición se utiliza la variante de Mealy, o si la acción ocurre con el estado se utiliza la variante de Moore. Ambas son matemáticamente equivalentes y pueden combinarse; la elección de cuál variante usar depende del sistema, del contexto y del analista. Además, los DTE se pueden complementar con tablas de Transición de Estados, las que también se distinguen según la variante utilizada. Los elementos de una Máquina de Estados Finitos son:

- **Estado:** Es un modo observable de comportamiento o circunstancia en que se puede encontrar el sistema y que lo caracteriza. Puede ser una espera que algo ocurra en el exterior o que una actividad externa concluya. Los estados no esperan acciones realizadas por el sistema.
- **Transición:** Cambio de estado producido en función de un evento.
- **Evento:** Hecho relevante (*input*) generado externa o internamente. Pueden ser simples o compuestos, incorporándole operadores lógicos (AND, OR, NOT).
- **Acción:** Es una operación o actividad instantánea realizada por el sistema (*output*).

La figura 2.7 muestra un DTE, indicando en ella los estados, acciones, eventos, transiciones y cómo se jerarquizan los estados.

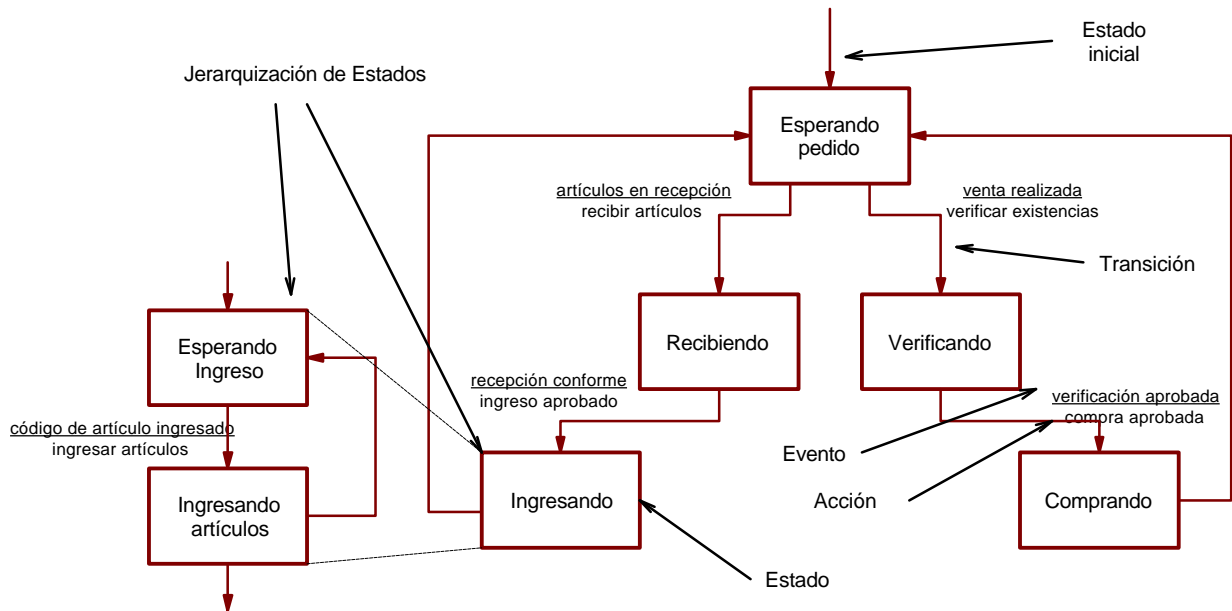


Figura 2.7: Diagrama de Transición de Estados variante Mealy⁸

2.2.3.2 Red de Petri (RP)

Las Redes de Petri son una herramienta gráfica y matemática para describir sistemas concurrentes. Existen muchas aplicaciones de Redes de Petri, con sus respectivas variaciones. Como se muestra en la figura 2.8, una Red de Petri elemental está compuesta de Lugares, Conexiones, Arcos, Ramas restauradores y Anotaciones opcionales, en el Anexo 1 se describe este modelo con mayor detalle.

- **Lugar:** Es un estado local del sistema y determina características propias.
- **Conexión:** Es un evento que produce que el sistema pase de un lugar a otro o a otros.
- **Arco:** Conecta lugares con conexiones.
- **Anotaciones opcionales:** Anotaciones que se pueden agregar al modelo, con el objetivo de esclarecer algún punto que no esté claro.
- **Ramas restauradoras:** Caso especial de Arcos, las que verifican estados del sistema sin modificarlos.

⁸ Para mayor información, véase [Pressman98], [Yourdon94].

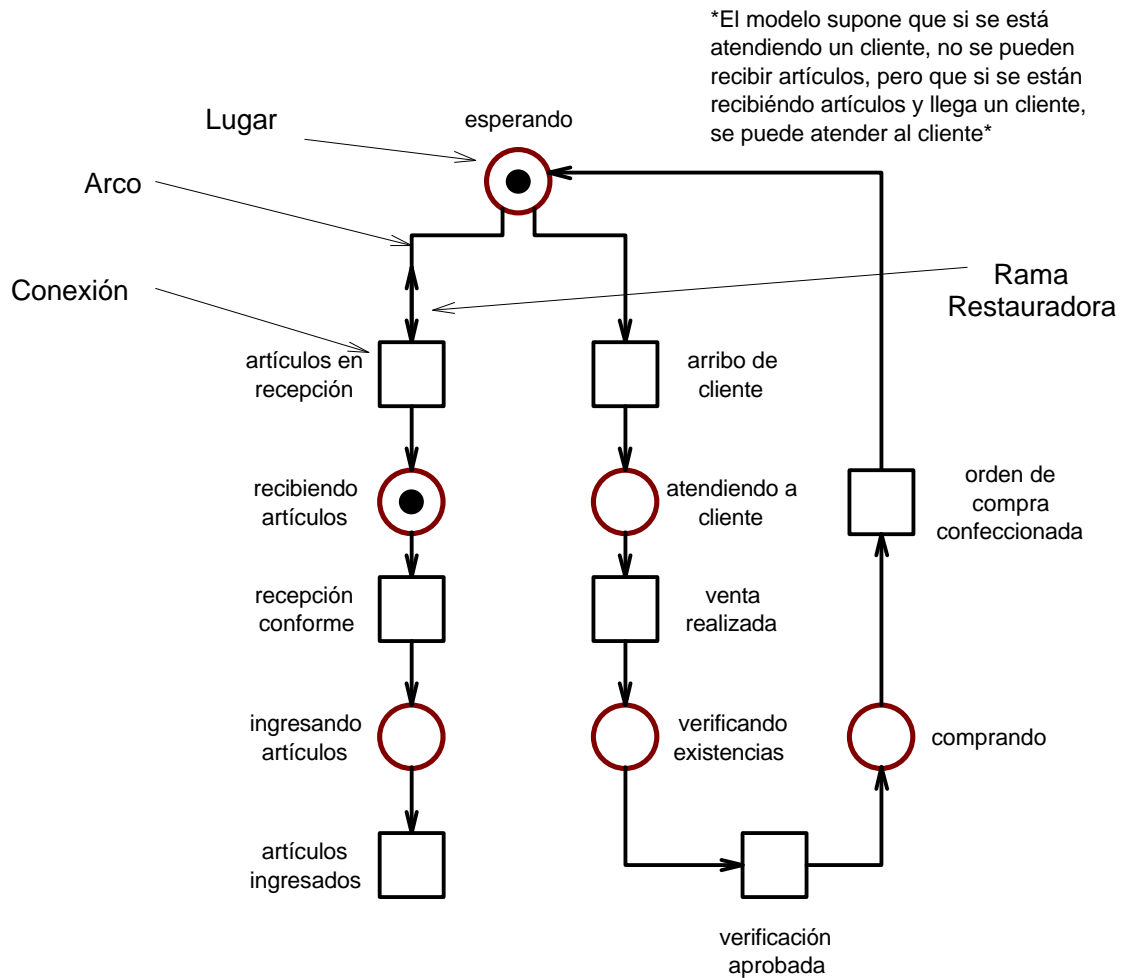


Figura 2.8: Red de Petri⁹

2.2.3.3 Diagrama de Estados (DE)

El Diagrama de Estados es un modelo que describe el comportamiento del sistema o de un objeto, es una variación de los *Statecharts* desarrollados por [Harel87]. Describe todos los posibles estados de un objeto particular y cómo los demás objetos pueden cambiar el estado de un objeto. También es usado para mostrar el ciclo de vida del comportamiento de un objeto.

Se usa el término “acción” para la transición y “actividad” para el estado. Las acciones son asociadas a transiciones, son procesos que ocurren rápidamente y no se pueden interrumpir. Las actividades son asociadas con estados y pueden durar más tiempo. Una actividad puede ser interrumpida por un evento.

⁹ Para mayor información, véase [Davis93], [Heuser90].

En este tipo de diagramas, así como en los *Statecharts*, se puede representar la concurrencia. Un Diagrama de Estado concurrente es usado cuando un objeto tiene un set de comportamientos independientes. Se puede dar una orden chequeando si un objeto se encuentra en un estado particular. Además, como se muestra en la figura 2.9, se puede incluir modularidad, composición, y jerarquización.

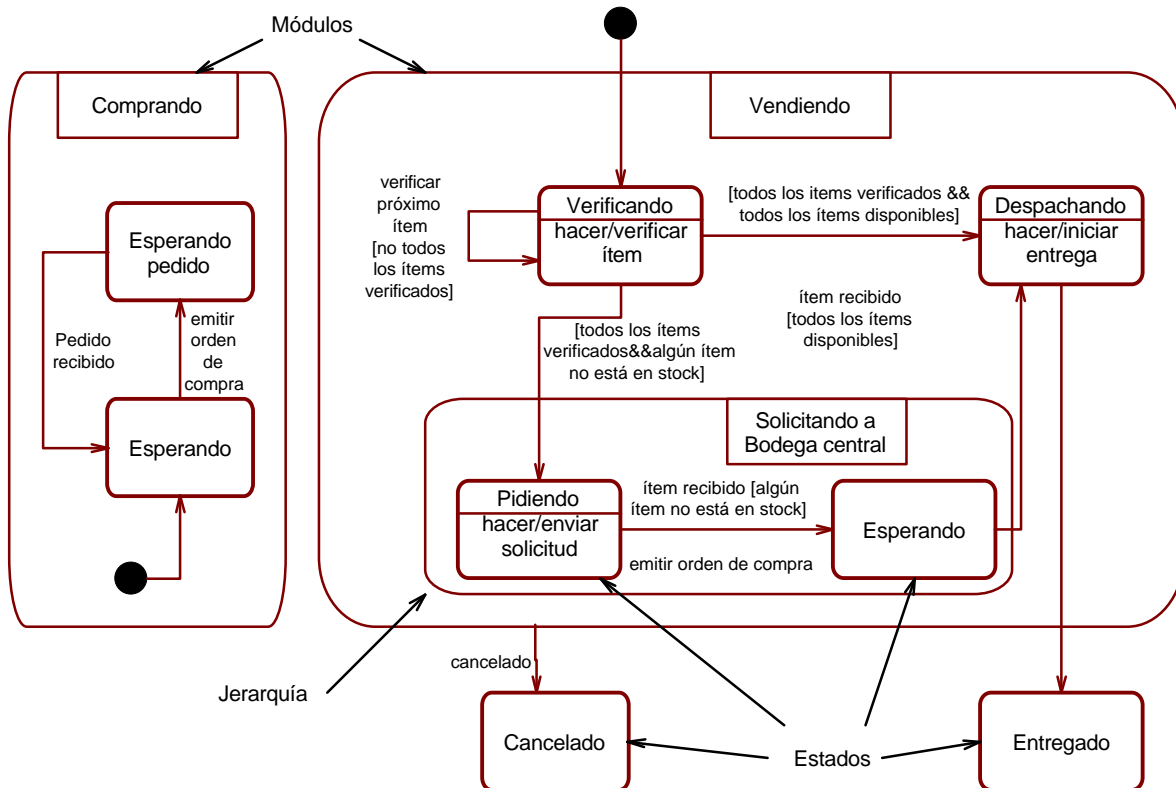


Figura 2.9: Diagrama de Estados¹⁰

2.2.3.4 Diagrama de Interacción (DI)

[Fowler&Scott97], pág. 103, define los Diagramas de Interacción como: “Modelos que describen cómo grupos de objetos colaboran en el comportamiento. Generalmente, un Diagrama de Interacción captura el comportamiento de un caso de uso. El diagrama muestra un número de objetos y los mensajes que se traspasan entre ellos, dentro del caso de uso”.

Hay dos tipos de diagramas de Interacción: los Diagramas de Secuencia y los Diagramas de Colaboración, a continuación se explica cada uno de ellos.

¹⁰ Para mayor información, véase [Fowler&Scott97], [Harel87].

2.2.3.4.1 Diagrama de Secuencia (DS)

En el Diagrama de Secuencia, un objeto es mostrado como una caja en el tope de una línea vertical. Esta línea vertical es llamada “línea de vida” o “*lifeline*”. La línea de vida representa los objetos durante la interacción.

Como se muestra en la figura 2.10, cada mensaje es representado por una flecha entre las líneas de vida de dos objetos. El orden en que los mensajes ocurren es mostrado de arriba hacia abajo. Cada mensaje es etiquetado como mínimo con su nombre, se pueden incluir los argumentos y la información de control, también se pueden mostrar las autodelegaciones, es decir, mensajes que un objeto envía a sí mismo.

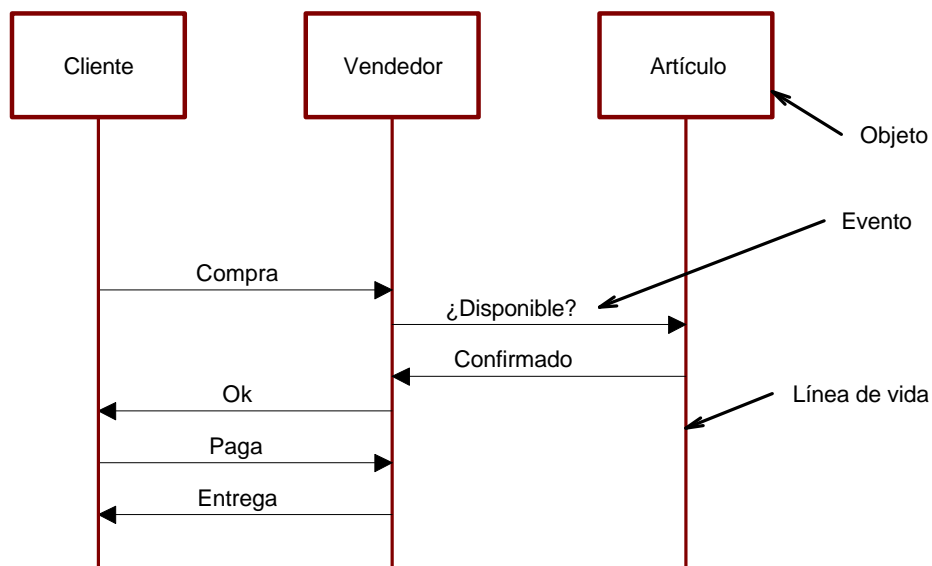


Figura 2.10: Diagrama de Secuencia¹¹

2.2.3.4.2 Diagrama de Colaboración (DCO)

En el Diagrama de Colaboración los objetos son representados con cajas, como se muestra en la figura 2.11. Tal como en un Diagrama de Secuencia, los mensajes enviados entre los objetos se indican por medio de flechas. Esta vez, la secuencia es indicada por medio de la numeración de los mensajes.

Al numerar los mensajes se hace más difícil ver la secuencia que tocan las líneas de vida. Por otra parte el orden espacial muestra otras cosas más fácilmente. Se puede ver cómo los objetos se enlazan y se usa el *layout* de los objetos para ordenar conjuntos de información.

[Fowler&Scott97], recomienda usar el Diagrama de Colaboración cuando se quiere ver el comportamiento de los objetos dentro de un caso de uso. Si se quiere ver el comportamiento

¹¹ Para mayor información, véase [Fowler&Scott97].

de un objeto en varios casos de uso se recomienda usar un Diagrama de Estados. Si se quiere ver el comportamiento de muchos casos de uso, se puede usar el Diagrama de Actividades.

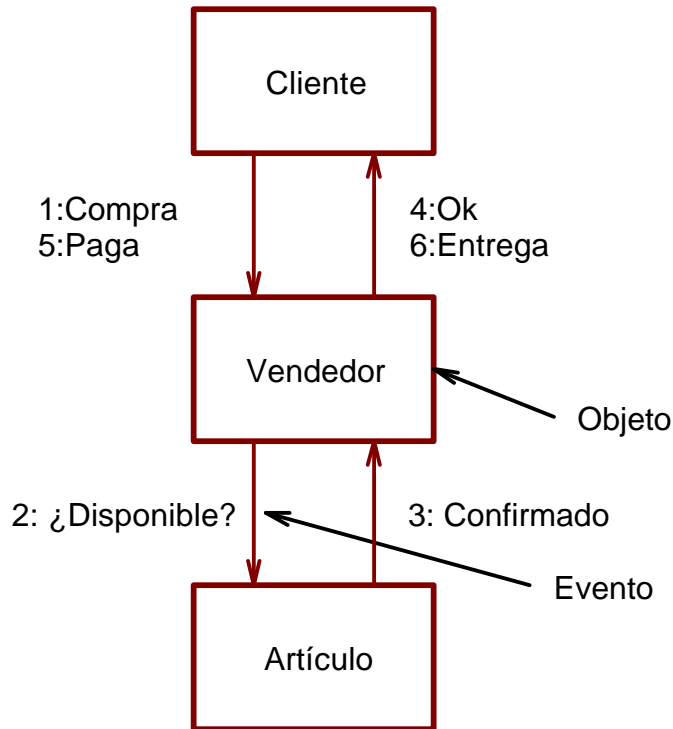


Figura 2.11: Diagrama de Colaboración¹²

Para finalizar este capítulo, se muestra la figura 2.12 que intercepta las tablas 2.1 y 2.2 mostradas al comienzo del capítulo, sintetizando de esta forma los modelos descritos. En esta figura se clasifican los diagramas de acuerdo a las dos formas: según la dimensión que modelan y según el nivel de abstracción con que éstos representan al sistema. Además, en la figura se muestran los elementos que se modelan en cada dimensión y cómo se puede obtener un nivel de abstracción mayor utilizando la jerarquización de los elementos.

Se debe considerar que hay modelos que por su naturaleza se pueden utilizar para representar elementos de dos dimensiones, como es el caso de los Diagramas de Casos de Uso que representan procesos y los eventos con que interactúan. Por otra parte, hay modelos que se pueden utilizar tanto para describir los elementos como para describir las relaciones entre ellos, en el caso del Diagramas de Actividades o el Diagrama de Flujo de Datos.

Finalmente, en relación con la figura 2.12, hay que aclarar que la sobreposición entre el DFD c/FC y el DA sólo se produce por efecto de utilización de espacio, es decir, ambos

¹² Para mayor información, véase [Fowler&Scott97].

modelos se pueden utilizar para describir elementos o relaciones entre los elementos, tanto de la dimensión funcional como de la dimensión dinámica.

NIVEL DIMENSIÓN	ELEMENTOS A MODELAR	RELACIONES ENTRE ELEMENTOS	DESCRIPCIÓN DE LOS ELEMENTOS	JERARQUIZACION DE LOS ELEMENTOS
DIMENSIÓN ESTÁTICA	Entidades	<div style="border: 1px solid blue; padding: 5px; display: inline-block;"> DER Relacionamientos </div>	<div style="border: 1px solid blue; padding: 5px; display: inline-block;"> Identificadores DD Atributos </div>	Jerarquización del DER: <i>Clusters</i>
	Objetos	<div style="border: 1px solid blue; padding: 5px; display: inline-block;"> Asociaciones Atributos DC Operaciones </div>		
DIMENSIÓN FUNCIONAL	Procesos	<div style="border: 1px solid blue; padding: 5px; display: inline-block;"> Asociaciones Atributos DC Operaciones </div>	<div style="border: 1px solid magenta; padding: 5px; display: inline-block;"> EP Especificación de Procesos Descripción Narrativa Subprocesos </div>	Jerarquización del DFD DFD de nivel más alto
		<div style="border: 1px solid magenta; padding: 5px; display: inline-block;"> Flujos de datos DFD Subprocesos </div>	<div style="border: 1px solid magenta; padding: 5px; display: inline-block;"> Flujos de datos Subprocesos DA Actividades </div>	
DIMENSIÓN DINÁMICA	Procesos de Control	<div style="border: 1px solid magenta; padding: 5px; display: inline-block;"> Usos y Extensiones DCU Eventos </div>	<div style="border: 1px solid magenta; padding: 5px; display: inline-block;"> Flujos de datos Subprocesos DFD c/FC </div>	Jerarquización de RP: RP Canal/Actividad Jerarquización del DTE: DTE de nivel más alto Jerarquización del DE: DE de nivel más alto o Superestado
	Sistema	<div style="border: 1px solid red; padding: 5px; display: inline-block;"> DI Flujos de Interacciones </div>	<div style="border: 1px solid red; padding: 5px; display: inline-block;"> Transiciones RP Estados </div>	
	Objetos		<div style="border: 1px solid red; padding: 5px; display: inline-block;"> Transiciones DTE Estados </div> <div style="border: 1px solid red; padding: 5px; display: inline-block;"> Transiciones DE Estados </div>	

Figura 2.12: Mapa dimensión – nivel de abstracción

3 Principales estrategias de integración de modelos

3.1 Fundamentos

Una decisión que debe tomarse antes de modelar es escoger una metodología. La metodología, de alguna manera, define la forma en que se abordará el problema, los elementos que se busca representar en cada dimensión, los modelos que se pueden utilizar para representarlos. Una metodología debiera contener implícitamente una estrategia de integración.

En el capítulo anterior fueron descritos los principales modelos que se utilizan para representar y describir un sistema desde las tres dimensiones o perspectivas. Cómo se relacionan y construyen los modelos que representen el sistema, depende del enfoque utilizado para modelar sistemas y consecuentemente de la estrategia de integración que se escoja.

Tal como muestra la figura 3.1, el desarrollo de sistemas se puede dividir en cuatro grandes etapas, éstas son: Definición del problema, Análisis, Diseño e Implementación.

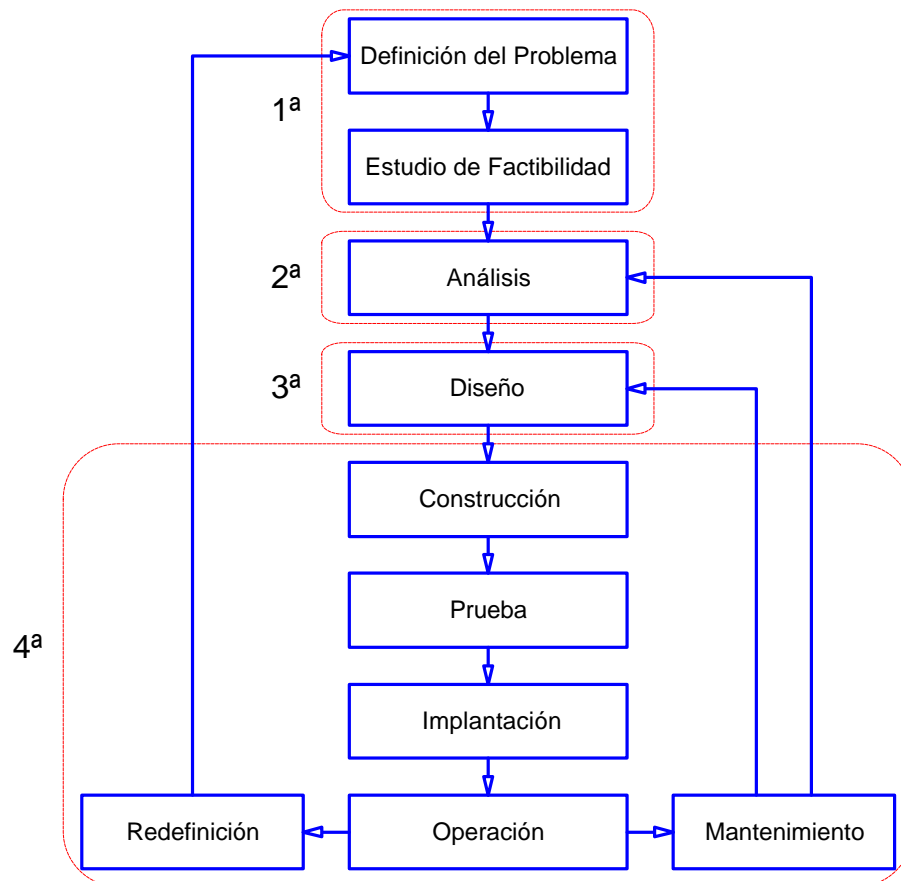


Figura 3.1: Etapas del desarrollo de Sistemas

En la primera etapa, se define cuál es el problema a resolver y se verifica, bajo distintos criterios, si es conveniente resolverlo mediante el desarrollo de un sistema de información. El resultado de esta etapa debe ser la aprobación o el rechazo del proyecto de desarrollo de sistemas y la definición clara del problema y de sus alcances.

En la segunda etapa se deben construir los modelos conceptuales que representen al sistema. El resultado de esta etapa es el diseño lógico o conceptual del sistema de información, constituido por un conjunto de modelos que representan de manera consistente al sistema desde las perspectivas dinámica, estática y funcional. Tal como se mencionó en el capítulo 2, estos modelos deben centrarse en “qué” hará el sistema y no en “cómo” lo debe hacer, sin tener en cuenta las restricciones de implementación.

En la etapa siguiente, es donde el modelo conceptual es transformado en un modelo de implementación. Como resultado se obtiene el diseño físico, en el que se incorporan las restricciones organizacionales y/o tecnológicas, cambios organizacionales y los requisitos de implementación.

En la etapa final, se implementa el sistema de información y se implanta en la organización, obteniendo así la solución al problema planteado.

La calidad de la solución desarrollada va a depender en gran medida de la claridad y exactitud en la definición del problema que se pretende resolver, de su representación conceptual, y de la solución implementada para resolverlo. Como la implementación responde a la concepción que se obtuvo del sistema en la segunda etapa, la importancia de contar con instrumentos adecuados y criterios de modelado para asegurar un modelo conceptual completo, correcto, expresivo y consistente, entre otros atributos, es un requisito ineludible de la calidad de la solución implantada.

Se han utilizado o propuesto muchos enfoques para modelar sistemas, los que han tenido diferentes grados de éxito. Inicialmente se utilizó el análisis documentado de sistemas, el cual describió el sistema en lenguaje natural. Esto acarreaba escribir muchas páginas intentando describir el sistema y, dada su ambigüedad e imprecisión, generalmente no lograba describirlo correctamente. Posteriormente, fueron propuestos distintos enfoques. En general, estos enfoques se pueden dividir en dos grandes categorías: los “dirigidos por métodos” y los “dirigidos por modelos”. Los primeros indican la metodología que se debe seguir para modelar un sistema, indicando el orden en que se deben construir los modelos en cada dimensión y las reglas de balanceo entre ellos, mientras que los segundos indican los modelos a utilizar en cada dimensión y las reglas de balanceo entre ellos, sin indicar el orden en que se deben construir.

Los objetivos de enmarcarse dentro de una estrategia definida son: asegurar consistencia entre los modelos, integrar la Especificación de Requerimientos, y de esta forma, evitar errores en las fases siguientes. Por otra parte, la estrategia a adoptar depende del contexto, de la experiencia del analista, de las políticas y planes informáticos de la organización, etc.

Se describirán tres formas de integrar los modelos: Análisis Estructurado Moderno (AEM), que sigue las reglas de balanceo según la metodología de Yourdon; Estrategia Genérica (EG), que no depende de ninguna metodología; y Orientación a Objetos (OO), usando la notación estandarizada de *Unified Modeling Language* (UML). La principal diferencia entre estas estrategias es el foco de la dimensión dinámica, la rigurosidad en las reglas de balanceo y los distintos enfoques que adoptan.

La tabla 3.1 presenta los elementos básicos que se modelan en cada estrategia, así como los modelos utilizados para describirlos y para modelar las relaciones entre ellos.

Tabla 3.1 – Elementos, relaciones y descripción en las estrategias de integración.

		Niveles	DIMENSIÓN		
			Estática	Funcional	Dinámica
ESTRATEGIA	AEM	Elementos	Entidades	Procesos	Procesos de Control
		Relaciones	DER	DFD	DFD c/FC
		Descripción	DD	DFD EP	DTE
	EG	Elementos	Entidades	Procesos	Sistema
		Relaciones	DER	DFD	
		Descripción	DD	DFD EP	RP o DE
	UML	Elementos	Objetos	Procesos	Objetos
		Relaciones	DC	DCU	DI
		Descripción	DD	DA Descripción Narrativa	DA DE

A continuación se describen las estrategias de integración ya nombradas.

3.2 Análisis Estructurado Moderno (AEM)

Un enfoque para modelar, dirigido por método y que se apoya en la rigurosidad, es el Análisis Estructurado Moderno. Enmarcándose en esta estrategia, los analistas describen el sistema como una red de procesos que interactúan. Para esto se utiliza el Diagrama de Flujo de Datos incorporándole Procesos de Control, posteriormente se construye el Diagrama Entidad Relacionamiento y el Diccionario de Datos para describir la dimensión estática, y para describir el comportamiento de los Procesos de Control se utilizan Diagramas de Transición de Estados.

Uno de los problemas de modelar dirigido por método es que el analista puede dejar de lado el objetivo principal, que es modelar el sistema en estudio, y concentrarse en detalles y en cómo modelar el sistema con la metodología escogida.

Por otra parte, según [Embley92], la secuencia de pasos establecida, no puede ser siempre seguida, por lo que los analistas deben adaptar el método haciendo excepciones a las reglas del AEM. Por lo tanto, el éxito de los analistas al modelar un sistema radica en entender los principios del modelamiento y su experiencia para hacer las excepciones apropiadas. Como resultado, cuando los analistas inexpertos intentan seguir los pasos del AEM, normalmente se frustran.

La tabla 3.2 muestra cómo los modelos deben ser consistentes entre sí.

Tabla 3.2 – Balanceo entre modelos utilizados en AEM.

	DFD	EP	DER	DD	DTE
DFD	4	4	4	4	4
EP	4		4	4	
DER	4			4	
DD	4	4	4	4	
DTE	4				4

Fuente: [Bustos99]

[Bustos99] nombra las siguientes reglas de balanceo:

4 DFD - DD:

Todo depósito y flujo de datos del DFD debe estar definido en el DD.

4 DD - DFD:

Todo depósito y flujo de datos definido en el DD debe estar en el DFD.

4 DER - DD:

Toda entidad y relacionamiento del DER deben estar definidos en el DD.

4 DD - DER:

Toda entidad y relacionamiento con atributos definido en el DD debe estar en el DER.

4 DFD - DFD - EP:

Todo proceso del DFD tiene asociado un DFD de nivel menor o una EP, pero no ambos.

4 EP - DFD:

Toda EP describe a un proceso elemental del DFD.

Todas las acciones en la EP deben aparecer como entradas, salidas y accesos a depósitos del DFD.

Cada dato en la EP es flujo o depósito del proceso en el DFD.

4 EP - DD:

Cada dato en la EP es componente de flujo o depósito en el DD.

En la EP pueden haber términos locales al proceso.

4 DD - EP - DFD - DER:

Cada definición del DD debe tener referencia en una EP, DFD, DER o en el DD.

4 DFD - DER:

Todo depósito del DFD es una entidad o un relacionamiento con atributos en el DER.

Los nombres de los depósitos (en plural) y entidades (en singular) deben coincidir.

Debe existir algún proceso del DFD que define valores para cada atributo del DER.

4 DER - DFD:

Los nombres de entidades (en singular) y de los depósitos (en plural) deben coincidir.

Todas las entidades, subentidades y relacionamientos con atributos del DER deben también aparecer como depósitos en el DFD.

4 EP - DER:

En la EP existen acciones de creación y eliminación de instancias de entidades y relacionamientos del DER.

4 DFD - DTE:

Cada flujo de control que entra al proceso de control del DFD es un evento en el DTE.

Cada flujo de control que sale del proceso de control del DFD es una acción en el DTE.

4 DTE - DFD:

Cada evento en el DTE es un flujo de control que entra al proceso de control del DFD.

Cada acción en el DTE es un flujo de control que sale del proceso de control del DFD.

4 DTE - DTE:

Estados particionados tienen otro DTE asociado.

3.3 Estrategia Genérica (EG)

Otro enfoque de análisis es el orientado a los modelos, el cual se concentra en construir un modelo del sistema bajo estudio. Este modelo captura características específicas del sistema y posteriormente se complementa con otros modelos que apunten a describir mejor el sistema. A diferencia de la especificación del orden de construcción de los modelos, los analistas, guiados por los principios del modelamiento conceptual, construyen modelos como mejor les parezca según las necesidades, consiguiendo ganar más visión y conocimiento del sistema.

Dentro de este enfoque se inserta la Estrategia Genérica, la cual se basa en la existencia de un único modelo dinámico, a través de *Statecharts* o Redes de Petri, que representa el comportamiento del sistema como un todo. Además, muestra concurrencia entre sus roles y sus tareas, donde cada tarea responde a un estímulo. Por otra parte, posee reglas de balanceo flexibles y es recomendable la utilización de un Diagrama de Flujo de Datos.

La tabla 3.3 muestra qué modelos utiliza y cómo deben balancearse entre sí.

Tabla 3.3 – Balanceo entre modelos utilizados en EG.

	DFD	EP	DER	DD	RP
DFD	4	4	4	4	4
EP	4		4	4	
DER	4			4	
DD	4	4	4	4	
RP	4				

Fuente: [Bustos99]

Las reglas de balanceo que define [Bustos99] son:

4 DFD - DD:

Todo depósito y flujo de datos del DFD debe estar definido en el DD.

4 DD - DFD:

Todo depósito y flujo de datos definido en el DD debe estar en el DFD.

4 DER - DD:

Toda entidad y relacionamiento del DER deben estar definidos en el DD.

4 DD - DER:

Toda entidad y relacionamiento definida en el DD debe estar en el DER.

4 DFD - DFD - EP:

Todo proceso del DFD tiene asociado un DFD de nivel menor o un EP, pero no ambos.

4 EP - DFD:

Toda EP describe a un proceso elemental del DFD.

Todas las acciones en la EP deben aparecer como entradas, salidas y accesos a depósitos del DFD.

Cada dato en la EP es flujo o depósito del proceso en el DFD.

4 EP - DD:

Cada dato en la EP es componente de flujo o depósito en el DD.

En la EP pueden haber términos locales al proceso.

4 DD - EP - DFD - DER:

Cada definición del DD debe tener referencia en una EP, en un DFD, en el DER o en el mismo DD.

4 DFD - DER:

Todo depósito del DFD es una entidad o un relacionamiento con atributos en el DER.

Los nombres de los depósitos (en plural) y entidades (en singular) deben coincidir.

Debe existir algún proceso del DFD que define valores para cada atributo del DER.

4 DER - DFD:

Los nombres de entidades (en singular) y de los depósitos (en plural) deben coincidir.

Todas las entidades, subentidades y relacionamientos con atributos del DER deben también aparecer como depósitos en el DFD.

4 EP - DER:

En la EP existen acciones de creación y eliminación de instancias de entidades y relacionamientos del DER.

4 DFD - RP:

Las entradas y las salidas a los procesos elementales del DFD están asociadas a un evento en la RP.

Procesos elementales del DFD están asociados a uno o más estados de la RP.

Procesos concurrentes del DFD están asociados a porciones independientes de la RP.

4 RP - DFD:

Los eventos en la RP están asociados a las entradas y las salidas de los procesos elementales del DFD.

Uno o más estados de la RP están asociados a los procesos elementales del DFD.

3.4 Orientación a Objetos (OO)

Esta estrategia no puede ser clasificada como puramente orientada a los métodos, ni como orientada a los modelos, ya que existen algunos métodos de construcción, aunque a la fecha ninguno se ha consagrado, y existen autores que entregan herramientas sin proponer métodos.

La Orientación a Objeto se centra en el comportamiento individual de los objetos, los cuales poseen características y comportamientos propios. Está basada fuertemente en las dimensiones estática y dinámica. La dimensión funcional no es exigida, pero sí se recomienda modelarla. En [Resende&Araújo00] se han establecido algunas reglas de balanceo y de consistencia entre algunos de los modelos a utilizar.

Actualmente, con la notación de UML, se ha logrado una estandarización en la notación y en los modelos a utilizar.

La tabla 3.4 muestra qué modelos se utilizan en esta estrategia y posteriormente se indican reglas de balanceo para estos modelos.

Tabla 3.4 – Balanceo entre modelos utilizados en UML.

	DCU	DA	DE	DI	DC	DD
DCU				4		
DA					4	
DE	4	4	4		4	
DI				4	4	
DC			4	4	4	4
DD					4	

Fuente: Elaboración propia.

En [Resende&Araújo00] se fundamentan y proponen reglas de balanceo y consistencia intra e inter Diagramas de Clase (DC), Diagramas de Colaboración (DI) y Diagramas de Estado (DE), éstas son:

4 DC - DC:

Una clase asociativa no puede ser definida entre ella misma y algo más.

No está permitida la especificación de herencia circular.

Una superclase de una generalización no puede ser una clase hoja¹³, y una superclase de una generalización no puede ser una clase raíz.

¹³ Clase hoja es la última de la jerarquía (hija), clase raíz es la primera (padre).

Una rama¹⁴ puede tener calificadores sólo si ella pertenece a una asociación binaria.
 Una rama puede tener agregación o composición si ella pertenece a una asociación binaria y si otra extremidad de la asociación no posee agregación o composición.
 Un método de una determinada clase debe implementar exactamente una operación de esta clase.

4 DI - DI:

Debe existir una asociación entre objetos para que pueda ocurrir un intercambio de mensajes.

Un mensaje debe contener una definición de exactamente una acción.

Un emisor y un receptor de un mensaje deben participar de la misma colaboración.

Un objeto que recibe un mensaje con una acción de creación no puede enviar o recibir mensajes anteriores a su creación.

Un objeto que recibe un mensaje con una acción de destrucción no puede enviar o recibir mensajes posteriores a su destrucción.

4 DE - DE:

Un Diagrama de Estados debe tener como mínimo un estado inicial.

Un estado inicial debe tener como mínimo una transición de salida y ninguna transición de entrada.

Un estado final no puede tener transición de salida.

Un estado compuesto secuencial puede tener, por lo menos, dos submáquinas concurrentes.

Una submáquina de estado debe tener como mínimo un estado inicial.

Un segmento de una transición con múltiples estados fuentes no puede tener eventos o condiciones.

Un segmento de una transición con múltiples destinos no puede tener eventos o condiciones.

Los segmentos de una transición con múltiples estados fuentes deben originarse de estados concurrentes.

Los segmentos de una transición con múltiples destinos deben destinarse a estados concurrentes.

4 DC - DI (y DI - DC):

Un objeto o una clase representada por un Diagrama de Colaboración (o de secuencias) debe estar definida en un Diagrama de Clase correspondiente.

Un objeto de un Diagrama de Colaboración (o de secuencias) debe ser instancia de exactamente una clase del Diagrama de Clase, además, esa clase no puede ser una clase abstracta.

Una asociación entre objetos en un diagrama de colaboración debe ser instancia de una asociación existente entre las clases base de esos objetos.

Una acción de llamada, en un mensaje, debe invocar una operación que esté definida en una clase base del objeto receptor del mensaje.

¹⁴ Una rama esta formada por las clases entre un padre (no necesariamente la raíz) y una hoja.

Una acción de llamada, en un mensaje, debe invocar sólo operaciones públicas, cuando el emisor o el receptor no son instancias de la misma clase.

Un objeto puede enviar mensajes a otro objeto sólo cuando una asociación que transporta este mensaje se basa en una asociación cuya rama, conectada a una clase base del objeto receptor del mensaje, es navegable.

4 DC - DE (y DE - DC):

Una clase representada por el Diagrama de Estados debe estar definida en el Diagrama de Clases correspondiente.

Un evento de llamada debe invocar una operación de una clase que está representada en el Diagrama de Estados.

Un evento de llamada, en una transición entre diferentes estados, no debe corresponder a una operación de consulta.

Una acción de llamada debe invocar una operación definida en una clase del Diagrama de Clases. Si esta operación pertenece a una clase A diferente de la clase B representada en el Diagrama de Estado, entonces esta operación debe ser pública y debe existir una asociación entre A y B, cuya rama conectada a la clase A será navegable.

A continuación se proponen otras reglas de balanceo que no se mencionan en [Resende&Araújo00] en calidad de opcionales:

4 DA - DC:

Todas las actividades representadas en los DA deben estar definidas como operaciones de algún o algunos objetos en el DC.

4 DCU - DI:

Todos los casos de uso deben tener un DI asociado.

4 DE - DCU:

Los eventos externos del DE son interacciones de actores con un caso de uso del DCU.

4 DE - DA:

Las actividades asociadas a los estados del DE deben estar representadas en los DA.

4 DC - DD:

Todo objeto del diagrama de clases debe estar definido en el diccionario de datos.

4 DD - DC:

Todo objeto definido en el diccionario de datos debe estar en el diagrama de clases.

3.5 Consideraciones de cada estrategia

La estrategia de Análisis Estructurado Moderno, enmarcándose como enfoque dirigido por método, define claramente la metodología, los modelos a utilizar y las reglas de balanceo entre ellos.

La definición de los modelos se centra en la dimensión funcional, dándole mayor importancia a las transformaciones que el sistema realiza. En sistemas que son mayormente “transformacionales”, esta inclinación hacia la dimensión funcional no incomoda y es posible que ayude. Sin embargo, existen casos, en que puede ser poco práctico describir las transformaciones que realiza el sistema con un DFD y posteriormente con EP, cuando sólo bastaría con hacer DFD de nivel 1 ó 2. Además, el AEM le entrega una menor importancia relativa a la dimensión dinámica, siendo más bien sólo un complemento o extensión a la dimensión funcional, esto se contrapone abiertamente con la importancia proporcional que se debe buscar al modelar las tres dimensiones.

En cuanto a las reglas de balanceo, ellas ayudan al analista a modelar en forma ordenada y consistente, pero en ocasiones, como las mencionadas en el párrafo anterior, las reglas de balanceo del Análisis Estructurado Moderno son muy rígidas y no se justifican.

Respecto del Modelado Orientado a Objetos, un argumento muy citado por los adeptos a la Orientación a Objeto es que es natural, es decir, un modelo orientado a objeto está de acuerdo con lo que los analistas y usuarios perciben de la realidad.

Por otra parte, se aduce a que con la orientación a objeto se disminuye el *gap* entre el dominio del problema y el dominio de la solución. Ahora, si se recuerda que el modelamiento conceptual debe enfocarse en el problema y no en la solución, se concluye que el Modelado Orientado a Objeto no es “orientado al problema”.

Según [Høydalsvik&Sindre93], las principales críticas al AOO, entre otras, son:

- No se hará un buen análisis simplemente por un modelo está de acuerdo a la manera en que el hombre piensa, es decir, si el modelo es un diseño.
- Una representación orientada a objeto puede ser muy buena para algunos casos, pero muy mala en otros. Puesto que en el análisis debe describir los elementos desde cualquier perspectiva, el Análisis Orientado a Objeto sólo describirá una solución parcial.
- La principal motivación para cambiarse a AOO es que del análisis se pueden derivar las etapas siguientes de diseño y codificación. Esto es claramente orientado al objetivo.

Entonces, [Høydalsvik&Sindre93] concluyen que la encapsulación aparece como principal problema del AOO, ya que para decidir qué operaciones encapsula un objeto, necesariamente se toman decisiones de diseño e implementación.

También [Bustos99a], refiriéndose al mismo problema, concluye que: “Una forma de resolver este problema es creando mecanismos que permitan postergar el encapsulamiento tanto cuanto sea posible. En otras palabras, modelar el dominio del problema fuera del paradigma de la orientación a objetos, con suficiente poder de expresión y flexibilidad para poder derivar modelos orientados a objetos con relativa facilidad, si la implementación desea ser realizada bajo este paradigma. De esta forma, el modelado concluye antes de introducirse el encapsulamiento, punto en el cual comienza el diseño orientado a objetos”.

Por otra parte, la Orientación a Objetos utilizando *Unified Modeling Language* (UML), a diferencia del AEM, se basa fuertemente en las dimensiones estática y dinámica, dejando opcional la dimensión funcional y sin considerar la importancia proporcional con que debieran modelarse las tres dimensiones. Para esto considera que el sistema está compuesto de objetos que se asocian entre sí y que exhiben comportamientos.

La dimensión dinámica se acopla a la dimensión funcional, principalmente porque el tipo de modelos utilizados es mixto, es decir, modelan elementos de la dimensión dinámica y también de la dimensión funcional, por esta razón el modelamiento a través de un modelo funcional puro no se exige, pero puede ser útil en algunos casos.

En cuanto a las reglas de integración, el UML casi no tiene reglas establecidas, sólo se han planteado algunas por parte de ciertos investigadores, lo que deja al analista una libertad que, a la hora de mantener consistencia entre los modelos, puede ser perjudicial.

Finalmente, la Estrategia Genérica se enmarca dentro del enfoque dirigido por modelos. En ella las dimensiones estática, funcional y dinámica se ponen al mismo nivel, para ello utiliza un modelo dinámico que describe el comportamiento del sistema como un todo.

Se entrega una gran libertad para escoger las herramientas con las cuales modelar las distintas dimensiones y se proponen reglas de balanceo flexibles. Sin embargo esta misma libertad a la hora de escoger modelos puede llevar al analista a equivocarse en la decisión de qué modelos utilizar y complicar la consistencia entre los modelos.

4 Estrategia propuesta

4.1 Fundamentos

En los capítulos anteriores se han analizado los aspectos conceptuales del modelamiento de sistemas, se describieron las principales herramientas y modelos que se pueden utilizar para describir sistemas, considerando las tres perspectivas o dimensiones, y las estrategias de integración que se utilizan actualmente para construir y balancear los modelos, manteniendo consistencia entre ellos. En el presente capítulo se describe una estrategia de integración de modelos conceptuales de sistemas de información que intenta ser una alternativa a las actuales estrategias y paradigmas.

La estrategia propuesta determina qué modelos utilizar y cómo se relacionan entre ellos, sin proponer una metodología para elaborar los modelos, ni un orden en que se deben construir. Por lo tanto, basándose en las definiciones de [Høydalsvik&Sindre93], se define que la estrategia propuesta es orientada a los modelos.

Una estrategia de integración orientada a los modelos debe ser independiente del orden en que se modele y del método que se utilice para modelar, es decir, debe definir los modelos que se deben utilizar para describir el sistema desde cada perspectiva y cómo se balancean entre ellos, y el analista debe decidir cómo abordar el problema y qué modelo construir primero. Para cada dimensión hay que definir qué elementos se deben representar y qué modelos utilizar para representar los elementos, ya sea para modelar las relaciones entre los elementos como para describirlos. Especialmente importante es indicar las reglas de balanceo e integración que se deben considerar para construir modelos desde las distintas dimensiones, y de esta manera asegurar consistencia y evitar errores en las fases siguientes. Por último, una estrategia de integración debe ser clara en sus fundamentos, para que los analistas puedan definir alternativas si un modelo no se acomoda a la forma en que se quiere representar un elemento o una relación.

Para comenzar, se extraen de cada estrategia descrita anteriormente algunos aspectos y ventajas que se pueden utilizar en esta proposición:

- De la metodología de Análisis Estructurado Moderno se extrae sólo la definición clara de los modelos a utilizar y el balanceo entre ellos (no el proceso de construcción), ayudando al analista a modelar en forma ordenada y consistente.
- De la Estrategia Genérica se puede heredar la libertad para escoger modelos, entregando reglas de balanceo flexibles.
- Una de las ideas que se puede extraer de la estrategia de OO es el foco de la dimensión dinámica, en la cual se modela el ciclo de vida de un objeto. Pero, como se mencionó anteriormente, la encapsulación aparece como uno de los problemas de la OO ya que

para decidir qué operaciones encapsula un objeto, necesariamente se toman decisiones de diseño e implementación.

Anteriormente se ha mencionado a la encapsulación de operaciones como la principal desventaja de la Orientación a Objetos. Para resolver este problema [Bustos99a] y [Høydalsvik&Sindre93] proponen relajar el encapsulamiento de las operaciones durante la etapa de análisis, de esta forma se intenta suplir esta desventaja o contradicción.

Si se adopta la solución que estos autores proponen y se define un objeto que tenga las características comunes, es decir, identificador, atributos y comportamiento, pero sin incorporarle aún las operaciones, se obtiene un caso especial de objeto, o mejor, un tipo especial de entidad -en el sentido del DER- una “entidad dinámica”, porque además de encapsular atributos, también presenta comportamiento.

Como la estrategia propuesta se enmarca fuera del paradigma de la Orientación a Objeto, se desecha al “objeto sin encapsulación”, y por lo tanto, se define como elemento básico a la “**entidad dinámica**”.

De esta forma, a esta estrategia se le denominará: **MOBED**, sigla que nace a partir de Modelamiento Basado en Entidades Dinámicas.

Por una parte, si se recuerda la definición de entidad entregada en la sección 2.2.1.1, ésta dice: “Es la representación de cualquier composición de información que deba almacenar el sistema, generalmente tiene un gran número de propiedades o atributos diferentes. Una entidad, a diferencia de un objeto, sólo encapsula datos”.

Por otra parte, a partir de las descripciones realizadas en la sección 2.2.1.2, un objeto se puede definir como: Una cosa tangible y/o visible, que tiene estados, comportamiento e identidad. Además tiene operaciones que pueden cambiar su estado, el de sus pares y el del sistema.

A partir de ambas definiciones, se puede definir a la entidad dinámica de la siguiente forma: Es la representación de cualquier composición de información que deba almacenar el sistema y que tenga atributos, identidad, estados y comportamiento. A esta definición se le puede agregar explícitamente que no posee operaciones.

Entonces, para identificar una entidad dinámica en el sistema que se desee modelar, se debe utilizar el mismo proceso que se utiliza para definir las entidades del DER. Según [Bustos99] debe responderse a: “¿qué datos se requieren para manejar el negocio? y ¿qué datos deben ser recordados por el sistema?”. Y según [Pressman98], pág. 203, “...¿Cuáles son las entidades¹⁵ que viven o transitan en el sistema?, ¿Cuál es la composición de cada entidad y qué atributos la describen?, ¿Dónde residen actualmente las entidades?...”

¹⁵ [Pressman98] utiliza la definición de “objetos de datos” para referirse a las entidades.

Si además de determinar la composición de la entidad y qué atributos la describen, se determina que para representar el sistema es importante saber cómo se comporta alguna entidad -es decir, en qué estados se puede encontrar y cuáles son los eventos que la hacen cambiar de estado- entonces se tiene una entidad dinámica. Esto es, todas las entidades “tradicionales” pueden ser entidades dinámicas, si el analista determina que es necesario conocer o controlar su comportamiento durante su permanencia en el sistema.

De esta forma, existirán entidades que presenten comportamientos simples - tal como entidad ingresada, consultada, actualizada y eliminada¹⁶- y otras que presenten comportamientos más complejos. Como lo normal es que se desee controlar a las entidades que tengan comportamientos complejos, generalmente las entidades dinámicas serán complejas en cuanto a su comportamiento.

Por estas razones, cuando se modela un sistema de información utilizando la estrategia propuesta, se tendrán una o más entidades dinámicas que cambien de estados (producto de eventos), que se relacionan entre sí (a través de relacionamientos estáticos) y que adquieren distintas características (en los procesos) durante su permanencia en el sistema.

La Figura 4.1 muestra cómo la entidad dinámica se proyecta a cada dimensión: estática, dinámica y funcional. En un capítulo posterior se explicará cómo se relacionan los modelos de las distintas dimensiones.

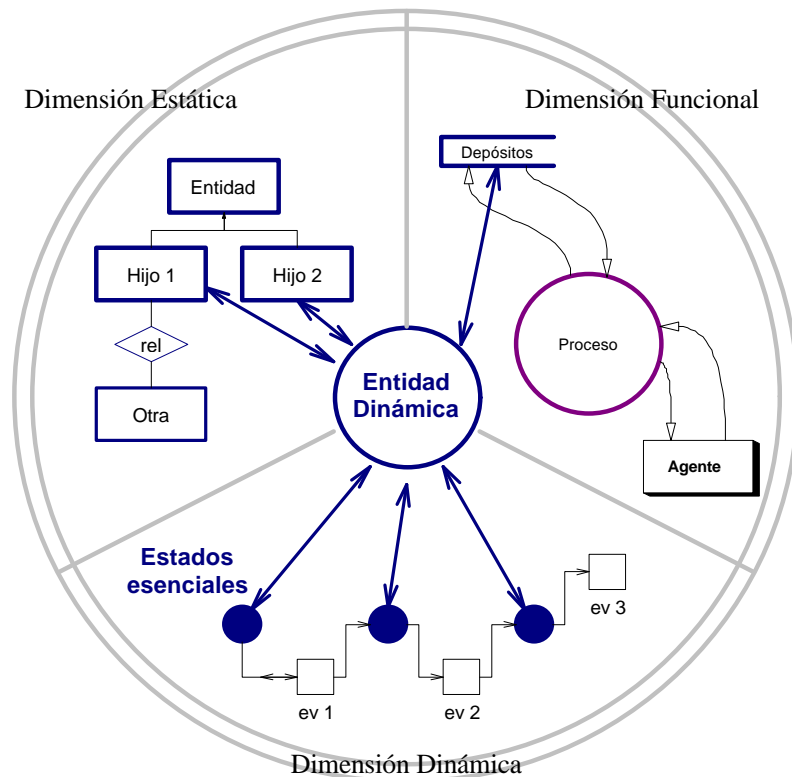


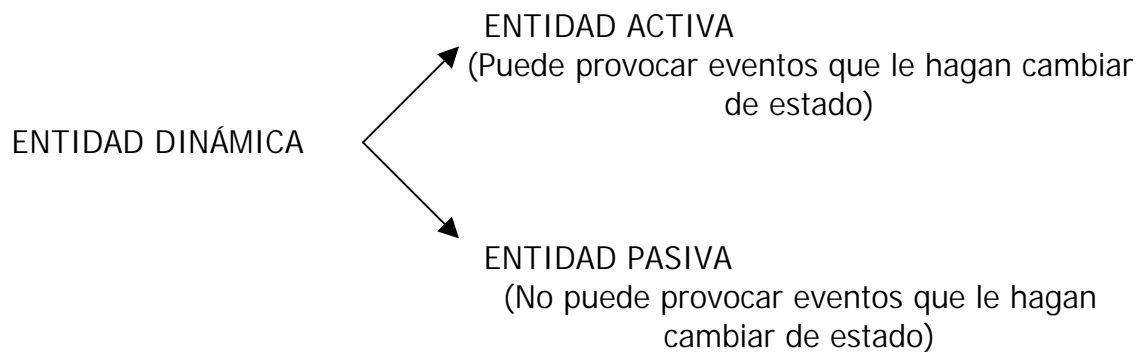
Figura 4.1: Entidad dinámica proyectada a cada dimensión.

¹⁶ Tal como el esquema CRUD: Create – Read – Uppdate – Delate.

4.2 Modelamiento de la dimensión dinámica

Una vez que se ha determinado que se debe modelar el comportamiento de alguna entidad, es decir, se ha definido alguna entidad dinámica, se debe modelar su ciclo de vida, en el que aparezcan todos los estados posibles de la entidad y todos los eventos que le puedan producir cambios de estados.

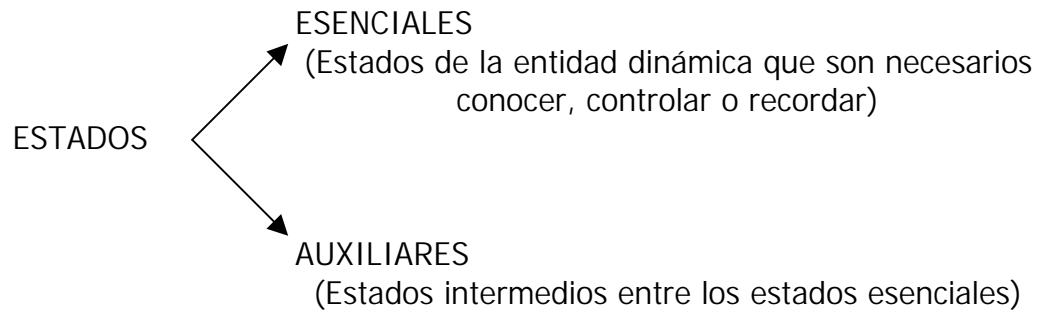
Dentro de este contexto, se puede definir a un evento como cualquier suceso que puede provocar que la entidad dinámica cambie de estado. Un evento puede ser producido por un agente externo, por otra entidad dinámica o por la misma entidad. A partir de los tipos de eventos que hagan cambiar de estado a una entidad surge la primera clasificación de las entidades dinámicas, ésta es: si una entidad puede provocar eventos que le hagan cambiar de estado se le define como “entidad activa”, y si no, como “entidad pasiva”. Estas clasificaciones serán explicadas más adelante.



Por otra parte, cada instancia de la entidad dinámica –es decir, cada ocurrencia de ella– podrá tener distintos estados posibles y, en cada estado, podrá tener distintas particularidades. Por otra parte, mientras viva en el sistema podrá estar en distintos estados, en uno o más a la vez, ya que éstos podrían ser concurrentes, excluyentes o secuenciales.

Debido a la complejidad que pueda presentar el modelo del ciclo de vida de la entidad dinámica, el modelador podrá distinguir los estados de la entidad que el sistema deba recordar o controlar y los estados de la entidad que no es necesario recordar, todo esto, de acuerdo al contexto y al problema que se modela. Por lo tanto, se pueden clasificar los estados de la entidad dinámica en dos tipos: estados esenciales y estados auxiliares.

Los estados esenciales son los estados de la entidad dinámica que son necesarios conocer, controlar o recordar y los estados auxiliares son los estados intermedios entre los estados esenciales. Estas definiciones serán ejemplificadas más adelante.



Para representar el sistema desde esta dimensión, se debe utilizar un modelo que describa el ciclo de vida de cada entidad dinámica, representando todos los estados en que ella se pueda encontrar y todos los eventos que puedan provocar los cambios de estado.

Para modelar el ciclo de vida de una entidad, teóricamente se podría utilizar cualquier modelo que describa aspectos dinámicos de los sistemas, tales como: Diagramas de Transición de Estados, Diagramas de Estados o *Statecharts*, o Redes de Petri. A continuación se analizan los modelos que pueden utilizarse para lograr el fin deseado.

Los Diagramas de Transición de Estados se desechan, ya que no pueden representar estados concurrentes.

El Diagrama de Estados o *Statecharts*, modelo que se propone en UML para modelar el ciclo de vida de un objeto, tiene las ventajas de representar jerarquización, modularización y concurrencia, pero su desventaja para modelar el ciclo de vida está justamente en esta última, la concurrencia de estados, ya que en este modelo no es fácil reconocer visualmente el encadenamiento de eventos o eventos que producen varios estados concurrentes, lo que se conoce como “*broadcasting*” [Allen&deChampeaux95].

La Red de Petri, por otra parte, tiene las ventajas de representar claramente la concurrencia de estados y eventos, y de ser también jerarquizable considerando su variante Canal/Actividad¹⁷, pero tiene la desventaja de crecer explosivamente, lo que dificulta el establecimiento de los límites.

Entonces, debido a las descripciones anteriores, se determina que desde la dimensión dinámica se prefiere modelar el ciclo de vida de cada entidad dinámica con una Red de Petri elemental, con algunas incorporaciones que resuelven en alguna medida el problema de los límites, tal como los *swimlanes* utilizados en los Diagramas de Actividades.

En la figura 4.2 se muestra el ciclo de vida de un proyecto público, el cual debe ser licitado, a continuación se debe contratar un contratista ejecutor de entre los que participaron en la licitación y posteriormente se debe controlar su ejecución.

¹⁷ Esta variante es explicada en detalle en el Anexo 1: Redes de Petri

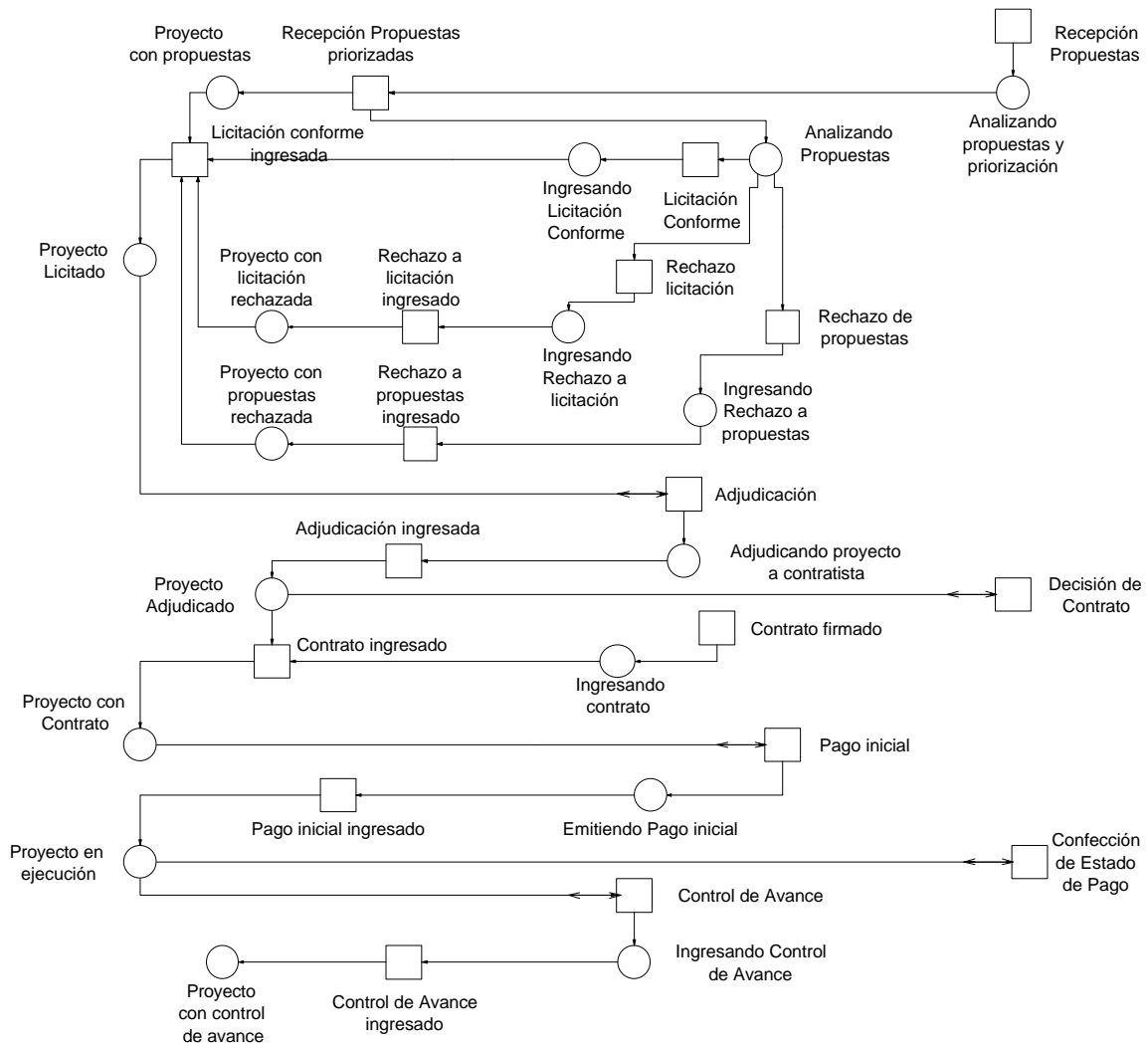


Figura 4.2: Ejemplo de ciclo de vida una entidad dinámica.

Anteriormente se mencionó que los eventos que produzcan cambios de estados a la entidad podrán ser producidos por la misma entidad o por agentes, esto se debe representar en la Red de Petri con carriles o *swimlanes* como los usados en los Diagramas de Actividades, que fueron descritos en el capítulo 2, donde cada *swimlane* representará un agente que interactúa con la entidad produciendo eventos que potencialmente le hagan cambiar de estado, no interesa mostrar los estados en los que se puede encontrar un agente, sino los eventos que afecten a la entidad dinámica. Un agente puede representar a otra entidad dinámica o a un agente externo, tal como: una persona, un departamento u otro sistema.

La figura 4.3 muestra otro ejemplo, a partir del ejemplo anterior, de un ciclo de vida de un proyecto público, en ella se pueden observar los distintos estados en que se puede encontrar la entidad, utilizando ramas restauradoras para relacionar estados que son concurrentes y secuenciales. Además, en la figura se aprecia cómo los agentes interactúan con la entidad produciendo o recibiendo eventos.

La cantidad reducida de eventos en el ciclo de vida de la figura 4.3 se debe a que es una entidad pasiva y sólo reacciona a eventos externos sin generarlos internamente.

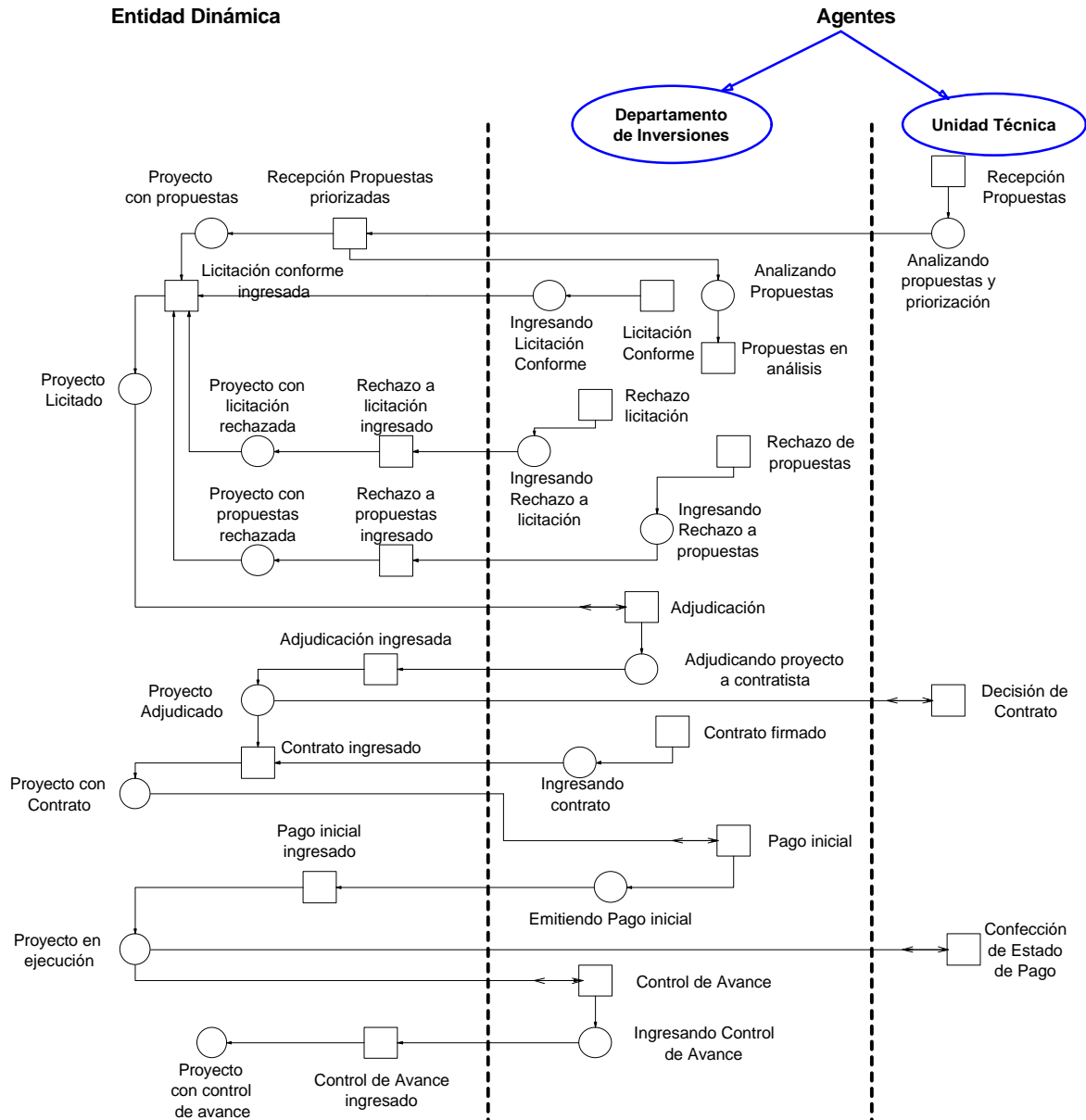


Figura 4.3: Ejemplo de ciclo de vida una entidad dinámica con *swimlanes*.

En la figura 4.4 se presenta otro ejemplo de ciclo de vida de una entidad, en este caso se presenta el comportamiento de un paciente de una clínica. En ella también se pueden observar los distintos estados en que se puede encontrar la entidad dinámica “paciente”. En este caso se puede observar que la entidad -a diferencia de la figura 4.3- es activa, ya que también genera eventos que la hacen cambiar de estados, los que se representan

principalmente en las decisiones (marcadas con un círculo rojo) que la entidad puede realizar, y que la cambian de estado.

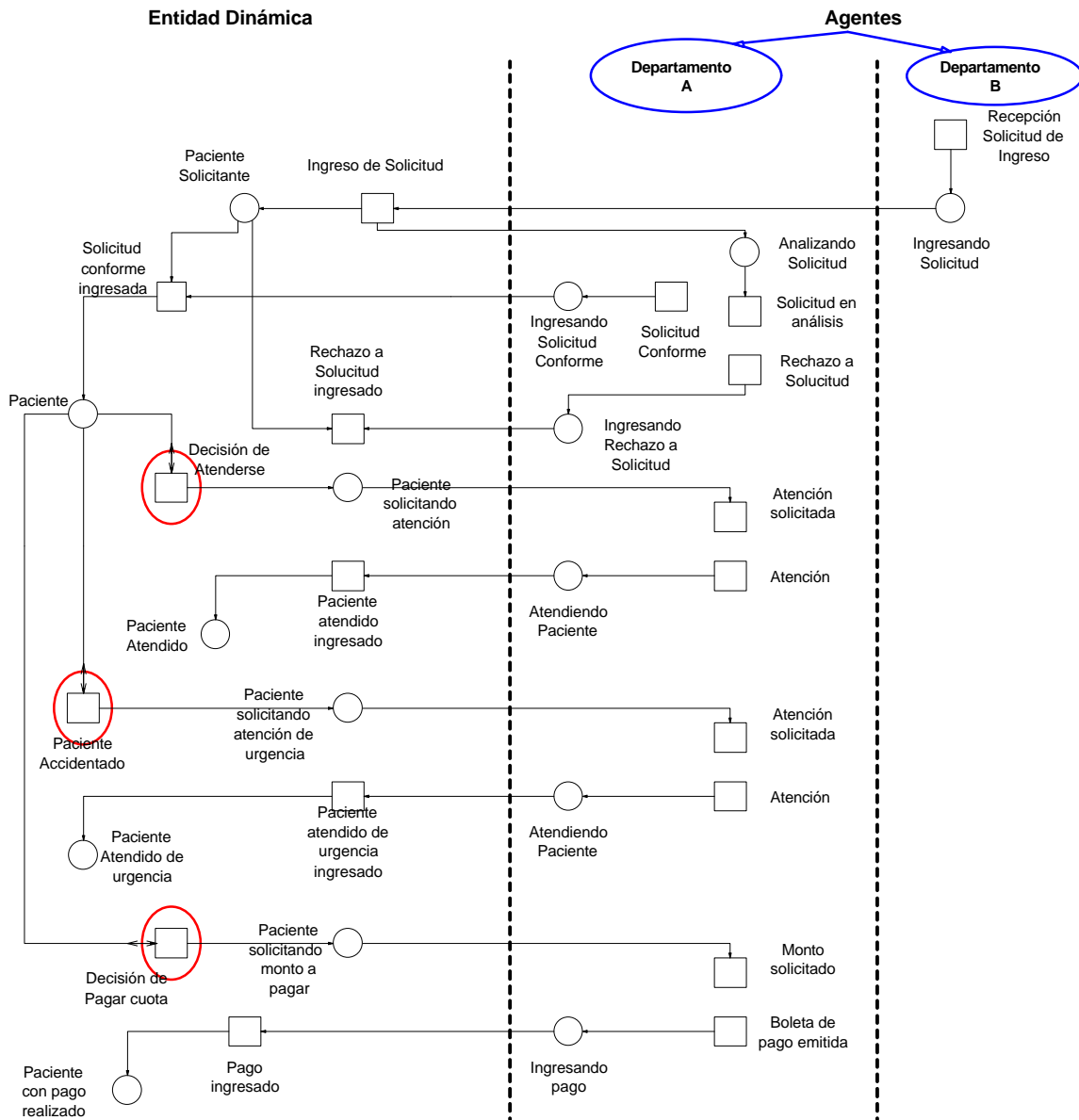


Figura 4.4: Ejemplo de ciclo de vida una entidad dinámica activa.

De acuerdo con lo expuesto en la sección anterior, algunos estados deberán recordarse y otros no, dependiendo del sistema y del problema que se modela. Esta diferencia entre los estados de la entidad hace necesario distinguir dos estados posibles: estados esenciales y estados auxiliares, los estados esenciales son los estados de los cuales se quiere tener control e información, y los estados auxiliares son estados intermedios, que sirven de transición entre los estados esenciales. La Figura 4.5 incorpora al ciclo de vida de la entidad dinámica la clasificación de estados esenciales y auxiliares.

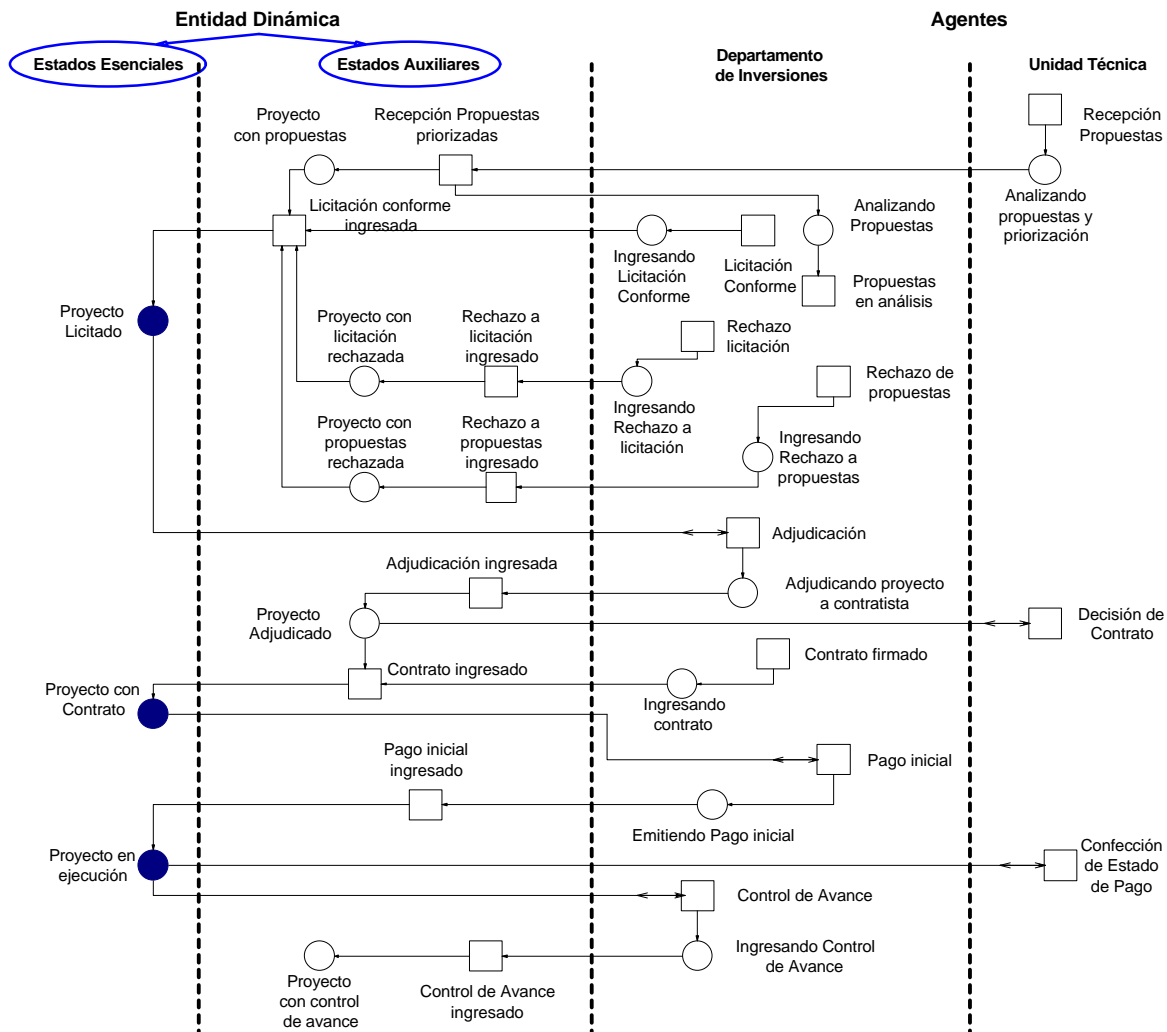


Figura 4.5: Estados esenciales de una entidad dinámica.

La entidad dinámica tendrá distintos estados posibles y mientras viva en el sistema podrá estar en uno o varios estados a la vez. Ya se ha mencionado que los estados esenciales son más importantes para el sistema y el problema, y que por lo tanto es necesario recordarlos. Por otra parte, como estos estados deben controlarse, también interesa describir cómo se logran estos estados, es decir, qué procesos o actividades debe realizar el sistema para hacer que la entidad pase a un nuevo estado esencial.

Para lograr un estado esencial a partir de otro estado esencial, la entidad pasa por estados auxiliares -o de transición-. Los eventos y los estados auxiliares de la Red de Petri elemental, que en conjunto hacen que una entidad dinámica pase de un estado esencial a otro, se agrupan y representan actividades bajo la abstracción de Red de Petri Canal/Actividad, según lo definido en el Anexo 1.

La figura 4.6 muestra cómo agrupar los estados y eventos para identificar actividades.

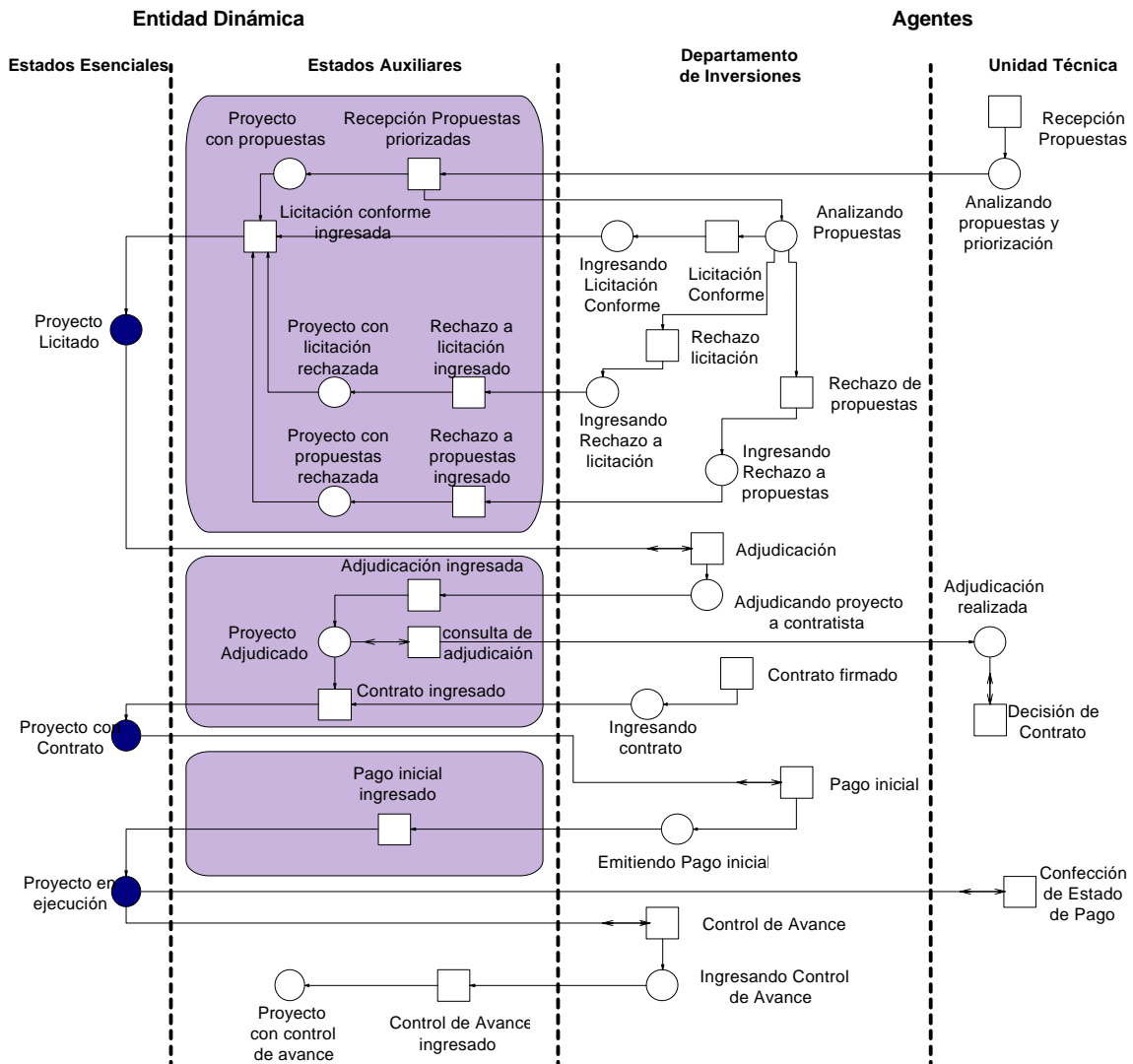


Figura 4.6: Ciclo de vida una entidad dinámica con actividades.

Como criterio de agrupamiento para identificar las actividades en la red, se puede utilizar el concepto de cohesión, adaptado del diseño estructurado de sistemas, descrito en [Page-Jones80] y [Stevens88]. Con este criterio se pueden jerarquizar los relacionamientos que existan entre los estados y eventos que forman la actividad. Se pueden identificar distintos grados de cohesión, tales como: Funcional, Secuencial, Comunicacional, Procedimental, Temporal, Lógica y Coincidental, entre otros. El analista no deberá preocuparse de categorizar la cohesión en un proceso específico. Más bien, debe entenderse el concepto global y evitar niveles inferiores de cohesión al definir las actividades.

En [Page-Jones80] y [Stevens88], se define la cohesión como uno de los criterios de calidad del diseño estructurado de sistemas, que es una medida de fuerza de asociación funcional de elementos dentro de un módulo.

El término cohesión se tomó de la sociología, donde significa la *relatedness*, que es el grado de conexión o afinidad entre los seres humanos en un grupo.

A partir de estos estudios y sus posteriores refinamientos, otros investigadores, desarrollaron una escala de cohesión que ha resultado una buena medida del mantenimiento de un módulo. A partir de las definiciones de [Yourdon&Constantine78] se pueden adaptar seis tipos de cohesión, que son:

1. Coincidental.
2. Lógica
3. Temporal
4. Comunicacional
5. Secuencial
6. Funcional

Las que se explican a continuación:

1. **Cohesión coincidental:** La cohesión coincidental es la de menor nivel, donde no existe una explicación para su agrupamiento. La figura 4.7, muestra una actividad con cohesión coincidental.

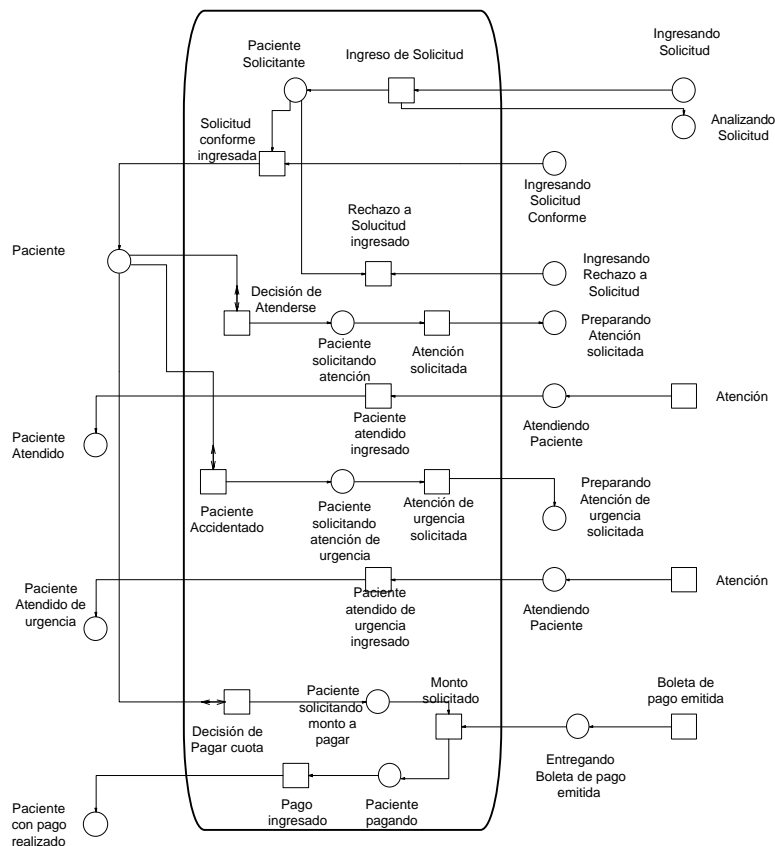


Figura 4.7: Ejemplo de una actividad con cohesión coincidental.

2. **Cohesión lógica:** Corresponde a actividades que pertenecen a una categoría arbitraria que puede tener características comunes. Tal como se muestra en la figura 4.8, todos los eventos se relacionan con la atención quirúrgica del paciente, se debe suponer que además el paciente puede recibir atención al momento de pagar o solicitar información, los eventos relacionados con esas situaciones también podrían cohesionarse lógicamente.

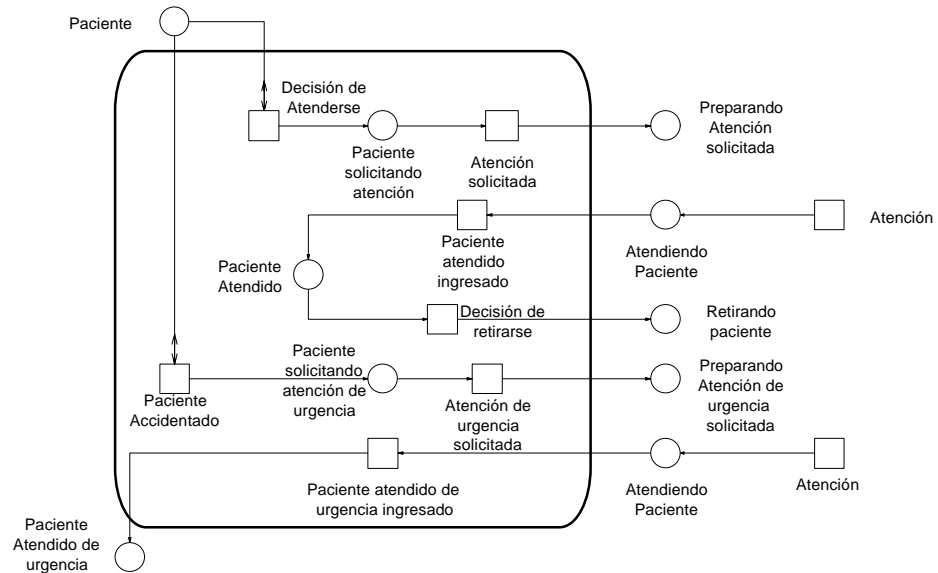


Figura 4.8: Ejemplo de una actividad con cohesión lógica.

3. **Cohesión secuencial:** La cohesión secuencial se aplica, a las actividades del ciclo de vida, cuando dentro de una actividad, los estados y eventos siguen una única secuencia lógica. La figura 4.9, muestra una actividad con cohesión secuencial, en este caso los eventos siguen una única secuencia lógica, es decir, no hay opcionalidad.

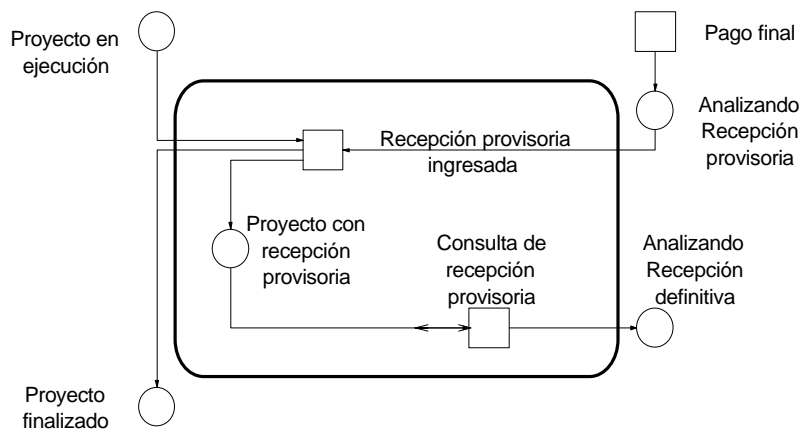


Figura 4.9: Ejemplo de una actividad cohesionada secuencialmente.

4. **Cohesión temporal:** Una actividad está cohesionada temporalmente si sus elementos tienden a ocurrir al mismo tiempo. La figura 4.10, muestra una actividad con cohesión temporal, donde cada uno de sus eventos suelen ocurrir en un tiempo relativamente similar.

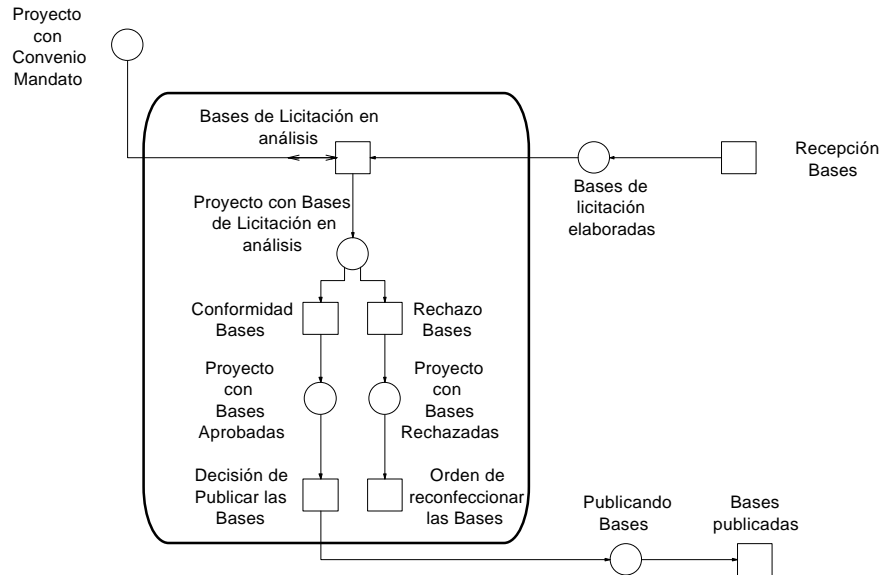


Figura 4.10: Ejemplo de una actividad con cohesión temporal.

5. **Cohesión comunicacional:** Una actividad está cohesionada comunicacionalmente si sus elementos contribuyen a actividades que usan las mismas entradas o salidas. Como se ejemplifica en la figura 4.11, se caracteriza por poseer entradas y/o salidas en común.

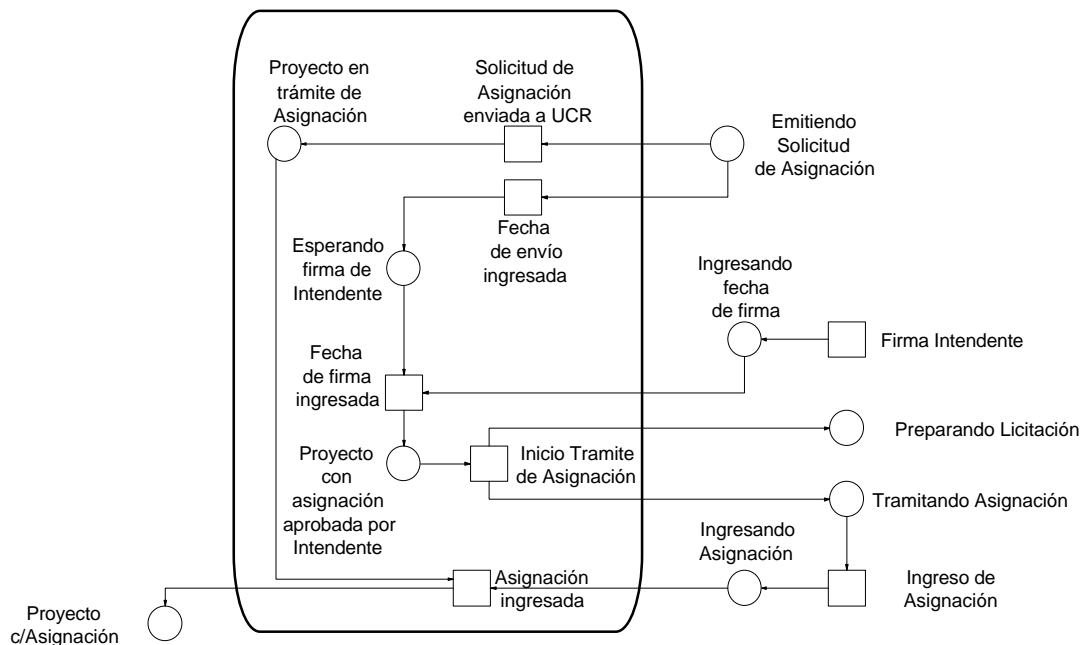


Figura 4.11: Ejemplo de una actividad con cohesión comunicacional.

6. **Cohesión funcional:** Este tipo de cohesión se aplica, con mayor grado, cuando el conjunto de estados auxiliares y eventos cumple una función específica y genera sólo una salida. En la figura 4.12 se muestra una actividad que puede ser cohesionada con el tipo de cohesión funcional, debido a que existen dos eventos que cumplen sólo una tarea, la de producir un Proyecto con Contrato. Si alguno de los eventos tuviera además otra salida, entonces este evento no podría pertenecer a una actividad cohesionada funcionalmente.

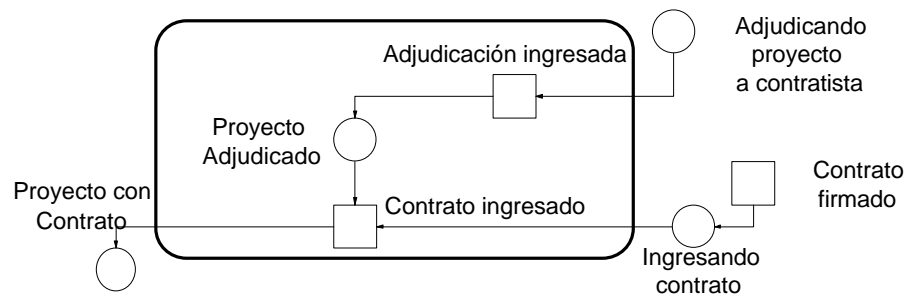


Figura 4.12: Ejemplo de una actividad cohesionada funcionalmente.

En definitiva, al agrupar los procesos en la Red de Petri elemental se deben privilegiar aquellos criterios de cohesión de más alto nivel.

Otra característica o clasificación para cada actividad que puede ser importante a la hora de modelar es lo que se ha definido como “Multiplicidad” (o Paralelismo e Individualismo). Se introduce esta nueva característica al modelamiento de actividades para diferenciar entre dos tipos de actividades. El paralelismo se define para aquellas actividades que **siempre** ocurren para conjuntos de instancias simultáneamente. Y por complemento, los procesos que no sean paralelos, serán individuales. Por ejemplo, en una institución bancaria, se puede definir una actividad para realizar los ajustes de cuenta al final del día en forma automatizada para todos los clientes a la vez, el conjunto de estados y eventos que se asocian a esta actividad que puede tener cohesión secuencial, y además se puede definir como una actividad que se realiza para un conjunto de clientes paralelamente. Por otra parte, una actividad manual realizada por un agente de la plataforma de mercado de capitales, sólo se realiza en forma individual, a un cliente a la vez.

La Multiplicidad y la Cohesión son dos características independientes y ortogonales entre sí, es decir, una actividad puede tener cualquier nivel de cohesión de los explicados anteriormente y además, puede ser individual o paralela.

La figura 4.13, muestra los distintos tipos de clasificaciones en que se pueden agrupar las actividades de la Red de Petri Canal/Actividad.

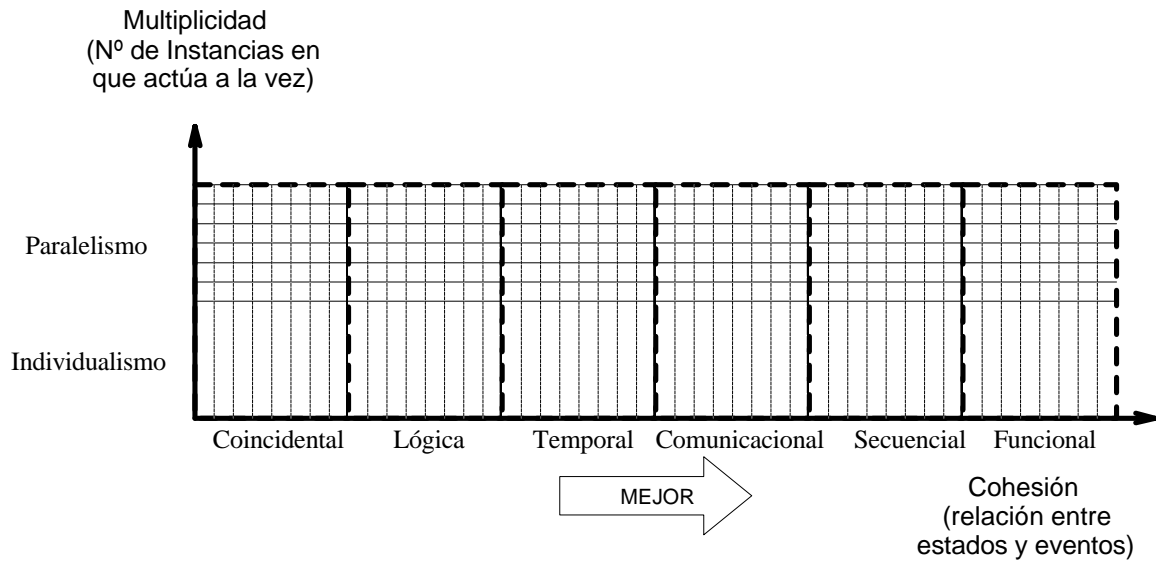


Figura 4.13: Espectro de cohesión y multiplicidad.

4.3 Modelamiento de la dimensión estática

Desde la perspectiva estática interesa representar las características estructurales y las relaciones permanentes del sistema, por lo tanto, se debe representar con un modelo que describa los elementos estructurales del sistema y sus relaciones estáticas.

Dado que esta estrategia está enmarcada fuera del paradigma de la Orientación a Objeto, se escoge el Diagrama Entidad Relacionamiento y se complementa con el Diccionario de Datos, en que se describirán los elementos de esta dimensión y algunos elementos estáticos de los modelos de las otras dimensiones.

Para describir el sistema desde esta perspectiva, se deben modelar las entidades tradicionales y dinámicas, se deben privilegiar el modelamiento de entidades dinámicas a través de jerarquías de tipos, generando subentidades de acuerdo a las distintas características y relaciones que la entidad pueda tener y a la importancia que tenga para el problema recordar las características de estas entidades. De esta forma, se facilitan y precisan los relacionamientos entre entidades dinámicas, con otras entidades tradicionales que sean relevantes para el sistema y se construye un modelo más expresivo donde se evitan los relacionamientos del tipo (0,1) o (0,n).

En la figura 4.14 se presenta un DER en el que se distingue una jerarquía de tipos y se observan las relaciones de las subentidades con otras. En este modelo también se pueden representar otras jerarquías que se relacionen directamente o a través de entidades tradicionales.

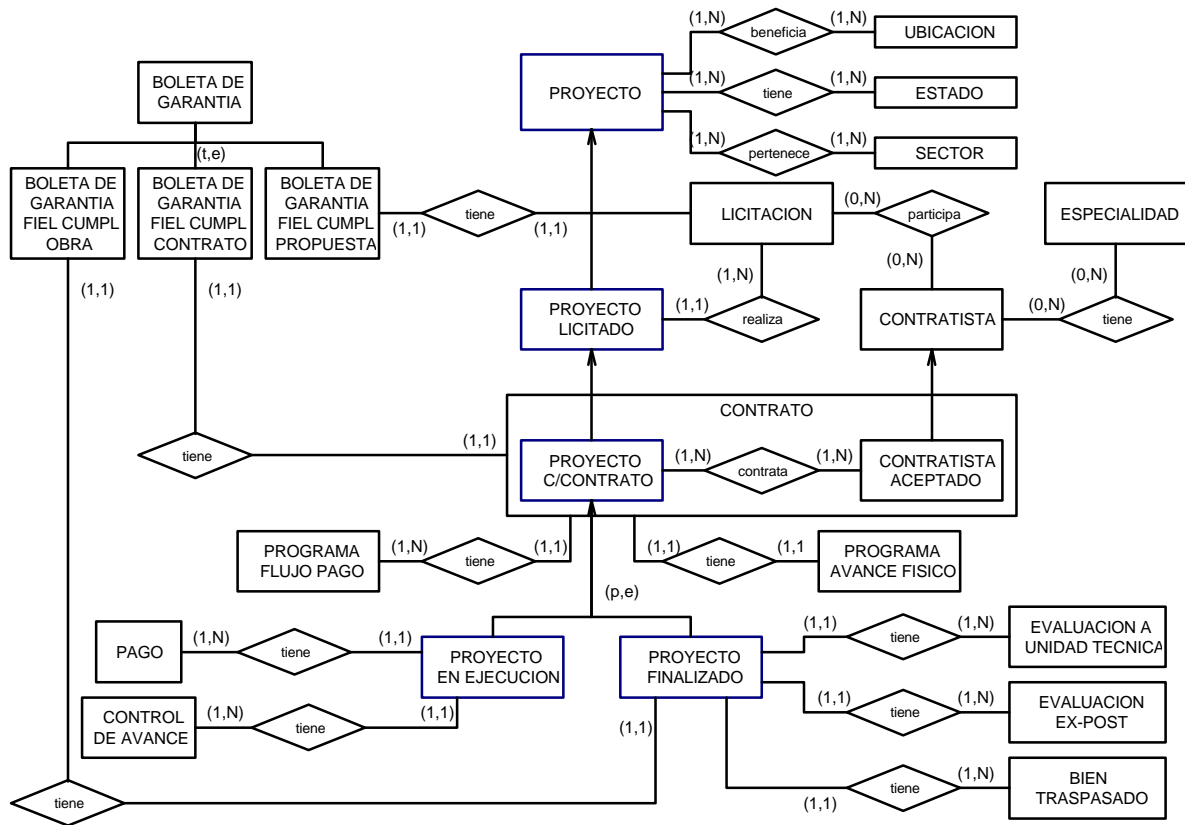


Figura 4.14: Ejemplo de DER con la jerarquía de una entidad dinámica.

Al construir el DER, se deben privilegiar los relacionamientos de las subentidades de la jerarquía en el menor nivel posible, de igual forma, los atributos son puestos a las entidades de mayor nivel posible, como un criterio de mejorar la calidad del modelo. Para terminar de describir esta dimensión, se debe incluir un Diccionario de Datos, en que se describirán las entidades, relacionamientos, atributos e identificadores.

4.4 Modelamiento de la dimensión funcional

Desde la perspectiva funcional interesa representar los procesos y transformaciones que el sistema debe realizar, las relaciones entre los procesos y de éstos con los agentes externos, lo que a su vez define los límites del sistema.

Para modelar el sistema en esta dimensión, se deben modelar los procesos que el sistema realiza, que producen transformaciones a las entidades que permanecen en el sistema.

Conceptualmente, la dimensión funcional puede modelarse con Diagramas de Flujo de Datos, Diagramas de Actividades o con Diagramas de Casos de Uso (*Use Cases*), complementados con descripciones.

El Diagrama de Actividades se utiliza para representar relaciones entre procesos o actividades concurrentes. Este diagrama, al incorporar temporalidad, modela aspectos dinámicos que se superponen a la Red de Petri. Como se ha dicho, la importancia de modelar el sistema desde la perspectiva funcional radica principalmente en la necesidad de representar aspectos funcionales, como la transformación de entradas en salidas, por lo tanto este diagrama tampoco cumple con los objetivos deseados.

Por otra parte, el Diagrama de Casos de Uso, aunque no pertenece al paradigma de la Orientación a Objeto, es sobrepasado por el Diagrama de Flujo de Datos, ya que este último tiene una mejor relación con el DER y, lo que es muy importante, describe todos los procesos del sistema, no así el Diagrama de Casos de Uso, en el cual sólo interesa modelar los procesos que son visibles desde el exterior.

Por lo tanto, para describir el sistema desde esta perspectiva se utiliza un Diagrama de Flujo de Datos, en el que se modelan los procesos que el sistema realiza, los depósitos y los agentes externos o terminadores.

La figura 4.15 es un ejemplo de un Diagrama de Flujo de Datos en el que se muestra cómo deben relacionarse los procesos entre sí, con los depósitos y cómo se relacionan con los terminadores.

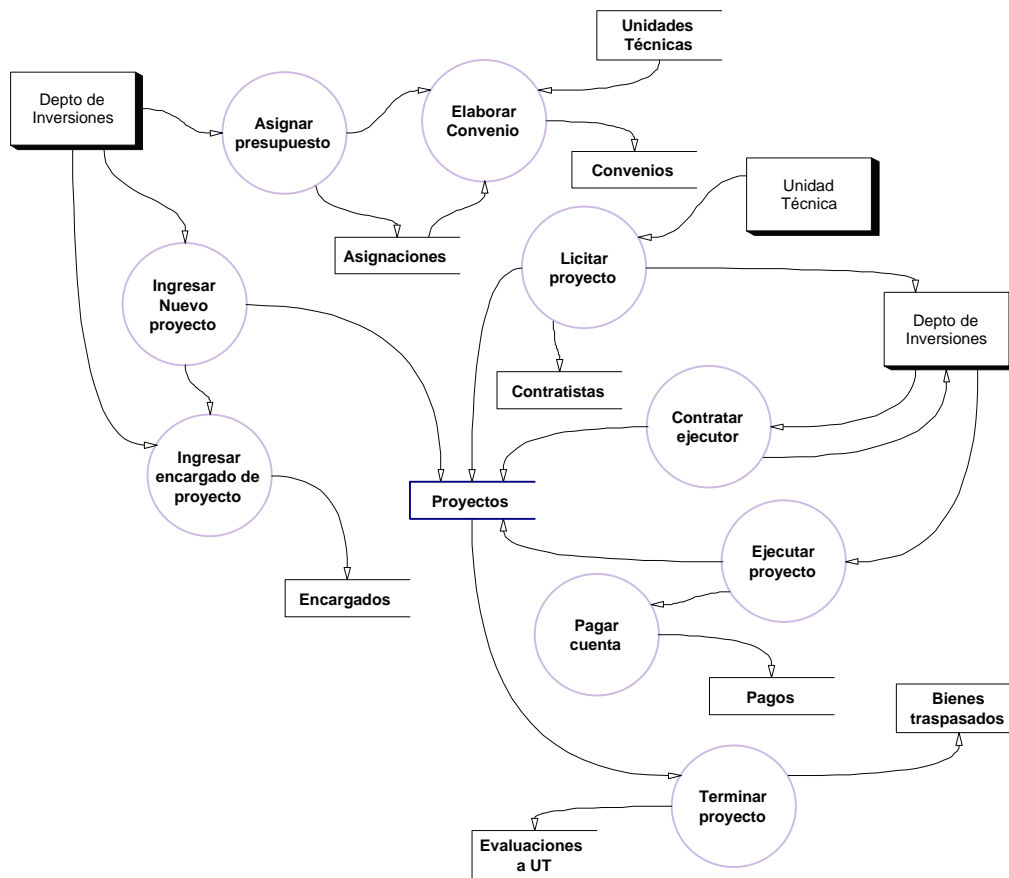


Figura 4.15: Ejemplo de Diagrama de Flujo de Datos.

El diagrama debe construirse utilizando la estrategia de agregación por datos o *bottom-up*, esto es, identificando procesos elementales que operen sobre depósitos determinados, y posteriormente agregando procesos que accesen los mismos depósitos hasta llegar al diagrama 0 y finalmente construir el diagrama de contexto. De esta forma, se facilita la relación funcional-estática.

Desde esta dimensión también puede ser necesario describir los procesos. La Especificación de Procesos (EP) podrá realizarse a través de Pre y Post Condiciones, con las cuales se indica qué condiciones son necesarias para que realizar un proceso y cuáles son las condiciones resultantes del proceso.

4.5 Relación dinámica - estática

Utilizando esta estrategia, el analista debe representar al sistema desde la perspectiva dinámica a través del modelamiento del ciclo de vida de la entidad dinámica y desde la perspectiva estática con el Diagrama Entidad Relacionamiento, tal como lo recuerda la figura 4.16.

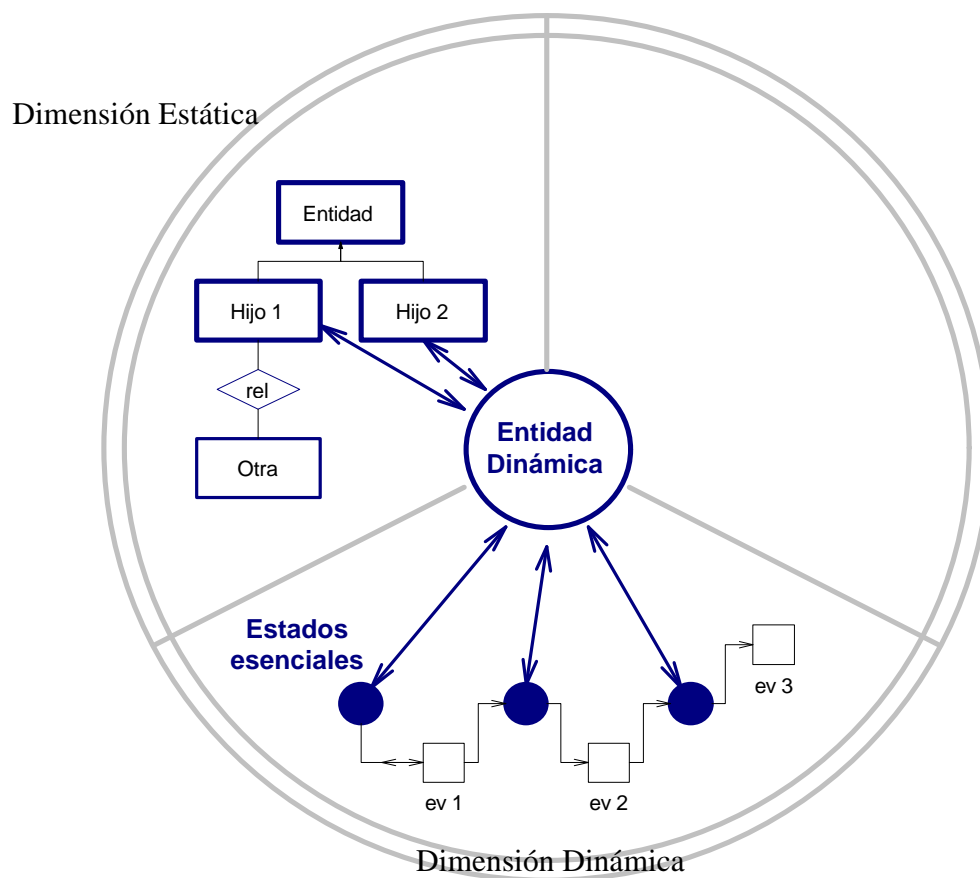


Figura 4.16: Entidad dinámica en las dimensiones estática y dinámica.

La forma “clásica” de representar el conjunto de instancias de la entidad “Proyecto” es a través de una entidad en el DER, tal como se muestra en la figura 4.17. En su parte superior se presenta el conjunto de instancias y en la parte inferior se muestra su representación estática, a través de una entidad con el identificador “Número” y para el ejemplo, se le agregaron los atributos “C” y “D”.

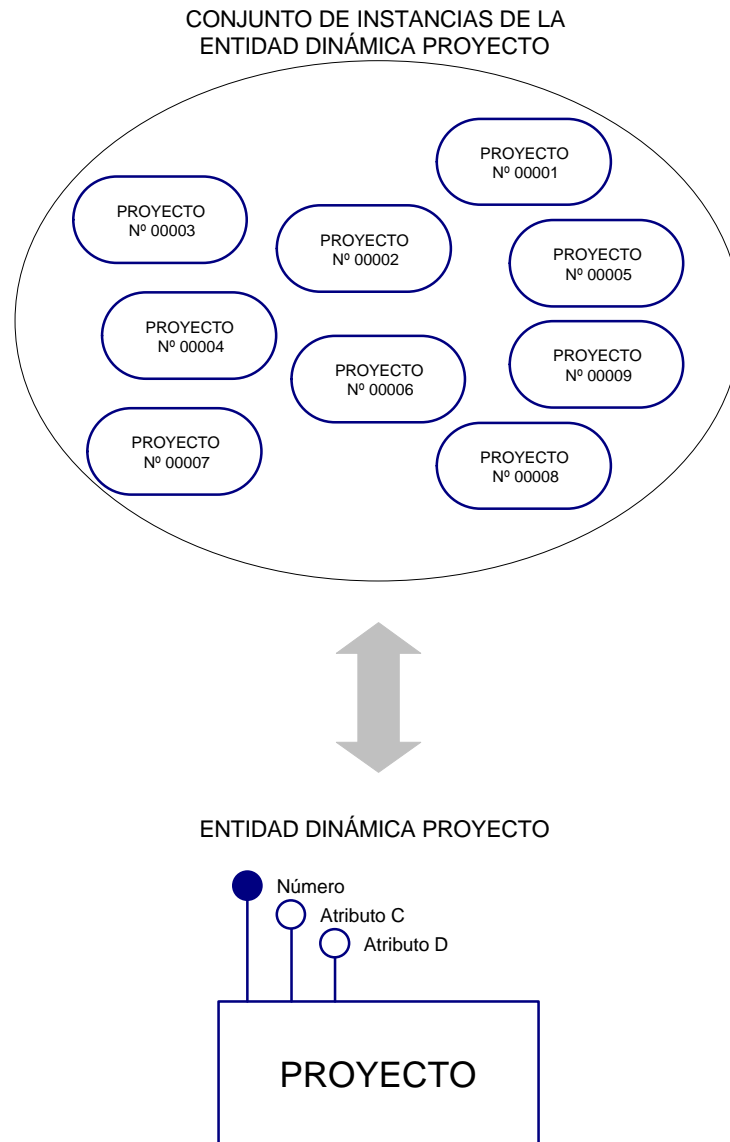
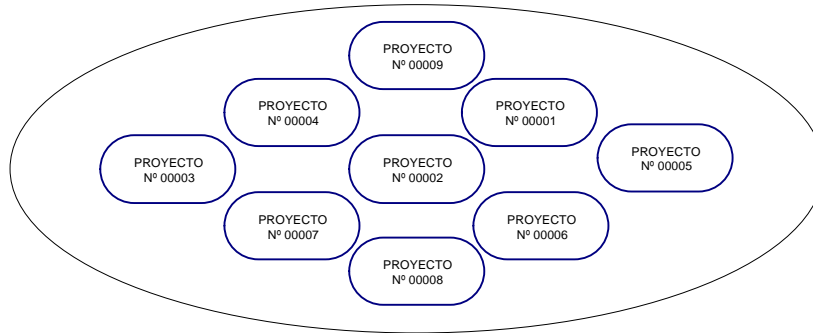


Figura 4.17: Representación de la entidad dinámica en el DER.

Por otra parte, si la entidad “Proyecto” es una entidad dinámica, y de acuerdo con la definición de entidad dinámica, es necesario conocer el comportamiento del conjunto de instancias que ella representa, esto se puede representar como muestra la figura 4.18, en ella se presenta el estado en que se encuentran las instancias de la entidad dinámica “Proyecto” en un momento dado.

CONJUNTO DE INSTANCIAS DE LA ENTIDAD DINÁMICA PROYECTO



COMPORTAMIENTO DE LAS INSTANCIAS DE LA ENTIDAD DINÁMICA PROYECTO

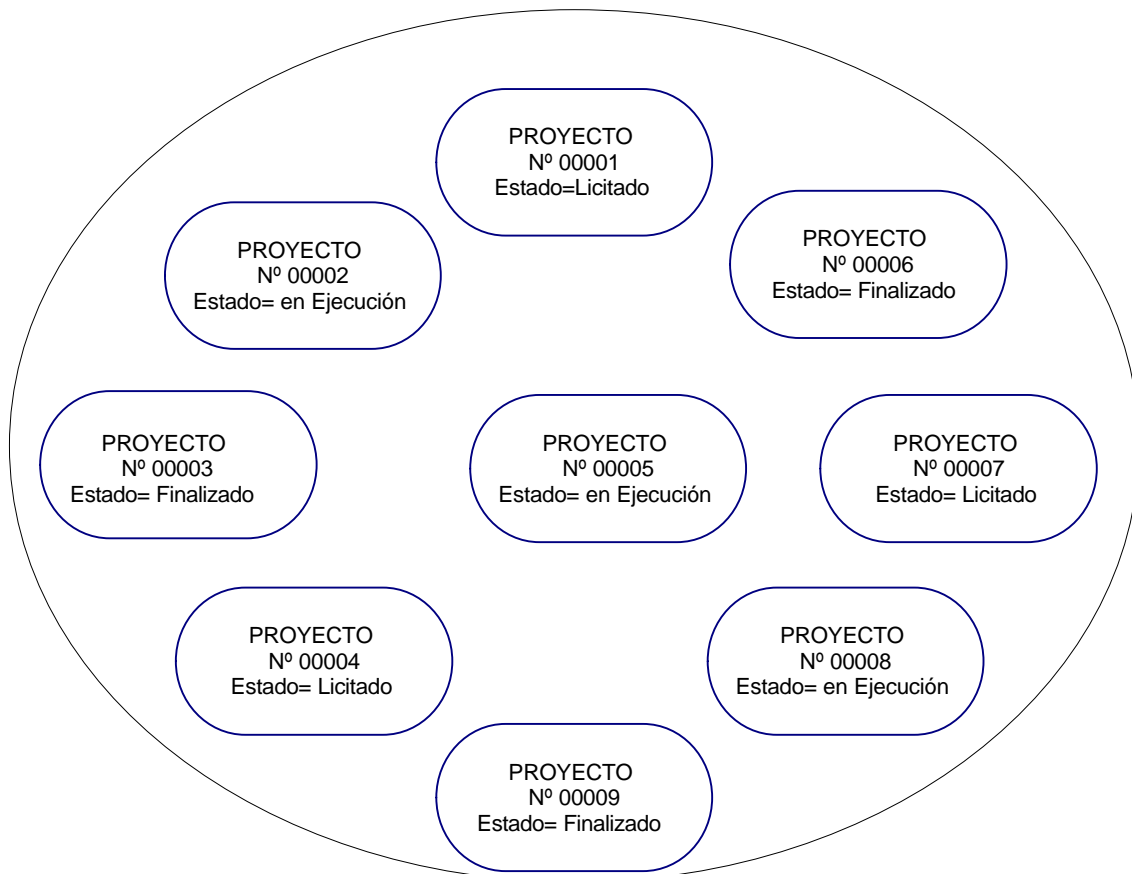


Figura 4.18: Comportamiento de las instancias de la entidad dinámica.

Tal como se ha mencionado, desde la perspectiva dinámica se debe representar el comportamiento de todas las instancias, para esto, se modela el ciclo de vida con una Red de Petri, tal como se muestra en la parte inferior de la figura 4.19, que representa el modelo genérico de comportamiento para todas las instancias de la entidad dinámica, ya que se modelan todos los estados en que puede estar la entidad dinámica y la relación entre ellos, a través de los eventos que pueden provocar cambios de estado. Además, se debe destacar y recordar que, de acuerdo a las definiciones entregadas anteriormente, los estados que deben ser recordados y/o controlados son clasificados como estados esenciales.

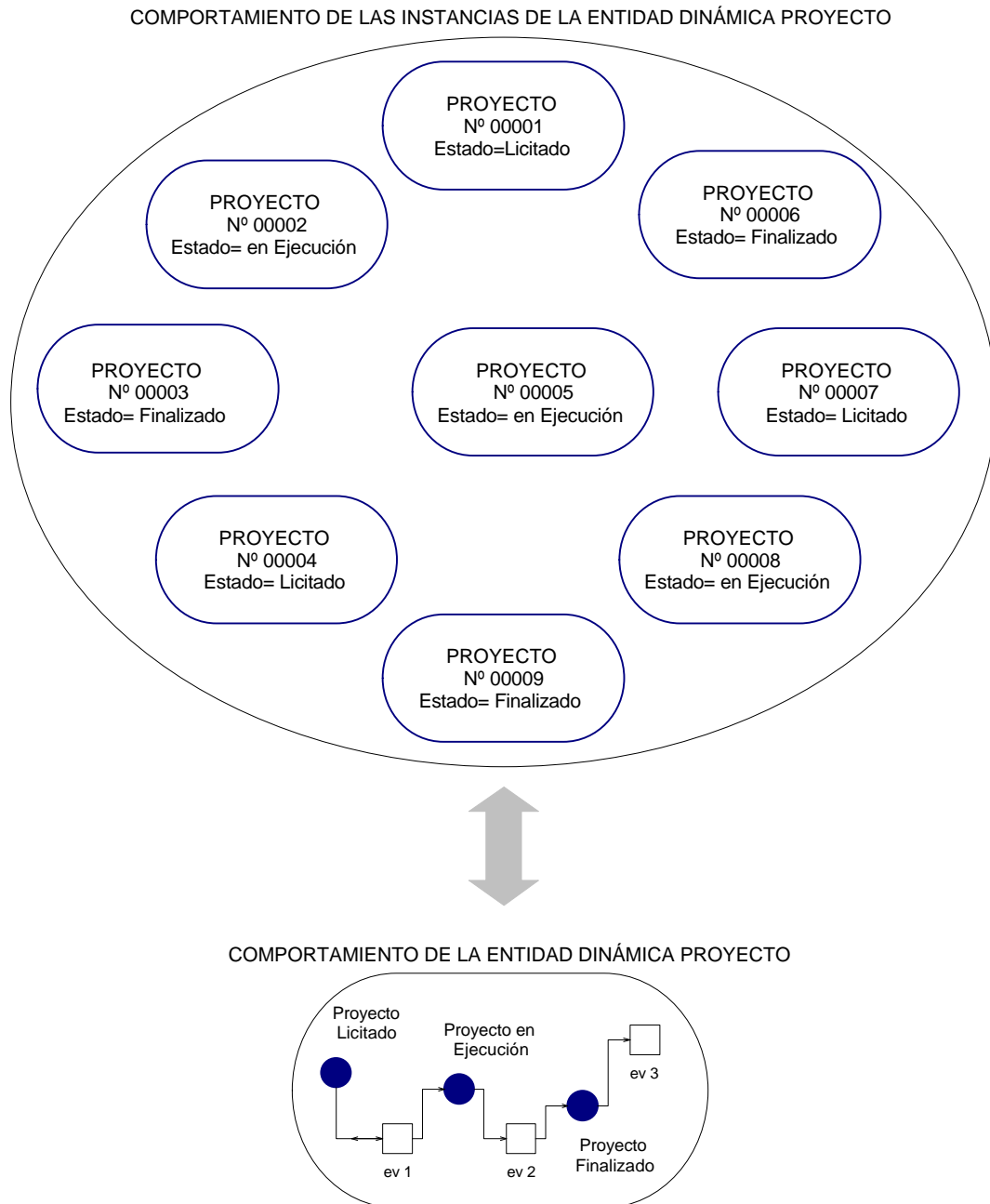


Figura 4.19: Modelo genérico de comportamiento de la entidad dinámica.

Por otra parte, la entidad dinámica -dada su condición de entidad- es un elemento que debe ser modelado estáticamente. Por lo tanto, el modelo de comportamiento de la entidad dinámica, representado en el ciclo de vida modelado con la Red de Petri, debe ser consistente con la representación estática de la entidad en el Diagrama Entidad Relacionamiento. Una forma de lograr esto, que es coherente con la representación de la figura 4.17, es como se muestra en la figura 4.20, en la que los estados esenciales de la entidad dinámica son representados como un atributo de la entidad en el DER, que puede tener los valores correspondientes a los estados esenciales.

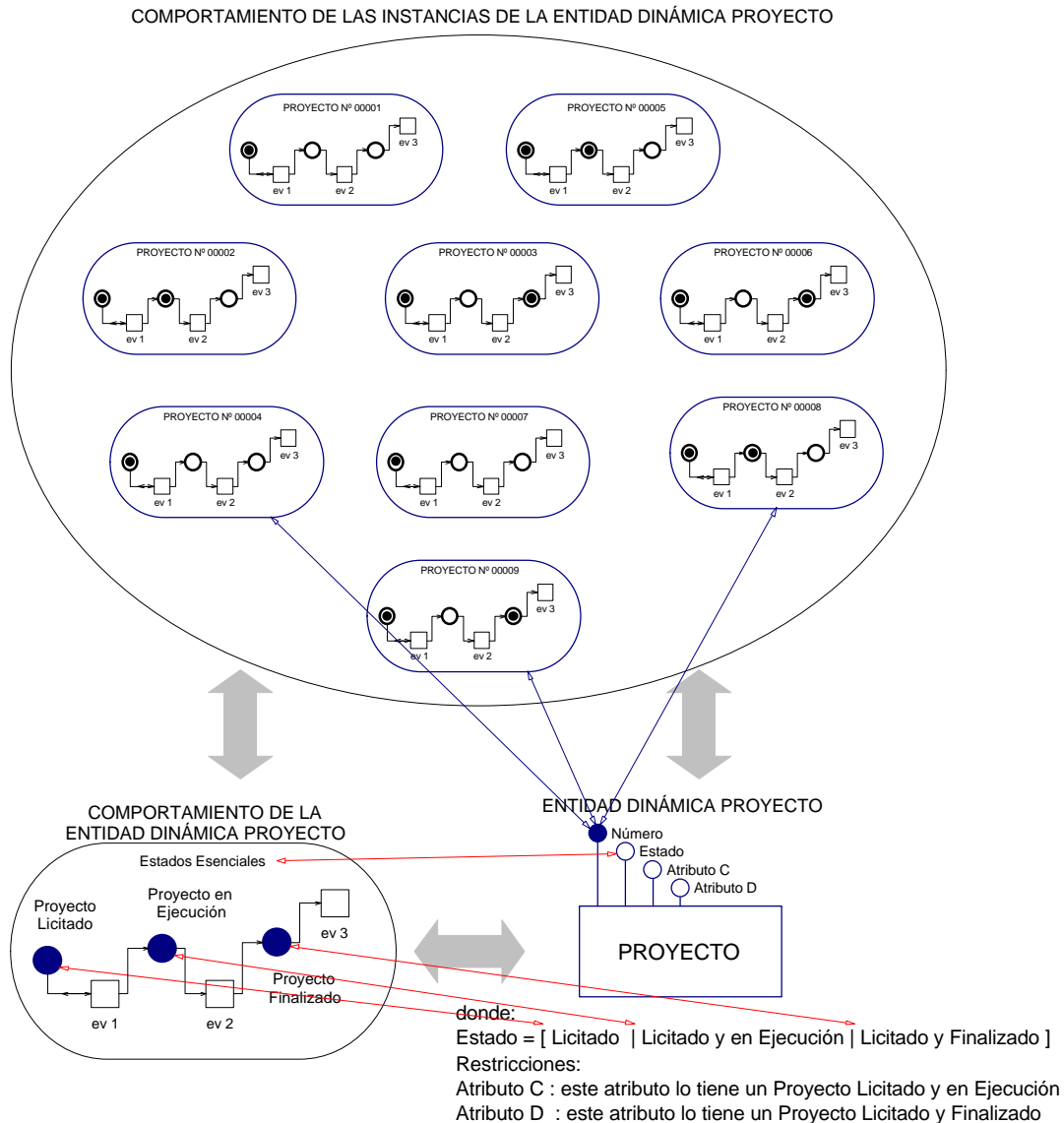


Figura 4.20: Modelo genérico de comportamiento v/s representación estática de la entidad.

Algunas de las deficiencias del DER presentado en la figura 4.20 son: no se representan ninguna relación entre los estados en que pueden estar las instancias de una entidad; se deben incorporar restricciones en el modelo para representar qué atributos puede tener la instancia durante un estado específico.

Otro criterio para clasificar las instancias de la entidad dinámica es de acuerdo al estado en que se encuentre, tal como se muestra en la figura 4.21, con este criterio se pueden clasificar las instancias del ejemplo en tres tipos, que son: Proyectos licitados, Proyectos en ejecución y Proyectos finalizados.

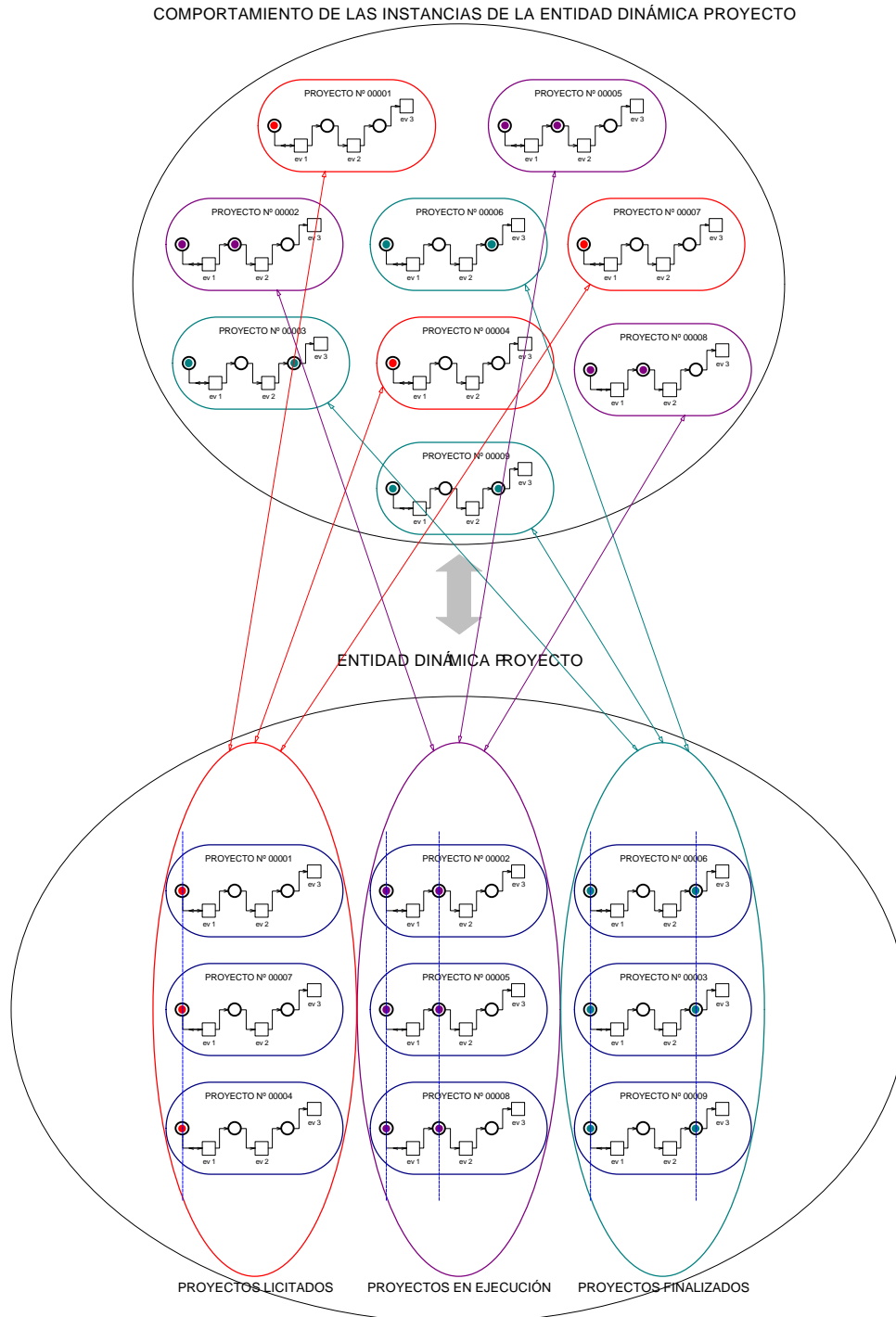


Figura 4.21: Criterio de clasificación según los estados de las instancias.

Esta forma alternativa de clasificar las instancias puede representarse en el DER de manera distinta a la presentada en la figura 4.20, esto es, pueden representarse los subconjuntos de las instancias de la entidad dinámica como subconjuntos de la entidad en una jerarquía de tipos, tal como se muestra en la figura 4.22.

De esta forma, se deben clasificar las subentidades de acuerdo a las características que tenga la entidad en los estados respectivos. Las instancias de la entidad dinámica podrán clasificarse en las distintas subentidades de la jerarquía durante su permanencia en el sistema, de acuerdo al estado en que se encuentre en ese momento, a la transición entre las subentidades se le denomina clasificación dinámica, según lo expuesto en [Odell98].

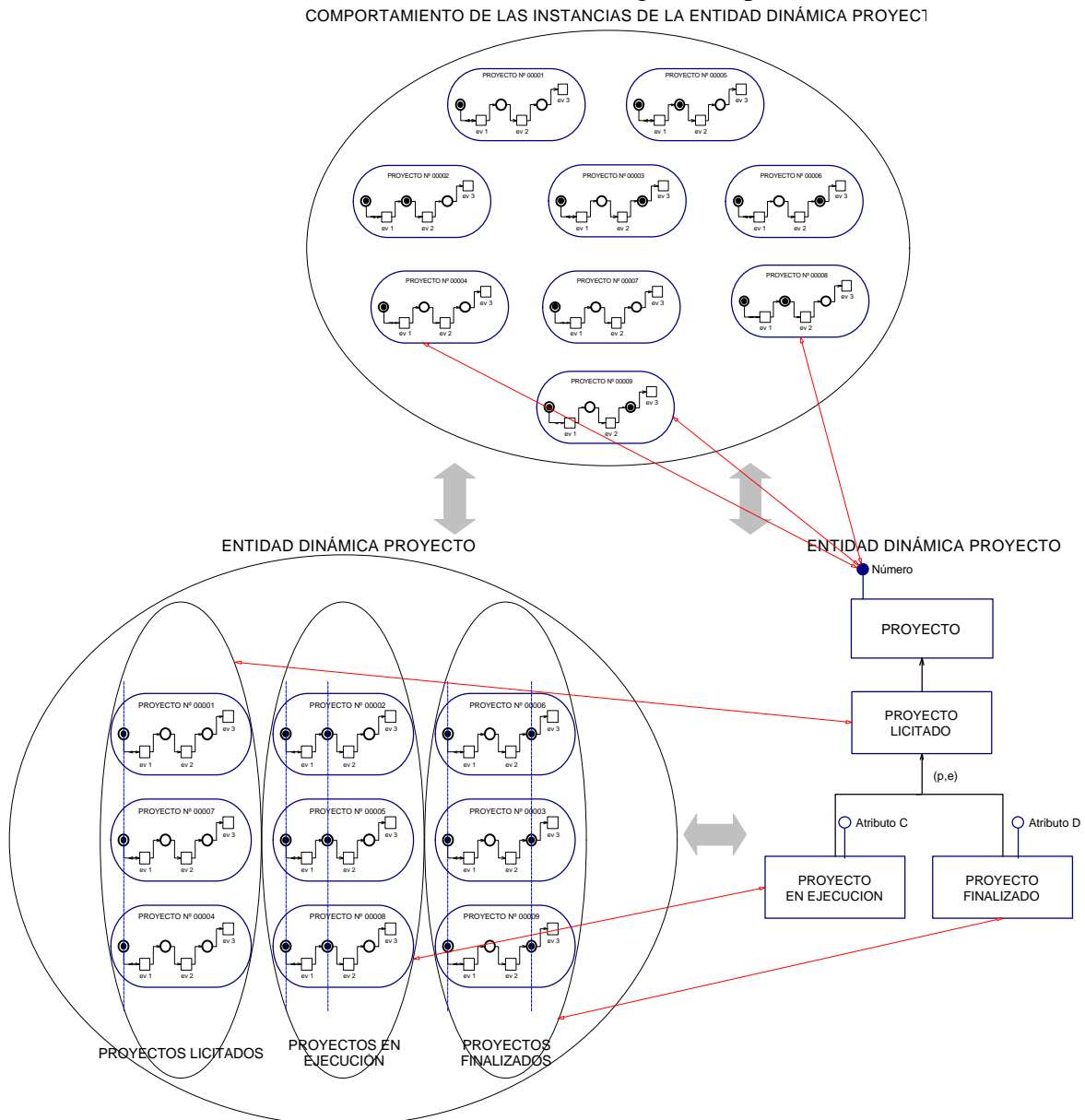


Figura 4.22: Criterio de clasificación de estados v/s representación estática de la entidad.

Por lo tanto, la representación estática de la entidad dinámica se puede realizar de las dos formas mostradas en la figura 4.23. Ambas formas representan la entidad desde la perspectiva estática, pero la jerarquía de tipos es más general que la entidad con atributos, ya que la jerarquía puede ser implementada como una entidad con atributos, pero la entidad con el atributo “estado” no puede ser implementada como una única jerarquía de tipos.

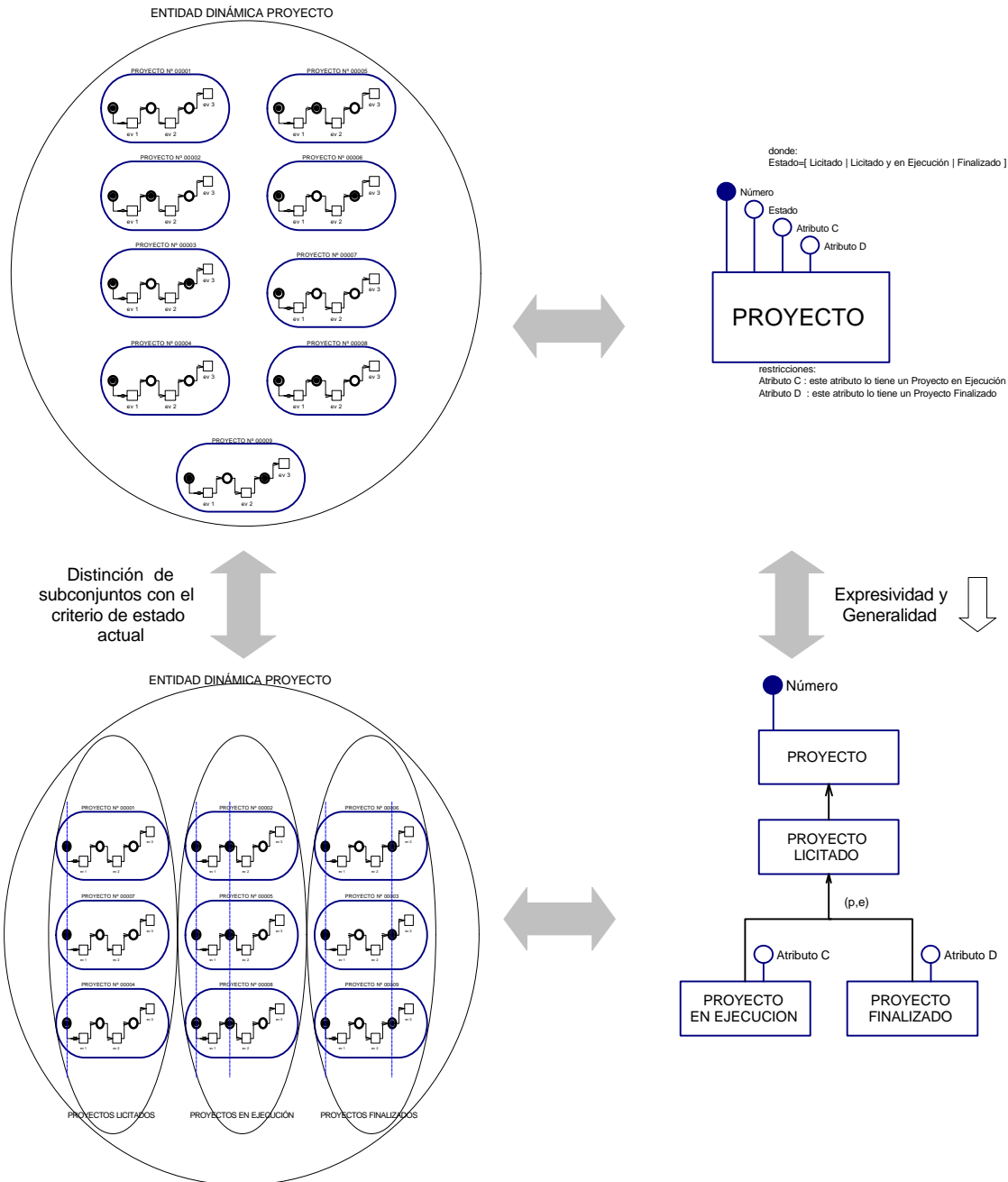


Figura 4.23: Resumen de criterios de clasificación v/s representación estática de la entidad dinámica.

Si se considera además, que la entidad dinámica se relaciona también con otras entidades que no fueron clasificadas como dinámicas. Al utilizar la jerarquía de tipos se puede visualizar de mejor forma la relación entre la entidad y otras entidades del sistema, obteniendo un modelo con mayor expresividad, ya que se eliminan las cardinalidades mínimas del tipo (0,1), y con mayor generalidad, ya que se evitan las restricciones, se observan de mejor forma los atributos y los estados a los que corresponden, tal como se muestra en la figura 4.24.

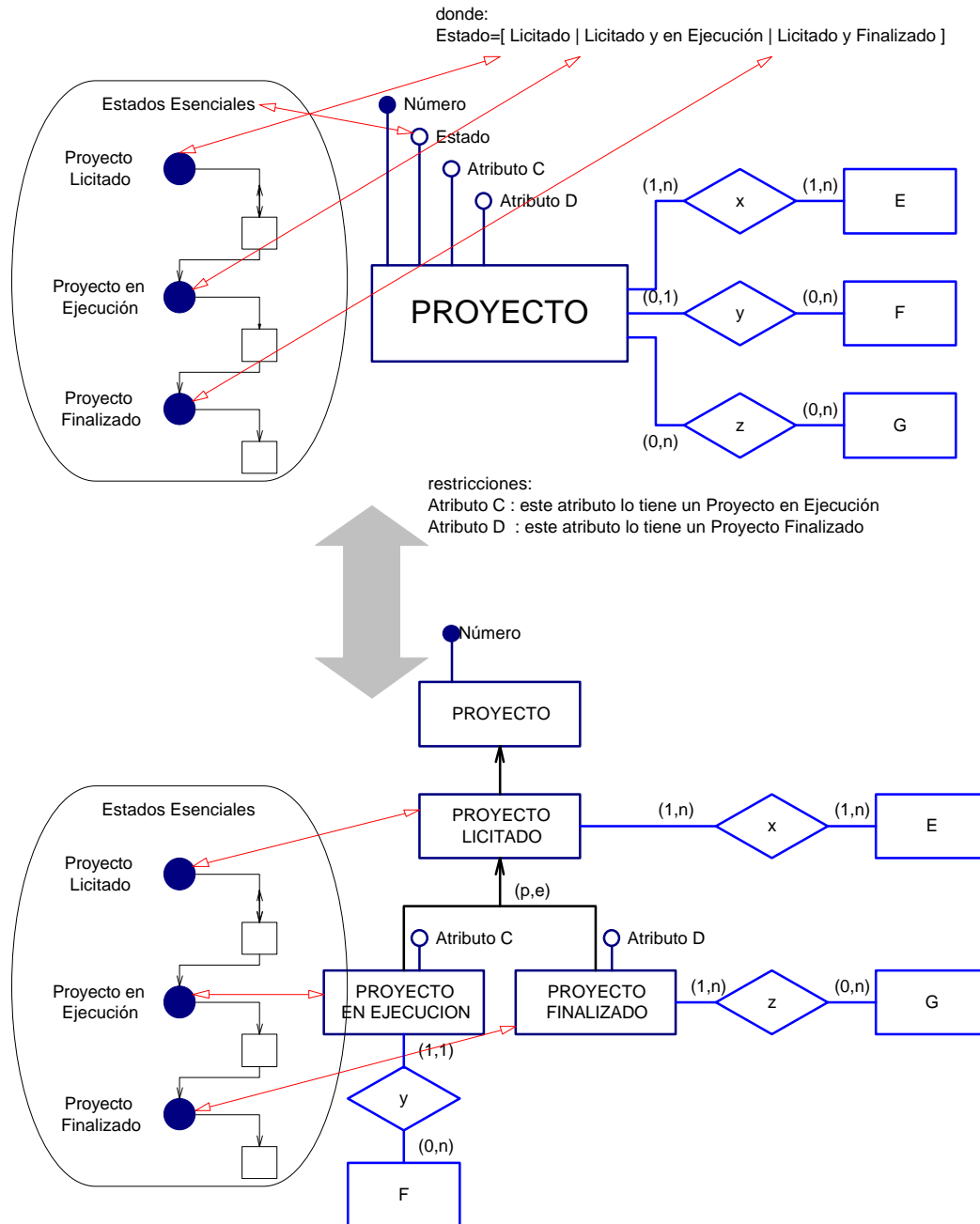


Figura 4.24: Representación de la entidad dinámica en el DER incluyendo relaciones con otras entidades.

Se establece entonces que los estados esenciales de la entidad dinámica deberán representarse desde la perspectiva estática como subentidades de una jerarquía de tipos en un Diagrama Entidad Relaciónamiento.

Para mantener consistencia entre los modelos, habrá que balancear las relaciones de totalidad o parcialidad y de exclusividad o sobreposición de la jerarquía del DER con las relaciones de concurrencia, secuencia y exclusividad de los estados esenciales del ciclo de vida, para lo que se pueden usar ramas restauradoras.

De acuerdo con lo expuesto inicialmente, en esta estrategia ningún modelo prima sobre otro y el orden en que se construyan no es establecido previamente, lo importante es que entre los modelos finales exista coherencia.

4.6 Relación dinámica – funcional

Para mantener consistencia entre los modelos dinámico y funcional, tal como muestra la figura 4.25, deberán balancearse principalmente tres elementos:

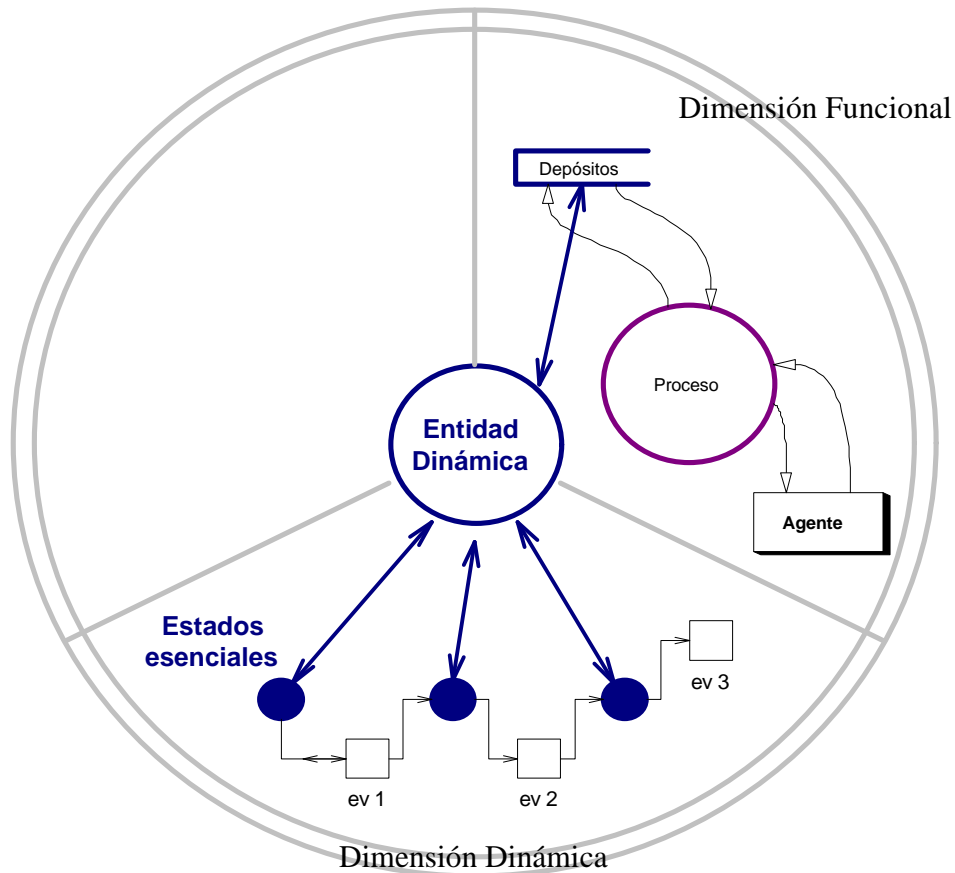


Figura 4.25: Entidad dinámica en las dimensiones funcional y dinámica.

Actividades	↔	Procesos
Agentes Externos	↔	Terminadores
Entidades dinámicas	↔	Depósitos

Cada actividad identificada en el ciclo de vida deberá estar representada como proceso elemental en un Diagrama de Flujo de Datos, tal como se propone en [Heuser90]. Si se recuerda que una actividad es un conjunto de estados y eventos que cambia de estado a una entidad o le crea uno nuevo, la relación actividad-proceso se presenta de manera intuitiva. Además, se pueden asociar los arcos de entrada y salida de las actividades a los flujos de entrada y salida de los procesos, aunque esta no es una relación uno a uno, debido a que los flujos del DFD son flujos de datos y los arcos de la Red de Petri son eventos que pueden o no contener datos.

Los agentes externos que interactúan con la entidad dinámica en la Red de Petri son representados como terminadores en el DFD.

Para efectos de lograr una mayor claridad del DFD, y por lo mismo, una mejor comprensión de éste, en el DFD se representarán los estados esenciales de la entidad dinámica como un sólo depósito, tal como lo muestra la figura 4.26.

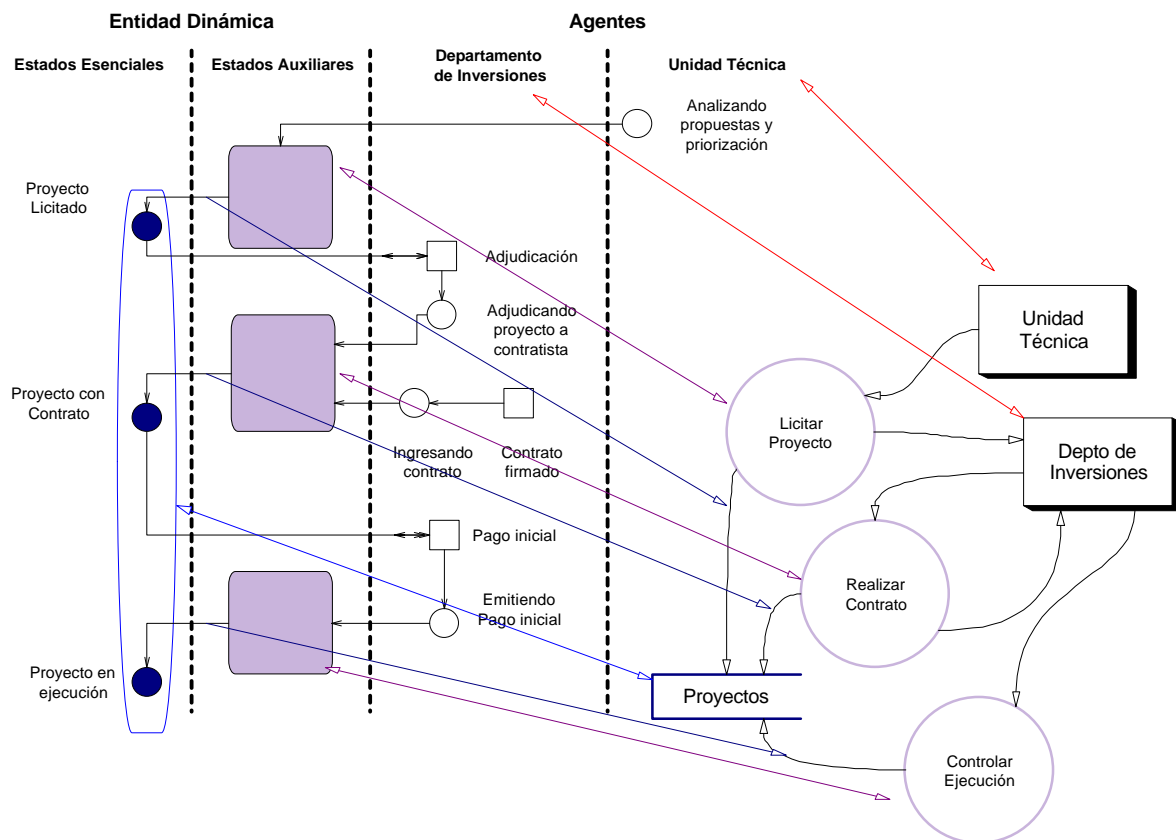


Figura 4.26: Relación dinámica-funcional.

La figura 4.26 muestra cómo se deben construir los modelos para mantener consistencia, con color rojo se muestran las relaciones entre las actividades y agentes del ciclo de vida con los procesos y terminadores del DFD y con color azul la relación entre los estados esenciales y el depósito que los representa.

4.7 Relación estática - funcional

Para mantener consistencia entre los modelos estático y funcional, tal como muestra la figura 4.27, deberán balancearse el DER y el DFD tal como en el Análisis Estructurado Moderno.

De esta forma, todas las entidades y relacionamientos con atributos del DER deberán representarse en el DFD como depósitos. Sin embargo, para este efecto, las entidades de la jerarquía de herencia, se clusterizan y se representan con un solo depósito en el DFD, como un criterio para mejorar la comprensión del mismo.

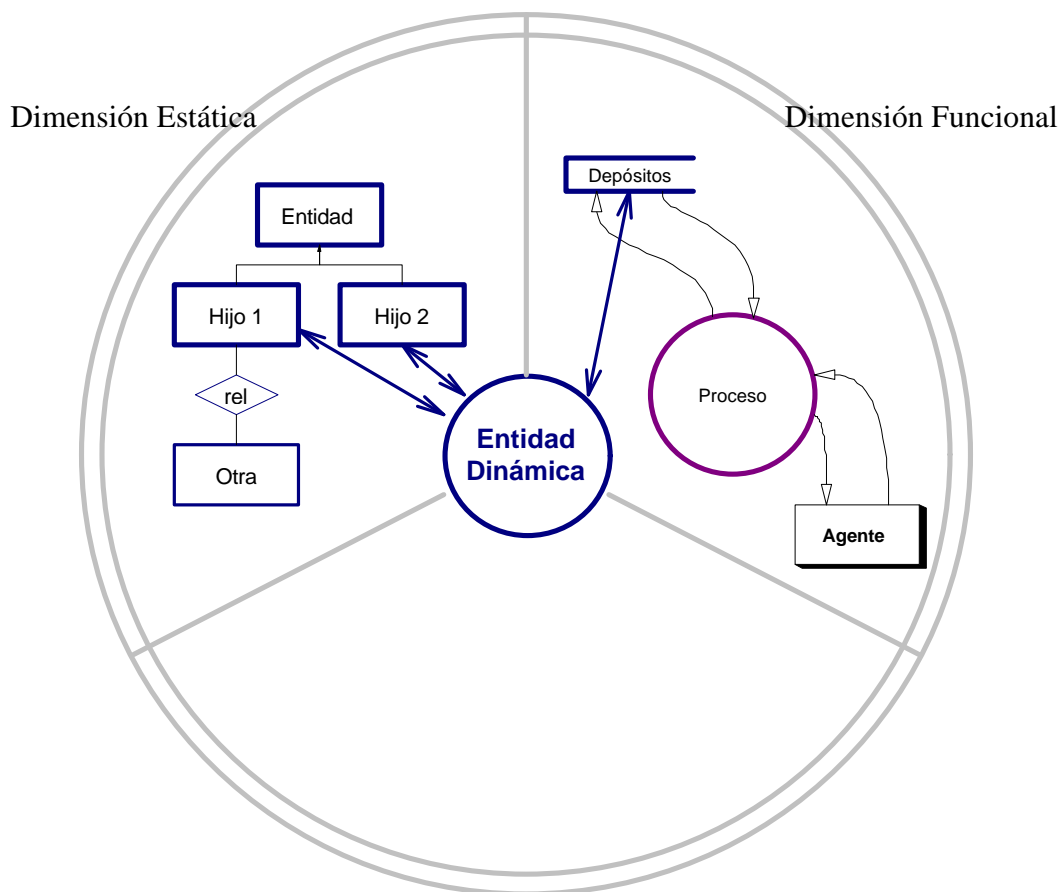


Figura 4.27: Entidad dinámica en las dimensiones estática y funcional.

En la figura 4.28 se muestra un ejemplo simplificado de la relación entre una porción del DFD con el DER.

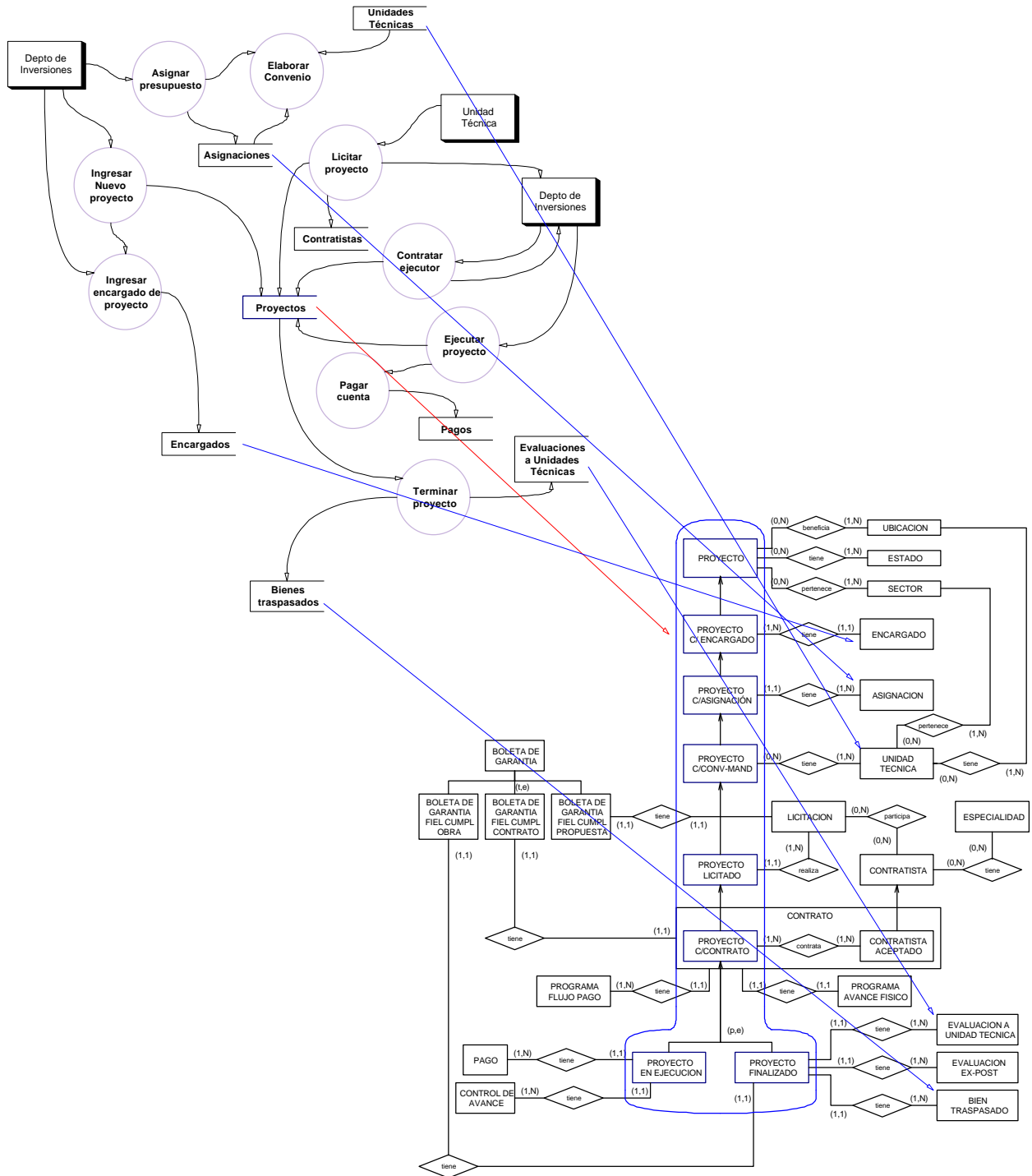


Figura 4.28: Relación funcional-estática.

Finalmente y de manera análoga a la figura 2.12, en la que se indican todos los modelos, con su nivel de abstracción y la dimensión que describen, se puede obtener un mapa de la estrategia propuesta considerando la descripción realizada en este capítulo. La figura 4.29 muestra cómo son utilizados los modelos en esta estrategia, según el nivel y la dimensión.

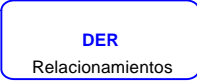
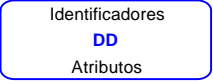
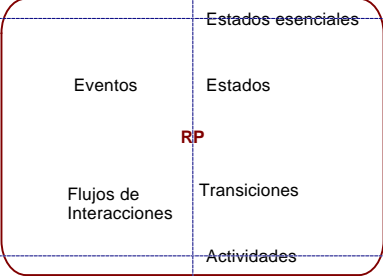
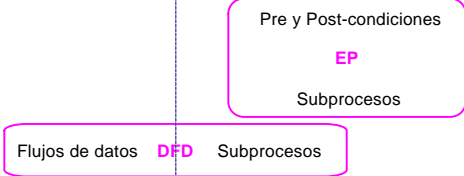
NIVEL DIMENSIÓN	ELEMENTOS A MODELAR	RELACIONES ENTRE ELEMENTOS	DESCRIPCIÓN DE LOS ELEMENTOS	JERARQUIZACIÓN DE LOS ELEMENTOS
DIMENSIÓN ESTÁTICA	Entidades			Jerarquización del DER: <i>Clusters</i>
DIMENSIÓN DINÁMICA	Entidad Dinámica			Jerarquización de RP: RP Canal/Actividad
DIMENSIÓN FUNCIONAL	Agentes Procesos			Jerarquización del DFD DFD de nivel más alto

Figura 4.29: Mapa dimensión – nivel de abstracción de la estrategia propuesta.

4.8 Reglas de balanceo entre los modelos utilizados

En esta sección se describen de manera detallada las reglas de balanceo y consistencia que se debe mantener entre los modelos. La figura 4.30 muestra una síntesis de los tres modelos y sus relaciones.

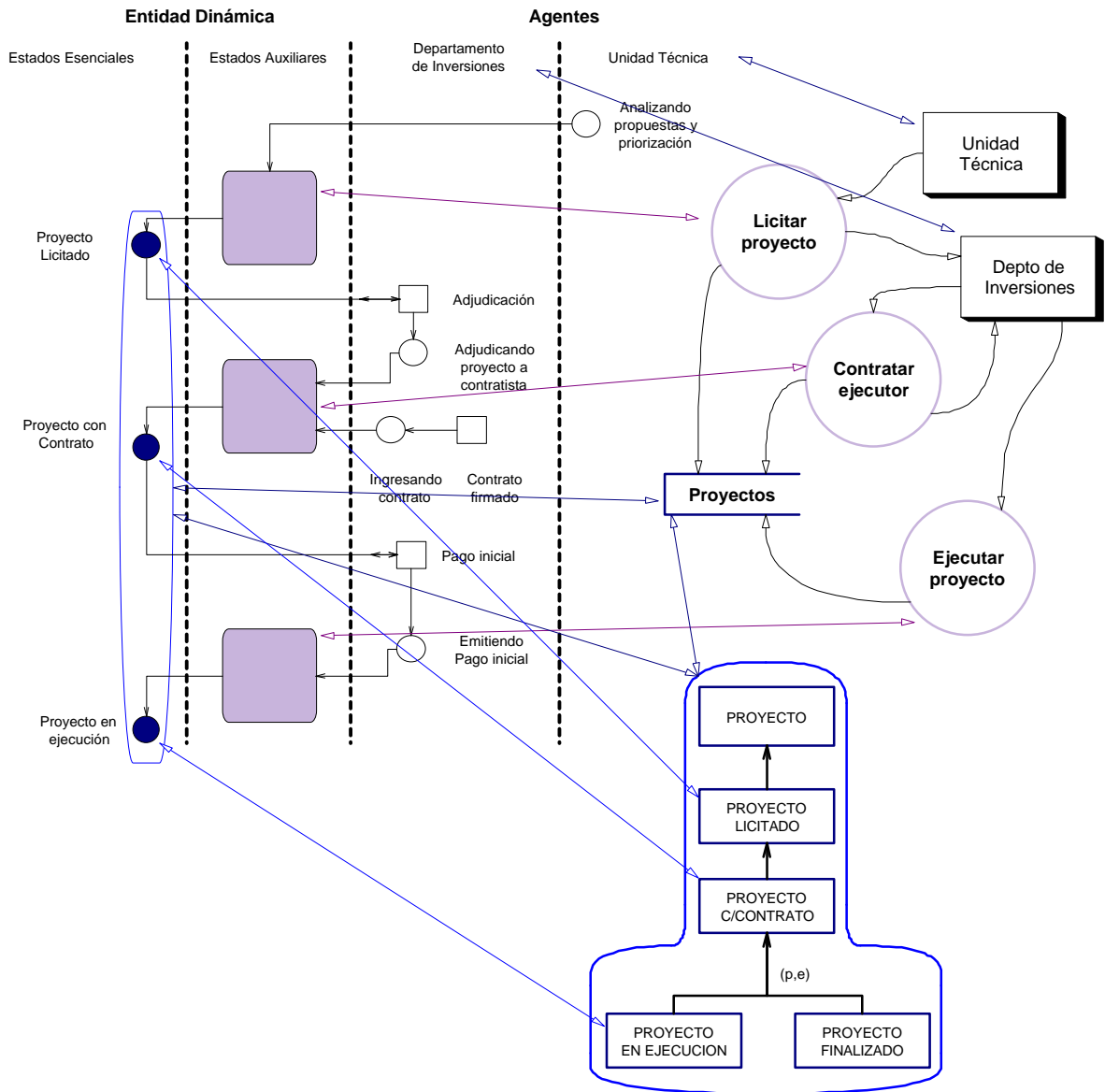


Figura 4.30: Relación tridimensional entre los modelos utilizados.

La tabla 4.1 resume los modelos que deben mantener consistencia entre sí.

Tabla 4.1 – Balanceo entre modelos utilizados en la Estrategia propuesta.

	DFD	DER	DD	RP	EP
DFD	4	4	4	4	4
DER	4	4	4	4	
DD	4	4	4		
RP	4	4		4	4
EP	4		4	4	

4 DFD - DFD:

Todo proceso de un DFD interactúa con otros procesos y puede pertenecer a procesos de mayor o menor abstracción. Esta regla es la misma que se establece en el AEM con este modelo (DFD).

4 DFD - DER:

Los depósitos del DFD corresponden a entidades o relacionamientos con atributos del DER, tal como se define en el AEM. En el caso la jerarquía de herencia de alguna entidad dinámica, ésta (la jerarquía) se representará en el DFD como un solo depósito, para efectos de simplificar el modelo y lograr mejor comprensión.

4 DFD - DD:

Todo terminador, depósito y flujo de datos del DFD debe estar definido en el DD. En el caso del depósito que representa una jerarquía de herencia, éste deberá definir en el DD a todas las entidades y subentidades que representa.

4 DFD - RP:

Los procesos elementales del DFD están asociados a una actividad de la RP. Asimismo, los flujos desde los procesos a los depósitos que representan la jerarquía de una entidad dinámica, están asociados a arcos entre un evento, al interior de la actividad asociada a ese proceso, y un estado esencial de la entidad dinámica.

Los procesos concurrentes del DFD están asociados a porciones independientes de la Red de Petri elemental.

La activación de los procesos del DFD está asociada a las entradas que provienen de estados esenciales de RP.

Por definición, las salidas de los procesos del DFD (actividades de la RP) hacia los depósitos que representan la jerarquía de herencia de una entidad dinámica, corresponden a estados esenciales de la RP.

4 DFD - EP:

Todo proceso elemental del DFD tiene asociado una EP.

4 DER - DFD:

Las entidades y relacionamientos con atributos del DER deben ser depósitos en el DFD, tal como se establece en el AEM. En el caso la jerarquía de una entidad dinámica, ésta se representará como un solo depósito, para efectos de simplificar el modelo y lograr mejor comprensión.

4 DER - DER:

El DER asociado a la jerarquía de herencia de una entidad dinámica se relaciona con la jerarquía de herencia de otra entidad dinámica con un relacionamiento simple o a través de otra entidad. Las jerarquías de herencia de las entidades dinámicas se *clusterizan* bajo la cohesión del tipo abstracción.

4 DER - DD:

Toda entidad simple, agregada o *clusterizada*, relacionamiento con atributos, atributos de entidades, atributos de relacionamientos e identificadores de entidades del DER debe estar definida en el DD.

4 DER - RP:

Todos las subentidades de la jerarquía de herencia de cada entidad dinámica en el DER deben estar representadas por estados esenciales en el ciclo de vida de la entidad dinámica modelada con la RP.

Las definiciones de totalidad o parcialidad y exclusividad o sobreposición de la jerarquía de herencia del DER deben ser consistentes con las definiciones de estados esenciales concurrentes o excluyentes.

En general, todas las subentidades que están asociadas a estados esenciales deben definirse como parciales, a excepción del padre, él que representa una abstracción de toda la jerarquía. Si se define como total, la superentidad estaría asociada a un estado esencial en el cual la entidad dinámica no estaría nunca.

Por otra parte, la exclusividad en las subentidades se asocia a estados esenciales excluyentes (secuenciales o no) y la sobreposición a estados esenciales concurrentes (secuenciales o no).

4 DD - DFD:

Todo terminador, depósito y flujo de datos definido en el DD debe estar en el DFD. En el caso del depósito que representa una jerarquía de herencia, éste deberá definir en el DD a todas las entidades y subentidades que representa.

4 DD - DER:

Toda entidad simple, agregada o *clusterizada*, relacionamiento con atributos definidos en el DD deben estar representados en el DER

Todos los atributos de entidades, atributos de relacionamientos e identificadores de entidades definidos en el DD pueden estar representados en el DER.

4 DD - DD:

En el DD se pueden usar definiciones del mismo DD.

4 RP - DFD:

Una o más actividades definidas en el ciclo de vida de la entidad dinámica están asociadas a los procesos del DFD.

Asimismo, arcos entre un evento, al interior de la actividad asociada a ese proceso, y un estado esencial de la entidad dinámica, están asociados a flujos desde los procesos a los depósitos que representan la jerarquía de una entidad dinámica

Las actividades definidas en base a porciones independientes de la Red de Petri elemental están asociadas a procesos concurrentes del DFD.

4 RP - DER:

Todos los estados esenciales de una entidad dinámica modelada en la RP deben representarse como subentidades de la jerarquía de herencia asociada a esa entidad en el DER.

Los estados esenciales concurrentes o excluyentes deben coincidir con las definiciones de sobreposición y exclusión, respectivamente, en la jerarquía de herencia asociada a esa entidad en el DER.

4 RP - RP:

En el caso que se desee modelar más de una entidad dinámica, éstas deben representarse como agentes en el ciclo de vida de otra entidad dinámica, tal como se muestra en el Anexo 2. Aunque sería posible modelar ambos ciclos de vida en un solo modelo que los integre, esto puede complicar innecesariamente los modelos cuando los ciclos de vida sean grandes y complejos.

4 RP - EP:

Cada actividad de la Red de Petri Canal/Actividad se asocia a la EP que describe a un proceso elemental del DFD.

Los estados de los que provienen los arcos de entrada a la actividad son pre condiciones de la EP que describe al proceso asociado a la actividad.

Los estados a los que van los arcos de salida de la actividad son post condiciones de la EP que describe al proceso asociado a la actividad.

4 EP - DFD:

Toda EP describe a un proceso elemental del DFD.

Todas las acciones en la EP deben aparecer como entradas, salidas y accesos a depósitos del DFD.

Cada dato en la EP es flujo o depósito del proceso en el DFD.

4 EP - DD:

Cada dato en la EP es componente de flujo o depósito en el DD.

En la EP pueden haber términos locales al proceso.

4 EP - RP:

Toda EP describe a un proceso elemental del DFD asociado a una actividad de la Red de Petri Canal/Actividad.

Las EP tienen como pre condiciones a los estados de los que provienen los arcos de entrada a la actividad asociada al proceso que especifican.

Las EP tienen como post condiciones a los estados a los que van los arcos de salida de la actividad asociada al proceso que especifican.

5 Consideraciones finales

Siempre es bueno hacer algunas consideraciones al final de una exposición con el objetivo de destacar algunos puntos que no fueron desarrollados en el transcurso de la investigación. En este capítulo, se exponen algunas reflexiones, que pueden tener en cuenta los investigadores y analistas a la hora de modelar un sistema de información utilizando la Estrategia de Modelamiento Basado en Entidades Dinámicas.

5.1 Consideraciones respecto a la estrategia

Para partir, sin duda hay que recalcar que, tal como se ha expuesto, una estrategia no es una metodología y, por lo tanto, al construir los modelos no existe un orden ni modelos que primen sobre los otros.

En este documento se intentó explicar cómo debía modelarse cada perspectiva, independientemente del orden que se siguiera para modelar el sistema completo. Sin embargo, puede influir el inevitable orden lineal en que hay que describir las ideas cuando se utiliza el medio escrito y pensar que la dimensión dinámica es la primera que se debe modelar y posteriormente la estática y la funcional. Esto, categóricamente, no es así, perfectamente se puede comenzar modelando la dimensión estática o bien la funcional.

Además, al construir los modelos, éstos pueden sufrir cambios de acuerdo a lo modelado en otra dimensión. Por ejemplo, puede ser que se haya modelado el ciclo de vida de una entidad dinámica y posteriormente, una vez modelada la dimensión estática, sea necesario agregar o quitar un estado esencial del ciclo de vida, lo que a su vez, modificaría algún depósito del Diagrama de Flujo de Datos o, tal vez, al modelar la dimensión funcional, sea necesario reagrupar las actividades del ciclo de vida.

Quizás, un buen ejercicio, que pondría a prueba la claridad de la exposición que se realiza en este documento, sería leer las secciones del Capítulo 4, en un orden aleatorio distinto al que fue establecido. De esta forma, podría evitarse la influencia recibida producto del orden de la lectura.

Lo importante, para que el modelamiento y el desarrollo completo sea un producto de calidad, es que los modelos finales mantengan consistencia interna y entre sí, de acuerdo a las reglas de balanceo establecidas.

5.2 Consideraciones respecto al modelamiento

Por otra parte, una decisión que debe tomar el analista en las etapas iniciales del modelamiento desde la perspectiva dinámica es determinar cuál entidad es dinámica y cuál no; se debe recordar que es una decisión exclusiva del analista que se enfrenta al problema. Una regla general es que si para representar el sistema es necesario conocer cómo se comporta alguna entidad, entonces debe modelarse su ciclo de vida como se propone en esta estrategia.

Sin embargo, si se desea, cualquier entidad que ha sido modelada en el DER, puede ser modelada dinámicamente, aunque su comportamiento no sea muy complejo.

Un caso especial de entidad que se puede modelar en el DER es una Entidad Agregada. Esta entidad debe considerarse desde el punto de vista dinámico como una entidad más, vale decir, si es necesario conocer su comportamiento debe modelarse su ciclo de vida. De todas formas, en estos casos en particular debe tenerse especial consideración en la consistencia interna de los modelos de ciclos de vida y ser consistentes con los modelos de las entidades que agregan.

Otro aspecto que se detectó, cuando se modeló el caso de estudio presentado en el Anexo 2, es que aparecen eventos en el carril o *swimlane* de estados auxiliares de la entidad, que pueden pertenecer a dos o más ciclos de vida. Esto es, un evento puede provocar que dos o más entidades cambien de estado. El analista puede resolver estas situaciones realizando algunas modificaciones a los modelos, de manera de mantener consistencia.

A modo de ejemplo, el evento “asignación ingresada”, que pertenece al *swimlane* de estados auxiliares de proyecto, produce que la entidad pase de “Proyecto en trámite de asignación” a “Proyecto con asignación”, pero además hace que la entidad cuenta presupuestaria pase de “Cuenta presupuestaria solicitada” a “Cuenta presupuestaria creada”. Esta situación aparece en el caso de estudio, Figura A2.1, y se hicieron algunas modificaciones para poder representarla. En la figura 5.1 se muestra la situación inicial, y en la figura 5.2 se muestra como se resolvió.

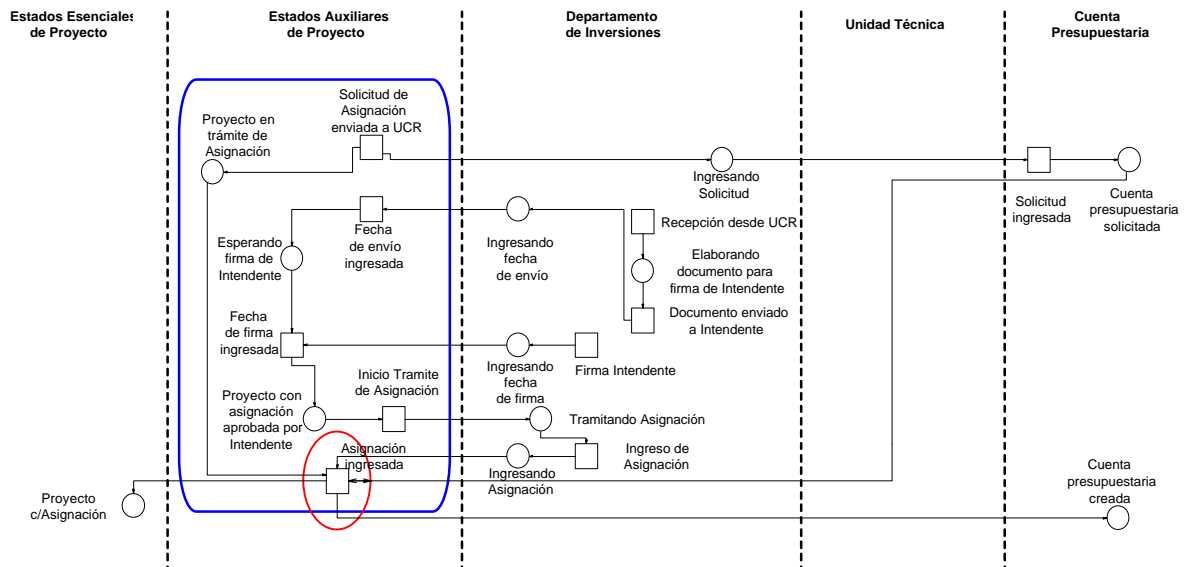


Figura 5.1: Evento que produce cambios de estados en dos entidades.

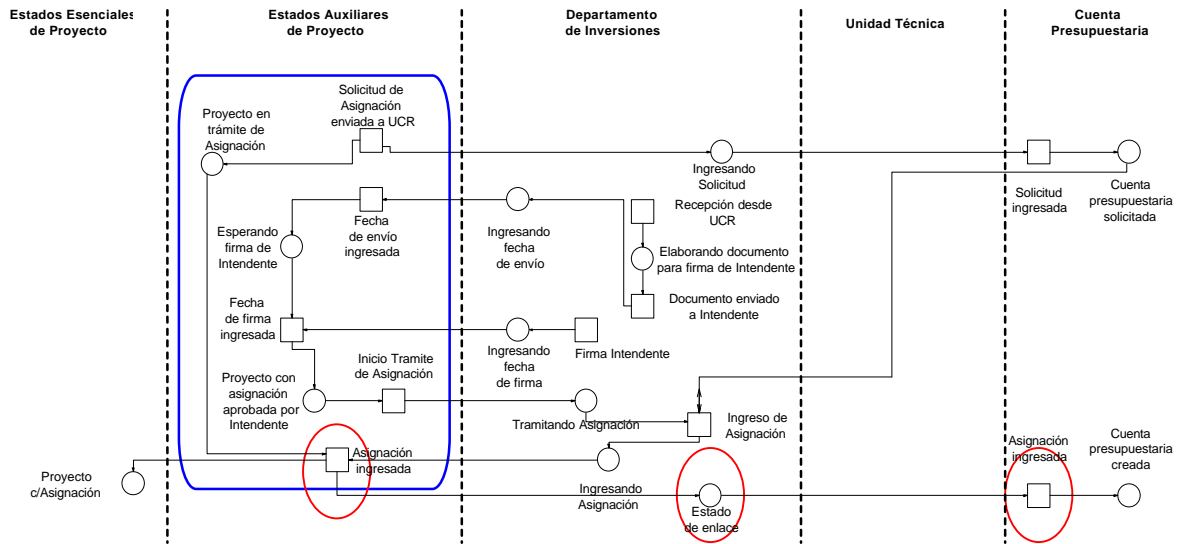


Figura 5.2: Solución al evento que produce cambios de estados en dos entidades.

Tal como se muestra en la figura 5.2, se creó un estado intermedio para poder representar la situación, a este estado se le denominó “estado de enlace”, ya que lo que hace es enlazar los ciclos de vida de dos o más entidades dinámicas.

Tal vez, se podría analizar de manera rigurosa la relación entre los modelos de ciclo de vida de varias entidades, con el fin de resolver conflictos como el presentado en este ejemplo y analizar mejor el caso de las entidades agregadas.

Por otra parte, una situación que se repite en el ciclo de vida de la entidad “Cuenta presupuestaria” del caso de estudio, es que aparecen eventos en el *swimlane* de estado auxiliares que son considerados como una actividad, es decir, además son representados como procesos en el Diagrama de Flujo de Datos. Generalmente, esta situación se produce con los eventos que crean nuevos estados esenciales y que en el DFD representan ingresos de datos o actualizaciones de depósitos.

Otra situación que puede producirse al modelar un ciclo de vida, que tiene relación con la explicada anteriormente es que pueden aparecer estados auxiliares y eventos que no pertenecen a ninguna actividad; esta situación en particular no se produce en el caso de estudio. En general, estas situaciones deberían revisarse y verificar la consistencia con el Diagrama de Flujo de Datos, en términos que realmente esos estados y eventos no pertenezcan a ningún proceso existente o si debiesen formar una actividad y consistentemente un nuevo proceso. Sin embargo, este tipo de estados y eventos en un ciclo de vida podrían modelarse sin problemas, y se pueden interpretar como estados por los que pasa la entidad dinámica que no están entre los estados de transición para lograr un estado esencial.

Otra particularidad, es que pueden modelarse actividades del ciclo de vida que tengan salidas no representadas, es decir, eventos que hacen que la entidad salga de un estado auxiliar, pero que no producen ningún otro estado. Debe tenerse en consideración esta

situación especialmente para mantener consistencia con el diagrama de Flujo de Datos, ya que estas salidas no representadas en el Modelo de ciclo de vida, pueden requerir representación en el proceso respectivo del Diagrama de Flujo de Datos. En general se produce esta situación cuando, en un ciclo de vida, una actividad finaliza y no produce ningún estado relevante para la entidad respectiva. La figura 5.3 muestra un caso en que la actividad tiene una salida que no se representa, por las razones mencionadas.

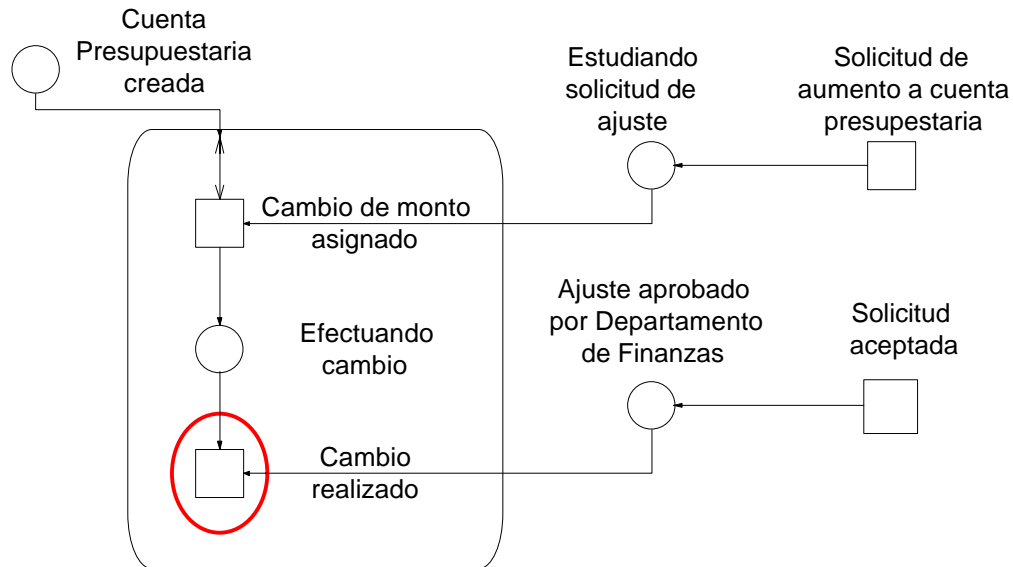


Figura 5.3: Salidas no representadas en una actividad.

Para finalizar, el modelamiento desde la perspectiva funcional especifica cómo se producen las transformaciones en el sistema, relacionando elementos de la perspectiva estática con elementos de la perspectiva dinámica. El ciclo de vida modelado con la Red de Petri entrega una visión dinámica de la transformación, por lo que, dependiendo del sistema a modelar, puede que no se requiera modelar el Diagrama de Flujo de Datos y que con la perspectiva dinámica de los procesos sea suficiente. Según [Heuser90], esto ocurre porque el DFD está orientado a una aplicación específica, en tanto que las Redes de Petri Canal/Actividad son más abstractas e independientes de las aplicaciones específicas.

Se puede mencionar además que existe relación entre los arcos de entrada y salida de las actividades del ciclo de vida y los flujos de datos del Diagrama de Flujo de Datos. Aunque se tiende a hacer una relación uno a uno, se debe destacar que la relación no es de equivalencia, ya que los arcos representan estímulos que pueden o no contener datos, por los que pueden asociarse a flujos de datos, en un caso, y a flujos de control, en otro, sin considerar los casos de las salidas no representadas, mencionadas anteriormente. En el caso que el estímulo contenga datos, debe mantenerse consistencia entre la Red de Petri y el Diagrama de Flujo de Datos.

6 Conclusiones

La estrategia presentada en este documento es una alternativa a las actuales estrategias y metodologías de modelado conceptual, que utiliza modelos consagrados y los relaciona de una forma distinta a las conocidas. Estos modelos son el Diagrama Entidad - Relacionamiento; la Red de Petri, con algunas variaciones; y el Diagrama de Flujo de Datos.

Se han propuesto los fundamentos de la estrategia y se ha establecido cómo utilizar los modelos para modelar los elementos de cada dimensión -Estática, Funcional y Dinámica- y cómo deben relacionarse para mantener consistencia.

A continuación se expondrán las conclusiones clasificándolas en: Conclusiones respecto de la estrategia, extensiones, comparaciones, experiencia personal y futuras investigaciones.

6.1 Conclusiones respecto de la estrategia

La estrategia propuesta tiene como elemento básico de construcción a la entidad dinámica, la cual tiene la particularidad de poseer atributos y comportamiento. Esta entidad dinámica es proyectada a todas las dimensiones: como entidad a la dimensión estática, como depósito a la dimensión funcional y como entidad dinámica a la dimensión dinámica. Basándose en esta unidad de construcción y utilizando herramientas y modelos existentes, se establecen relaciones entre todas las perspectivas, es decir, relaciones Estática - Funcional, Funcional - Dinámica y Dinámica - Estática. Esta última relación se debe resaltar, ya que ni el AEM, ni la Estrategia Genérica la establecen.

Con la integración de las dimensiones Dinámica y Estática se logra modelar conceptualmente sistemas de información haciendo énfasis en el comportamiento, con lo que se obtiene una visión más global y conceptual del sistema y su dominio, lo que no se logra con el Análisis Estructurado Moderno y se ha logrado en parte en el paradigma de la Orientación a Objeto.

Con la exposición de un caso práctico se ha intentado aclarar el uso de esta estrategia, en cuanto a cómo balancear los modelos y cómo resolver algunas complicaciones que pueden aparecer. Para esto, la descripción del caso práctico se realizó hasta la especificación de procesos y el diccionario de datos, con lo que se ha pretendido mostrar el modelamiento completo y detallado del caso.

6.2 Extensiones

Al ser ésta la primera propuesta formal de esta estrategia, se pueden detectar aplicaciones en otras áreas que se relacionan con los sistemas de información. A modo de ejemplo, se pueden mencionar el Modelamiento de Procesos de Negocio, Reingeniería de Procesos, el Control de Gestión y la Simulación.

Una extensión con la cual se pueden facilitar la aplicación en otras áreas y también en el área de sistemas, es utilizar otros modelos y comparar cómo se comportan en relación a los propuestos en este documento. De esta forma se puede mencionar la utilización de Casos de Uso en lugar del Diagrama de Flujo de Datos para modelar la perspectiva funcional, lo cual se podría complementar con el uso de Diagramas de Interacción para describir los Casos de Uso, tal como se propone en [Fowler&Scott97]. Intuitivamente, pareciera que estas relaciones se podrían establecer sin modificar la relación Dinámica - Estática, aunque debería resolverse la relación Funcional - Estática, entre los Casos de Uso y el Diagrama Entidad - Relacionamiento.

Por otra parte, una extensión de mayor complejidad sería desarrollar una metodología de modelamiento. Una metodología está compuesta de un lenguaje de modelamiento y un proceso. Por una parte, el lenguaje de modelamiento está dado por los modelos que utiliza esta estrategia, los que se deben relacionar según se ha establecido. Para desarrollar una metodología a partir de esta estrategia, debe indicarse el proceso a seguir para elaborar los modelos. Una forma para estandarizar el proceso puede ser al considerar distintos escenarios tipos y ver cómo se debería enfrentar el problema en cada uno para lograr una generalización o bien describir cómo proceder en cada caso.

De todas formas, al formular una metodología, debiera tenerse en cuenta, por una parte, que no se debe concentrar al analista en modelar la solución al problema, sino que lo debe ayudar a cumplir el objetivo principal del análisis conceptual, cual es, modelar el dominio del problema y, por otra parte, que debiera ser lo suficientemente flexible para que los analistas puedan utilizarla en distintos casos sin preocuparse de adaptarla a cada uno.

6.3 Futuras investigaciones

Como última consideración se presentan algunas investigaciones que se vislumbran a futuro como producto de lo expresado en este documento y de la experiencia adquirida en la investigación.

Debido principalmente al aporte que puede entregar la relación Dinámico-Estática en el modelamiento conceptual, relación que en esta investigación se muestra bastante coherente y consistente, se derivan nuevas investigaciones en torno a ella.

Una opción es investigar y elaborar una fundamentación más profunda, tal vez en forma matemática, entre los estados esenciales del ciclo de vida de la Entidad Dinámica y las subentidades de la jerarquía de tipos del DER.

Otra investigación que podría complementar este estudio, es aplicar la estrategia a un caso más complejo que el analizado en este documento, con el objetivo de entregar mayor respaldo y validar de alguna forma la implementación de la solución que se resuelva del análisis y modelamiento conceptual.

También se podría investigar la transición desde el Modelamiento Basado en Entidades Dinámicas al Diseño e Implementación de una solución. No se debe olvidar que el Análisis y el Modelamiento de Sistemas se encuentran insertos dentro de un proceso mayor de Desarrollo de Sistemas. Para cuantificar la transición del Análisis al Diseño se podrían considerar implementaciones Orientadas a Objeto como implementaciones que no pertenezcan a este paradigma.

Una forma de potenciar el uso de esta estrategia es realizando una comparación con otras estrategias. En el futuro se podría comparar con las principales estrategias y metodologías, tales como el Análisis Estructurado Moderno o la Orientación a Objeto utilizando UML o con alguna metodología anterior de este paradigma.

La comparación se puede realizar basándose en distintos criterios, tales como calidad, economía, eficiencia, etc. En cuanto a la calidad, se pueden comparar los modelos obtenidos por los distintos métodos o estrategias en cuanto a representación del dominio del problema, satisfacción de los requerimientos, flexibilidad y adaptabilidad. También se pueden definir criterios económicos como rapidez y tamaño. Finalmente, combinando los criterios anteriores se pueden obtener razones de eficiencia como, por ejemplo, tiempo/adaptabilidad.

Aunque puede que no resulte fácil cuantificar la flexibilidad o el tamaño, lo importante es establecer los parámetros y que los analistas tengan capacidades y experiencias similares, para que los resultados no se afecten por factores externos a los que se quieren comparar.

Para finalizar, en este capítulo se han expuesto sólo algunas extensiones e investigaciones que se pueden realizar a partir de esta investigación. Debido a lo novedoso de la estrategia propuesta, resulta difícil acotar las aplicaciones, extensiones y estudios que podrían derivarse.

Por último, y de acuerdo con el objetivo planteado al comienzo de la investigación, este documento entrega el primer planteamiento formal de la estrategia de Modelamiento Basado en Entidades Dinámicas -MOBED- fundamentando las relaciones de integración entre modelos conceptuales de sistemas de información y estableciendo las bases para futuras investigaciones y extensiones.

Anexo 1: Redes de Petri

Como anexo, se entrega una descripción más profunda y detallada de uno de los modelos que se utilizarán en esta estrategia y que suele ser el más desconocido, la Red de Petri. Para esto se utilizarán las descripciones existentes en [Bustos99].

1 Elementos básicos

Las Redes de Petri fueron desarrolladas por Carl A. Petri en 1961 para modelar sistemas concurrentes. Es una red que consiste en:

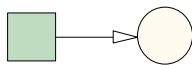
- lugares (estados)



- conexiones (eventos)



- arcos que conectan los lugares y las conexiones (transiciones)



- anotaciones opcionales
- reglas de funcionamiento específicas.

La figura A1.1 muestra un ejemplo de Red de Petri elemental.

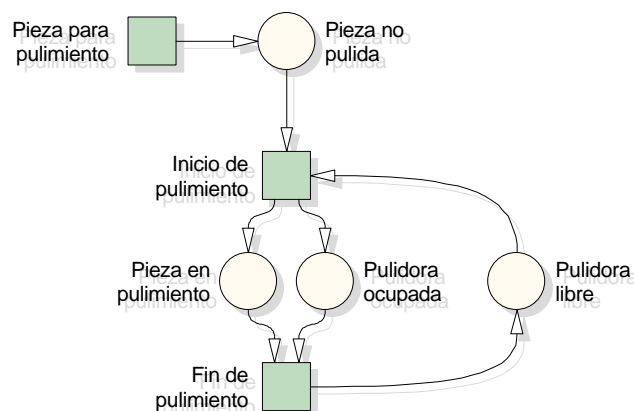


Figura A1.1: Ejemplo de una Red de Petri.

Fuente: [Bustos99]

2 Reglas de funcionamiento

La red, una vez construida, funciona de acuerdo a las marcas o fichas (●) que están presentes en uno o varios estados.

Para cambiar de estado, debe producirse un evento. Los eventos podrán consumir una o varias fichas, podrán producir fichas o podrán cambiar fichas a otros estados, de acuerdo a reglas de habilitación específicas, como las que se muestran en la figura A1.2.

A continuación se explica cada una de estas situaciones: en la figura A1.2.a) el evento toma ambas fichas y las pone en los estados siguientes; en la figura A1.2.b) debe llegar una ficha al estado que está desocupado antes del evento para habilitar el cambio; en la figura A1.2.c) hay que esperar que se desocupe el estado siguiente al evento para habilitar los cambios de estados; la figura A1.2.d) muestra dos porciones independientes de la red que pueden ser concurrentes; la figura A1.2.e) muestra dos eventos que pueden tomar la ficha y ponerla en el estado siguiente, en este caso, el evento que se produzca primero cambia la ficha de estado; finalmente la figura A1.2.f) el evento que se produzca primero pone una ficha en el estado siguiente, y la red queda no habilitada, tal como la figura A1.2.c).

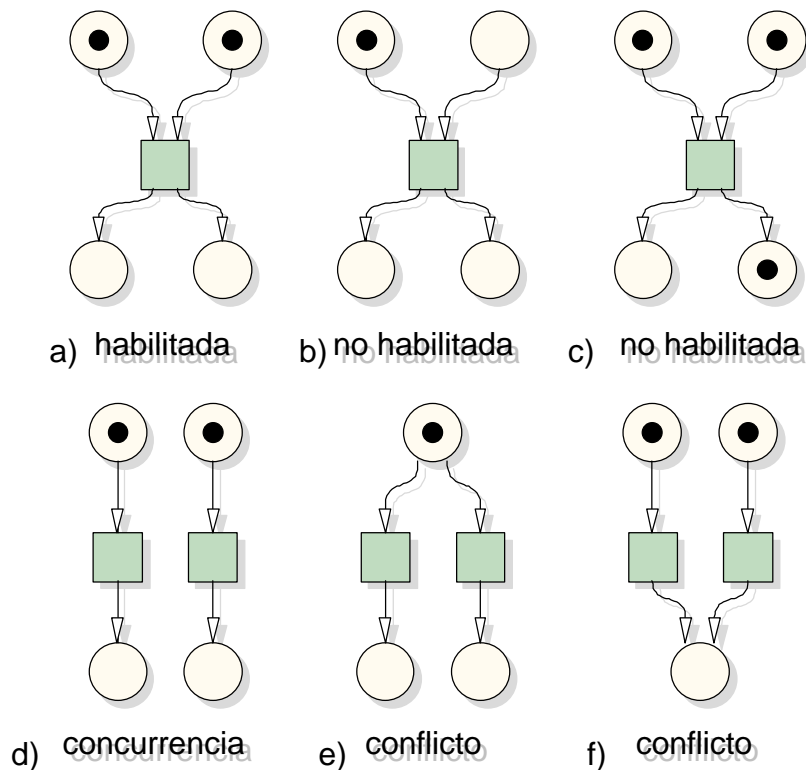


Figura A1.2: Marcación en las redes de Petri.

Fuente:[Bustos99]

3 Relaciones causales entre eventos

En ocasiones, al construir la red, aparecen problemas con la habilitación de estados, tal como el ejemplo de “cruza semáforo” presentado en la figura A1.3. Se desea representar que el cruce del auto depende del estado del semáforo, pero que su ocurrencia no modifica el estado del semáforo. Para solucionar este problema se utilizan ramas restauradoras. Existen dos tipos de ramas restauradoras, las cuales se describen a continuación:

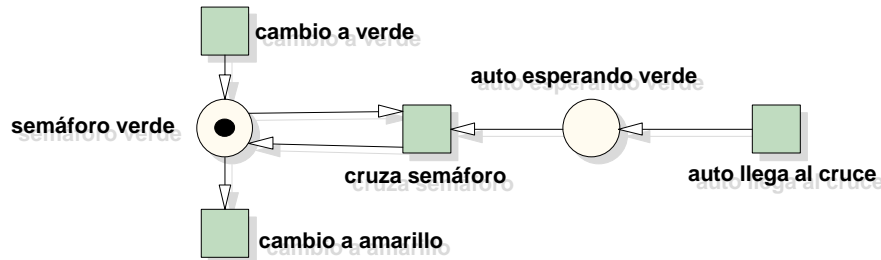


Figura A1.3: Problemas con la habilitación de “cruza semáforo”.

Fuente: [Bustos99]

3.1 Rama restauradora de entrada en una alteración

Habilita la alteración con la presencia de la marca de entrada, retira y coloca instantáneamente esta marca, ver figura A1.4.

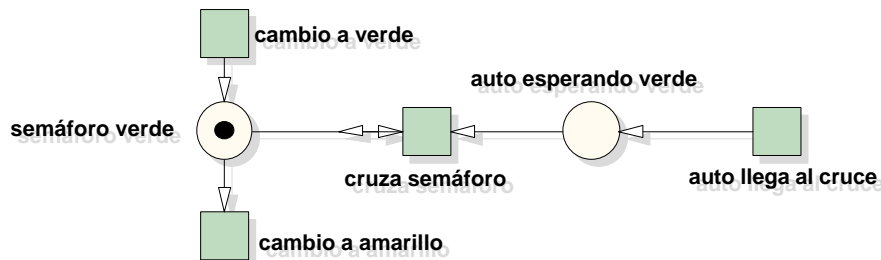


Figura A1.4: Ramas restauradoras.

Fuente: [Bustos99]

3.2 Rama restauradora de salida en una alteración

Habilita la alteración con la ausencia de la marca de salida, coloca y retira instantáneamente esta marca, ver figura A1.5.

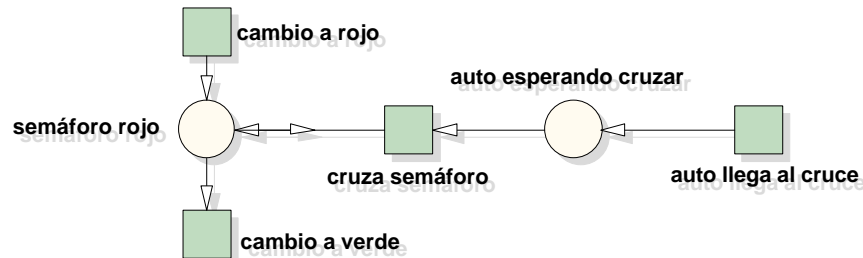


Figura A1.5: Rama restauradora de salida.

Fuente: [Bustos99]

4 Jerarquización en las Redes de Petri: Redes Canal/Actividad

Finalmente, un tipo especial de Redes de Petri, son las Redes Canal/Actividad. Estas redes son útiles para jerarquizar las redes elementales, al diferencia de las redes elementales, esta variante no posee reglas de funcionamiento. Sus componentes son:

- canal:



- actividad:



- arcos: conectan actividades y canales



Existen algunas reglas que se utilizan para relacionar las redes detalladas (elementales) y las abstractas, éstas son: los elementos abstractos representan a todos los elementos detallados (relación 1 a n), los arcos abstractos deben respetar el sentido de los arcos detallados, los elementos detallados de un elemento abstracto deben ser consistentes con el ambiente y los límites del mismo, ver figuras A1.6 y A1.7.

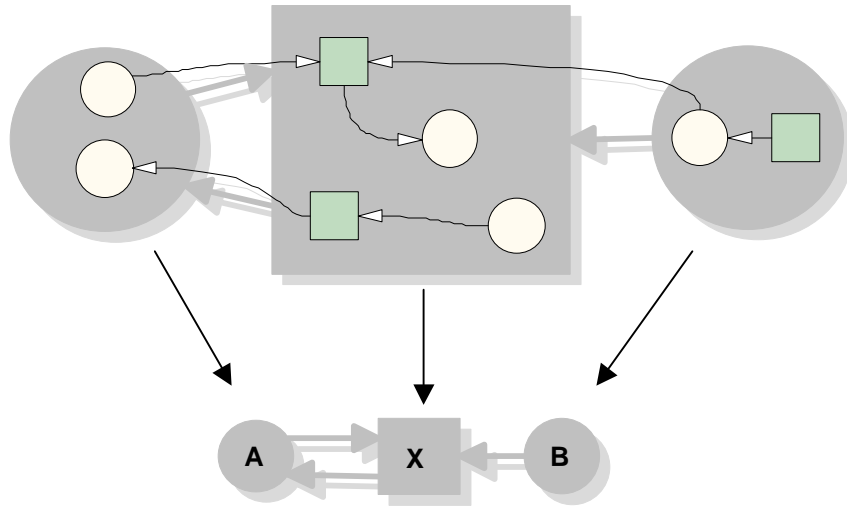


Figura A1.6: Cómo formar una Red de Petri Canal/Actividad.

Fuente: [Bustos99]

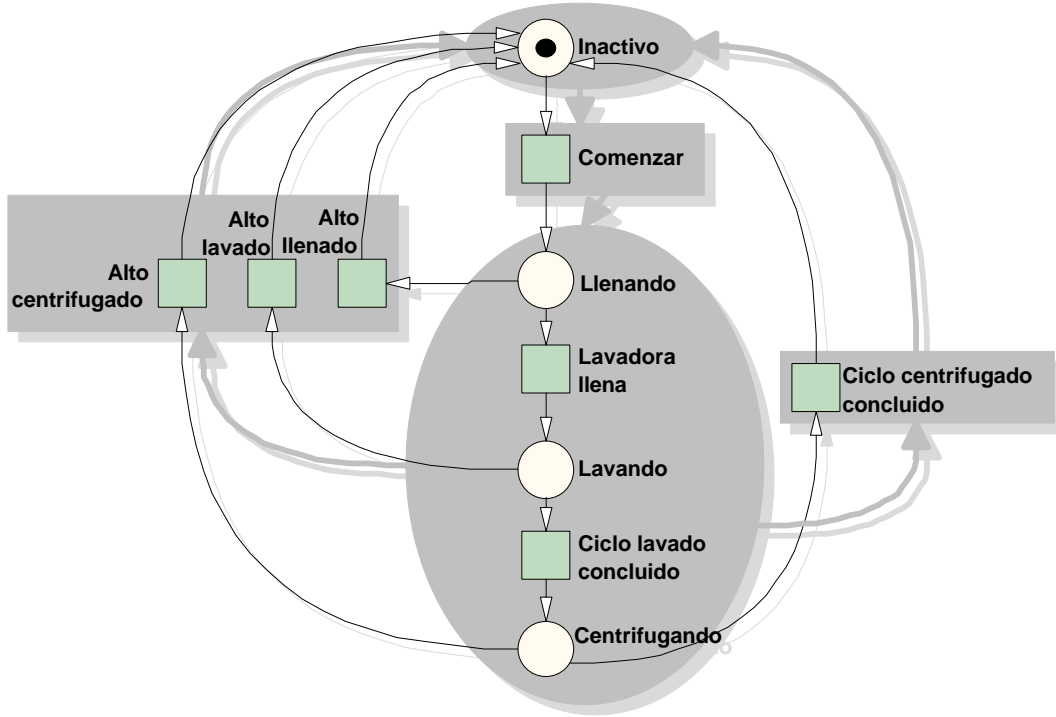


Figura A1.7: Ejemplo de una Red de Petri Canal/Actividad.
Fuente: [Bustos99]

La Red abstracta resultante para el ejemplo anterior es la que se muestra la figura A1.8.

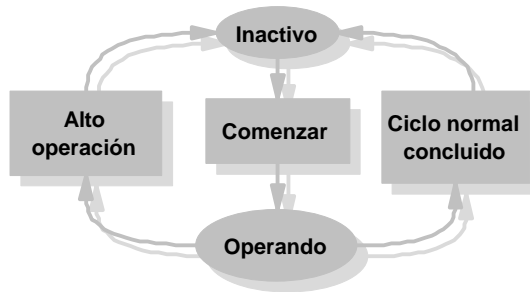


Figura A1.8: Red de Petri Abstracta Canal/Actividad.
Fuente: [Bustos99]

Anexo 2: Caso de estudio

Una forma de entender mejor la estrategia descrita en este informe, es explicándola con un caso práctico. El objetivo de este capítulo es entregar al lector un problema real modelado con la estrategia propuesta.

1 Descripción del problema

El Gobierno Regional de Valparaíso se encarga de administrar los recursos monetarios del Fondo Nacional de Desarrollo Regional, con los cuales se financian proyectos de distintos tipos que contribuyan a mejorar la calidad de vida de los habitantes de la Región de Valparaíso.

El sistema utilizado en la actualidad, dificulta la obtención de información oportuna y completa, que apoye el control y la gestión de proyectos financiados con el FNDR. Con el fin de utilizar mejor la información con que cuentan los niveles de gestión, se ha decidido crear un sistema informático que apoye eficientemente decisiones tales como: cuánto asignar a un proyecto, a qué unidad técnica se debe entregar la responsabilidad de supervisar la ejecución de un proyecto, o a quién contratar para ejecutarlo. Además, el sistema debe mejorar el control de la ejecución de un proyecto, ya que debido a la gran cantidad de proyectos que se ejecutan simultáneamente, es muy probable que el promedio de las decisiones que se tomen sean buenas, pero en casos puntuales pueden ser muy pobres.

Un proyecto financiado con el FNDR pasa por varias etapas, a grandes rasgos estas son: Priorización, Asignación presupuestaria, Convenio Mandato, Licitación, Adjudicación y Contrato, Ejecución y Control y Término. A continuación se describe brevemente cada una de estas etapas:

1. **Priorización:** Durante el transcurso del año, SERPLAC -que es una institución dependiente del Ministerio de Planificación y Cooperación- recibe las propuestas y proyectos de entidades públicas. Posteriormente este organismo revisa su formulación y se evalúa la factibilidad de ejecutarlo. Una vez al año, cuando el Gobierno central otorga el presupuesto correspondiente al FNDR, el departamento de Desarrollo Regional elabora una cartera de proyectos factibles -considerando en ella el presupuesto y los proyectos pendientes- a partir de la cual el Consejo Regional aprueba una cierta cantidad de proyectos.
2. **Asignación presupuestaria:** En esta etapa, se asigna un encargado al proyecto y se solicita autorización a la Unidad de Control Regional para asignarle un presupuesto inicial, este presupuesto puede variar, con la autorización de la UCR, a medida que surjan inconvenientes o gastos extraordinarios.
3. **Convenio Mandato:** Se le asigna la responsabilidad de ejecutar y controlar el avance del proyecto a alguna Unidad Técnica que está capacitada para hacerlo, el conjunto de

Unidades Técnicas de la Región ya está establecido, es decir, de las UT que existen hay que elegir una que se responsabilice de realizar la licitación, contratar, controlar la ejecución y la recepción final.

4. **Licitación:** La Unidad Técnica elabora bases de licitación, las que deben ser aprobadas por el Departamento de Inversiones del Gobierno Regional, y posteriormente deben publicarse a través de un medio determinado.
5. **Adjudicación y Contrato:** Luego de recibir y analizar las propuestas recibidas, la Unidad Técnica entrega una lista priorizada de las propuestas que debe ser revisada por el encargado del proyecto y finalmente realizar el contrato.
6. **Ejecución y Control:** Posterior a la firma de contrato y al pago inicial, comienza la ejecución del proyecto. La Unidad Técnica envía Estados de Pago al Departamento de Inversiones, el cual –salvo situaciones de excepción- paga al contratista.
7. **Término:** Una vez finalizado el proyecto y se ha realizado el último pago, se realiza su recepción provisoria y recepción definitiva. Posteriormente el Gobierno Regional evalúa el desempeño de la Unidad Técnica y finalmente se realiza una evaluación Ex-Post para medir el impacto real del proyecto en los habitantes de la Región.

Con la descripción del dominio del problema a modelar, el siguiente paso es presentar los modelos que describen el sistema desde las distintas perspectivas.

2 Modelamiento de la perspectiva dinámica, RP

Desde la perspectiva dinámica se construirá un modelo de Ciclo de Vida de dos entidades que son relevantes para el problema, estas son: Proyecto y Cuenta presupuestaria. Interesa modelar estas dos entidades ya que el objetivo del sistema es, por una parte, controlar y obtener información del estado de los proyectos, y por otra parte, mantener un registro del presupuesto y los pagos realizados por cada proyecto.

Es así como se presenta a continuación el modelo de Ciclo de Vida de un Proyecto y posteriormente el modelo de Ciclo de Vida de una Cuenta presupuestaria. En cada ciclo de vida se identifican las actividades y su nivel de cohesión, según se explicó anteriormente en el capítulo 4. En estos modelos es posible apreciar que existen estados y eventos que aparecen en ambos ciclos de vida. Por ejemplo, “solicitud ingresada” aparece en el ciclo de vida de la cuenta presupuestaria en el *swimlane* de estado auxiliar de cuenta presupuestaria y en el ciclo de vida de proyecto aparece en el *swimlane* del rol cuenta presupuestaria. Esto muestra que ambos ciclos de vida podrían modelarse en una única gran red, pero aún con este ejemplo simple, comprender eso sería difícil y complicado. Principalmente para los usuarios finales que, conociendo el dominio del problema, pero no estén familiarizados con la notación. A continuación se presentan en las figuras A2.1, A2.2 y A2.3 los ciclos de vida de las entidades dinámicas Proyecto y Cuenta Presupuestaria.

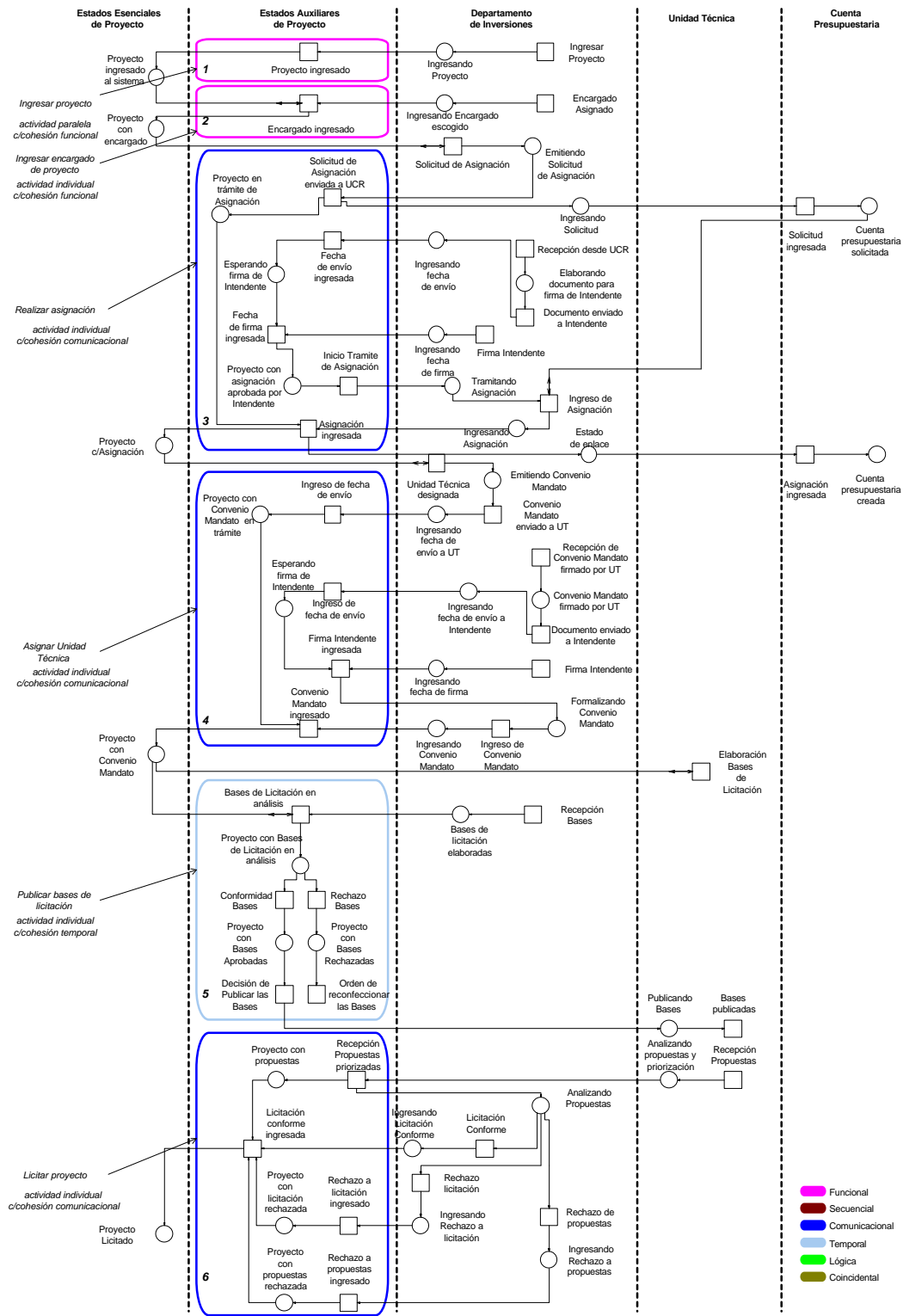


Figura A2.1: Ciclo de vida de la entidad Proyecto, parte 1.

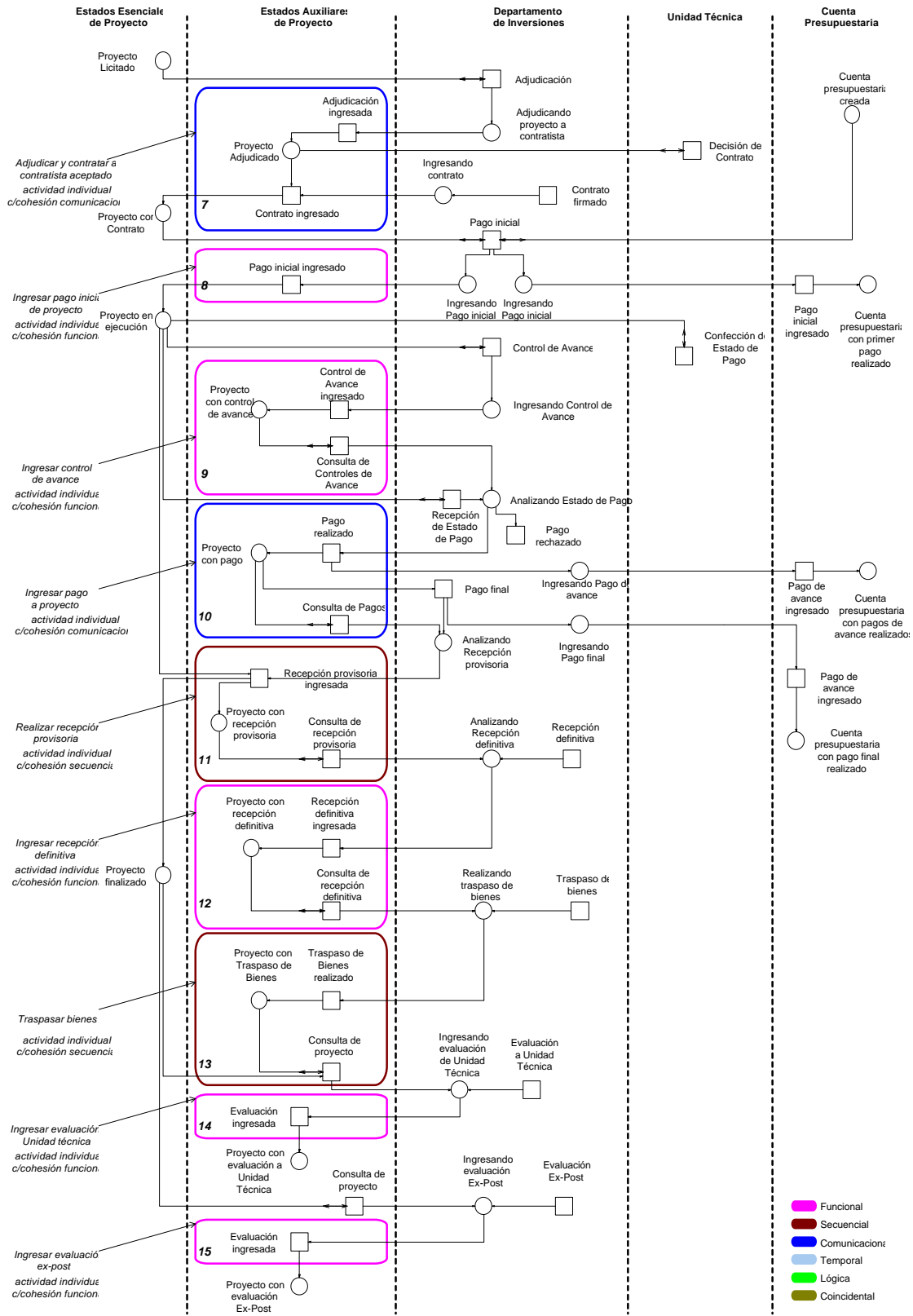


Figura A2.2: Ciclo de vida de la entidad Proyecto, parte 2.

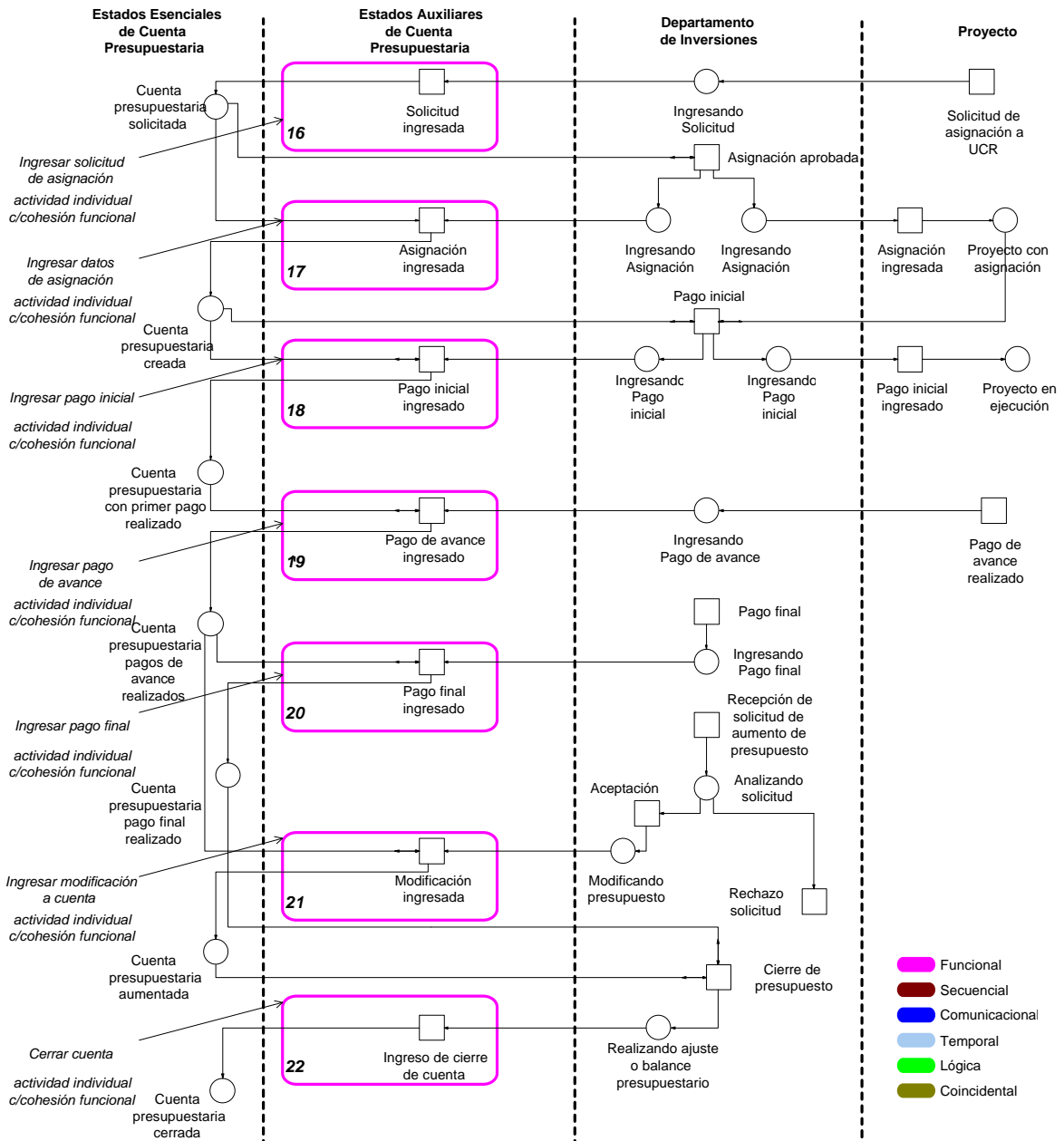


Figura A2.3: Ciclo de vida de la entidad Cuenta Presupuestaria.

3 Modelamiento de la perspectiva funcional

Desde esta perspectiva se modelan los procesos y transformaciones que realiza el sistema con un Diagrama de Flujo de Datos. La figura A2.4 representa todos los procesos que se asocian a actividades del ciclo de vida modelado con la red de Petri. En ella se puede apreciar la consistencia existente entre los procesos, los terminadores, los flujos desde y hacia los terminadores y los flujos desde y hacia los *depósitos esenciales* (asociados a entidades dinámicas).

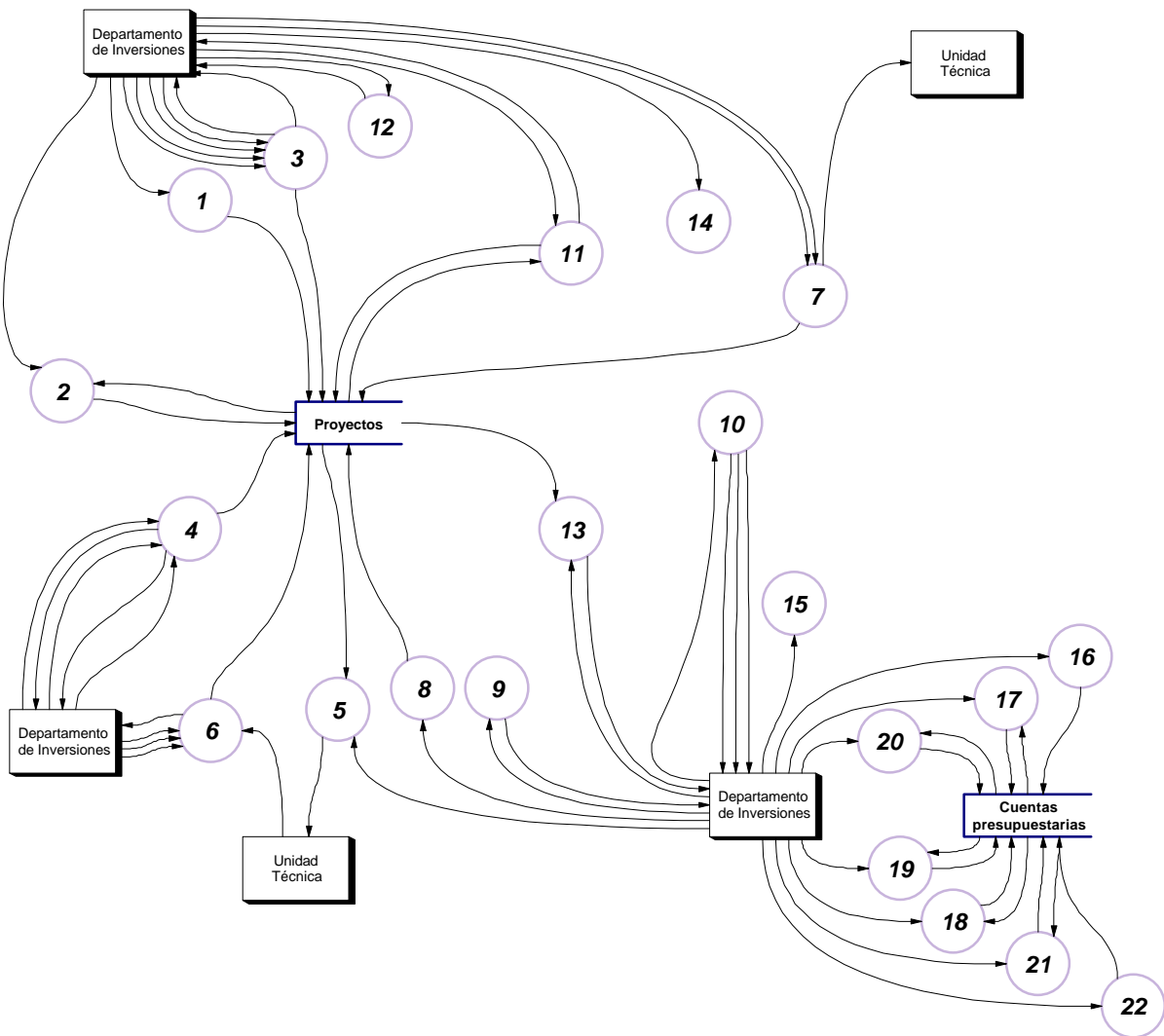


Figura A2.4: Diagrama de Flujo de Datos asociado al ciclo de vida.

En la figura A2.5 se representa el DFD extendido, es decir, el DFD que además de mostrar las relaciones con los terminadores y los *depósitos esenciales*, muestra las relaciones con los demás depósitos del sistema que han sido representados en el MER. Además, este DFD define procesos de actualización de algunos depósitos, los que continúan con la numeración: **23, 24, 25, 26, 27, 28 y 29.**

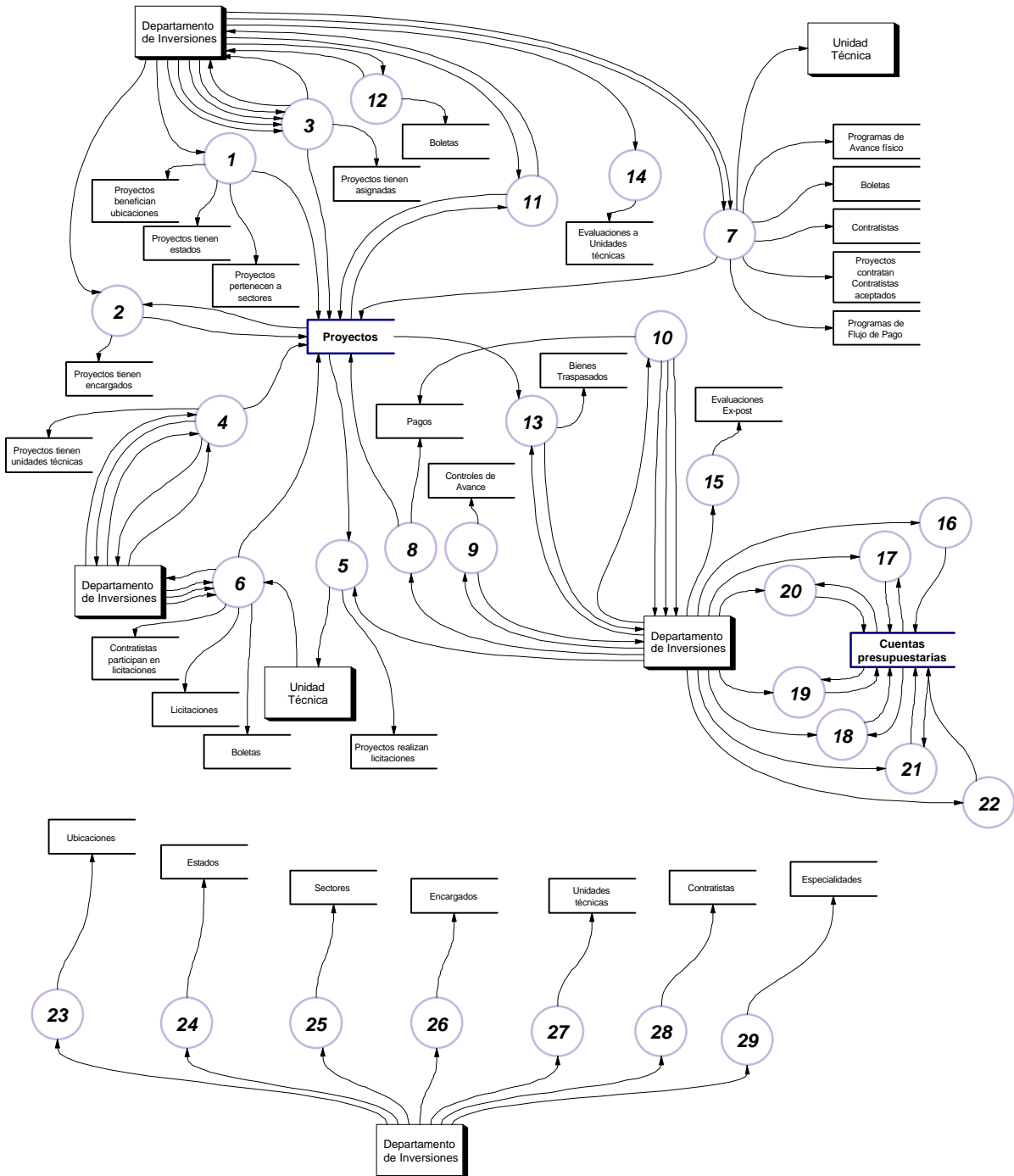


Figura A2.5: Diagrama de Flujo de Datos extendido.

3.1 Diagrama de Flujo de Datos

A continuación se presentará el DFD de manera *Top down*, es así como en la figura A2.6 se muestra el diagrama de contexto del Sistema de Información de Control de Proyectos, en el que se representa la relación entre el sistema y su entorno. Para el caso, el sistema se relaciona con el Departamento de Inversiones y la Unidad Técnica.

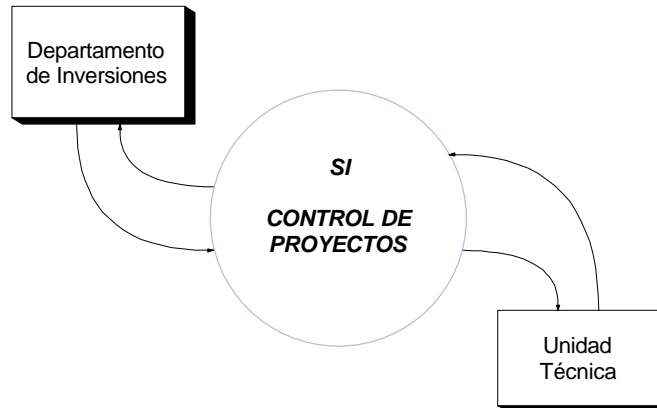


Figura A2.6: Diagrama de Contexto o de nivel cero.

A continuación, en la figura A2.7, se presenta el diagrama cero, en él se muestran los principales procesos que realiza el sistema. Como una forma de simplificar el modelo, se le ha asignado una cardinalidad a cada flujo representado entre los terminadores y los procesos, esta cardinalidad representa el número de flujos de información existentes para cada proceso, ya sea de entrada o de salida.

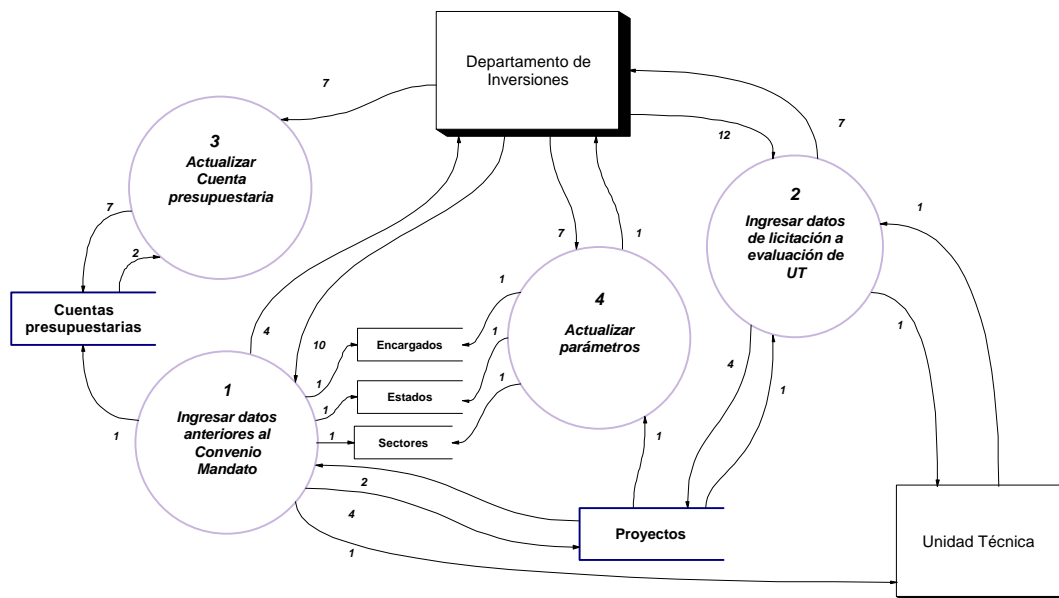


Figura A2.7: Diagrama cero o de nivel uno.

La figura A2.8 muestra el diagrama uno, en él se describe el proceso 1 a través de subprocesos que interactúan entre sí, con los depósitos y con los terminadores, posteriormente se describirá cada uno de los estos procesos con especificación de procesos de Pre y Post Condiciones. Además, en este nivel se nombran los flujos de datos, los que son definidos en el Diccionario de Datos junto con los depósitos.

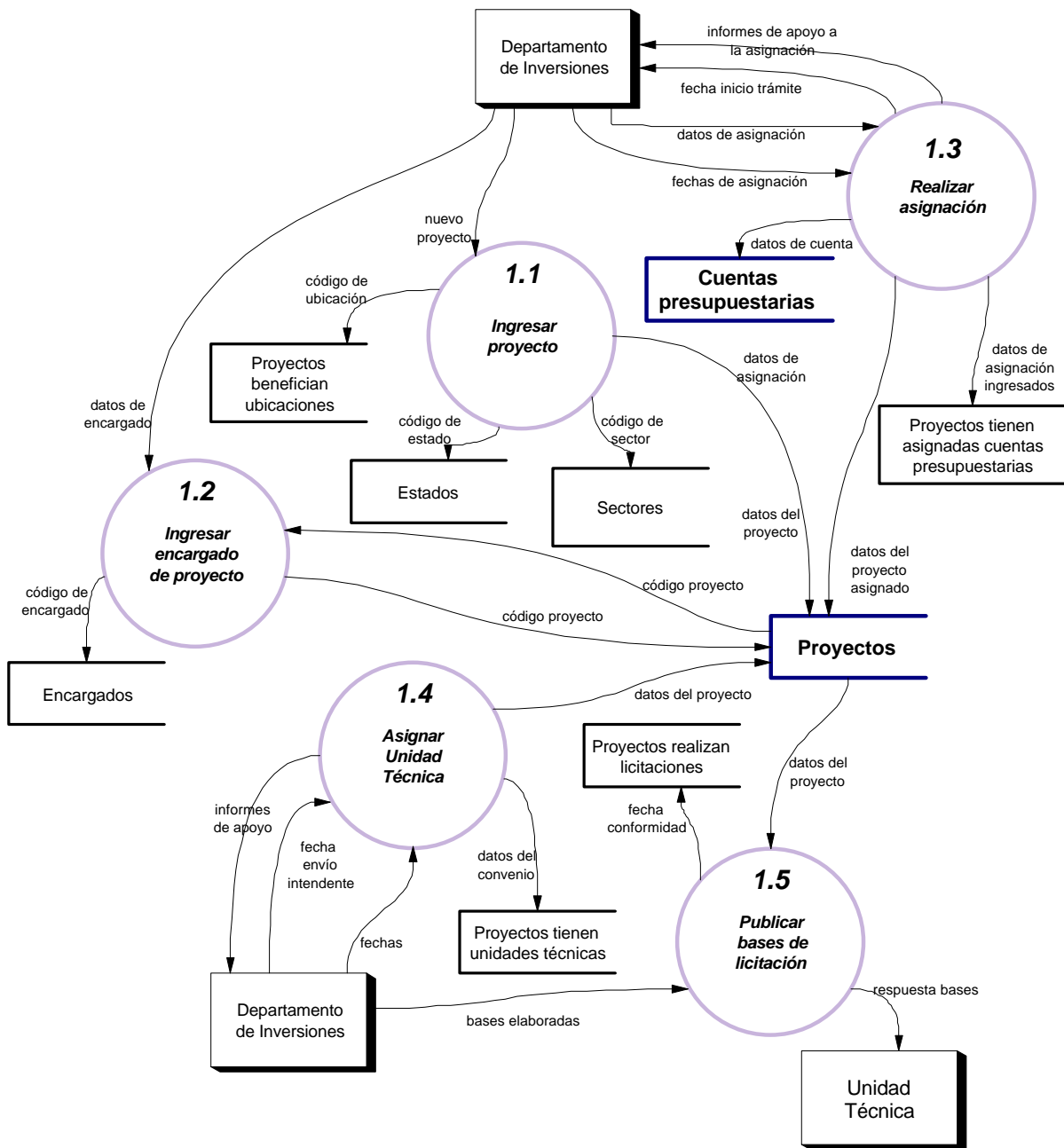


Figura A2.8: Diagrama uno.

La figura A2.9 muestra el diagrama dos, en el que se describe el proceso 2, al igual que en el diagrama 1, se describirá cada uno de los estos procesos con especificación de procesos de Pre y Post Condiciones, los flujos de datos y depósitos en el Diccionario de Datos.

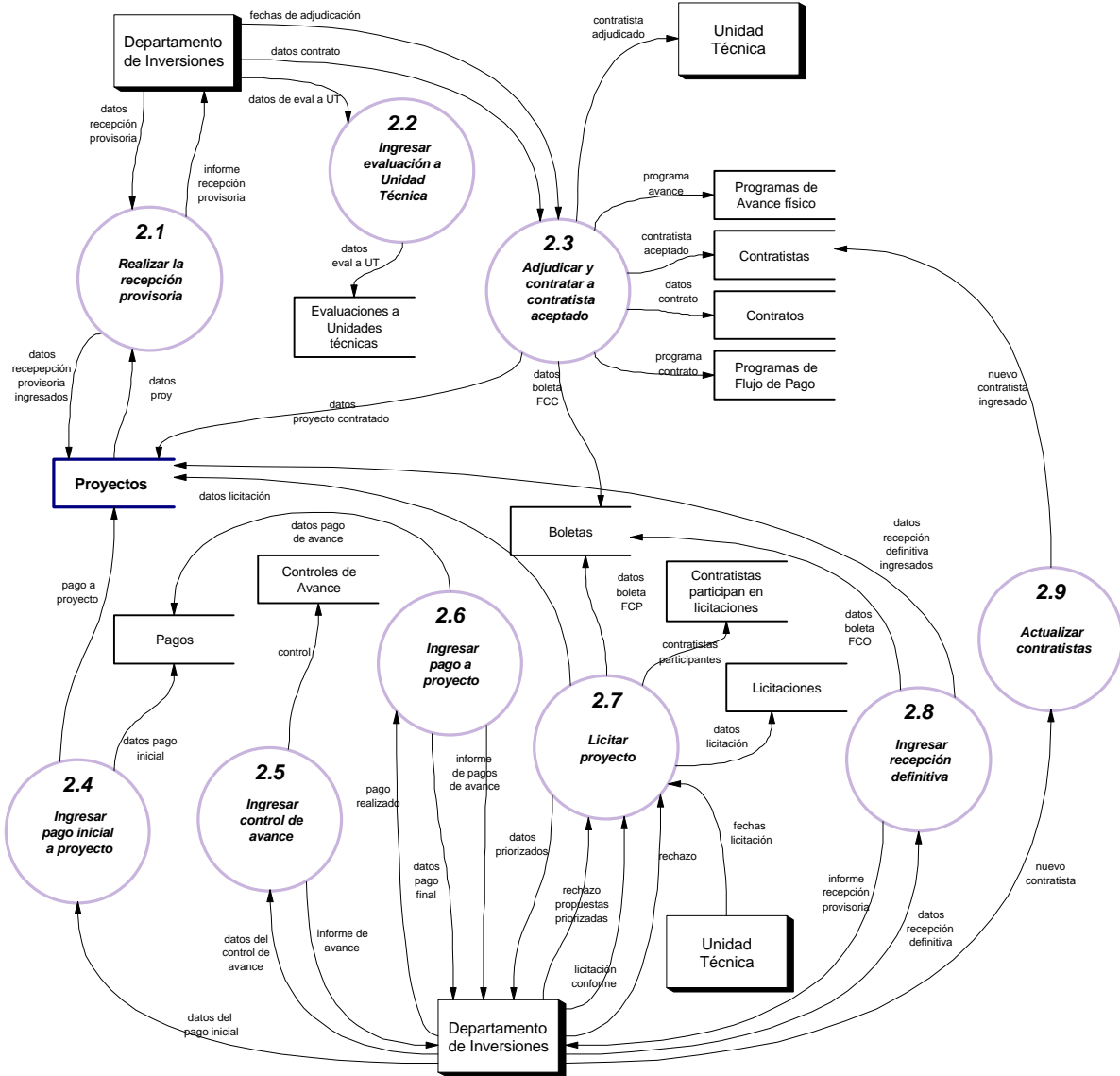


Figura A2.9: Diagrama dos.

La figura A2.10 muestra el diagrama tres, en el que se describe el proceso 3. Al igual que en los diagramas anteriores, se describirá cada uno de los estos procesos con especificación de procesos de Pre y Post Condiciones y los flujos de datos y depósitos, en el Diccionario de Datos.

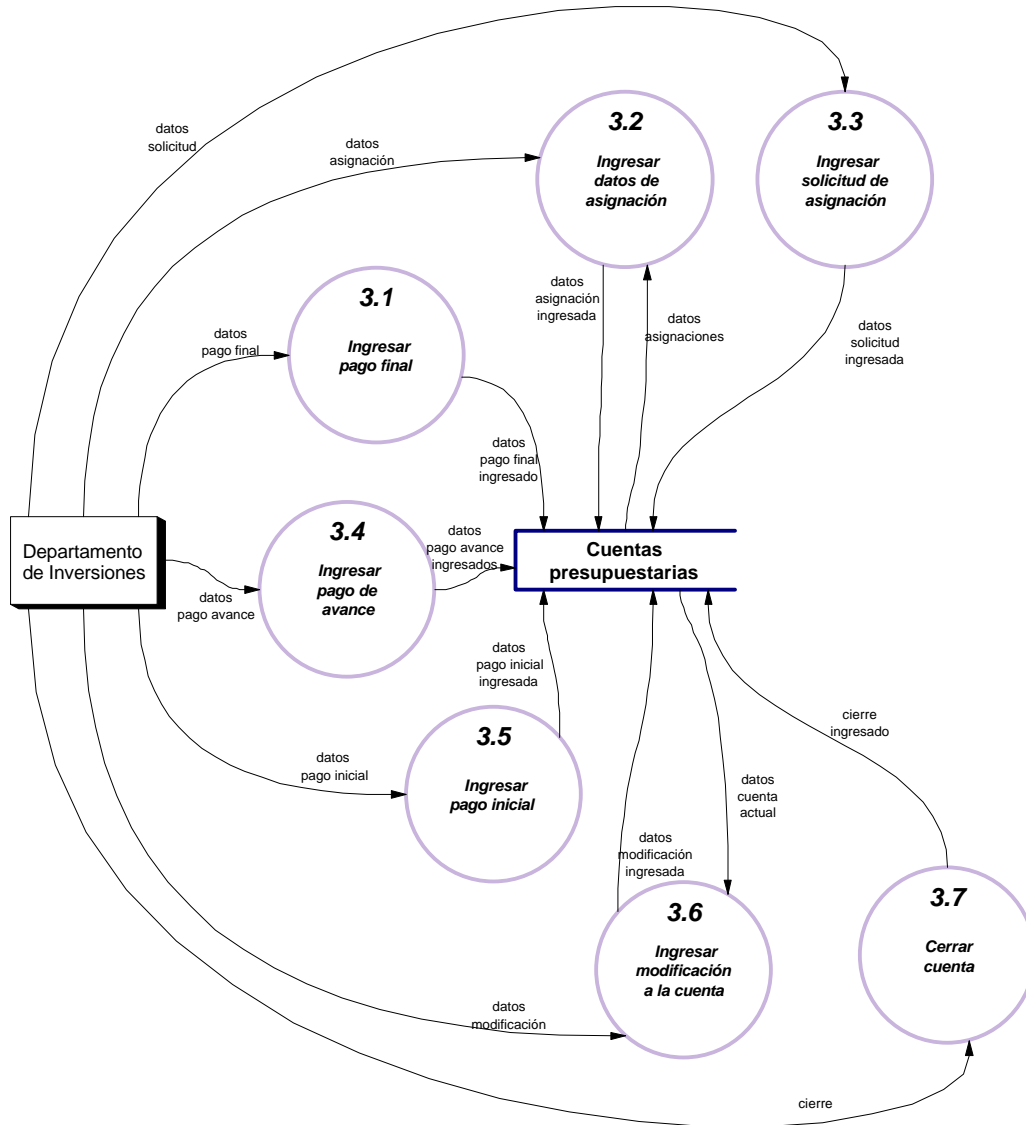


Figura A2.10: Diagrama tres.

La figura A2.11 muestra el diagrama cuatro, en el que se describe el proceso 4, los procesos de este diagrama también se describirán con especificación de procesos de Pre y Post Condiciones y los flujos de datos y los depósitos, en el Diccionario de Datos.

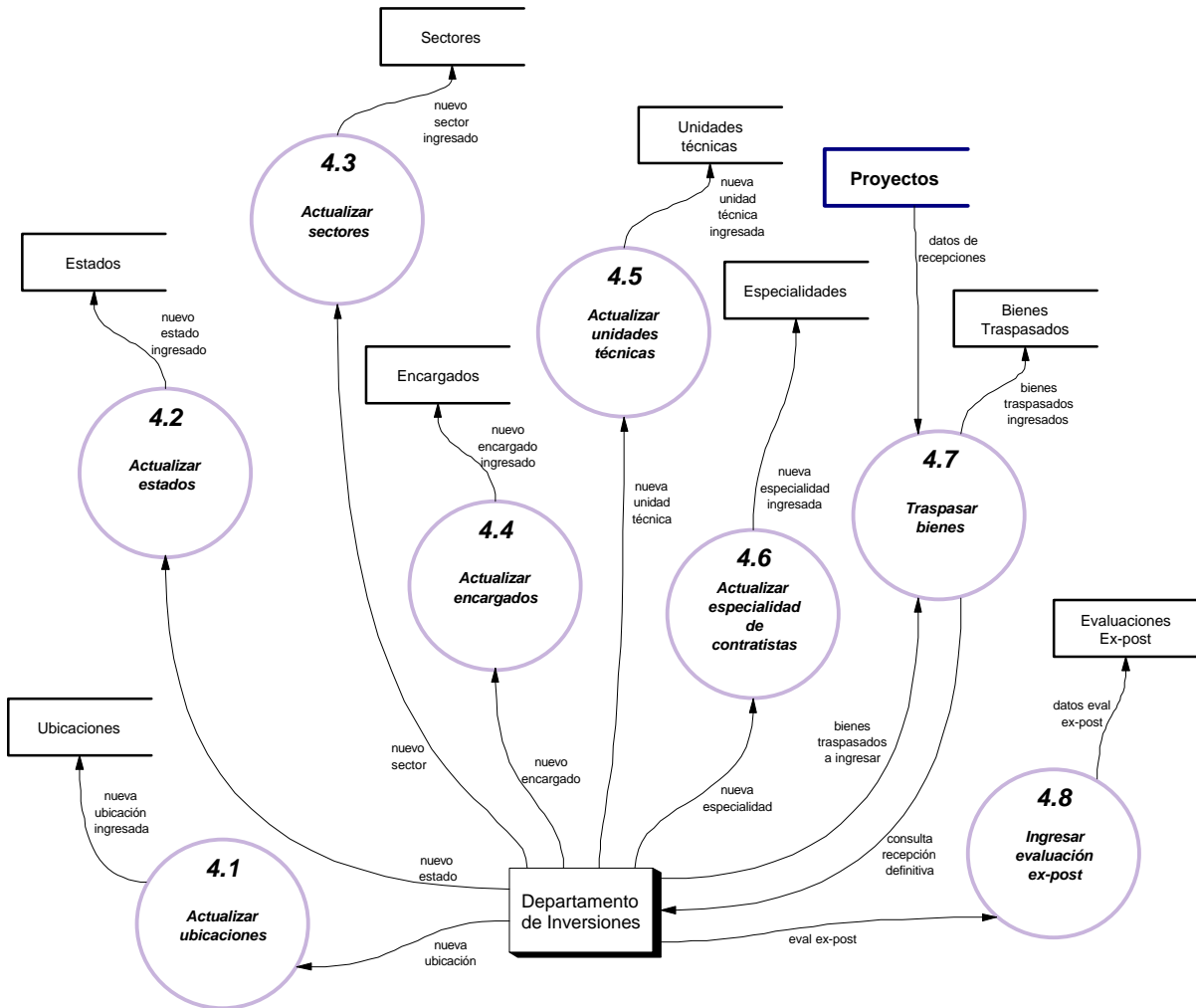


Figura A2.11: Diagrama cuatro.

3.2 Especificación de Procesos

Como se ha mencionado en la sección anterior, los procesos elementales serán descritos a través de especificación de Procesos del tipo Pre y Post Condiciones con las que se describe la función que realiza el proceso sin usar algoritmos, de esta forma se deja para la implementación la elección de algoritmos alternativos. A continuación se describe cada proceso, indicando nombre y su número.

Proceso 1

Proceso 1.1: Ingresar proyecto

PRE-CONDICIÓN

Existe un proyecto priorizado a ingresar al sistema

POST-CONDICIÓN

Proyecto ingresado al sistema

Proceso 1.2: Ingresar encargado de proyecto

PRE-CONDICIÓN

Existe un encargado asignado a un proyecto

POST-CONDICIÓN

Proyecto con encargado asignado

Proceso 1.3: Realizar asignación

PRE-CONDICIÓN

Existe un proyecto con encargado y cuenta presupuestaria creada para el proyecto

POST-CONDICIÓN

Proyecto con asignación

Proceso 1.4: Asignar Unidad Técnica

PRE-CONDICIÓN

Existe proyecto con asignación y Unidad Técnica capacitada para responsabilizarse de controlar su ejecución

POST-CONDICIÓN

Proyecto con Convenio Mandato

Proceso 1.5: Publicar bases de licitación

PRE-CONDICIÓN

Existe proyecto con Convenio Mandato y bases de licitación elaboradas

POST-CONDICIÓN

Unidad Técnica publicando bases

Proceso 2

Proceso 2.1: Realizar recepción provisoria

PRE-CONDICIÓN

Departamento de Inversiones aprueba recepción provisoria

POST-CONDICIÓN

Proyecto con recepción provisoria y Departamento de Inversiones analizando recepción definitiva

Proceso 2.2: Ingresar evaluación a Unidad Técnica

PRE-CONDICIÓN

Unidad Técnica evaluada

POST-CONDICIÓN

Proyecto con unidad técnica evaluada

Proceso 2.3: Adjudicar y contratar a contratista aceptado

PRE-CONDICIÓN

Proyecto adjudicado a contratista y contrato firmado

POST-CONDICIÓN

Proyecto con contrato

Proceso 2.4: Ingresar pago inicial a proyecto

PRE-CONDICIÓN

Pago inicial realizado y Proyecto con contrato

POST-CONDICIÓN

Proyecto en ejecución

Proceso 2.5: Ingresar control de avance

PRE-CONDICIÓN

Proyecto en ejecución y control de avance realizado

POST-CONDICIÓN

Control de avance ingresado

Proceso 2.6: Ingresar pago a proyecto

PRE-CONDICIÓN

Estado de pago aceptado y pago realizado

POST-CONDICIÓN

Pago de avance ingresado

Proceso 2.7: Licitar proyecto

PRE-CONDICIÓN 1

Recepción de propuestas priorizadas y licitación conforme

POST-CONDICIÓN 1

Proyecto licitado

PRE-CONDICIÓN 2

Recepción de propuestas priorizadas y licitación rechazada

POST-CONDICIÓN 2

Proyecto con licitación rechazada y Publicar bases de licitación

Proceso 2.8: Ingresar recepción definitiva

PRE-CONDICIÓN

Departamento de Inversiones aprueba recepción definitiva

POST-CONDICIÓN

Proyecto con recepción definitiva y Departamento de Inversiones traspasando bienes

Proceso 2.9: Actualizar contratistas

PRE-CONDICIÓN

Existe un contratista nuevo

POST-CONDICIÓN

Contratista ingresado al depósito CONTRATISTAS

Proceso 3**Proceso 3.1:** Ingresar pago final

PRE-CONDICIÓN

Cuenta con pagos de avance realizados y Pago final realizado

POST-CONDICIÓN

Cuenta con pagos de final realizado

Proceso 3.2: Ingresar datos de asignación

PRE-CONDICIÓN

Cuenta presupuestaria solicitada y Asignación aprobada

POST-CONDICIÓN

Cuenta presupuestaria creada

Proceso 3.3: Ingresar solicitud de asignación

PRE-CONDICIÓN

Solicitud de Asignación enviada a UCR

POST-CONDICIÓN

Cuenta presupuestaria solicitada

Proceso 3.4: Ingresar pago de avance

PRE-CONDICIÓN

Cuenta presupuestaria con primer pago realizado y Pago de avance realizado

POST-CONDICIÓN

Cuenta presupuestaria con pago de avance realizado

Proceso 3.5: Ingresar pago inicial

PRE-CONDICIÓN

Cuenta presupuestaria creada y Pago inicial realizado

POST-CONDICIÓN

Cuenta presupuestaria con primer pago realizado

Proceso 3.6: Ingresar modificación a cuenta

PRE-CONDICIÓN

Cuenta con pagos de avance realizados y Aceptación a modificación

POST-CONDICIÓN

Cuenta presupuestaria aumentada

Proceso 3.7: Cerrar cuenta

PRE-CONDICIÓN

Cuenta presupuestaria creada y cierre de presupuesto

POST-CONDICIÓN

Cuenta presupuestaria cerrada

Proceso 4**Proceso 4.1:** Actualizar ubicaciones

PRE-CONDICIÓN

Existe una nueva ubicación

POST-CONDICIÓN

Ubicación ingresada al depósito UBICACIONES

Proceso 4.2: Actualizar estados**PRE-CONDICIÓN**

Existe un nuevo estado

POST-CONDICIÓN

Estado ingresado al depósito ESTADOS

Proceso 4.3: Actualizar sectores**PRE-CONDICIÓN**

Existe un nuevo sector

POST-CONDICIÓN

Sector ingresado al depósito SECTORES

Proceso 4.4: Actualizar encargados**PRE-CONDICIÓN 1**

Existe un encargado adicional a los existentes

POST-CONDICIÓN 1

Encargado adicionado al depósito ENCARGADOS

PRE-CONDICIÓN 2

Existe un encargado que reemplaza a alguno de los existentes

POST-CONDICIÓN 2

Encargado adicionado al depósito ENCARGADOS y Encargado reemplazado en el depósito ENCARGADOS

Proceso 4.5: Actualizar unidades técnicas**PRE-CONDICIÓN 1**

Existe una Unidad Técnica adicional a los existentes

POST-CONDICIÓN 1

Unidad Técnica adicionada al depósito UNIDADES TECNICAS

PRE-CONDICIÓN 2

Existe una Unidad Técnica que reemplaza a alguna de los existentes

POST-CONDICIÓN 2

Unidad Técnica adicionada al depósito UNIDADES TECNICAS y Unidad Técnica reemplazada al depósito UNIDADES TECNICAS

Proceso 4.6: Actualizar especialidad de contratistas**PRE-CONDICIÓN**

Existe una nueva especialidad para los contratistas

POST-CONDICIÓN

Especialidad ingresada al depósito ESPECIALIDADES

Proceso 4.7: Traspasar bienes

PRE-CONDICIÓN

Proyecto con recepción definitiva aceptada y existen bienes a traspasar

POST-CONDICIÓN

Proyecto con traspaso de bienes

Proceso 4.8: Ingresar evaluación ex-post

PRE-CONDICIÓN

Existe proyecto finalizado y existe evaluación ex-post

POST-CONDICIÓN

Proyecto con evaluación ex-post

4 Modelamiento de la perspectiva estática

4.1 Diagrama Entidad Relacionamiento

Para describir el sistema desde esta perspectiva, se construye un Diagrama Entidad Relacionamiento. En la figura A2.12 se muestra el diagrama en que se modela una jerarquía de herencia de la entidad Proyecto y de esta manera se facilitan los relacionamiento con otras entidades. En este mismo diagrama o en otro se construye la jerarquía de herencia de la Cuenta presupuestaria y del mismo modo, se relaciona esta jerarquía con otras entidades.

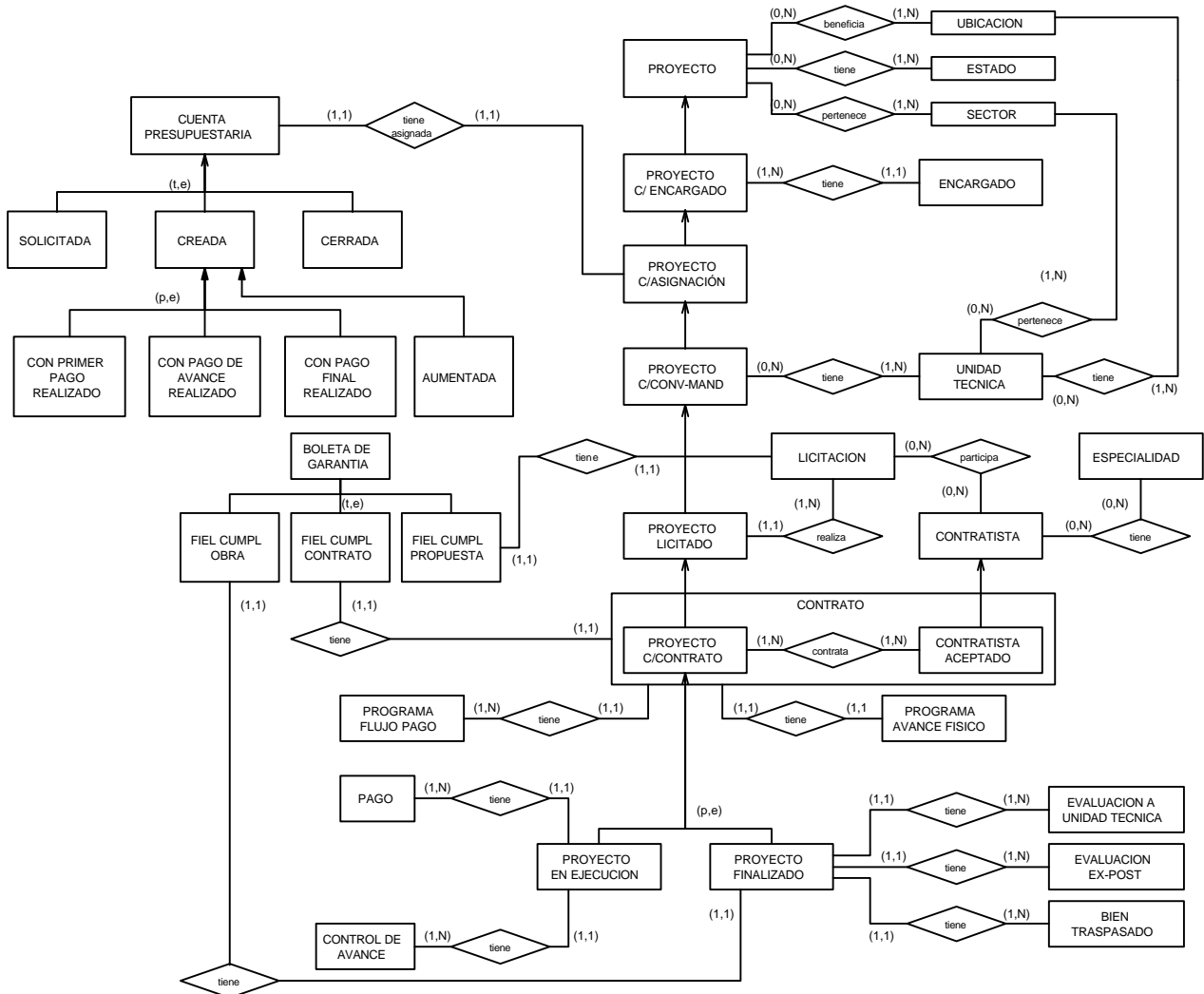


Figura A2.12: Diagrama Entidad Relacionamiento.

4.2 Diccionario de Datos

Otro modelo que se debe usar para describir esta perspectiva es el Diccionario de Datos, en el cual deben describirse las entidades, atributos, identificadores, flujos de datos y depósitos, el cual es presentado a continuación:

A

Actividad = Texto

Actividades a desarrollar = {Actividad}

Año = [1990..2999]

AUMENTADA = Monto + fecha

B

Banco = Texto

BIEN TRASPASADO = @Código de Traspaso + N° de oficio Traspaso + Fecha Traspaso + Detalle del traspaso + Encargado GORE.

bienes traspasados a ingresar= N° de oficio Traspaso + Fecha Traspaso + Detalle del traspaso + Encargado GORE.

Bienes Traspasados = {BIEN TRASPASADO}

bienes traspasados ingresados = N° de oficio Traspaso + Fecha Traspaso + Detalle del traspaso + Encargado GORE.

BOLETA GARANTIA = *superentidad* FIEL CUMPLIMIENTO DE CONTRATO + FIEL CUMPLIMIENTO DE PROPUESTA + FIEL CUMPLIMIENTO DE OBRA

Boletas = {BOLETA GARANTIA}

C

CERRADA = Monto total + fecha cierre

cierre = N° asignación + fecha cierre

cierre ingresado = N° asignación + fecha cierre

Código de Detalle = Número

Código de Encargado = Número

Código de Encargado asignado = {Código encargado}

Código de Especialidad = Número

Código de Estado = Número

Código de Evaluación = Número

Código de Sector = Número

Código de Traspaso = Número

Código de Ubicación = Número

Código Detalle = Número

Código Estado = Código Estado

Código proyecto = Número

Código sector = Código Sector

Comuna = Texto

CON PAGO DE AVANCE REALIZADO = {Monto} + {fecha}

CON PAGO FINAL REALIZADO = Monto + fecha

CON PRIMER PAGO REALIZADO = Monto + fecha

consulta recepción definitiva = Fecha recepción definitiva + Encargado + Observaciones + Código proyecto + N° Boleta Garantía + Fecha Emisión + Banco + Monto + Fecha Vencimiento

CONTRATISTA = @Rut contratista + Nombre + Dirección + Teléfono + Fax + E-mail

CONTRATISTA ACEPTADO = @Código de Proyectos en ejecución + @Rut Contratista

CONTRATISTA participa en LICITACION = Código proyecto + N° de Licitación + Fecha Recepción Propuestas en UT + Fecha Recepción Propuestas en GORE + N° de oficio Recepción Propuestas + Lugar [1° | 2° | 3°] + Rut + Monto + Boleta de Garantía FCP + Observaciones

Contratistas = {CONTRATISTA}

Contratistas participan en licitaciones = {CONTRATISTA participa en LICITACION}

Contratistas participantes = Código proyecto + Fecha Recepción Propuestas en UT + Fecha recepción Propuestas en GORE + N° de licitación + {Lugar} + {RUT} + {Monto} + {Observaciones} + {N° Boleta Garantía FCP}

CONTRATO = @Código proyecto + @Rut contratista + Fecha resolución de adjudicación + Fecha Contrato + Fecha Inicio obras + Tipo Contrato + Plazo estipulado + Fecha de Término + Tipo de Obra + Monto del Contrato + Retención + N° de oficio resolución de adjudicación.

Contratos = {CONTRATO}

control = Código proyecto + % avance + etapas terminadas + etapas en ejecución + Fecha Control de Avance + Observaciones + Monto Viático + Responsable avance + Foto digitalizada

CONTROL DE AVANCE = @N° Programa de Avance + Fecha Control de Avance + Etapas terminadas + Etapas en ejecución + Foto Digitalizada + Monto Viático + Responsable avance + Observaciones

Controles de Avance = {CONTROL DE AVANCE}

CREADA = *superentidad* CON PRIMER PAGO REALIZADO + CON PAGO DE AVANCE REALIZADO + CON PAGO FINAL REALIZADO + AUMENTADA

CUENTA PRESUPUESTARIA = *superentidad* SOLICITADA + CREADA + CERRADA

Cuentas Presupuestarias = {CUENTA PRESUPUESTARIA}

D

datos aceptación definitiva = Fecha recepción definitiva + Encargado + Observaciones + Código proyecto + N° Boleta Garantía + Fecha Emisión + Banco + Monto + Fecha Vencimiento

datos aceptación definitiva ingresados = Fecha recepción definitiva + Encargado + Observaciones + N° Boleta Garantía

datos asignación = Monto solicitado + Monto aprobado + Situación + Monto Gastos Administrativos + Monto Equipamiento + Monto Obra + Monto Terreno + Monto construcción + Otros montos + Fechas de asignación

datos asignación ingresada = Monto solicitado + Monto aprobado + Situación + Monto Gastos Administrativos + Monto Equipamiento + Monto Obra + Monto Terreno + Monto construcción + Otros montos + Fechas de asignación

datos asignaciones = {Monto solicitado + Monto aprobado + Situación + Monto Gastos Administrativos + Monto Equipamiento + Monto Obra + Monto Terreno + Monto construcción + Otros montos + Fechas de asignación}

datos boleta = N° Boleta Garantía + Fecha Emisión + Banco + Monto + Fecha Vencimiento

datos boleta FCC = datos boleta

datos boleta FCO = datos boleta

datos boleta FCP = datos boleta

datos contrato = Código proyecto + Rut contratista + Fecha Contrato + Fecha Inicio Obras + Tipo Contrato + Plazo estipulado + Fecha Término + Tipo de Obra + Detalle + N° Boleta Garantía Fiel Cumplimiento de Contrato + Fecha Emisión + Banco + Monto

datos cuenta actual = Monto aprobado + Monto gastos administrativos + Monto Equipamiento + Monto Obra + Monto Terreno + Monto Construcción + Otros

datos de aceptación provisoria = Fecha recepción provisoria + Encargado + Observaciones + Código proyecto

datos de aceptación provisoria ingresados = Fecha recepción provisoria + Encargado + Observaciones

datos de asignación = N° resolución + N° de Asignación + Monto solicitado + Monto aprobado + Situación + Monto Gastos Administrativos + Monto Equipamiento + Monto Obra + Monto Terreno + Monto construcción + Otros montos + Fechas de asignación

datos de asignación ingresados = N° resolución + N° de Asignación + Monto solicitado + Monto aprobado + Situación + Monto Gastos Administrativos + Monto Equipamiento + Monto Obra + Monto Terreno + Monto construcción + Otros montos + Fechas de asignación

datos de cuenta = N° resolución + N° de Asignación + Monto solicitado + Monto aprobado

datos de encargado = {Nombre + apellido}

datos de eval a UT = Código proyecto + Fecha Evaluación + Encargado + Detalle de la Evaluación a U.T.

datos del control de avance = Código proyecto + Fecha Control de Avance + Etapas terminadas + Etapas en ejecución + Observaciones + (Monto Viático) + Responsable Avance + Foto digitalizada

datos del convenio = N° de resolución + Código proyecto + Rut Unidad Técnica + (Nombre Unidad Técnica) + Monto gastos Administrativos + Monto Equipamiento + Monto Obra + Monto terreno + Monto Construcción + Otros montos

datos eval a UT Ingresados = Código proyecto + Fecha Evaluación + Encargado + Detalle de la Evaluación a U.T.

datos eval ex-post = Fecha Evaluación + Encargado + Detalle de la Evaluación Social

datos licitación = Código proyecto + Tipo de Licitación + Tipo propuesta + Precio de cada Base + N° de Bases Vendidas

datos modificación = Datos de la Asignación + Pagos del Convenio Mandato + Datos del Contrato + Programa de Pago

datos modificación ingresada = Datos de la Asignación + Pagos del Convenio Mandato + Datos del Contrato + Programa de Pago

datos pago de avance = {(N° Estado de Pago) + N° Egreso + Fecha de Egreso + N° Pago en la secuencia + Tipo pago + Monto inicial + (Retención anticipo) + (Retención contrato) +

(Resolución contrato) + (Devolución contrato) + Neto + IVA + (Multas) + Líquido a cancelar + Saldo contrato}

datos pago de avance ingresados = {(N° Estado de Pago) + N° Egreso + Fecha de Egreso + N° Pago en la secuencia + Tipo pago + Monto inicial + (Retención anticipo) + (Retención contrato) + (Resolución contrato) + (Devolución contrato) + Neto + IVA + (Multas) + Líquido a cancelar + Saldo contrato}

datos pago final = {(N° Estado de Pago) + N° Egreso + Fecha de Egreso + N° Pago en la secuencia + Tipo pago + Monto inicial + (Retención anticipo) + (Retención contrato) + (Resolución contrato) + (Devolución contrato) + Neto + IVA + (Multas) + Líquido a cancelar + Saldo contrato}

datos pago final ingresado = {(N° Estado de Pago) + N° Egreso + Fecha de Egreso + N° Pago en la secuencia + Tipo pago + Monto inicial + (Retención anticipo) + (Retención contrato) + (Resolución contrato) + (Devolución contrato) + Neto + IVA + (Multas) + Líquido a cancelar + Saldo contrato}

datos pago inicial = {(N° Estado de Pago) + N° Egreso + Fecha de Egreso + N° Pago en la secuencia + Tipo pago + Monto inicial + (Retención anticipo) + (Retención contrato) + (Resolución contrato) + (Devolución contrato) + Neto + IVA + (Multas) + Líquido a cancelar + Saldo contrato}

datos pago inicial ingresado = {(N° Estado de Pago) + N° Egreso + Fecha de Egreso + N° Pago en la secuencia + Tipo pago + Monto inicial + (Retención anticipo) + (Retención contrato) + (Resolución contrato) + (Devolución contrato) + Neto + IVA + (Multas) + Líquido a cancelar + Saldo contrato}

datos recepciones = Fecha recepción definitiva + Encargado + Observaciones + Código proyecto + N° Boleta Garantía + Fecha Emisión + Banco + Monto + Fecha Vencimiento + Fecha recepción provisoria + Encargado + Observaciones + Código proyecto

datos solicitud = Fecha de envío a UCR + monto

datos solicitud ingresada = Fecha de envío a UCR + monto

Descripción de actividades a realizar = Texto

Descriptor = Texto

Detalle del Traspaso = Texto

Detalle Evaluación de U.T. = Texto

Detalle Evaluación social = Texto

Devolución Contrato = Número

Día = [1..31]

Dígito = [1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0]

Dirección = Texto

Duración = Número

E

E-mail = Texto

ENCARGADO = @Código de Encargado + Nombre + Primer Apellido + Segundo Apellido

Encargado GORE = Texto

Encargado RD = Texto

Encargado RP = Texto

Encargados = {ENCARGADO}

ESPECIALIDAD = @Código de Especialidad + Nombre de Especialidad

Especialidades = {ESPECIALIDAD}

Estado (flujo) = Nombre Estado

ESTADO = @Código de Estado + Nombre Estado

Estados = {ESTADO}

Etapas en ejecución = Actividad

Etapas terminadas = Actividad

eval ex-post = Código proyecto + Fecha Evaluación + Encargado + Detalle de la Evaluación Social

EVALUACION A UNIDAD TECNICA = @Código de Evaluación + Fecha Evaluación + Encargado + Detalle de la evaluación de U.T.

EVALUACION EX POST = @Código de Evaluación + Fecha Evaluación + Encargado + Detalle de la evaluación social + Detalle de la evaluación de U.T.

Evaluaciones a Unidades Técnicas = {EVALUACION A UNIDAD TECNICA}

Evaluaciones Ex-Post = {EVALUACION EX POST}

F

Fax = Texto

Fecha elaboración solicitud de asignación = Fecha

Fecha Rechazo a licitación = Fecha

Fecha = día + mes + año

fecha conformidad = (Fecha Recepción Bases) + (Fecha Conformidad Bases) + (Fecha Publicación) + (N° de licitación)

Fecha Conformidad = Fecha

Fecha Conformidad Bases = Fecha

Fecha Contrato = Fecha

Fecha Control de Avance = Fecha

Fecha de adjudicación = Fecha resolución de adjudicación + Código Proyecto + Rut Contratista

Fecha de envío a la UCR para visado = Fecha

Fecha de envío toma de razón = Fecha

Fecha de recepción toma de razón = Fecha

Fecha de Término = Fecha

Fecha Egreso = Fecha

Fecha Emisión = Fecha

Fecha entrega mandato a UT = Fecha

Fecha Evaluación = Fecha

Fecha firma Intendente = Fecha

Fecha ingreso al S.N.I = Fecha

Fecha inicio = Fecha

Fecha Inicio obras = Fecha

Fecha Mandato firmado por Intendente = Fecha

Fecha Priorización = Fecha

Fecha Publicación = Fecha

Fecha Real de Término = Fecha

Fecha recepción asignación aprobada = Fecha

Fecha Recepción Bases = Fecha

Fecha recepción de mandato firmado por la UT = Fecha

Fecha Recepción Definitiva = Fecha

Fecha Recepción Propuestas en GORE = Fecha

Fecha Recepción Propuestas en UT = Fecha

Fecha Recepción Provisoria = Fecha

Fecha Rechazo a licitación = Fecha

Fecha Rechazo a propuestas = Fecha

Fecha resolución de adjudicación = Fecha

Fecha término = Fecha

Fecha Traspaso = Fecha

Fecha Vencimiento = Fecha

Fecha visado en UCR = Fecha

fechas = Convenio Mandato + (Fecha entrega mandato a UT) + (Fecha recepción de mandato firmado por la UT) + (Fecha Mandato firmado por Intendente)

fechas de adjudicación = Fecha resolución de adjudicación + Código Proyecto + Rut Contratista

fechas de asignación = (Fecha elaboración solicitud de asignación) + (Fecha de envío a la UCR para visado) + (Fecha visado en UCR) + (Fecha firma Intendente) + (Fecha recepción asignación aprobada)

fechas licitación = (Fecha Recepción Bases) + (Fecha Conformidad Bases) + (Fecha Publicación) + (N° de licitación)

FIEL CUMPLIMIENTO DE CONTRATO = @N° Boleta de Garantía + Fecha Emisión + Banco + Monto + Fecha Vencimiento

FIEL CUMPLIMIENTO DE LA OBRA = @N° Boleta de Garantía + Fecha Emisión + Banco + Monto + Fecha Vencimiento

FIEL CUMPLIMIENTO DE PROPUESTA = @N° Boleta de Garantía + Fecha Emisión + Banco + Monto + Fecha Vencimiento

Foto Digitalizada = *algún formato de fotos como gif ó jpg*

G

H

I

informe de avance = Suma pagos todos los proyectos + Todos los pagos presupuestados + Porcentaje de avance + programa inicial de avance + Costo del proyecto + Programa de pago

informe de pagos de avance = Código Proyecto + N° Programa de Avance + N° Programa Pago

informes de apoyo = {Rut + Nombre + Dirección + Ubicación + (Especialidad) + Teléfono + Responsable + N° de Proyectos en Ejecución + Monto proyectos en Ejecución + Observaciones}

informes de apoyo a la asignación = Código proyecto + nombre proyecto + justificación + tiempo aproximado de ejecución + (Van privado) + (Van social) + (Tir privada) + (Tir social) + Sector + Estado + Ubicación + Prioridad + Tipo financiamiento) + pago presupuestado

Instituciones que participan = Texto

IVA = Número

J

Justificación = Texto

K

L

LICITACION = @Código Proyecto + @N° de Licitación + Tipo de licitación + Tipo propuesta + Precio de cada Base + N° Bases vendidas + Fecha Conformidad + Fecha Rechazo + Razón Rechazo + N° de oficio Conformidad a licitación + N° de oficio Rechazo a licitación + N° de oficio Rechazo a propuestas

Licitaciones = {LICITACION}

Líquido a cancelar = Número

Localización = Texto

Lugar = [1° | 2° | 3° | 4° | 5°]

M

Mes = [Enero, Febrero,..., Diciembre]

Monto = Número

Monto aprobado = Número

Monto Construcción = Número

Monto del Contrato = Número

Monto diseño = Número

Monto Equipamiento = Número

Monto gastos administrativos = Número

Monto inicial = Número

Monto Obra = Número

Monto presupuestado = Número

Monto solicitado = Número

Monto Terreno = Número

Monto total gastado = Número

Monto Viático = Número

Multas = Número

N

N° Bases vendidas = Número

N° Boleta de Garantía = Número
N° Control de Avance = Número
N° de Asignación = Número
N° de Egreso = Código proyecto + {N° Egreso}
N° de Licitación = Número
N° de Licitaciones a realizar = Número
N° de oficio = Número
N° de oficio Conformidad a licitación = Número
N° de oficio Conformidad Bases = Número
N° de oficio entrega mandato a UT = Número
N° de oficio firma de intendente = Número
N° de oficio mandato firmado por Intendente = Número
N° de oficio Recepción Bases = Número
N° de oficio recepción de asignación de UCR = Número
N° de oficio recepción de mandato firmado por la UT = Número
N° de oficio Recepción Definitiva = Número
N° de oficio recepción Estado de Pago = Número
N° de oficio Recepción Propuestas = Número
N° de oficio Recepción Provisoria = Número
N° de oficio Rechazo a licitación = Número
N° de oficio Rechazo a propuestas = Número
N° de oficio resolución de adjudicación = Número
N° de oficio solicitud de asignación a UCR = Número
N° de Resolución = Número
N° Egreso = Número
N° Estado de Pago = Número
N° pago en la secuencia = Número
N° pagos = N° Egreso + Código proyecto
N° Programa = N° Programa de Avance + N° Control de Avance
N° Programa de Avance = Número
N° Programa de Pago = Número
N° Programa Flujo Pago = N° Programa de Pago
Neto = Número
Nombre = Texto
Nombre de Especialidad = Texto
Nombre detalle = Texto
Nombre Estado = Texto
Nombre Proyecto = Texto
Nombre Sector = Texto
nueva especialidad = Especialidad
nueva especialidad ingresada = Especialidad
nueva ubicación = (Región) + Provincia + Comuna
nueva ubicación ingresada = (Región) + Provincia + Comuna
nueva unidad técnica = {Rut + Nombre + Dirección + Ubicación + (Especialidad) + Teléfono + Responsable + N° de Proyectos en Ejecución + Monto proyectos en Ejecución + Observaciones}

nueva unidad técnica ingresada = {Rut + Nombre + Dirección + Ubicación + (Especialidad) + Teléfono + Responsable + N° de Proyectos en Ejecución + Monto proyectos en Ejecución + Observaciones}

nuevo encargado = Nombre + Apellido

nuevo encargado ingresado = Nombre + Apellido

nuevo estado = Estado

nuevo estado ingresado = Estado

nuevo proyecto = {Código proyecto + nombre proyecto + Tipología + descriptor + Instituciones que participan + descripción de actividades a realizar + justificación + (Van privado) + (Van social) + (Tir privada) + (Tir social) + Sector + Estado + Ubicación + Prioridad + Tipo financiamiento + Tiempo aproximado de término + Pago presupuestado + fecha ingreso al S.N.I. + Fecha Priorización + Resultado análisis técnico

nuevo sector = Nombre Sector

nuevo sector ingresado = Nombre Sector

nuevos contratista = (Rut Contratista) + (Nombre) + (Dirección) + (Teléfono) + (Observaciones)

nuevos contratista ingresado = (Rut Contratista) + (Nombre) + (Dirección) + (Teléfono) + (Observaciones)

Número = {Dígito}

Ñ

O

Observaciones = Texto

Observaciones del Resultado = Texto

Otros montos = Número

P

PAGO = @N° Egreso + N° Estado de Pago + Fecha Egreso + N° de pago en la secuencia + Tipo de Pago + Monto inicial + Retención Anticipo + Retención Contrato + Devolución Contrato + Neto + IVA + Multas + Líquido a cancelar + Saldo contrato + N° de oficio recepción Estado de Pago

pago realizado = (N° Estado de Pago) + N° de Egreso + Fecha de Egreso + N° Pago en la secuencia + Tipo pago + Monto inicial + (Retención anticipo) + (Retención contrato) + (Resolución contrato) + (Devolución contrato) + Neto + IVA + (Multas) + Líquido a cancelar + Saldo contrato

Pagos = {PAGO}

Plazo estipulado = Número

Precio de cada Base = Número

Predecesoras = Actividad

Primer Apellido = Texto

Prioridad = Número

PROGRAMA FLUJO DE PAGO = @N° de Programa de Pago + N° pago en la secuencia + Fecha + Monto + Saldo

programa avance = Código Proyecto + N° Programa de Avance + Rut contratista + {Actividades a Desarrollar} + {Predecesoras} + {Duración} + {Fecha inicio} + {Fecha Término}

PROGRAMA AVANCE FISICO = @N° Programa de Avance + N° Control de Avance + Actividades a desarrollar + Predecesoras + Duración + Fecha inicio + Fecha término

programa contrato = N° Programa de Pago + N° Pago en la secuencia + Monto + Fecha

Programas de Avance Físico = {PROGRAMA DE AVANCE FISICO}

Programas de Flujo de pago = {PROGRAMA DE FLUJO DE PAGO}

Proyectos realizan licitaciones = {PROYECTO LICITADO realiza LICITACION}

Provincia = Texto

PROYECTO = *superentidad* @Código proyecto + Nombre Proyecto + Tipología + Descriptor + Instituciones que participan + Descripción de actividades a realizar + Justificación + Fecha ingreso al S.N.I + Fecha Priorización + Resultado Análisis técnico económico + Observaciones del Resultado + VAN Privado + VAN Social + TIR privada + TIR social + Prioridad + Tipo de Financiamiento + Tiempo aproximado de término + Monto presupuestado + PROYECTO CON ENCARGADO + PROYECTO CON ASIGNACIÓN + PROYECTO CON CONV MANDATO + PROYECTO LICITADO + PROYECTO CON CONTRATO + PROYECTO EN EJECUCIÓN + PROYECTO FINALIZADO.

PROYECTO beneficia UBICACIÓN = Localización

PROYECTO CON ASIGNACION = @Código Proyecto

PROYECTO CON ASIGNACION tiene asignada CUENTA PRESUPUESTARIA = @N° de Asignación + N° de Resolución + Monto solicitado + Monto aprobado + Monto gastos administrativos + Monto Equipamiento + Monto Obra + Monto Terreno + Monto Construcción + Otros montos + Situación + Fecha elaboración solicitud de asignación + Fecha de envío a la UCR para visado + Fecha visado en UCR + Fecha firma Intendente + Fecha recepción asignación aprobada + N° de oficio solicitud de asignación a UCR + N° de oficio recepción de asignación de UCR + N° de oficio firma de intendente.

PROYECTO CON CONTRATO = @Código proyecto + @Rut contratista

PROYECTO CON CONV MANDATO = @Código Proyecto

PROYECTO CON CONV MANDATO tiene UNIDAD TECNICA = @N° de Resolución + Código proyecto + Rut Unidad Técnica + Fecha entrega mandato a UT + Fecha recepción de mandato firmado por la UT + Fecha Mandato firmado por Intendente + Monto gastos administrativos + Monto Equipamiento + Monto Obra + Monto Terreno + Monto Construcción + Monto Diseño + Otros monto + N° de oficio entrega mandato a UT + N° de oficio recepción de mandato firmado por la UT + N° de oficio mandato firmado por Intendente.

PROYECTO CON ENCARGADO = @Código Proyecto

PROYECTO CON ENCARGADO tiene ENCARGADO = Código proyecto + Código de Encargado

PROYECTO EN EJECUCION = @Código proyecto

PROYECTO EN EJECUCION tiene CONTROL DE AVANCE = Código proyecto + N° Control de Avance

PROYECTO EN EJECUCION tiene PAGO = Código proyecto + N° Egreso

PROYECTO FINALIZADO = @Código proyecto + Rut Contratista + Rut Unidad Técnica + Fecha Recepción Provisoria + EncargadoRP + Observaciones + Fecha Recepción Definitiva + Encargado RD+ Observaciones + Fecha Traspaso de Dominio + N° de oficio Recepción Provisoria + N° de oficio Recepción Definitiva

PROYECTO FINALIZADO tiene BOLETA GARANTIA FCO = Código proyecto + N° Boleta de Garantía

PROYECTO LICITADO = @Código Proyecto + N° de Licitaciones a realizar

PROYECTO LICITADO realiza LICITACION = Código Proyecto + N° de Licitación + Fecha Recepción Bases + N° de oficio Recepción Bases + Fecha Conformidad Bases + N° de oficio Conformidad Bases + Fecha Publicación.

Proyecto tienen unidades técnicas = {PROYECTO CON CONV MANDATO tiene UNIDAD TECNICA }

Proyectos = {PROYECTO }

Proyectos benefician ubicaciones = {PROYECTO beneficia UBICACIÓN }

Proyectos contratan contratistas aceptados = {PROYECTO contrata CONTRATISTA ACEPTADO }

Proyectos pertenecen a sectores = {PROYECTO pertenece a SECTOR }

Proyectos tienen asignadas cuentas presupuestarias = {PROYECTO CON ASIGNACION tiene asignada CUENTA PRESUPUESTARIA }

Proyectos tienen encargados = {PROYECTO CON ENCARGADO tiene ENCARGADO }

Proyectos tienen estados = {PROYECTO tiene ESTADO }

Q

R

Razón Rechazo = Texto

Región = [V Región]

Responsable = Texto

Responsable avance = Texto

Respuesta bases = [Publicar | Reconfecionar]

Resultado Análisis técnico económico = [RS | FI]

Retención = Número

Retención Anticipo = Número

Retención Contrato = Número

Rut = Texto

Rut contratista = Texto

Rut Unidad Técnica = Texto

S

Saldo = Número

Saldo contrato = Número

SECTOR = @Código de Sector + Nombre Sector

Sectores = {SECTOR }

Segundo Apellido = Texto
Situación = [Creación | Aumento | Disminución]
SOLICITADA = Monto
Suma pagos todos los proyectos = {PAGOS}

T

Teléfono = Texto
Tiempo aproximado de término = Número
Tipo Contrato = *tipos de contrato que se puedan realizar*
Tipo de financiamiento = [Tradicional | BID | PER | FIE]
Tipo de licitación = [Nacional | Internacional]
Tipo de Obra = Texto
Tipo de Pago = [Inicial | Avance | Final]
Tipo propuesta = [Pública | Privada | Dirigida]
Tipología = [53 | 65 | 77]
TIR privado = Número
TIR social = Número
Todos los pagos presupuestados = {FLUJO DE PAGO}
Toma de razón = [exento | afecto]

U

UBICACION = @Código de Ubicación + Región + Provincia + Comuna
Ubicaciones = {UBICACIÓN}
UNIDAD TECNICA = @Rut de Unidad Técnica + Nombre + Dirección + Teléfono + Fax +
 E-mail + Responsable + Observaciones
Unidades Técnicas = {UNIDAD TECNICA}

V

VAN privado = Número
VAN social = Número

W**X****Y****Z**

Bibliografía

[Acevedo90] Acevedo, Héctor. **El análisis estructurado de sistemas y el desarrollo de proyectos informáticos**. Valparaíso, Chile: Ecogestión, 1990.

[Aktas87] Aktas, A. Ziya. **Structured Analysis & Design of Information Systems**. USA: Prentice Hall, 1987.

[Allen&deChampeaux95] Extending the Statechart Formalism: Event Scheduling & Disposition. **ACM SIGPLAN Notices**, New York, v.30, n.10, Octubre 1995.

[Batini94] Batini, Carlo. **Diseño Conceptual de Bases de Datos: Un Enfoque de Entidades y Relaciones**. Madrid: Addison-Wesley/Diaz de Santos, 1994.

[Bustos99] Bustos, Guillermo. Profesor. **Modelamiento de Sistemas de Información**. Valparaíso: UCV, 1999. Diapositivas.

[Bustos99a] Bustos, Guillermo. Modelado Orientado a Objetos: Una Evaluación Crítica. **Revista de la Facultad de Ingeniería de la Universidad Católica de Valparaíso**, Valparaíso, vol. 10, Octubre 1999.

[Davis93] Davis, Alan. **Software Requirements: Objects, Functions, & States**. Englewood Cliffs: Prentice Hall, 1993.

[deChampeaux94] de Champeaux, Dennis. **Object-Oriented System Development**. USA: Addison-Wesley, 1994.

[Embley92] Embley, David. **Object-Oriented System Analysis**. A Model-Driven Approach. Englewood Cliffs, USA: Prentice Hall, 1992.

[Feldman&Miller89] Feldman, P. & Miller, D. Entity Model Clustering: Structuring A Data Model by Abstraction. **The Computer Journal**, Cambridge, v. 29, n. 4, Agosto 1986.

[Fowler&Scott97] Fower, Martin & Scott, Kendall. **UML Distilled**. Applying The Standard Object Modeling Language. USA: Addison-Wesley, 1997.

[Harel87] Harel, D. Statecharts: A visual Formalism for Complex Systems. **Science of Computer Programming**, v 8, n 3, p. 231-274, June 1987, Amsterdam.

[Heuser90] Heuser, Carlos. **Modelagem Conceitual de Sistemas: Redes de Petri**. Buenos Aires: Kapelusz, 1990.

[Høydalsvik&Sindre93] Høydalsvik, Geir & Sindre, Guttorm. On the Purpose of Object-Oriented Analysis. **ACM SIGPLAN Notices**, New York, v.28, n.10, Octubre 1993

[Jacobson95] Jacobson, Ivar. **The Object Advantage: Business Process Reengineering with Object Technology**. Wokingham: ACM Press, 1995.

[Odell98] Odell J. **Advanced Object-Oriented Analysis & Design Using UML**. New York: SIGS Books, 1998.

[Page-Jones80] Page-Jones, Meilir. **The Practical Guide to Structured Systems Design**. New York: Yourdon Press, 1980.

[Pressman98] Pressman, Robert. **Ingeniería del Software: Un enfoque práctico**. 4^{ta} Edición, España: McGraw Hill / Interamericana de España, 1998.

[Resende&Araújo00] Resende, R. & Araújo, B. **Consistência de Diagramas UML**. III Jornadas Iberoamericanas de Ingeniería de Requisitos y Ambientes de Software - Ideas 2000. Centro Nacional de Investigación y Desarrollo Tecnológicos – CENIDET. Cancún, México, Marzo 2000.

[Stevens88] Stevens, Wayne. **Projeto Estruturado de Sistemas**. 3^a Edición, Rio de Janeiro: Editora Camus, 1988.

[Teorey89] Teorey, Toby; Wei, Guangping; Bolton, Deborah; Koenig, John. ER Model Clustering as an Aid for User Communication and Documentation in Database Design. **Communication of the ACM**, v. 32, n. 8, Agosto 1989.

[Yourdon95] Yourdon, Edward. **Mainstream objects: An Analysis and Design Approach for Business**. New York: Prentice Hall, 1995.

[Yourdon94] Yourdon, Edward. **Análisis Estructurado Moderno**. México: Prentice Hall, 1994.

[Yourdon&Constantine78] Yourdon, Edward & Constantine, Larry. **Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design**. 2^a Edición, New York: Yourdon Press, 1978.