

METODOLOGÍA DE DESARROLLO DE SISTEMAS I

UNIDAD 3 NOCIONES GENERALES DEL MODELO ESTRUCTURADO

Autor de contenidos:
Carlos Neil



PRESENTACIÓN

Las especificaciones funcionales de los sistemas de información expresan las necesidades de información de los usuarios que utilizan el sistema para la toma de decisiones. Estas especificaciones se deben traducir en términos convertibles a las soluciones virtuales que se desarrollan al elaborar un sistema informático.

Se deberá realizar un salto conceptual y formal desde la ambigüedad de los lenguajes utilizados para expresar los problemas de decisión (v.g. lenguaje escrito) a términos carentes de ambigüedad, factibles de ser tratados automáticamente.

Nos introduciremos en este Módulo en las herramientas del análisis estructurado que facilitan esta conversión, este pasaje de expresiones redundantes a expresiones formales con redundancia mínima. A lo largo de esta unidad recorrerá las diferentes técnicas de modelado utilizando para ello diversos ejemplos de sistemas.

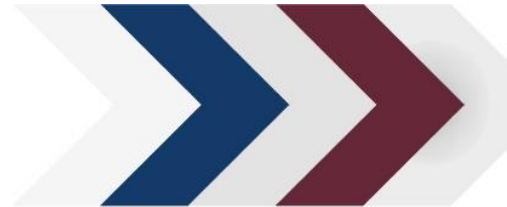
Le sugerimos entonces que preste atención a las recomendaciones de su tutor/a y a las fechas de estipuladas en el cronograma de actividades.

A través del estudio de esta unidad esperamos que usted, como alumno de esta asignatura, adquiera capacidad para:

- Comprender los componentes del diagrama de flujo de datos y aplicarlos en el modelado funcional de sistemas.
- Conocer y aplicar la simbología básica para documentar sistemas de información.
- Ejercitar habilidades para modificar conceptual y lógicamente la base de datos a partir de nuevos requerimientos.
- Corroborar la versatilidad del modelo para ser modificado fácilmente.
- Aplicar las herramientas para especificar procesos estructurados.
- Decodificar y codificar utilizando la simbología específica y aplicar las herramientas de modelado de datos de sistemas de información.
- Relacionar y articular las distintas visiones del sistema representadas por los diferentes modelos del análisis estructurado.

A continuación, le presentamos un detalle de los contenidos y actividades que integran esta unidad. Usted deberá ir avanzando en el estudio y profundización de los diferentes temas, realizando las lecturas requeridas y elaborando las actividades propuestas, algunas de desarrollo individual y otras para resolver en colaboración con otros estudiantes y con su profesor tutor.





REQUERIMIENTOS Y DIAGRAMA DE FLUJO DE DATOS

En esta unidad la propuesta será abocarnos al estudio de tres herramientas que se utilizan para modelar los datos: **diagrama de flujo de datos, diccionario de datos y especificación de procesos.**

Para abordar cada una de ellas, la propuesta será que lean los contenidos de cada uno de los temas y realice luego la guía de aprendizaje correspondiente.

¡Comencemos!

DIAGRAMA DE FLUJO DE DATOS

Todos los métodos de análisis de los requisitos de información se basan en la construcción de un conjunto de modelos del sistema que se pretende desarrollar. Utilizando alguna notación, propia de cada método, se crean modelos que reflejen el sistema y aplicando las técnicas de descomposición mediante un proceso de construcción descendente o ascendente, se establece la esencia del sistema a desarrollar.

La construcción de modelos presenta ventajas claras, tales como centrarse en determinadas características del sistema, dejando de lado otras menos significativas. Esto, además, permite enfocar las discusiones con el usuario en los aspectos más importantes del sistema, sin distracciones en características que sean, por el momento, irrelevantes.

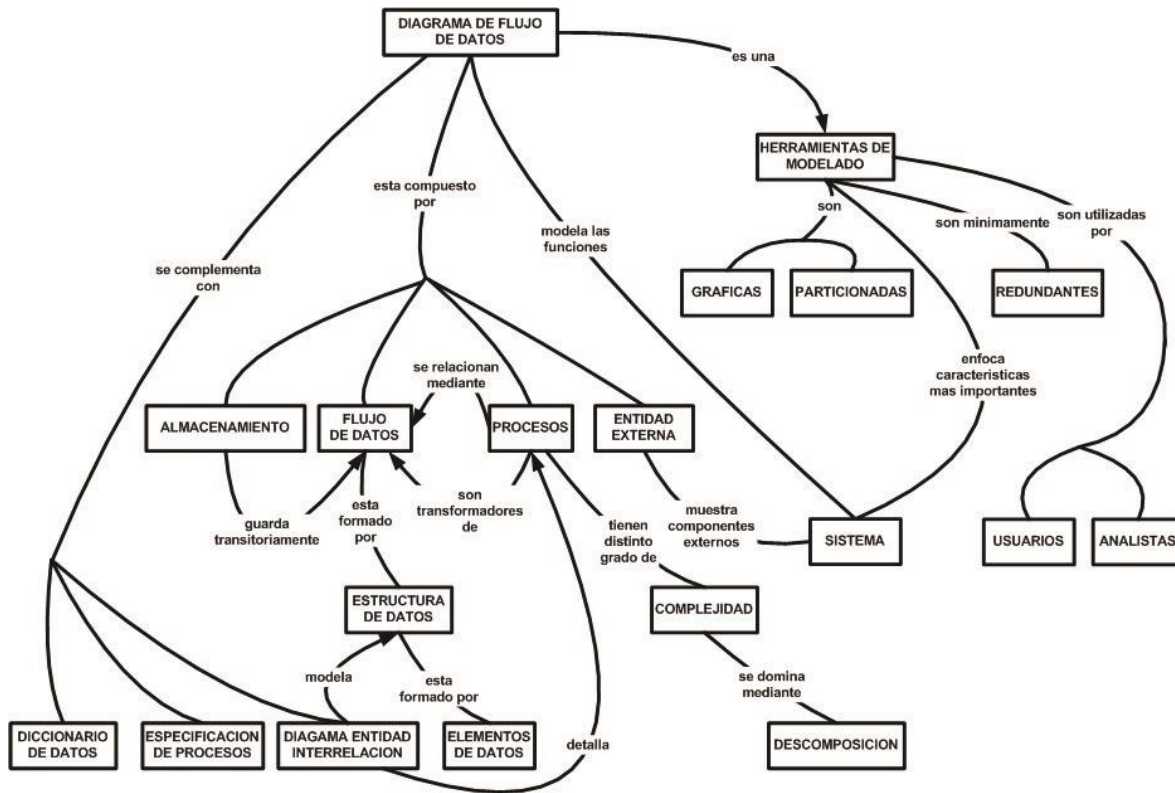


Figura 7.1: Mapa conceptual de “Diagrama de Flujo de Datos”

Los modelos, además, permiten realizar cambios y correcciones en los requisitos a un bajo costo y sin correr ningún riesgo. Si se perciben desentendimientos con el usuario o si éste ha cambiado de idea acerca de los requisitos del sistema, se puede cambiar el modelo o incluso desecharlo y empezar de nuevo. Si no se hiciesen modelos, los cambios en los requisitos sólo se efectuarían después de construir el software, y el costo sería mucho mayor.

Permite, también, verificar que el analista haya entendido correctamente las necesidades del usuario y que las documente de forma tal que los diseñadores y programadores pueden construir el software a partir de él.

1. EL ENFOQUE CLÁSICO

No todas las técnicas de análisis logran los objetivos antes descriptos. Una descripción del sistema de cientos de páginas constituye un modelo, pero oculta las características del sistema, tanto las relevantes como las irrelevantes, su desarrollo puede ser muy costoso, es difícil de modificar y no permite verificar si se han establecido correctamente los requerimientos del usuario.



El análisis de requisitos de información clásico, utilizado en los albores de esta disciplina, consistía en redactar especificaciones funcionales en forma de documentos textuales. Estos tenían una serie de características indeseables.

Por un lado, para entender el sistema había que leer la especificación completa. No había posibilidad de que ni el analista ni el usuario pudiesen centrarse en una determinada parte de la especificación sin tener que leer el resto.

Las especificaciones, con frecuencia, eran redundantes. La misma información se podía encontrar en diferentes partes del documento. Debido a esto, cualquier cambio que se hiciese en la especificación debía reflejarse en varios puntos del documento. Esta situación producía, con frecuencia, inconsistencias en la medida que, si no se realizaban cambios en todas las partes del documento, se tendrían distintas definiciones para los mismos objetos del sistema.

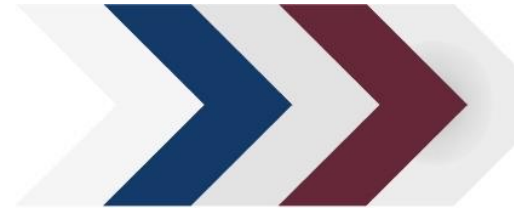
Otra característica indeseable era la ambigüedad, al estar escritas en lenguaje natural, podían ser interpretadas de forma distinta por analistas, usuarios, diseñadores o programadores. Estudios realizados muestran que la mayoría de los errores encontrados en el sistema final con un alto costo asociado a la corrección de esos errores surgía de este tipo de malentendidos.

Debido a esto, la mayor parte del software de esa época carece de una documentación fiable, incluso aunque se hubiese realizado análisis y redactado una especificación.

Por estos motivos fueron surgiendo nuevos métodos de análisis, cuyo objetivo era obtener especificaciones gráficas, formadas por una colección de diagramas, acompañados de información textual detallada, que sirviese de material de referencia; particionadas, de forma que fuese posible leerse o trabajar sobre partes individuales de la especificación sin tener que leerla completamente; mínimamente redundantes, de forma que los cambios en los requisitos se realicen en un sólo punto de la especificación y, además, transparentes, tal que fuesen fáciles de leer y de entender. Los sistemas son complejos, las especificaciones tienen que ser claras y sencillas.

Bajo el nombre genérico de análisis estructurado, se engloban una serie de propuestas de diversos autores, con las características antes nombradas. Entre otros se destacan De Marco, Yourdon, Gane y Sarson. Cada uno de ellos ha desarrollado su propio método de análisis, mejorando, ampliando o adaptando los anteriores a algún campo de aplicación específico.





Existen diferentes técnicas y notaciones de modelado de sistemas que puede utilizar un ingeniero del software. Combinando todas ellas se puede establecer un modelo completo del sistema, pero lo importante es usar aquellos diagramas que sirvan en una situación determinada. Habrá escenarios en que lo importante sea modelar el flujo de datos, mientras que el control sea obvio. En otras situaciones los datos serán lo suficientemente complejos como para justificar un modelo de datos. En otras, por ejemplo, en sistemas de tiempo real, será necesario especificar el comportamiento del sistema en función del tiempo.

En general, la mayoría de los sistemas requerirán utilizar varios modelos para representarlos. Cada uno de ellos se centra en un conjunto limitado de aspectos del sistema, dejando de lado otros. Combinando los diversos modelos se puede tener una visión detallada de todas las características del sistema. Esto es importante ya que los sistemas utilizan estructuras de datos complejas, realizan funciones complejas y tienen pautas de comportamiento complejas.

2. EL DIAGRAMA DE FLUJO DE DATOS

Un sistema de procesamiento de datos incluye tanto datos como procesos, y cualquier análisis de un sistema debe incluir ambos aspectos. Se necesita una técnica para modelar sistemas que describa qué funciones son las que realiza, qué interacción se produce entre esas funciones, qué transformaciones de datos realiza el sistema y qué datos de entrada se transforman en qué datos de salida.

A medida que la información se mueve a través del sistema, va siendo modificada mediante una serie de transformaciones. El diagrama de flujo de datos es una técnica gráfica que representa el flujo de datos y las transformaciones que se aplican a ellos.

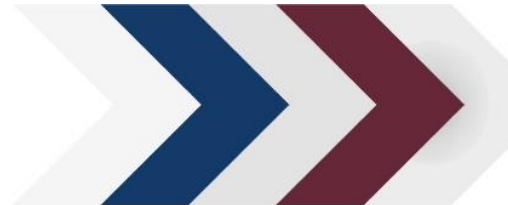
Representa, esencialmente, qué funciones o qué transformaciones se realizan sobre los datos pero no cuándo se realizan o en qué secuencia.

3. LOS COMPONENTES DEL DIAGRAMA DE FLUJO DE DATOS

Para representar el sistema mediante diagrama de flujo de datos se utilizará la notación propuesta por Yourdon, posiblemente la más extendida. Se detallará a continuación los elementos que se utilicen en el diagrama de flujo de datos.

LOS PROCESOS





Los procesos o funciones son, básicamente, transformadores de flujos que muestran una parte del sistema que transforma datos de entrada en datos de salida. Se representa gráficamente con un círculo (figura 7.1).



Figura 7.1: Representación de un proceso

Son los componentes que realizan cada una de las funciones del sistema. Los procesos en el diagrama de flujo de datos evitan mostrar detalles procedimentales, estos se aclaran mediante las especificaciones de procesos.

LAS ENTIDADES EXTERNAS

Las entidades externas (figura 7.2), representan a los elementos externos al sistema. Estos pueden ser personas, organizaciones u otros sistemas que interactúan con él y que están fuera de los límites del sistema. Las entidades externas proporcionan datos que serán transformados por el sistema o son las que consumen los datos que fueron transformados por el sistema.



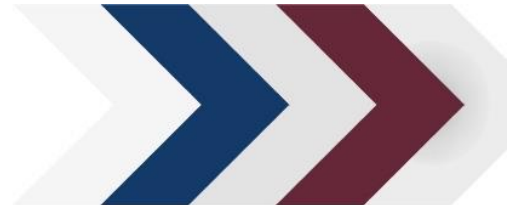
Figura 7.2: Representación de una entidad externa

Los flujos de datos que comuniquen el sistema con las entidades externas representan las interfaces del mismo. Las entidades externas aparecen, principalmente, en el diagrama de contexto y pueden repetirse en los niveles inferiores.

La forma de reconocerlas es preguntando qué harán esas entidades con los datos que les ofrece el sistema o de donde obtiene los datos que la entidad externa entrega al sistema. Si la respuesta a ambas preguntas es “no interesa”, se está en presencia de una entidad externa.

Es evidente que ni el analista ni el diseñador están en posibilidades de cambiar los contenidos de una entidad externa o la forma en que trabaja. Sin duda existirán muchas entidades externas,





pero solamente se deben incluir, como objetos de estudio, aquellas que se conectarán con el sistema.

LOS FLUJOS DE DATOS

Los flujos de datos representan datos o estructuras de datos que fluyen a través del sistema (figura 7.3). La flecha indica el sentido de flujo. Los flujos de datos conectan procesos entre sí, con entidades externas o con almacenamientos de datos.



Figura 7.3: Representación de un flujo de datos

Los flujos de datos representan datos en movimiento. Puede tratarse de un elemento de dato simple o compuesto o incluso de una colección de datos de estructura compleja y van etiquetados con un nombre que identifica a los datos que transportan y, posiblemente, con el estado de dicha información. Por ejemplo, número de teléfono, número de teléfono correcto, número de teléfono erróneo.

LOS ALMACENAMIENTOS DE DATOS

Mientras que los flujos de datos representan datos en movimiento los almacenamientos representan datos estáticos o en reposo.

Los almacenamientos describen a los datos persistentes que serán utilizados por el sistema (figura 7.4). Permiten guardar, temporalmente, datos que luego podrán ser procesados por el mismo proceso que los creó o por otro distinto. En la mayoría de los casos, se utilizarán almacenamientos de datos cuando dos procesos intercambien datos pero sus funciones no se ejecuten en forma simultánea.



Figura 7.4: Representación del un almacenamiento



Los almacenamientos se comunican solamente con los procesos mediante flujos de datos, de modo tal que éstos pueden entrar o salir de él (figura 7.5). Si un flujo de datos ingresa a un almacenamiento se interpreta como una modificación del contenido de su estructura, esto es, una alta, baja o una actualización de uno o más valores de los elementos de datos que lo componen. Un flujo de datos que parte desde un almacenamiento a un proceso expresa la lectura de uno o más elementos, no hay modificación del contenido de su estructura de datos.



Figura 7.5: Flujos entrantes y salientes a un almacenamiento

Los almacenamientos de datos no detallan ningún tipo de implementación ulterior y van etiquetados con un nombre significativo que haga referencia a los datos que contienen, generalmente en plural.

4. LAS RELACIONES ENTRE COMPONENTES DEL MODELO

No todos los componentes del modelo pueden relacionarse entre sí. Los procesos pueden comunicarse con almacenamientos, otros procesos o entidades externas. Los almacenamientos y las entidades externas sólo pueden comunicarse con procesos (figura 7.6).

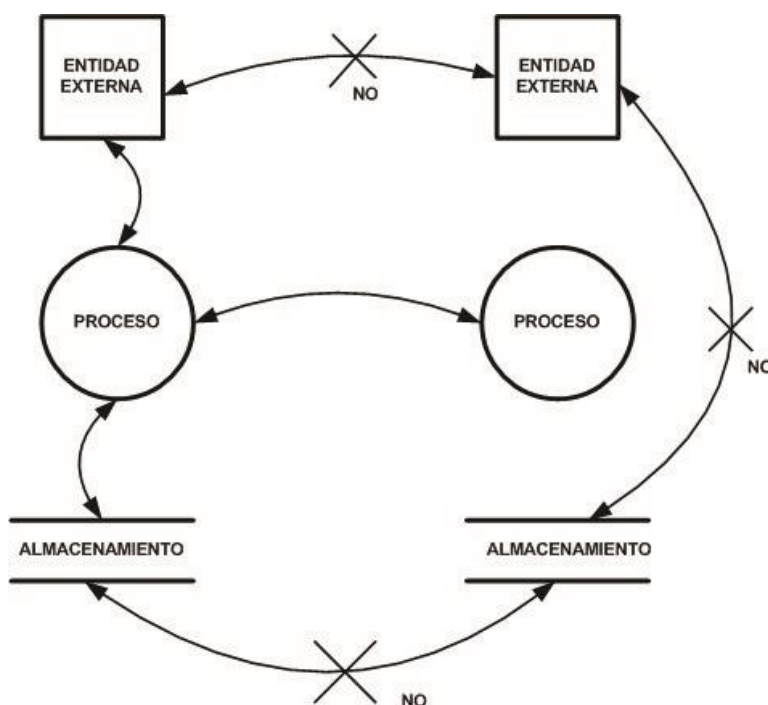


Figura 7.6: Relaciones entre componentes del modelo

5. EL DIAGRAMA DE CONTEXTO

Se puede utilizar el diagrama de flujo de datos para representar el sistema en cualquier nivel de abstracción. El diagrama de flujo de datos de nivel 0 se lo denomina diagrama de contexto (figura 7.7) y en él el sistema está representado por un sólo proceso, que identifica cuál es la función principal del sistema, mostrando además los flujos de datos que lo relacionan con las entidades externas.

El diagrama de contexto tiene una gran importancia ya que, aunque simple, resume el requisito principal del sistema, esto es, recibir entradas, luego las procesa de acuerdo con determinadas funciones y genera, por último, las salidas. A partir del diagrama de contexto se pueden ir construyendo nuevos diagramas, en un proceso de descomposición descendente, que vaya definiendo con mayor nivel de detalle los flujos de datos y procesos de transformación que ocurren en el sistema, de forma tal que al final se obtendrá una jerarquía de diagramas.

En general, cualquier proceso que aparezca en un diagrama de flujo de datos puede ser descripto, más detalladamente, en un nuevo diagrama. A esto se denomina explosión de un proceso. Cada proceso que se está describiendo aparece descompuesto en una serie de subprocesos o subsistemas, cada uno de los cuales se encarga de detallar un aspecto determinado del proceso original.



Los flujos de datos que entran y salen del proceso que se está describiendo deben entrar y salir del diagrama de flujo de datos que lo desarrolla. Además de estos flujos, el diagrama contendrá, por lo general, nuevos flujos que comunican los procesos que figuran en él y posiblemente también almacenamientos de datos.



Figura 7.7: El diagrama de contexto

6. LA CONSTRUCCIÓN DE UN DIAGRAMA DE FLUJO DE DATOS

Dado que en el diagrama de contexto el sistema estará representado por un sólo proceso, este diagrama de flujo de datos de nivel 0 sólo dará lugar, luego del proceso de explosión, a un diagrama de flujo de datos de nivel 1. A su vez, éste puede dar lugar a tantos diagrama de flujo de datos de nivel 2 como procesos contenga, y así sucesivamente hasta que se haya alcanzado un nivel en el que los procesos sean lo suficientemente simples como para no necesitar una descripción más detallada en un diagrama de flujo de datos. De esta forma, se construye una jerarquía de diagrama de flujo de datos.

7. LOS PROCESOS POR NIVEL

Un diagrama de flujo de datos debe contener, por simplicidad, no más de 9 procesos. Es razonable tener 7 ± 2 procesos por nivel. Una de las características más importantes de las herramientas de modelado en la facilidad de su lectura. Un diagrama de flujo de datos con muchos procesos sería complicado para su lectura y comprensión,

LOS NOMBRES DE LOS PROCESOS

Cada proceso de un diagrama de flujo de datos debe tener un nombre corto y significativo. Normalmente un verbo más un sustantivo. Si bien la descripción de los componentes es una de las tareas encomendada al diccionario de datos, es conveniente que los nombres que se propongan a los procesos del diagrama de flujo de datos sean significativos en el contexto y cortos en su extensión.



LA NUMERACIÓN DE LOS PROCESOS

Los procesos deben ser numerados. Todo proceso, en cualquier nivel, debe tener un número que lo identifique. La numeración de los procesos en el diagrama de flujo de datos no implica secuencialidad, solamente es una referencia para identificarlo a él y a sus procesos descendientes.

8. LA EXPLOSIÓN DE LOS PROCESOS

Para modelar sistemas complejos se utiliza la descomposición, ésta da como resultado un diagrama de flujo de datos a distintos niveles de detalle. La descomposición o explosión en subprocesos es uno de los mecanismos que se utilizan para administrar la complejidad. El límite de la descomposición de los procesos está dado por la complejidad del mismo.

Los diagrama de flujo de datos de niveles inferiores desarrollan, de forma más concreta, los procesos de niveles superiores. La explosión de procesos implica la especificación más detallada de los procesos más complejos y se realiza hasta alcanzar un nivel de especificación mínimo y sencillo.

9. EL BALANCEO EN LAS EXPLOSIONES

En cada diagrama de flujo de datos proveniente de la explosión de un proceso de orden superior debe representarse los mismos flujos de datos que en el proceso padre. La explosión implica el detalle de un proceso complejo dividido en subprocesos, por tal motivo los flujos de entrada y de salida en el proceso superior debe corresponderse con los de los niveles inferiores.

LA NUMERACIÓN EN LA EXPLOSIÓN DE PROCESOS

Normalmente, el sistema que se está modelando será lo suficientemente complejo como para necesitar varios diagramas de flujo de datos. Se precisará construir, entonces, una jerarquía de diagrama de flujo de datos.

Cada diagrama de flujo de datos de un nivel “n” será el resultado de la explosión de un proceso de un diagrama de flujo de datos de nivel “n-1” (figuras 7.8 y 7.9). Es necesario que el nombre del diagrama de flujo de datos sea el nombre del proceso que desarrolla, que la numeración de los procesos en el diagrama de flujo de datos “n” derive del número del diagrama de flujo de datos





“n-1”, por ejemplo, 3, 3.1, 3.2, etc., además debe mantenerse la consistencia entre los flujos de datos en ambos niveles.

La regla de equilibrio dice que los flujos de datos que entran o salen del proceso de orden superior deben aparecer en el diagrama de flujo de datos de orden inferior, manteniendo, además, el nombre y el sentido.

En los diagramas de flujo de datos de niveles superiores generalmente es necesario, cuando los flujos de datos pasan al nivel inferior, descomponerlos. Para realizar esta descomposición se puede hacer divergir el flujo en dos o más o, al menos, dejar reflejada esta descomposición en el diccionario de datos.

El diagrama de flujo de datos debe ser leído y comprendido, no sólo por el analista que construyó el modelo, sino también por los usuarios que son los expertos en la materia de aplicación.

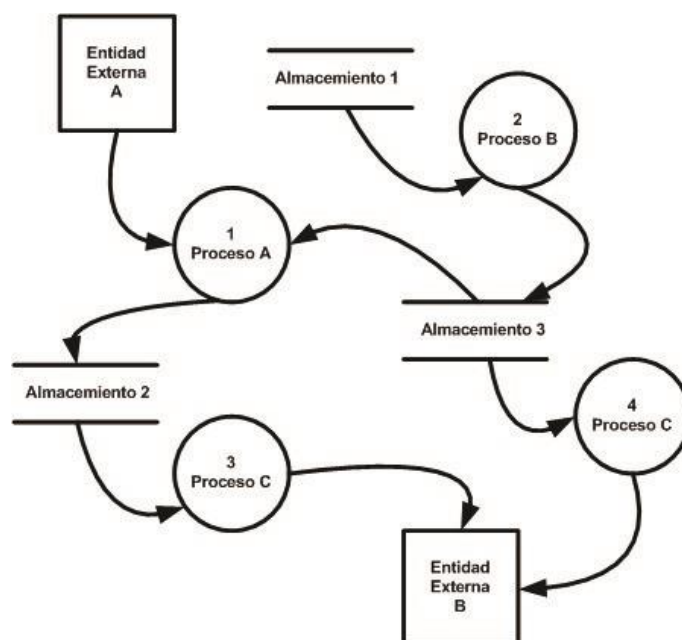


Figura 7.8: Diagrama de flujo de datos de primer nivel

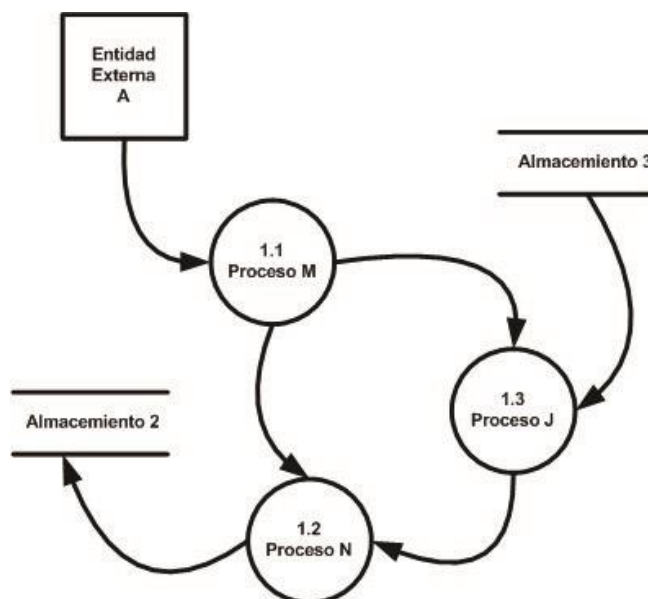


Figura 7.9: Diagrama de flujo de datos de segundo nivel

10. LAS ESPECIFICACIONES DE LOS PROCESOS PRIMITIVOS

Los diagramas de flujo de datos permiten identificar las principales funciones o transformaciones que realiza un sistema y las relaciones entre éstas, pero no indican nada acerca de los detalles de cómo se realizan estos procesos.

Para definir los detalles de qué datos de entrada se transforman en qué datos de salida y cómo se realiza esta transformación se necesita una descripción detallada de los procesos. Para esto se utilizará la especificación de proceso.

En los diagramas de flujo de datos de menor nivel, esto es, los más altos en la jerarquía, los procesos se describen mediante un nuevo diagrama de flujo de datos que define, más detalladamente, las funciones que realiza y los flujos que maneja. Este proceso de descomposición debe continuar hasta que se alcance un nivel en el que un proceso pueda ser descrito de forma sencilla y no ambigua. Estos procesos se denominan primitivos.

Una forma de especificar un proceso primitivo es mediante un algoritmo. Esto no quiere decir que el algoritmo propuesto en el análisis se corresponda con la implementación final del proceso, sino que es simplemente una forma de describir la operación. La elección del algoritmo definitivo se



hace en la fase de diseño, considerando además de la especificación, criterios de eficiencia y las características del lenguaje de implementación.

11. RESUMEN

El diagrama de flujo de datos es una herramienta de modelado que describe el comportamiento de un sistema independientemente de su implementación. Está compuesto por procesos, que son transformadores de flujos de datos; almacenamientos, que constituyen depósitos temporales donde se guardan los datos para su posterior utilización; flujos de datos, que representan conductos por donde pasan los datos del sistema y entidades externas que son objetos externos al sistema con los cuales éste se comunica.

El diagrama de flujo de datos es un conjunto de procesos interrelacionados mediante flujo de datos que se utiliza en sistemas de información donde la descripción de los procesos es importante para la etapa de modelado.

Especificación de Procesos Estructurados

La especificación de requisitos de los usuarios es el documento que describe las características funcionales de un sistema de información que va a ser desarrollado. Es un documento fundamental ya que, por un lado representa el contrato acordado con el cliente, y por otro sirve de base a todas las etapas posteriores de la ingeniería de software.

El principal objetivo de la especificación de procesos estructurados es definir, de forma clara y no ambigua, las funciones y restricciones del sistema, de forma tal de evitar problemas en las etapas de diseño y codificación.

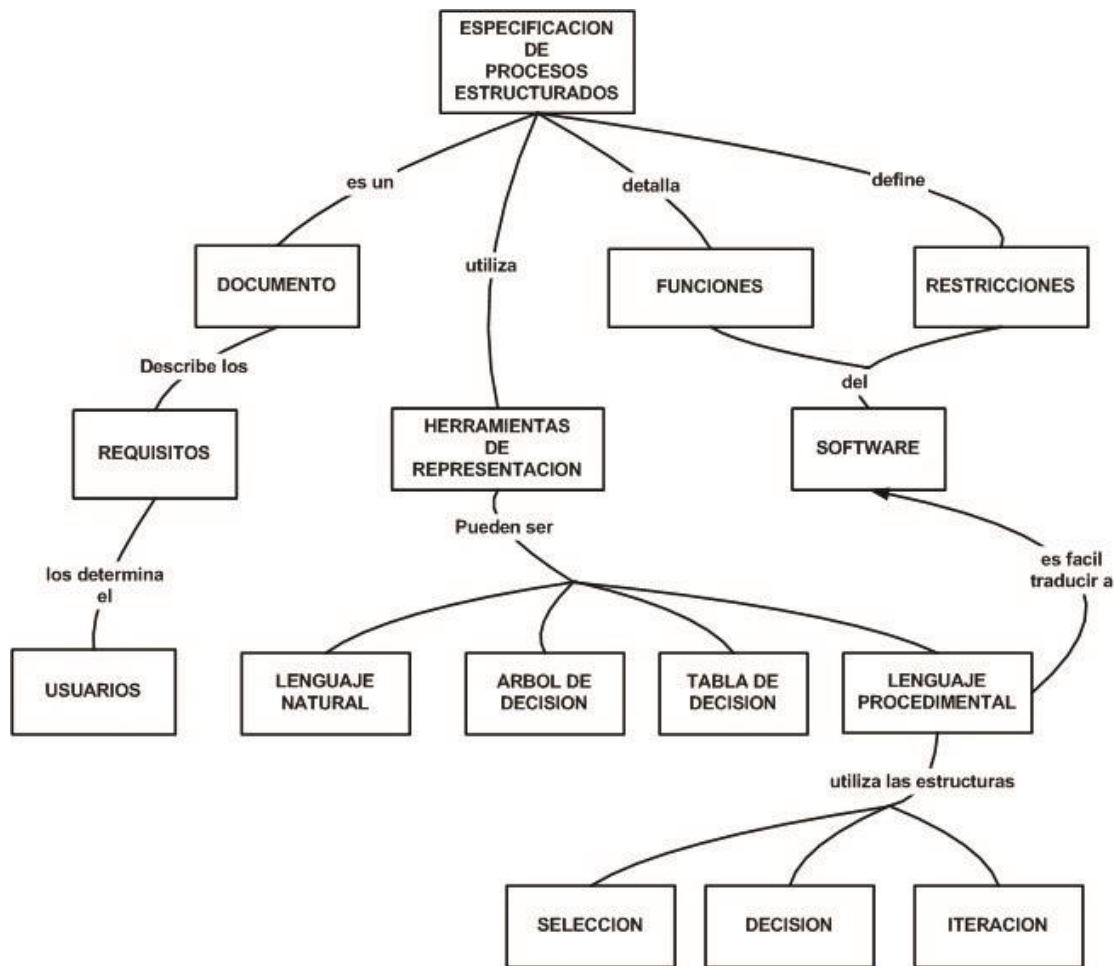


Figura 9.1: Mapa conceptual de “Especificación de Procesos Estructurados”

La especificación de procesos estructurados es el resultado de las tareas de análisis. Para comprender mejor lo que el usuario necesita, se divide el problema en partes y se desarrollan representaciones o modelos que muestren la esencia de los requisitos. Si éste no refleja los requerimientos del cliente entonces, inevitablemente, el diseñador construirá un sistema incorrecto. Por otra parte, si la especificación es incompleta, ambigua o inconsistente, aunque haya sido aceptada por el cliente, no se podrá satisfacer las necesidades en forma adecuada.

Cada una de las dudas, producto de una mala especificación, plantea una serie de alternativas y obliga a elegir entre varias posibilidades durante el diseño y la implementación. Pero estas decisiones no se tomarán con un entendimiento completo del problema, debido a que las realizan diseñadores y programadores que nunca han estado en contacto con el cliente, por lo que el resultado será una implementación que, con seguridad, no cumplirá las necesidades del usuario. Por lo tanto, la forma de especificar un sistema tiene una gran influencia en la calidad de la solución implementada finalmente.



12. LOS LENGUAJES Y HERRAMIENTAS PARA LA ESPECIFICACIÓN DE LOS PROCESOS

La especificación consiste en describir un sistema de forma tal que queden expresadas su funcionalidad, sus restricciones y su rendimiento de la forma más clara y precisa posible. Para ello se debe utilizar algún tipo de lenguaje, existen varias alternativas, cada una de ellas con sus ventajas e inconvenientes.

13. EL LENGUAJE NATURAL

La solución más intuitiva para la descripción de procesos es utilizar el lenguaje natural, esta es la opción que se ha utilizado tradicionalmente. Entre sus ventajas se puede citar la facilidad de uso y entendimiento, no se debe aprender ningún nuevo lenguaje y cualquier persona puede leer la especificación, para luego comentarla o criticarla. Entre los inconvenientes están la imprecisión y la ambigüedad.

Aunque el análisis de requisitos se haya realizado correctamente, una especificación en lenguaje natural, por las características antes señaladas, puede dar lugar a que la implementación final no cumpla con los requerimientos. Además, debido a su propia facilidad de uso e imprecisión, las especificaciones suelen ocultar características del sistema que sólo se pondrán de manifiesto a la hora de implementarlas, es decir, al traducir la especificación a un lenguaje de programación.

14. LOS LENGUAJES PROCEDIMENTALES

La especificación del proceso debe expresarse de manera tal que pueda ser verificado tanto por el usuario como por el analista. Precisamente, por esta razón, se evita el lenguaje narrativo como herramienta de especificación, ya que es ambiguo, sobre todo si describe acciones alternativas y repetitivas. Por su naturaleza, también tiende a causar gran confusión cuando expresa condiciones booleanas compuestas, esto es, combinaciones de los operadores and, or y not.

Los lenguajes de programación tienen una sintaxis carente de ambigüedad, más una semántica bien definida. Con el fin de poder acercar la especificación a la implementación, se podrían utilizar esos mismos lenguajes, o un pseudocódigo basado en ellos, para especificar sistemas. De esta forma, al aproximar la especificación a la implementación se reducen los errores en la codificación.

Otra ventaja es que, al utilizar un lenguaje de programación, es mucho más fácil definir la interfaz de un proceso, ésta está formada por el conjunto de los parámetros de dicha especificación y, además, se podría comprobar si todos los datos necesarios para realizar la transformación están indicados en dicha interfaz.





Los principales inconvenientes son tres. Por un lado, los lenguajes de programación no son lo suficientemente formales como para poder deducir, a partir de él, propiedades de completitud, consistencia o corrección de una especificación. Por otra parte, el uso de lenguajes procedimentales obliga a definir un algoritmo a la hora de especificar un proceso, pero el objetivo de la especificación no es definir el cómo se ha de implementar el sistema sino definir qué se ha de implementar. Otro inconveniente es que el lenguaje utilizado para la especificación sea empleado como lenguaje final de implantación no siendo, quizá, el más apropiado.

Otra alternativa para la descripción de las funciones de un sistema es la especificación del proceso mediante pseudocódigo. Este es un lenguaje similar a los convencionales, con la diferencia de que no tiene una sintaxis tan rígida. Permite la descripción de qué es lo que sucede en cada burbuja primitiva del diagrama de flujo de datos, sin exigir decisiones prematuras. Su propósito es definir lo que debe hacerse para transformar entradas en salidas.

La descripción mediante pseudocódigo pueda ser comunicada adecuadamente al amplio público que está involucrado en el desarrollo del sistema. A pesar de que el analista es típicamente quien escribe la especificación del proceso, habitualmente serán diversos tipos de usuarios, tales como los administradores, los auditores, el personal de control de calidad y otros, los que leerán la especificación del proceso.

Las construcciones utilizadas en las especificaciones de procesos mediante pseudocódigo son las estructuras de secuencia, iteración y decisión.

LA ESTRUCTURA DE SECUENCIA

La estructura de secuencia (figura 9.2), se caracteriza por tener una entrada y una salida dentro de la cual se encuentran una serie de acciones cuya ejecución es lineal y en el orden en que aparecen. A su vez, todas las acciones tienen una única entrada y una única salida.



Figura 9.2: Estructura de secuencia

LA ESTRUCTURA DE DECISION

La estructura de decisión también tiene una sola entrada y una sola salida, pero dentro de la misma se realiza una acción de entre varias, según una condición preestablecida. Esta condición puede ser simple o compuesta. Las estructuras de decisión pueden ser de dos salidas, en la que una de ellas puede ser la acción nula o de tres o más salidas, que se denominan múltiples.





Se detallarán a continuación las distintas estructuras de decisión.

La estructura “SI ENTONCES SI-NO FIN-SI” (figura 9.3), admite dos salidas y comienza con la palabra reservada SI seguida de la condición que debe probarse. La alternativa que debe ser ejecutada, si la condición resulta verdadera, se encuentra precedida por la palabra ENTONCES. La otra alternativa que debe realizarse, si la condición es falsa, esta precedida por la palabra SI-NO

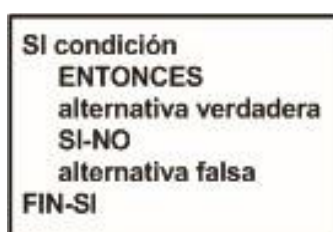


Figura 9.3: Estructura de decisión

Puede suceder (figura 9.4) que, si la condición es falsa, no existan acciones a ejecutar luego del SI-NO, entonces

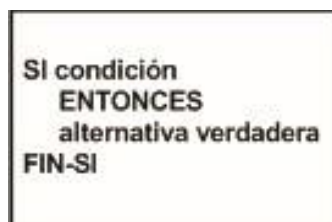


Figura 9.4: Estructura de decisión reducida

Tanto la alternativa verdadera como la falsa, pueden tener a su vez estructuras de decisión.

La estructura de decisión generalizada o múltiple (figura 9.5), tiene el siguiente formato general:

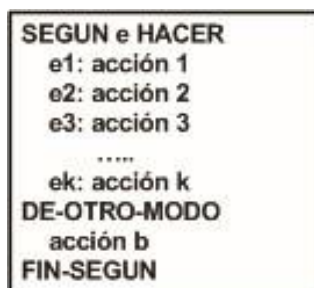




Figura 9.5: Estructura de decisión

En esta estructura, existe una expresión “e” cuyos resultados pueden ser los valores e_1, e_2, \dots, e_k . En función de estos valores se ejecutará una y sólo una de las acciones indicadas. Si el valor de “e” es distinto de e_1, e_2, \dots, e_k , se ejecutará la acción b.

LA ESTRUCTURA DE REPETICIÓN

Otra de las construcciones utilizadas en la estructura de repetición. En esta estructura existe una entrada y una salida dentro la cual se repite una acción que generalmente es una estructura de secuencia, un número determinado o indeterminado de veces, dependiendo en este caso del cumplimiento de una condición.

Las estructuras de repetición pueden ser:

La estructura PARA (figura 9.6), es una estructura que ejecuta un conjunto de acciones un número “n” de veces conocido anticipadamente.

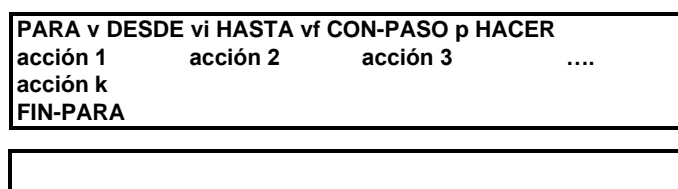


Figura 9.6: Estructura de repetición “PARA”

Donde v es una variable numérica, llamada variable de condición, v_i y v_f son variables de tipo numérico, constantes numéricas o expresiones aritméticas. v_i es el valor inicial, v_f es el valor final y p es el paso. p puede ser positivo o negativo, pero no nulo.

La estructura MIENTRAS (figura 9.7) es una estructura que repite una acción mientras se cumpla la condición que controla el bucle. La característica principal de esta estructura es que la condición es evaluada siempre antes de cada repetición.

El número de repeticiones de esta construcción oscila entre 0 e infinito, dependiendo de la evaluación de la condición cuyos argumentos, al menos una vez, deberán modificarse dentro del bucle, pues de no ser así el número de repeticiones será infinito constituyéndose en un bucle sin salida.

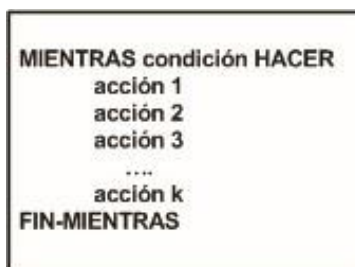


Figura 9.7: Estructura de decisión “MIENTRAS”

La estructura REPETIR HASTA (figura 9.8), es una estructura que repite una acción hasta que se cumpla la condición que controla el bucle, la cual se evalúa después de cada ejecución del mismo. El número de repeticiones oscila entre 1 e infinito, dependiendo de la evaluación de la condición, cuyos argumentos deberán modificarse dentro del bucle, pues de no ser así, también, el número de repeticiones será infinito estando nuevamente en presencia de un bucle sin salida.

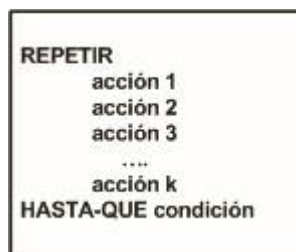


Figura 9.8: Estructura de decisión “REPETIR”

15. EL ÁRBOL DE DECISIÓN

El árbol de decisión (figura 9.9), es un diagrama que muestra en forma secuencial condiciones y acciones, presenta qué condiciones se consideran en primer lugar, cuál en segundo y así sucesivamente hasta que al final se despliega una acción determinada.

Esta herramienta de especificación se utiliza principalmente para organizar la información recopilada en la etapa de requerimientos y poder entender las combinaciones de condiciones cuando éstas no son muy complejas y no todas son posibles.

La construcción del árbol comienza desde la raíz, donde se inicia también la secuencia de decisión, la rama a seguir dependerá de las condiciones y de la decisión que deba tomarse. Las hojas, al final del árbol, representan las acciones a seguir.

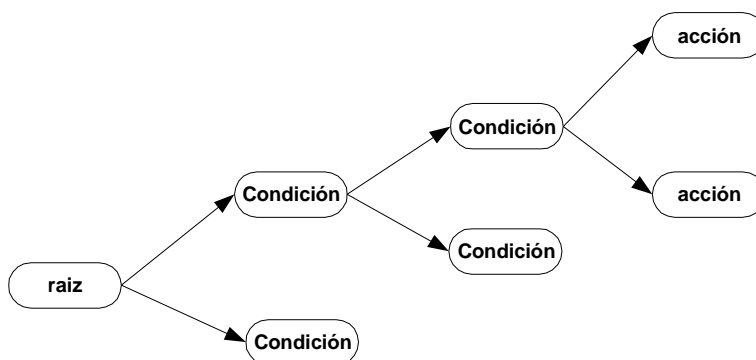


Figura 9.9: Árbol de decisión

16. LA TABLA DE DECISIÓN

La tabla de decisión (figura 9.10) se utiliza cuando se deben especificar procesos complejos. Está compuesta por cuatro cuadrantes. En el primer cuadrante, está la combinación de condiciones, aquí se enlistan todas las posibles combinaciones de condiciones, que por ser éstas booleanas tendrá 2^n columnas.

El segundo cuadrante, denominado identificación de condiciones, señala las condiciones que se consideran relevantes, éstas deben ser booleanas, es decir, condiciones que solamente pueden ser verdaderas o falsas.

En el tercer cuadrante, se ubica la identificación de acciones, en él se detallan todas las posibles acciones a realizar.

Por último, En el cuarto cuadrante se ubican las acciones elegidas. Aquí se realiza una marca en la intersección de la combinación de las condiciones y la acción a realizar después de concluir el análisis.

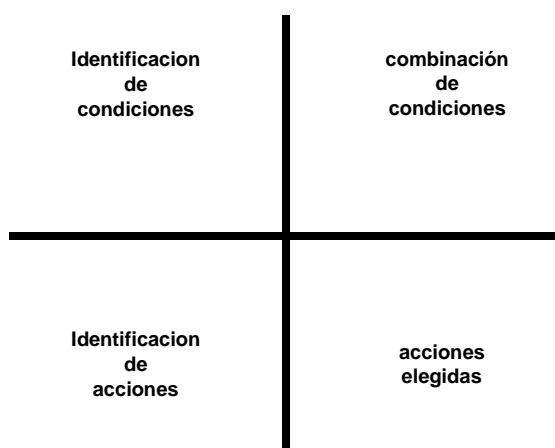




Figura 9.10: Tabla de decisión

Para construir la tabla (figura 9.11), se parte de la identificación de todas las condiciones relevantes, la respuesta a estas condiciones debe ser verdadera o falsa. La cantidad de condiciones limita la construcción práctica de la tabla de decisión; el número de combinaciones de condiciones crece en forma exponencial (2 condiciones, 4 combinaciones, 3 condiciones, 8 combinaciones, 4 condiciones, 16 combinaciones, etc.)

Luego se identifican las acciones a seguir. La cantidad de acciones no constituye un limitante práctico en la construcción de la tabla. Por último, se marcan, para cada combinación de condiciones, la o las acciones a realizar.

Luego de terminada la construcción de la tabla, se analizan las redundancias y contradicciones que pudieran existir. Hay redundancia cuando aparecen las mismas condiciones para dos o más combinaciones y ésta difieren en solo una condición. Existe contradicción cuando un conjunto de condiciones no puede ocurrir nunca. Esto puede suceder cuando se trabaja con rango de valores, por ejemplo fechas o valores numéricos.

Condición 1	V	V	V	V	F	F	F	F
Condición 2	V	V	F	F	V	V	F	F
Condición 3	V	F	V	F	V	F	V	F
Acción 1		X				X		
Acción 2	X			X	X			X
Acción 3			X				X	

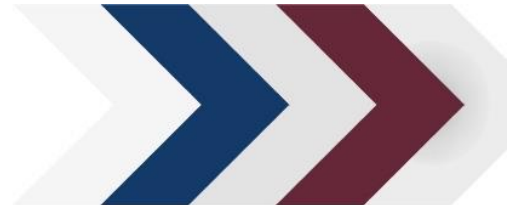
Figura 9.11: Ejemplo de tabla de decisión con tres condiciones

9.5. RESUMEN

El objetivo principal de la especificación de procesos es definir de forma clara y no ambigua las funciones y restricciones del sistema, de forma que se eviten problemas en las etapas de diseño y codificación. La especificación es el resultado del proceso de análisis. Una especificación es, por definición, una descripción de lo que se quiere realizar, no de cómo se va a realizar o implementar. Se pueden utilizar, para su descripción, un conjunto de herramientas tales como el lenguaje natural, al lenguaje procedimental, el árbol de decisión y la tabla de decisión.

Diccionario de Datos





Los diferentes modelos del análisis estructurado permiten tener una perspectiva del sistema desde diferentes puntos de vista. Cada uno de estos modelos enfoca una visión determinada del sistema en estudio. Toda la información que complementa a los modelos gráficos se encuentra organizada y detallada en el diccionario de datos.

El diccionario de datos, por lo tanto, es un listado organizado que contiene los detalles del sistema, con definiciones precisas y rigurosas para que, tanto el usuario como el analista, tengan un entendimiento común de todas las entradas, salidas, componentes de los almacenamientos y procesos intermedios que existen en el sistema.

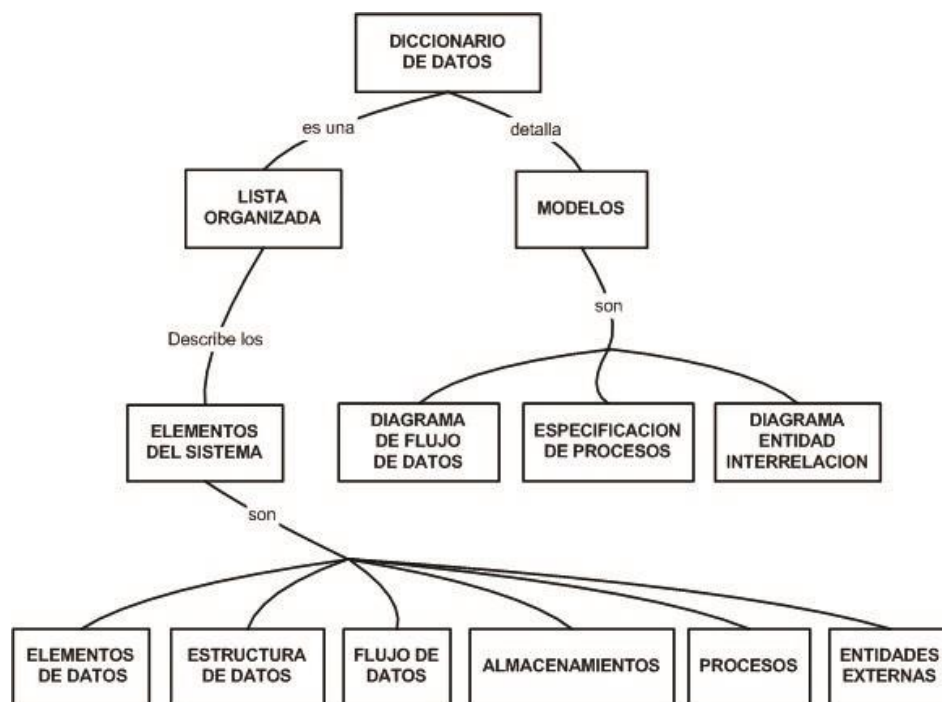


Figura 10.1: Mapa conceptual de “Diccionario de Datos”

El diccionario de datos define el significado de los flujos y almacenamientos que se muestran en los diagramas de flujo de datos; detalla la composición de la agrupación de estructura de datos que se mueven a lo largo de los flujos, es decir, estructuras complejas que pueden descomponerse en unidades más elementales. Describe, además, la composición de los almacenamientos y especifica los valores y unidades relevantes de los datos elementales que componen los flujos de datos y los almacenamientos de datos. Detalla, también, las relaciones entre entidades que se enfatizan en el diagrama de entidadinterrelación



17. LA NOTACIÓN DEL DICCIONARIO DE DATOS

El diccionario de datos posee una notación simple que permite describir cada uno de los componentes del sistema en estudio.

EL DATO ELEMENTAL

El dato elemental es la mínima unidad indivisible, participa como componente en las estructuras de datos del sistema. Cualquier descomposición del dato elemental carecerá de sentido dentro del ámbito de estudio. Son sinónimos de datos elementales: elemento de dato, campo, atributo, etc. El dato elemental es el componente con mayor nivel de detalle, de él se especifica el nombre, su descripción, el alias, su longitud y tipo y el dominio de valores admisibles. El nivel de detalle dependerá de las necesidades del analista y muchas veces de la herramienta utilizada para la documentación.

Se deben asignar, a los datos elementales, nombres que sean significativos en el contexto del desarrollo del sistema. Por ejemplo, son nombres válidos: apellido, fecha-factura, no es conveniente utilizar, en la descripción de los datos elementales, nombres enigmáticos o carentes de significado.

Por ejemplo: XXY, fec-noc-ter.

La descripción del elemento de datos indica, de manera breve, lo que éste dato elemental representa para el sistema. Permite tener una descripción del significado que se le atribuye en el sistema al elemento de dato. El símbolo que se utiliza para realizar comentarios textuales en la definición de los componentes es “* *”.

Por ejemplo: Apellido = *apellido del empleado*”, DNI = *documento nacional de identidad*.

El alias se utiliza cuando existe, en la organización, más de una forma de denominar a un dato elemental.

Por ejemplo: fecha de nacimiento = *alias de fech_nac*

El longitud y el tipo detalla, independientemente de la implementación en un lenguaje de programación determinado, la cantidad de espacio asignado a él y el tipo de dato que representa. Este puede ser, entre otros, alfanumérico, numérico; fecha, boolean, etc.





Por ejemplo: Nombre, alfanumérico (30), Edad, numérico (2).

Existen otras formas de representar tanto la longitud como el tipo de datos, esto es, mediante un comentario.

Por ejemplo: Nombre = *alfanumérico de 30 espacios*, Edad = *numérico de 2 espacios*

El dominio detalla el conjunto de valores permitidos para cada dato elemental. El dominio puede ser de valores discretos o continuos. Cuando es de valores discretos se especifica detallando por extensión, mediante un par ordenado, los posibles valores y su significado. Por ejemplo: estado-civil = {(s, soltero); (c, casado), ...}. Cuando los valores permitidos son continuos se especifica el rango mediante el valor inicial y valor final que puede tomar.

Por ejemplo: fecha-nac = {vi:17/06/98; vf: 19/02/99}

LA ESTRUCTURA DE DATOS

Los datos elementales se agrupan en estructuras para describir componentes del sistema. Una estructura de datos está compuesta por elementos y/o estructuras de datos. Estas estructuras se construyen sobre un conjunto de relaciones entre los componentes. La definición de una estructura de datos comienza con el símbolo "=", este tiene el significado de "está compuesto por".

La relación secuencial define los componentes que siempre estarán incluidos en la estructura. Pueden ser datos elementales o estructuras de datos. A esta relación se denomina también concatenación. El símbolo que se utiliza para su descripción es "+".

Por ejemplo:

Dirección = calle + número + código_postal + localidad

Alumno = código + nombre + dirección

La relación de selección define distintas alternativas para datos elementales o estructura de datos incluida dentro de una estructura. De todas las opciones, solamente se elige una. La relación de selección se puede considerar, para los datos elementales, como un dominio de valores discretos.





El símbolo que se utiliza para su descripción es “[]”. Para separar opciones alternativas en la construcción de selección, se utiliza el símbolo “|”

Por ejemplo:

Documento = [dni | cedula de identidad | pasaporte] sexo = [Femenino | Masculino]

La relación de repetición define la iteración de un dato elemental o una estructura de datos cero o más veces dentro de una estructura. El símbolo que se utiliza para su descripción es “ $v_i\{ \}v_f$ ”; siendo v_i el valor inicial y v_f el valor final .

Por ejemplo:

Solicitud = nombre del cliente + domicilio de envío + 1{artículo}10. Significa que puede requerirse entre 1 y 10 artículos en la solicitud.

La relación opcional indica que el dato elemental o la estructura de datos puede estar o no presente dentro de una estructura. Es un caso especial de repetición con $v_i = 0$ y $v_f = 1$. El símbolo que se utiliza es “()” .

Por ejemplo:

domicilio del cliente = domicilio de envío + (domicilio de facturación) O, lo que es igual: domicilio del cliente = domicilio de envío + 0{domicilio de facturación}1 El identificador indica un campo único en los almacenamientos, que puede ser compuesto y tiene la particularidad de no poder repetirse y tampoco tener valores nulos. Corresponde, en el modelo relacional, a la clave primaria. El símbolo que se utiliza para su descripción es “@”

Por ejemplo:

Socio = @código + nombre + apellido + dirección

EL FLUJO DE DATOS

Los flujos de datos son conductos por donde pasan los elementos de datos o las estructuras de datos que comunican a los componentes del diagrama de flujo de datos. En este diagrama, los flujos se representan mediante líneas con flechas que indican su dirección. Se describe, en el diccionario de datos, mediante el nombre, contenido, origen y destino





El nombre se utiliza para referenciar un flujo de datos. Se deben asignar nombres de flujos que sean significativos en el contexto del desarrollo del sistema. Por ejemplo: Datos del socio. La descripción indica, de manera breve, lo que ese flujo representa para el sistema. Por ejemplo: Datos de los socios del club El contenido, describe los componentes del flujo, pueden ser elementos de datos o estructuras de datos.

Por ejemplo:

Datos del socio = nombre + apellido + ... + dirección

La fuente indica el origen del flujo de datos, puede ser una entidad externa, un proceso, o un almacenamiento. El destino indica hacia donde se dirige el flujo de datos, puede, al igual que la fuente, ser una entidad externa, un proceso, o un almacenamiento.

EL ALMACENAMIENTO

Los almacenamientos son flujos de datos en reposo. Se representan de la misma forma que los flujos de datos. Por lo tanto tienen nombre, descripción, contenido y además se detallan los flujos de entrada y salida.

El flujo de entrada indica cuáles son los flujos que alimentan al almacenamiento. Esto implica una alteración del contenido. Puede ser una inserción, borrado o modificación de los valores componentes de la estructura.

El flujo de salida indica cuáles son los flujos que se extraen del almacenamiento. Esto implica, solamente, la lectura de los valores componentes de la estructura sin alterar el contenido de los mismos.

LOS PROCESOS

Los procesos son los transformadores de flujos de datos del sistema. Los procesos primitivos del diagrama de flujo de datos, serán descriptos, además, en forma más detallada mediante alguna de las herramientas de especificación. En general, se define el nombre, la descripción y los flujos de entrada y salida. Se deben asignar nombres a los procesos que sean significativos en el contexto del desarrollo del sistema. En general se recomienda que sean verbos más objeto sobre el cual éste actúe. Por ejemplo: Inscribir socio





La descripción indica, de manera breve, lo que este proceso representa para el sistema. Por ejemplo: proceso de inscripción de socios al establecimiento deportivo El flujo de entrada indica cuáles son los flujos que alimentan al proceso. El flujo de salida indica cuáles son los flujos que se salen del proceso.

LAS ENTIDADES EXTERNAS

Las entidades externas representan a los objetos externos al sistema en estudio, de tal manera que determinan sus fronteras. Pueden ser personas, organizaciones u otros sistemas. Se detalla mediante el nombre, la descripción y los flujos de datos asociados.

Se deben asignar nombres a las entidades externas que sean significativos en el contexto del desarrollo del sistema. Por ejemplo: DGI

La descripción indica, de manera breve, lo que la entidad externa representa para el sistema. Por ejemplo: Dirección General Impositiva

Los flujos de datos asociados indican cuáles son los flujos de entrada y salida asociados a esa entidad externa

RESUMEN

El diccionario de datos es una lista organizada de todos los componentes del sistema, con definiciones precisas y rigurosas para que, tanto el usuario como el analista, tengan un entendimiento común de todas las entradas, salidas, componentes de los almacenamientos y procesos intermedios. Complementa los demás modelos del sistema.