1. Caso Práctico

Caso Práctico

*En **SI Andalucía** se enfrentan al desarrollo de la aplicación para la empresa **Rokemar.** Ya han hecho el estudio de viabilidad, y el cliente ha aceptado una de las soluciones que le han propuesto, pero para ello ha sido necesario determinar minuciosamente qué debe hacer esa aplicación, para lo cual han tenido que elaborar un **análisis de sistemas y requisitos**. Básicamente han tenido que representar un **modelo** del sistema, que les permita entender mejor su funcionamiento, y entender qué debe hacer. En la definición de los requisitos han trabajado conjuntamente todas las partes involucradas en un desarrollo: los desarrolladores del software de la empresa **SI Andalucía**, a través de los analistas, que en este caso han sido **María** y **Víctor**, también los clientes, representados por el gerente de **Rokemar**, que es el que ha tomado algunas decisiones sobre el diseño que más le convenía a su empresa, y por último los usuarios finales del sistema, que son básicamente los comerciales de la empresa **Rokemar**. Como resultado han obtenido el documento de Especificación de Requisitos Software (ERS).*

***Víctor** y **María** se plantearon qué forma de representar la ERS usar, ya que tenían varias alternativas, según la presentación, y otras tantas según el enfoque de la modelización, que no eran del todo excluyentes. Tras darle algunas vueltas, decidieron que necesitaban usar una mezcla de ellas, ya que cada una resultaba más indicada para describir alguna parte concreta del sistema, pero lo que tienen claro es que van a tener que usar intensivamente técnicas gráficas, sobre todo Diagramas de Flujo de Datos (DFD)*

2. Introducción al análisis y diseño estructurado

Introducción al análisis y diseño estructurado

Esta unidad debemos comenzarla comprendiendo qué significa su nombre.

- El término **análisis**, aplicado a sistemas, significa
- descomponer el sistema en sus componentes
- para estudiar cada uno de ellos tanto como un ente aislado como en interacción con el resto.

Cuando hablamos de una fase del ciclo de vida, **el análisis consiste en producir un documento de especificación de requisitos que describa lo que el futuro sistema debe hacer, pero no cómo debe hacerlo, es decir que se trata de ver el problema, no la solución**. El análisis de los requisitos es el proceso del estudio de las necesidades de los usuarios para llegar a una definición de los requisitos del sistema, de hardware o de software.

La definición de los requisitos de un sistema debe ser el resultado del trabajo conjunto de todas las partes involucradas en un desarrollo: los **desarrolladores** del software a través de los analistas, los **clientes** y usuarios finales del sistema; obteniendo el documento de Especificación de Requisitos Software (ERS) objetivo de esta fase. El análisis de requisitos proporciona al analista una representación de la información, de las funciones y del comportamiento del sistema que se puede traducir en un diseño posteriormente.

Durante el análisis de requisitos, el analista crea modelos del sistema para entender mejor:

- el procesamiento **funcional** (análisis funcional que veremos en esta unidad),
- el **contenido** de la información (análisis de datos que veremos en la siguiente unidad) y
- el **comportamiento** del sistema (análisis de control que veremos en esta unidad). Los modelos servirán de pilar para el diseño del software, y como base para la creación de una especificación del software.

La tarea de modelización es de las más importantes en el análisis de requisitos ya que los métodos de análisis son

realmente métodos de modelización. Estudiaremos en esta unidad algunos de los métodos de modelización más utilizados en el análisis de sistemas (funcional y de control, la unidad siguiente será para el de datos).

3. Análisis de requisitos: modelización

Análisis de requisitos: modelización

Anteriormente hemos dicho que **la definición de los requisitos de un sistema debe ser el resultado del trabajo conjunto de todas las partes involucradas en un desarrollo: los desarrolladores del software a través de los analistas, los clientes y usuarios finales del sistema; obteniendo el documento de Especificación de Requisitos Software (ERS) objetivo de esta fase**.

Pero ¿cuál es la definición de **Especificación de Requisitos**?

Para responder a esta pregunta veamos cómo lo define la IEEE:

- La ERS es la documentación de requisitos esenciales (**funciones, rendimiento, diseño, restricciones y atributos**) del software y de sus interfaces [IEEE, 1999a]
- Una especificación es un documento que define, de forma completa, precisa y verificable, los requisitos, el diseño, el comportamiento u otras características de un sistema o componente de un sistema [IEEE, 1999a]

Por tanto **la ERS es la culminación de la tarea de análisis**, y debe ser validada por el usuario. Además debe cumplir una serie de carácterísticas deseables, que son:

- Incluir información veraz.
- Comunicar la información de forma eficaz.
- Expresar qué hacer y no cómo hacerlo.
- Describir los requisitos software.
- Excluir cosas innecesarias.
- No debe incluir detalles del diseño, de la verificación o de la dirección, salvo las restricciones de diseño que afecten a los requisitos.

Veamos ahora una **clasificación** de las técnicas de modelización para la ERS. Se pueden intentar clasificar bajo dos enfoques diferentes:

- Según la **forma de representación** pueden ser:
  - gráficas,
  - textuales,
  - marcos.
- Según el **enfoque de modelización** bajo el que se crean modelos del sistema en base a su:
  - función (qué hace el sistema),
  - información (qué información utiliza el sistema) y
  - tiempo (cuándo sucede algo en el sistema).

4. 1Clasificación según la forma de representación

Clasificación según la forma de representación

¿Cuántas formas de representar una ERS podemos usar según este criterio?

Según la forma de representación, vamos a tener tres formas de representar una ERS, y son las siguientes:

- **Gráficas:**

La técnica gráfica se basa en la utilización de una serie de iconos **que representan un componente de un aspecto particular del modelo**. Son el tipo de técnicas de modelización más útiles cuando es importante resaltar la conexión entre los distintos componentes del modelo.
No se utilizan por separado en la modelización de un sistema, sino en combinación con los restantes tipos de técnicas. Un ejemplo, que veremos más adelante son los diagramas de flujo de datos (DFD) que se utilizan en combinación con el diccionario de datos (DD) y las especificaciones de proceso.

- **Textuales:**
  Se utilizan para especificar, con más detalle, los componentes definidos en los gráficos mediante una gramática definida. Por ejemplo, el pseudocódigo.
- **Formularios (o plantillas):**
  Especifican la información relativa a un componente de un modelo que ha sido declarado en un diagrama o en otra plantilla. Se representan mediante un formulario en el que se incluyen todas las características del componente utilizando campos en el que cada entrada puede tener una gramática particular. Un ejemplo es el Diccionario de datos (DD).

5. Clasificación según el enfoque de modelización

Clasificación según el enfoque de modelización

Imagínate que vamos a representar un **sistema** informático. Como podrás entender no es lo mismo que ese sistema informático automatize las funciones de un cajero de un supermercado, o el control de una central nuclear. Los **procesos** serán más delicados y dependerán de factores muy diferentes. Cuando hablamos de modelos, como una representación de una realidad física, se suelen tener en cuenta tres factores principales

- La **función** que hace el sistema.
- La **información** que maneja o crea el sistema.
- El **tiempo** en que sucede algo en el sistema.

Por lo tanto sería recomendable examinar cualquier sistema bajo estas tres visiones. Cada una de ellas tiene un conjunto de técnicas que se utilizan con frecuencia:

- **Dimensión de la función**: el diagrama de flujo de datos (DFD), se utiliza para mostrar las funciones del sistema y sus interfaces. Las técnicas en las que se apoya son el diccionario de datos y las especificaciones de procesos.
- **Dimensión de la información**: el diagrama entidad/interrelación se utiliza para señalar las entidades y las relaciones entre ellas.
- **Dimensión del tiempo**: la lista de eventos y diagrama de transición de estados (DTE) se utiliza para mostrar cualquier cosa que ocurra y sobre la que el sistema debe responder.

Al **modelizar** un sistema se le suele dar más importancia a unas dimensiones que a otras, ya que éstas presentan características diferentes.

- Por **ejemplo**, para la construcción de un sistema basado en el mantenimiento de sistemas de gestión, como los de las entidades bancarias, que tienen una gran base de datos, la dimensión más afectada es la información.
- Otro **ejemplo** podría ser el desarrollo de un sistema de tiempo real como el control del núcleo en una central nuclear, en el que evidentemente la dimensión predominante es el tiempo. Una aplicación hecha a medida para la gestión de un comercio en concreto viene determinada preferentemente por la dimensión función.

En la siguiente tabla, aunque todavía no sepáis de qué estamos hablando, tenéis las distintas técnicas, y cómo se solapan entre unas dimensiones y otras.

Información	Función	Tiempo

	Diagramas de entidad-relación (E/R)		
Información	Diagrama de estructura de datos (DED)		
	Matriz entidad/entidad		
	Diagramas de flujo de datos (DFD)	Diagramas de flujo de datos (DFD)	
	Matriz función/entidad	Diagramas de descomposición funcional	
Función		Diagramas de estructura	
		Diagramas de flujo	
		Diagramas HIPO	
		Diagramas de Warnier/Orr	
	Diagrama de Historia y vida de entidad	Redes de Petri	Lista de eventos
Tiempo	Matriz evento/entidad	Diagrama de transición de estados (DTE)	Diagrama de transición de estados (DTE)

A partir de ahora vamos a estudiar con más detalle las distintas técnicas de análisis funcional y de análisis de control.

Para saber más

Si quieres aprender más sobre otras técnicas y enfoques para el análisis de requisitos, te recomendamos el siguiente enlace, basado en un Modelo de Proceso de la Ingeniería de la usabilidad y de la accesibilidad.

MPIu+a

6. Análisis funcional: modelo de flujo de datos y técnicas

Análisis funcional: modelo de flujo de datos y técnicas

Caso Práctico.

María *y* ***Víctor*** *saben que el DFD les va a dar una información muy valiosa, y representada de forma muy gráfica, sobre los procesos que intervienen en el sistema, y los datos que fluyen entre esos procesos, junto a las transformaciones que experimentan esos datos.*

Para elaborar el DFD del sistema que están informatizando para la empresa ***Rokemar*** *tienen que establecer antes que nada:*

- *las* ***entidades externas*** *con las que se va a comunicar el sistema,*
- *los procesos que se llevan a cabo,*
- *los almacenes de datos que se usan, y*
- *los flujos de datos que comunican unos procesos con otros.*

*Lo que se pretende de nuestro sistema es básicamente que **actualice** y extienda con nuevas funcionalidades un sistema informático que ya estaba implantado, y que gestionaba los presupuestos de las obras, las ventas de promociones terminadas, la información de los clientes y proveedores, y la contabilidad. Los **cambios** que solicita el cliente se refieren a introducir la posibilidad de que los trabajadores puedan acceder a las **funcionalidades** que ya tiene el sistema vía web, desde cualquier ordenador con conexión a Internet, y que los clientes de la constructora puedan consultar alguna información también mediante conexión web.*

*Por tanto, como **entidad externa** podemos considerar el sistema informático preexistente, con el que debe interactuar nuestro sistema actual. Los clientes y trabajadores de la empresa, que ya eran entidades externas del sistema anterior, también lo van a ser del nuestro.*

*Como **procesos**, básicamente los mismos que hacía el sistema anterior, pero realizados a través de Internet (presupuestos, ventas, información de clientes y proveedores, y contabilidad*

*Como **flujos de datos**, las ofertas a los clientes, las construcciones realizadas, los pedidos de los proveedores, los contratos de contraventa, las nóminas de los trabajadores, los asientos contables, y un largo etc. En definitiva, todos los movimientos de información que se deben manejar en la empresa para conseguir sus objetivos.*

Como almacenes de datos, los establecidos en la base de datos de la empresa.

Toda esa información la representan en un diagrama de flujo de datos. Naturalmente, de forma inmediata se dan cuenta de que necesitan detallar mucho más, y aplican el procedimiento de descomposición en niveles de ese DFD, detallando cada proceso en sus subprocesos, hasta llegar a procesos elementales, que no son susceptibles de subdividirse en más subprocesos. Cada uno de esos diagramas, dará lugar a un conjunto de definiciones de procesos, o miniespecificaciones, para las que usan diferentes técnicas, sobre todo un lenguaje estructurado tipo pseudocódigo, pero también han usado en algún caso alguna otra técnica (árboles de decisiones, diagramas de acción y tablas de decisión)

Para realizar un **análisis fucional** nos basamos en tres técnicas principales:

- Diagrama de flujo de datos (DFD).
- Diccionario de datos (DD).
- Especificaciones de procesos.

El diagrama de flujo de datos (DFD) es la técnica más difundida dentro del análisis estructurado. Esto es debido a la necesidad de los analistas de una técnica que les ayude a modelizar las funciones del sistema y los datos que fluyen entre ellas.

Un DFD es un diagrama en forma de red que representa el flujo de datos y las transformaciones que se aplican sobre ellos al moverse desde la entrada hasta la salida del sistema. Se utiliza para modelizar las funciones del sistema y los datos que fluyen entre ellas a distintos niveles de abstracción.

El sistema, se modelizará mediante un conjunto de DFD´s nivelados en el que los niveles superiores definen las funciones del sistema de forma general y los niveles inferiores definen estas funciones en niveles de una forma más detallada.

Aquí tienes un esquema de DFD, no te preocupes si no lo entiendes, ya que en seguida entraremos a conocerlo en detalle:

7. Componentes de un DFD

Componentes de un DFD

Bueno, ya conocemos la definición, y el esquema general de un DFD, ahora vamos a ir viendo cada uno de los componentes que lo forman, que son los siguientes:

- **Procesos:** son los componentes funcionales del sistema.
- **Almacenes:** representan datos almacenados o en reposo.
- **Entidades externas:** representan la fuente y/o el destino de la información del sistema.
- **Flujos de datos:** representan los datos que fluyen entre las funciones.

Veamos cada uno más detalladamente.

8. Procesos

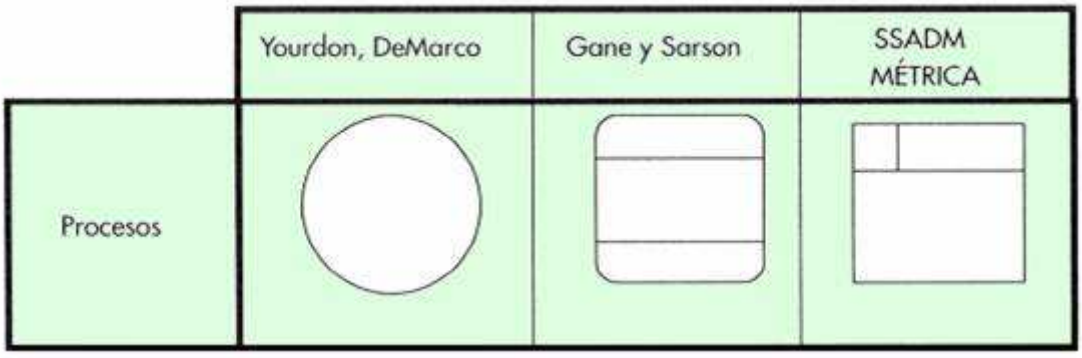
Procesos

El proceso es el componente del diagrama que representa una función que transforma los flujos de datos de entrada en uno o varios flujos de datos de salida. El término proceso no hay que tratarlo como un programa en ejecución sino como una función que tiene que realizar el sistema.

El **proceso** debe ser capaz de generar los flujos de datos de salida a partir de los flujos de datos de entrada más una información local (constante o variable) al proceso. Esto se conoce como la regla de conservación de datos.

Cuando al proceso no le llegan todos los datos necesarios para obtener los datos de salida diremos que hay un **error de conservación de datos**. Si esto ocurre significará que se ha olvidado incluir algunos datos de entrada. También podemos encontrarnos con el caso contrario, aquel en el que el flujo de datos o algún componente suyo no se utiliza para generar los flujos de salida, lo que se denomina pérdida de información.

Su representación gráfica puede variar según los autores.



En realidad el concepto de **representación** es el mismo usemos la notacion que usemos. Como nosotros debemos unificar criterios y usar una sola, nos decidimos por el uso de la representación de Yourdon, donde **la representación se realiza mediante un círculo**. En el interior del círculo, se incluyen un número y un nombre que debe cumplir las siguientes características:

- Ser lo más representativo posible de la función que representa. Por ello hay que intentar dar un nombre que englobe a toda la función y no sólo a parte de ella. Hay que suprimir nombres con poca significación tales como REALIZAR OPERACIÓN, GESTIONAR ACCIÓN, etc.

- Ser breve, normalmente formado por un verbo seguido de un sustantivo (por ejemplo: GENERAR ALBARAN).
- El nombre y el número del proceso deben ser únicos en el conjunto de DFD que representan el sistema.

9. Almacenes de datos

Almacenes de datos

El almacén de datos representa información del sistema almacenada de forma temporal. Si los flujos de datos representan datos en movimiento, los almacenes representan datos en reposo. El almacén es un depósito lógico de almacenamiento y, por tanto, puede representar cualquier dato temporalmente almacenado independientemente del dispositivo utilizado. En consecuencia, puede representar un cajón con papeles, un archivador manual, un fichero o una base de datos.

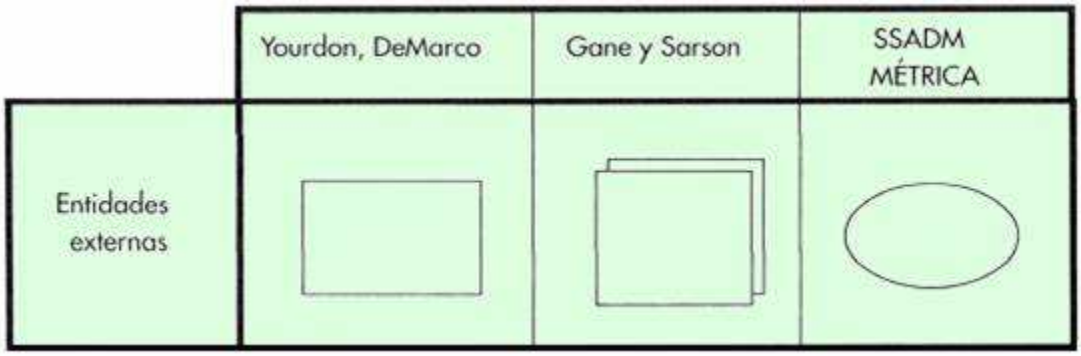
Al igual que antes, nosotros usaremos la notacion Yourdon. **Se representan por dos líneas paralelas**. Los almacenes tienen las siguientes características:

- Todos los almacenes de datos deberán llevar un **nombre**, que debe ser lo más representativo posible (de los datos que contienen)
- Un almacén de datos se puede **representar** varias veces en un DFD si con ello se mejora su legibilidad (indicándolo con un asterisco).
- Dentro de un conjunto de DFD nivelados, el almacén se situará en **el nivel más alto** de los que sirven de interconexión entre dos o más procesos en el que se representan todos sus accesos y además se representará en los niveles siguientes donde se usen.
- Si en un DFD hay un almacén que sólo tiene conexión con un proceso, se dice que el almacén es local a ese proceso y, por tanto, no debe aparecer en ese nivel. Dicho almacén se representará en el DFD en el que se especifique dicho proceso.
- Un almacén se dice que tiene estructura simple cuando es de tipo registro, esto es, está formado por una sucesión de atributos en el que uno o varios de ellos identifican cada ocurrencia del almacén. El contenido de los almacenes se define en el diccionario de datos.
- El contenido de un almacén con una estructura más compleja se puede representar mediante un diagrama entidad/interrelación. Quizás luego se convierta en un fichero físico.

10. Entidades externas

Entidades externas

Una entidad externa es el componente del DFD que representa un generador o consumidor de información del sistema y que no pertenece al mismo. Puede representar un subsistema, una persona, departamento, organización, etc. que proporciona datos al sistema o que los recibe de él.



Podemos considerar una serie de aspectos referentes a las entidades externas:

- Son, como su nombre indica, **externos al sistema** que se está modelizando. Los flujos que parten o llegan a ellas definen la interfaz entre el sistema y el mundo exterior.
- Las relaciones que haya entre las entidades externas no son objeto del estudio del modelo. Por tanto, no se representan los posibles flujos de información existentes entre ellas.
- **Se representan en el diagrama mediante un cuadrado** en el que se indica un nombre en su interior que debe ser representativo. Al igual que ocurría con los almacenes de datos, las entidades externas se pueden dibujar varias veces en un DFD con objeto de mejorar la legibilidad del mismo. Si esto es así, se puede marcar a las entidades duplicadas con un asterisco.
- Normalmente, las entidades externas sólo van a aparecer en el DFD de mayor nivel, que se llamará diagrama de contexto.

11. Flujos de datos

Flujos de datos

Podemos definir un flujo de datos como un camino a través del cual viajan datos de composición conocida de una parte del sistema a otra. Representan los datos en movimiento en un momento y con una cardinalidad determinados. Estos dos aspectos no se indican en el DFD.

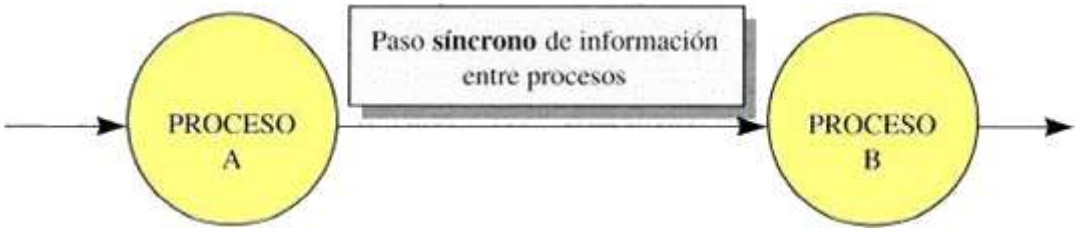
Los flujos de datos son el medio de conexión de los restantes componentes del DFD. **Se representan por arcos dirigidos, en donde la flecha indica la dirección de los datos**.

conexión entre los componentes de un DFD por medio de los flujos de datos existen una serie de restricciones. En la siguiente tabla vemos qué conexiones están permitidas:

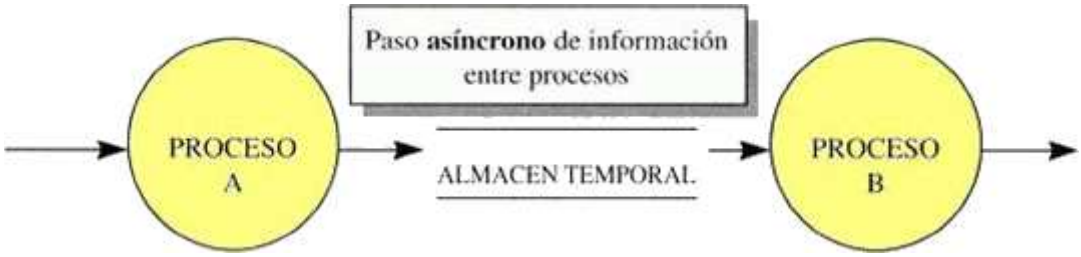
	Proceso	**Almacén**	**Entidad externa**
Proceso	**Si**	**Si**	**Si**
Almacén	**Si**	No	No

Externa	**Si**	No	No

La conexión directa entre dos procesos mediante un flujo de datos es posible siempre y cuando la información sea síncrona, es decir, que el proceso destino comienza en el momento en el que el proceso origen finaliza su función.



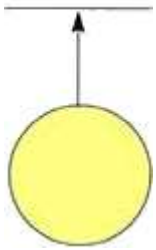
Si no ocurre así, es necesario que exista un almacén temporal que guarde los datos del proceso origen. Así, el proceso destino capturará estos datos cuando los necesite.



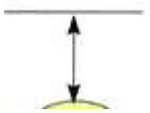
Las diferentes **conexiones** que se pueden hacer entre procesos y almacenes son:

- El flujo de consulta muestra la utilización de la información del almacén por el proceso para una de las siguientes acciones:

  - Utilizar los valores de uno o más atributos de una ocurrencia del almacén.
  - Comprobar si los valores de los atributos seleccionados cumplen unos criterios determinados.
- El flujo de actualización indica que el proceso va a alterar la información mantenida en el almacén para:



  - **Crear** una nueva ocurrencia de una entidad o interrelación existente en el almacén.
  - **Borrar** una o más ocurrencias de una entidad o interrelación.
  - **Modificar** el valor de un atributo.
- El flujo de diálogo entre un proceso y un almacén representa, como mínimo, un flujo de consulta y un flujo de actualización que no tienen relación directa.



Ejemplo:

Observar un proceso que se encarga de gestionar las peticiones de cd-rom´s de los usuarios de un videoclub. Hay un flujo de diálogo, ya que

- cuando un usuario realiza una PETICIÓN DE CD-ROM,
- el proceso GESTIONAR PETICIONES DE USUARIO consulta el almacén CD-ROM para ver si se encuentra disponible.
- Si no está el cd-rom en el videoclub se rechaza la petición,
- pero si el cd-rom está presente, el proceso actualiza el almacén indicando que tiene un ejemplar menos.
- Además actualizará el almacén PRÉSTAMOS (Flujo de actualizacion) para registrar el préstamo.
- Previamente realiza un flujo de consulta para comprobar que ese usuario no tiene penalizaciones por devolver cd´s fuera de plazo en el almacén USUARIOS

Como **resumen**, los flujos de datos tienen que tener las siguientes características:

- Deben **tener un nombre representativo** del contenido de la información que fluye a través de él. Todos los flujos de datos deben tener un nombre, excepto aquellos que entren o salgan de almacenes que tengan una estructura simple. En este caso, la estructura de estos flujos de datos es la misma que la del almacén.
- El **contenido de un flujo puede ser de varios tipos**:
- **Un flujo de datos se puede desdoblar** en varios flujos (cada uno con los mismos datos) o **puede aparecer repetido** varias veces en un DFD. Del mismo modo, varios flujos iguales se pueden unir en un único flujo
- **Los flujos de datos no indican el control de ejecución de los procesos ni cuándo va a comenzar o terminar de realizarse un proceso.**

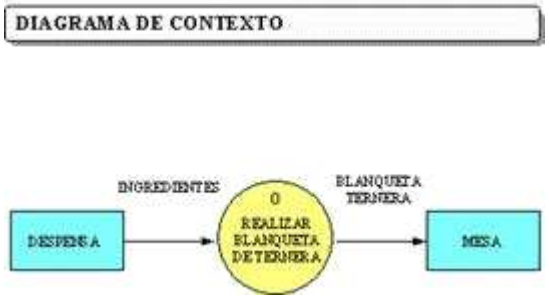
Para saber más

Si quieres empezar a practicar con los DFD´s aquí tienes una herramienta gratuita para ello:

Studio Case **[Versión en caché]**

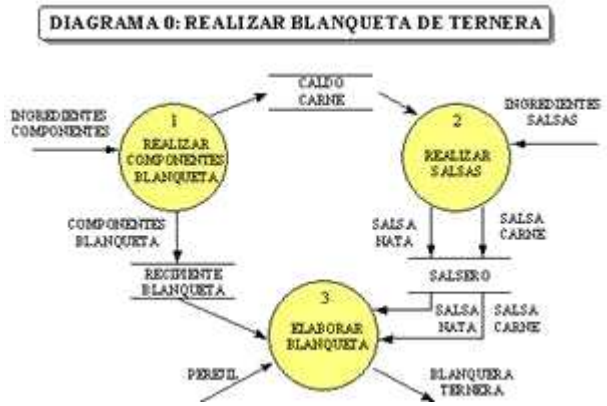
Como **ejemplo** para que practiques te proponemos que veas cómo se utiliza Studio Case en nuestro ejemplo de "la receta de la blanqueta de ternera" en las siguientes animaciones:

- En Primer lugar realizaremos el diagrama de contexto:



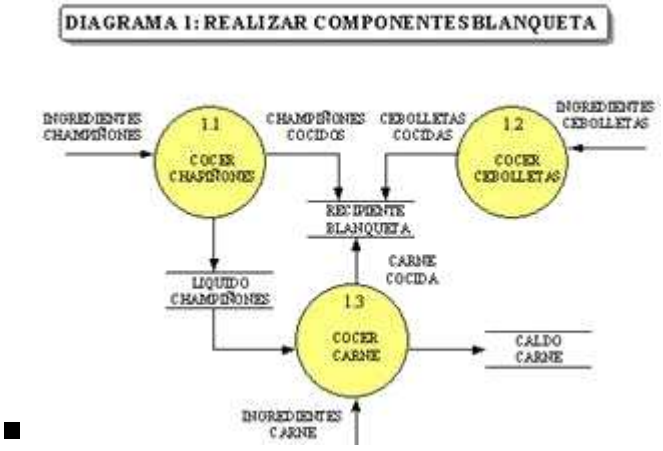
DEMO: Mira cómo se realiza el diagrama de contexto

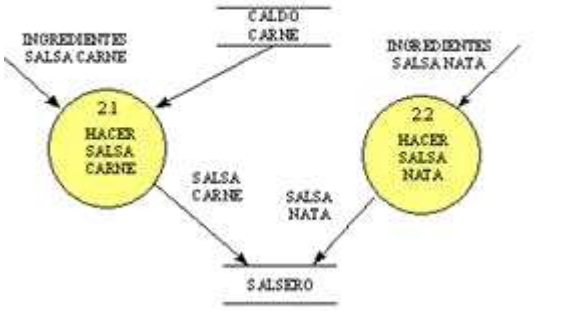
- En segundo lugar haremos el diagrama de sistemas

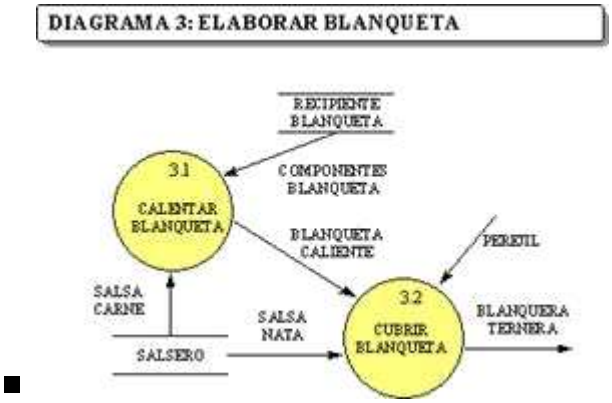


DEMO: Observa la creación del diagrama de sistemas

- Ahora llegó tu turno de practicar y podrías realizar los siguientes desarrollos







12. Descomposición en niveles de un DFD

Descomposición en niveles de un DFD

Como el modelo de un sistema grande no se puede representar en una sola página mediante un DFD, la idea es representarlo "por capas" (y cada capa queda definida mediante un DFD).

Se utiliza un esquema descendente (topdown) en el que cada nivel proporciona una visión más detallada de una parte definida en el nivel anterior.

Las ventajas de esta forma de estudio del sistema son las siguientes:

- Ayuda a construir la especificación de **arriba a abajo**. Esto es importante, ya que al comenzar el modelado de un sistema no se conocen todos los detalles sino sólo los aspectos más generales.
- Los distintos niveles pueden ir dirigidos a personas diferentes.
- Facilita el trabajo de los analistas que pueden trabajar paralelamente modelando funciones independientes del sistema.
- Facilita la documentación del sistema, ya que cada diagrama puede escribirse en una página.

Se comienza por el nivel más alto de la jerarquía mediante un DFD denominado diagrama de contexto. En este diagrama sólo hay un proceso que representa el sistema completo.

El diagrama de contexto en un DFD que es el primer diagrama de la jerarquía, también se le conoce como Diagrama de Nivel 0. El objetivo de este diagrama es delimitar la frontera entre el sistema y el mundo exterior, y definir sus interfaces, es decir, los flujos de datos de entrada y salida del sistema con el entorno, o lo que es lo mismo, el contexto.

El diagrama de contexto está formado exclusivamente por:

- un **proceso** que representa una "caja negra" del sistema completo,
- un conjunto de **entidades externas** que representan la procedencia y el destino de la información del sistema, y
- un conjunto de **flujos de datos** que representan los caminos por los que fluye dicha información.

En este diagrama es necesario que estén representadas todas las entidades externas y, según algunos autores, es el único diagrama en el que pueden aparecer.

A continuación, este diagrama de nivel 0 o **diagrama de contexto** se descompone en otro DFD que se denomina **diagrama del sistema** en el que se representan las funciones principales o subsistemas.

El diagrama en que se descompone el diagrama de contexto se suele denominar diagrama del sistema, porque en él se representan las funciones principales que debe realizar, así como la relación entre ellas (principales interfaces entre

estas funciones).

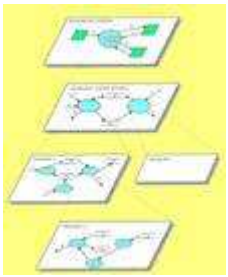
Es interesante que las funciones de este diagrama sean **conceptualmente independientes** entre sí, lo que facilita la descomposición de cada una por personas (analistas) diferentes. De esta forma, varios analistas pueden estar trabajando al mismo tiempo en el sistema sin que se solapen entre ellos las funcionalidades que tienen que modelar.

A continuación, se descomponen cada uno de los procesos en nuevos diagramas que representan funciones más **simples**. Se procede de la misma manera hasta que todas las funciones estén lo "suficientemente detalladas" como para que no sea necesaria la creación de nuevos DFD. **Los procesos primitivos** (o funciones primitivas) son aquellos procesos de un DFD que ya no se descomponen en más diagramas de nivel inferior. Es muy importante recalcar que por cada función primitiva habrá una especificación que la describa.

La decisión de no descomponer más es una responsabilidad del analista. Es por tanto una decisión subjetiva, aunque se pueden definir una serie de reglas que ayuden a la misma:

- Cuando un requisito funcional se puede especificar en menos de una página mediante un lenguaje de especificación o pseudocódigo (sea formal o no).
- Cuando los procesos del diagrama tienen pocos flujos de datos de entrada y salida.

De una manera gráfica, el DFD completo sería lo siguiente (Haz clic en la imagen para verla a pantalla completa):



Resumiendo, un conjunto de DFD queda definido por:

- **Diagrama** de contexto, único y en la parte superior.
- **Niveles** medios, formado por el resto de diagramas.
- **Funciones primitivas**, que están presentes tanto en los niveles intermedios como en los últimos niveles de la jerarquía y que se corresponden con procesos que no se pueden descomponer más, por ser ya básicos.

Un aspecto a tener en cuenta es la consistencia entre niveles, también llamado balanceo:

> **Es necesario comprobar la consistencia entre los distintos niveles de DFD, es decir, que la información que entra y sale de un proceso de nivel N, sea consistente con la información que entra y sale del DFD en el que se descompone.**

Para ello, se sigue la **regla del balanceo** entre niveles que se enuncia así:

- Todos los flujos de datos que entran en un diagrama hijo deben estar representados en el padre por el mismo flujo de datos entrando en el proceso asociado.
- Las salidas del diagrama hijo deben ser las mismas salidas del proceso padre.

La convención de la **numeración de diagramas y procesos** es la siguiente:

- Cada diagrama recibe el número y el nombre del proceso que descompone (proceso padre).
- El proceso del diagrama de contexto siempre es numerado como cero.
- Los procesos del diagrama del sistema se enumeran por un entero comenzando por 1 y de forma creciente hasta completar el número de procesos del diagrama.

- En los restantes niveles, los números de los procesos están formados por la concatenación del número de diagrama en el que están, más un punto y un número entero único para identificarlo dentro del diagrama.

Para saber más

Para saber más sobre DFD´s puedes ver algunos ejemplos resueltos:

Más sobre DFD **[Versión en caché]**

13. Recomendaciones en la creación de DFD

Recomendaciones en la creación de DFD

Ya hemos visto la técnica de creación de un DFD, pero quizás venga bien estudiar una serie de **recomendaciones generales** que os ayuden en esta tarea. El modelado de un sistema mediante un conjunto de DFD que tengan unas condiciones de calidad dadas no es una tarea fácil. Durante la construcción se realizan cambios continuos debido a dos razones principales:

- Normalmente la construcción se realiza mediante un proceso de **refinamiento iterativo** de los mismos. Es casi imposible crear los DFD correctamente en el primer intento.
- Al comenzar el modelado no se conocen todas las características del sistema. La inclusión de nuevos requisitos o los cambios en los existentes provocan la modificación de la mayoría de los DFD.

Las **recomendaciones** para la correcta creación de DFD´s son:

- Identificar las entidades externas.
- Dibujar los procesos.
- Dibujar los flujos de datos.
- Escoger nombres con significado.
- Evitar nombres de personas y papeles políticos.
- Etiquetar los procesos de forma que se identifiquen las funciones que lleven a cabo. Un sistema muy difundido es usar verbo + objeto. (Emitir Factura, por ejemplo)
- Deben provenir de un vocabulario con sentido para el usuario.
- Evitar utilizar nombres técnicos que no entiende el usuario.
- Numerar los procesos.
- Constancia a la hora de aplicar los números.
- El sistema de numeración puede parecer que implica una secuencia de ejecución, pero esto no es así. Un DFD es una red de procesos asíncronos que se comunican.
- Los números se convierten en base para la numeración jerárquica cuando existen niveles.
- Redibujar el DFD tantas veces como sea necesario.
- Debe ser técnicamente correcto. Es decir que cumpla todas las reglas de creación de DFD´s
- Debe ser aceptable por el usuario.
- Debe ser estéticamente aceptable.
- Evitar DFDs excesivamente complejos.
- Asegurarse que el DFD sea internamente consistente y que también lo sea con cualquier DFD relacionado con él.
- Evitar sumideros infinitos (que a un proceso sólo le lleguen flujos de entrada y ninguno de salida).
- Evitar procesos de generación espontánea (que de un proceso sólo salgan flujos de salida y no tenga ninguno de entrada de información).
- Cuidarse de no crear almacenes de sólo lectura o sólo escritura.
- Evitar redes desconectadas.
- No realizar Bucles
- Están prohibidos los Flujos entre almacenes de datos
- Están prohibidos los Procesos aislados (sin flujos de entrada o salida)

- Están prohibidos Flujos entre entidades externas

A continuación tienes un ejemplo de DFD completo:

Ejemplo de DFD

14. Diccionario de datos

Diccionario de datos

Podemos definir un diccionario de datos (DD) como una lista organizada de los elementos utilizados por el sistema que gráficamente se encuentran representados sobre el conjunto de DFD`s.

El DD se crea a la vez que los DFD durante el análisis del sistema. Las **entradas** deberán ser únicas para cada componente del DFD, es decir, habrá una entrada en el DD por cada componente que aparezca en el conjunto de DFD.

Veamos ahora cada uno de los elementos que lo forman:

15. Definición de flujos de datos

Definición de flujos de datos

La **definición de los flujos de datos** sigue una aproximación top-down. Las componentes son definidas, a su vez, mediante componentes más detalladas. Y se procede de esta forma hasta obtener partes indivisibles, es decir, datos elementales (atributos o campos). Por ejemplo, si sabemos que un flujo de datos A está compuesto por uno B y uno C, y que B está compuesto de B1, B2 y B3, mientras que el C es siempre C1 y C2, podríamos escribir la definición así:

A = B1 + B2 + B3 + C1 + C2

Pero para entender el significado del flujo de datos complejos es mejor definirlos en función de sus componentes subordinados. Estos componentes serán otras entradas al DD. Luego conviene definirlo así:

A = B + C; B = B1 + B2 + B3

Un flujo de datos (ya sea un grupo o flujo múltiple) se puede definir teóricamente mediante la inclusión, selección e iteración de sus componentes. Además se incluyen otros símbolos que aportan más significado a cada entrada del DD

Símbolo	Significado
=	**Composición**: esta compuesto de, o es equivalente a
+	**Inclusión**: y
[ ]	**Selección**: selección de una de las opciones encerradas entre corchetes, y separadas por el símbolo |
{ }	**Opción**: significa que el componente encerrado es opcional (puede estar presente o ausente)
Texto	**Comentario**: el texto entre asteriscos es un comentario aclarativo de una entrada del DD
@	**Identificador**: se utiliza para señalar un campo o conjunto de campos que identifican cada ocurrencia de un almacén

En los puntos de este mismo apartado vamos a ver ejemplos del uso de estos símbolos con su significado asociado.

Composición e inclusión

La composición o definición del flujo se introduce con el símbolo "=". Si tenemos un flujo de datos múltiple A, compuesto de un flujo B y uno C, la definición le representa así:

A = B + C

es decir, A está compuesto de B y C, o a cada ocurrencia de A le corresponde una ocurrencia de B y una de C.

Por ejemplo, si tenemos un cliente que realiza una petición de préstamo a un videoclub, debe entregar el carnet del videoclub y una ficha en donde se detallen las películas que pide, la definición del flujo queda representada:

```
PETICIÓN PELICULAS = CARNET VIDEOCLUB+FICHA PELICULAS
```

(es decir, a cada ocurrencia de petición le corresponde una ocurrencia de carnet y una ocurrencia de ficha de películas).

Supongamos que el carnet del videoclub indica una serie de campos asociados al cliente. Estos se definirían en el DD de la siguiente forma.

```
CARNET VIDEOCLUB = NUM. CARNET + APELLIDOS + NOMBRE + TIPO CARNET
```

Selección

La selección de un componente (sea un elemento o un conjunto de elementos) se representa entre corchetes, separando cada opción mediante el símbolo " | "

En el ejemplo anterior, podemos suponer que el cliente tiene uno de los diferentes tipos de carnet (joven, adulto, vip) en función de los cuales depende el número de días que puede disponer de las películas en préstamo. La definición quedaría así:

```
TIPO CARNET = [JOVEN | ADULTO | VIP]
```

Iteración

La iteración representa la repetición de los elementos incluidos entre llaves.

En el ejemplo, la ficha de películas está compuesta por una estructura repetitiva de películas de los que se recogen: el código, el título y el director. La definición queda así:

```
FICHA PELICULAS = { PELICULAS }

PELICULAS = CODIGO + TÍTULO + DIRECTOR
```

Es posible representar los límites inferior y superior sobre las ocurrencias de una estructura repetitiva. Si en el ejemplo el préstamo está restringido a cinco películas, se puede definir así:

```
FICHA PELICULAS = 1 { PELICULAS }
```

Opción

El dato opcional indica que puede o no estar presente. Por ejemplo, supongamos que en el carnet del videoclub se puede, opcionalmente, recoger el número de teléfono. Esto queda representado así:

```
CARNET VIDEOCLUB = NUM. CARNET + APELLIDOS + NOMBRE + TIPO CARNET + (NUMERO TELÉFONO)
```

16. Definición de almacenes

Definición de almacenes

Los almacenes se definen como entidades repetitivas de datos y/o grupos de datos. El analista o usuario selecciona uno o más datos para organizar la colección de entradas en un almacén, que se denomina identificador. Por **ejemplo**, un almacén que incluya los libros disponibles en una biblioteca, con los siguientes campos: signatura, título, autor y número de unidades, se define del siguiente modo:

```
LIBROS DISPONIBLES = @ SIGNATURA + TÍTULO + AUTOR + NUMERO UNIDADES
```

la signatura identifica cada ocurrencia del almacén.

A continuación tienes un ejemplo de DD completo:

DEMO: Ejemplo de DD

17. Alias (o sinónimos)

Alias (o sinónimos)

Muchas veces, al **modelar** un sistema, hay datos que se nombran de distinta forma y que, en realidad, representan el mismo dato. Esto se puede definir en el DD creando un **alias**, que es un sinónimo de una entrada del DD, ya se trate de un flujo de datos o de un elemento de datos.

No es conveniente su utilización, ya que se crean redundancias en la especificación estructurada. Sin embargo, cuando estamos realizando el modelo de un sistema, puede haber distintos usuarios en diferentes departamentos que llaman al mismo elemento de dos maneras diferentes.

Ejemplo: En los concesionarios suelen usar indistintamente el término "bastidor" o "chasis" para denominar al número de serie que trae el motor de fábrica.

18. Definición de procesos (miniespecificaciones):Introducción

Definición de procesos (miniespecificaciones):Introducción

La especificación de proceso (también denominada miniespecificación) es una técnica que define el procedimiento que realiza un proceso primitivo. Debe describir de una manera más o menos formal cómo se obtienen los flujos de datos de salida a partir de los flujos de datos de entrada, más una información local del proceso. Hay varias alternativas para describir este procedimiento:

- Lenguaje estructurado.
- Árboles de decisión.
- Tablas de decisión.
- Diagramas de acción.

En los siguientes apartados te presentamos ejemplos de cada una de estas alternativas.

19. Definición de procesos (miniespecificaciones): Lenguaje estructurado

Definición de procesos (miniespecificaciones): Lenguaje estructurado

Es un lenguaje formado por un subconjunto de palabras (del idioma elegido. español, inglés, etc.) de las que unas se utilizan para formar las **construcciones propias** de la programación estructurada (como la secuencia, repetitiva y alternativa, y otras

incluyen un conjunto de verbos que reflejan **acciones simples**.

Alternativa	**SI condición** **Bloque** **SI NO** **Bloque** **FIN SI**
Repetitiva	**MIENTRAS condición** **Bloque** **FIN MIENTRAS** **REPETIR** **Bloque** **HASTA condición**
Secuencia	**Formada por un conjunto de sentencias (bloque) y cada una de ellas puede ser una acción sencilla o una estructura de las anteriores**

Las acciones se especifican con un **verbo** como, por ejemplo, LEER, ESCRIBIR, BORRAR, ENCONTRAR, CALCULAR, VALIDAR, etc. Es habitual definir hasta un conjunto de 40 a 50 acciones en un lenguaje estructurado.

A continuación tienes un ejemplo:

DEMO: Ejemplo de definición de procesos

20. Definición de procesos (miniespecificaciones): Árboles de decisión

Definición de procesos (miniespecificaciones): Árboles de decisión

¿Qué es un **árbol** de decisión? ¿Cómo nos ayuda a definir un proceso?

> **Un árbol de decisión es "un modelo de una función discreta en la que se determina el valor de una variable y en función de su valor se lleva a cabo una acción".**

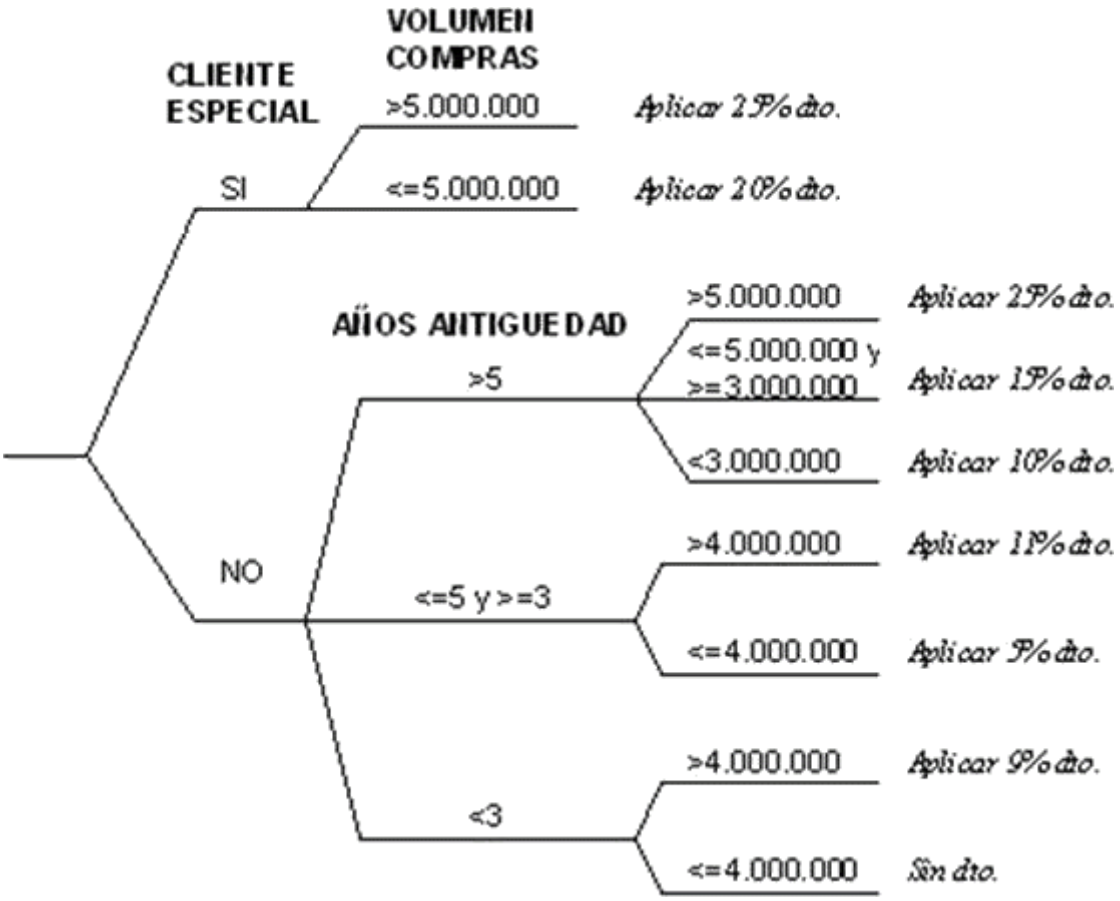
Es una representación en forma de **árbol** que representa:

- los valores de las **variables** y las **acciones** tomadas (que dependen del valor de la variable actual y de las acciones tomadas con anterioridad) para cada valor,
- así como el **orden** en el que se realiza la decisión. Se suele utilizar cuando el número de condiciones no es muy grande. En otro caso, es mejor utilizar una tabla de decisión.

Veamos un **ejemplo**:

Supongamos la política de descuentos que realiza una empresa sobre los pedidos de sus clientes dependiendo del volumen de compras del año anterior.

- Si se trata de clientes con más de 5 años de antigüedad se le aplica un descuento del 25% si el valor de los pedidos anuales es superior a 5.000.000 de unidades.
- Si el montante de los pedidos se encuentra entre los valores 3.000.000 y 5.000.000 de unidades, el descuento efectuado será del 15% y si no se alcanza la cifra de 3.000.000 de unidades, se aplicará el 10%.
- Para clientes entre 3 y 5 años de antigüedad se aplicará el 11% para compras por valor superior a 4.000.000 de unidades. y el 5% por valor igual o inferior.
- Si tienen menos años de antigüedad se aplicará el 9% si el valor de compras es superior a 4.000.000 de unidades.
- A los clientes clasificados como especiales se les aplicará un descuento de 25% si el volumen de compras supera los 5.000.000 de unidades. o del 20% en caso contrario.



21. Definición de procesos (miniespecificaciones): Tablas de decisión

Definición de procesos (miniespecificaciones): Tablas de decisión

¿Qué entendemos por **tabla de decisión**? ¿Cómo podemos hacer las miniespecificaciones usando tablas de decisión?

Una tabla de decisión es un modelo alternativo que muestra la función en forma tabular o matricial. Para ello, hay que definir la parte de condición, formada por un conjunto de condiciones y entradas de condiciones, y la parte de acción, formada por un conjunto de acciones y entradas de acciones.

Para el ejemplo anterior:

CONDICIONES	ENTRADAS DE CONDICIONES								
Cliente especial	SI	SI	NO	NO	NO	NO	NO	NO	NO
Compras > 5.000.000	SI	-	SI	-	-	-	-	-	-
Compras <= 5.000.000	-	SI	-	NO	-	-	-	-	-
3.000.000 <= Compras <= 5.000.000	-	-	-	SI	-	-	-	-	-
Compras < 3.000.000	-	-	-	-	SI	-	-	-	-
Compras > 4.000.000	-	-	-	-	-	SI	-	SI	-
Compras <= 4.000.000	-	-	-	-	-	-	SI	-	SI
Años antigüedad > 5	-	-	SI	SI	SI	-	-	-	-
3 <= Años antigüedad <= 5	-	-	-	-	-	SI	SI	-	-
Años antigüedad < 3	-	-	-	-	-	-	-	SI	SI
ACCIONES	**ENTRADAS DE ACCIONES**								
Aplicar 25% descuento	X		X						
Aplicar 20% descuento		X							
Aplicar 15% descuento				X					
Aplicar 11% descuento						X			
Aplicar 10% descuento					X				
Aplicar 9% descuento								X	
Aplicar 5% descuento							X		
Sin descuento									X

22. Definición de procesos (miniespecificaciones): Diagramas de acción

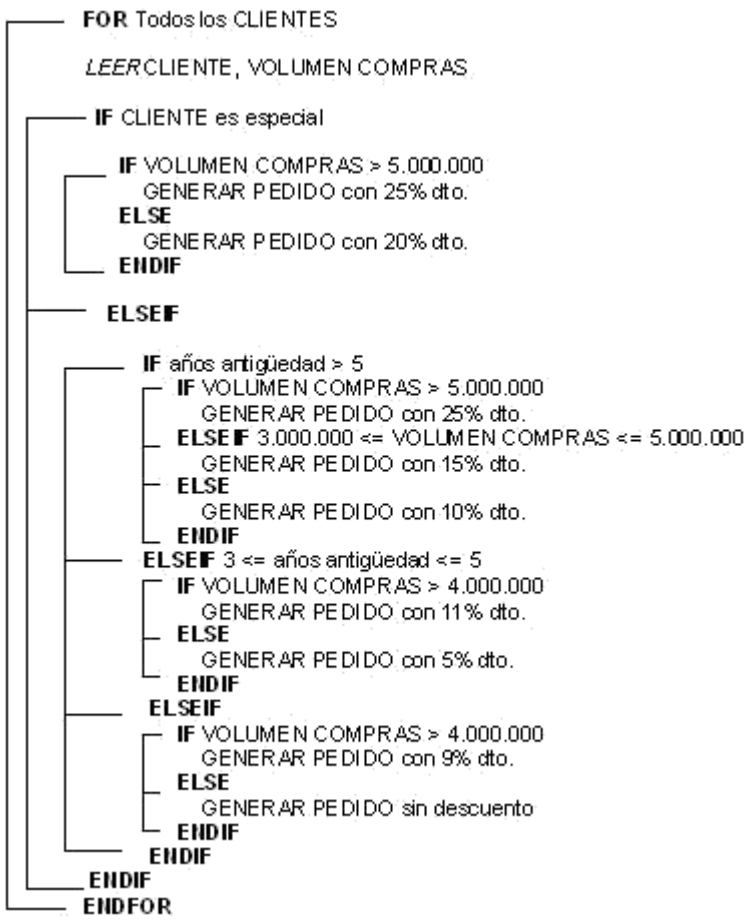
Definición de procesos (miniespecificaciones): Diagramas de acción

Ahora vamos a ver el último método de definición de procesos: Los **diagramas de acción**.

¿Qué es un diagrama de acción?

Un diagrama de acción es una técnica de especificación que utiliza niveles anidados de corchetes que representan la estructura lógica utilizada para transformar los datos de entrada en los datos de salida. Una ventaja de estos diagramas es que son útiles en todo el ciclo de vida. Durante la fase de análisis se preparan de una manera bastante general para especificar sólo las normas de transformación de datos de entrada en datos de salida. Durante la fase de diseño podremos detallar más estos diagramas.

Observa la definición del procedimiento de asignación de descuentos del ejemplo anterior.



Para saber más

Para saber más sobre especificación de procesos:

Más sobre especificación de procesos [Versión en caché]

Si quieres practicar con otra herramienta para crear DFD´s:

Easy Case

Disco 1

Disco 2

23. Análisis de control: modelo de especificación de control y técnicas

Análisis de control: modelo de especificación de control y técnicas

Caso Práctico.

Víctor *le pregunta a* ***María*** *si también es necesario hacer un modelo de especificación de control para apreciar algunos aspectos del comportamiento del sistema, sin quedarse sólo en un punto de vista funcional. Sabe que en este sentido las listas de eventos o los diagramas de transición de estados pueden proporcionarle ese punto de vista.*

María *le responde que ese análisis es complementario al que ya han realizado, y que la complejidad de su aplicación no*

lo hace necesario, en gran medida gracias a que ya se basa en un sistema preexistente, que facilita la comprensión del comportamiento del sistema, tanto para ellos como analistas y desarrolladores, como para los trabajadores como usuarios y para el gerente de la empresa, como cliente suyo. Por tanto, ése es un trabajo que no van a tener que desarrollar.

Al especificar un sistema se puede apreciar, además de aspectos funcionales o de datos, una **perspectiva de control** (o compartamiento del sistema).

Para describir este aspecto se requieren **técnicas adicionales** como el análisis de eventos, o los diagramas de transición de estados. A continuación vamos a describir de forma resumida estos dos conceptos, ya que su profundización no es materia de este módulo aunque si es conveniente que sepáis a grandes rasgos de que se trata.

24. Lista de eventos (sucesos)

Lista de eventos (sucesos)

Un evento es algo que ocurre en el mundo real y causa un cambio en la base de datos, es decir, causará actualizaciones en una o más entidades. Generalmente, aunque no siempre, se muestra el disparador como un flujo de datos que entra en el sistema. El evento no se refiere al procesamiento en sí mismo, sino a lo que causa el procesamiento en el mundo real.

Los **eventos** (o sucesos) pueden ser de tres tipos:

- **Generados externamente**. Se representan por medio de un flujo que entra en el sistema. Por **ejemplo**, en la llegada del evento petición de libro por parte del usuario, activa el proceso de gestionar peticiones de usuario, dando lugar a la creación de una nueva ocurrencia de préstamos.
- **Reconocidos internamente**, cuando, por ejemplo, se supera un valor umbral. En este caso se tendrá un disparador inicial que será un flujo que atraviesa el sistema, pero que por sí solo no es suficiente como para ser llamado un evento reconocido internamente, por ello es necesario que la base de datos esté en una determinada situación, y se representa por un flujo que va del almacén de datos al proceso. Por **ejemplo**, en el caso anterior si suponemos que es necesario que exista al menos un ejemplar del libro en la biblioteca, entonces nos damos cuenta de que la petición de libro por sí sola no es suficiente para desencadenar el proceso de peticiones de usuario, sino que además necesitamos tener la confirmación del almacén libros de que el libro está en stock.
- **Basados en el tiempo**, como, por ejemplo, la realización de facturas una vez al mes. En este caso se representa como un flujo que va del almacén de datos al proceso como el único disparador de ese proceso.

Los eventos se obtienen de los DFD, es decir, el nivel de DFD más bajo nos muestra los eventos. A efectos prácticos, la relación normal es un evento por cada función, aunque puede ser que una función tenga asociados más de un evento. Por tanto, cuando se nombran los eventos no se debe copiar el nombre del proceso o función del DFD que lleva la actualización.

25. Diagramas de transición de estados

Diagramas de transición de estados

El diagrama de transición de estados (DTE) es una técnica de modelado que se enfoca en el comportamiento dependiente del tiempo de un sistema. Normalmente se utiliza para representar el comportamiento de sistemas de tiempo real en el que el software debe responder a sucesos del mundo real en un tiempo muy limitado.

Los componentes de un DTE son :

- El **estado**, que representa un modo externo de comportamiento. El nombre del estado es el nombre del comportamiento exhibido por el sistema. Uno de los estados de un DTE será el inicial, que se representa con una

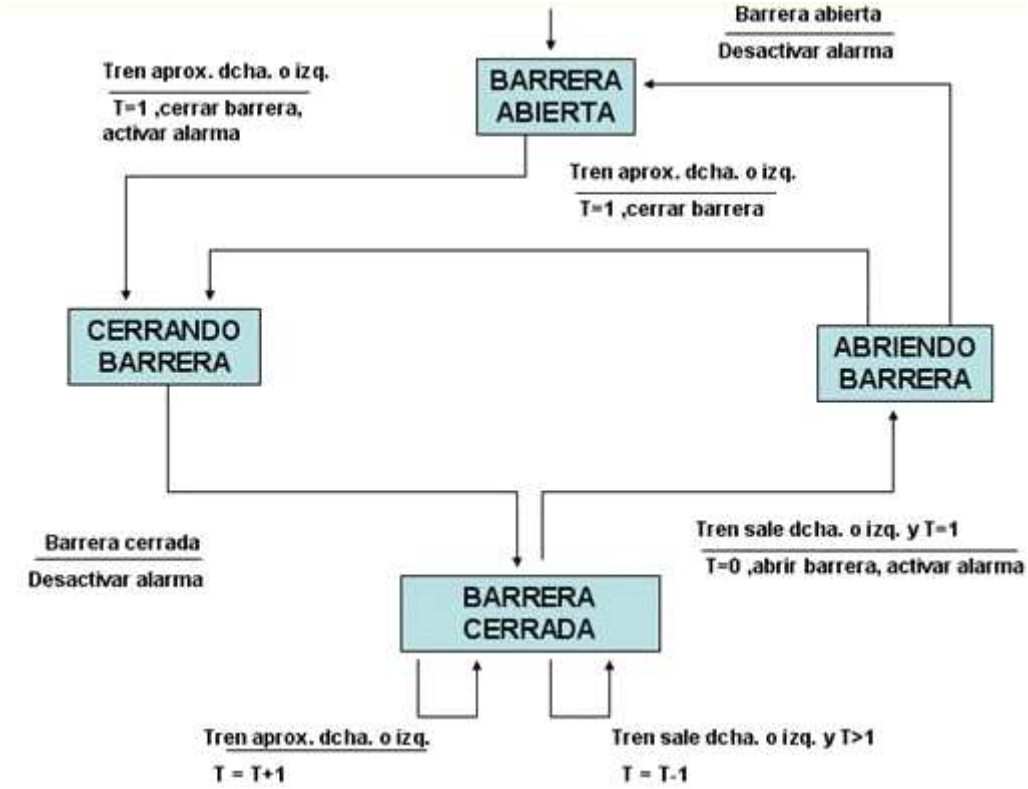
flecha de transición apuntando hacia él y sin ningún estado origen. Esta flecha puede tener, o no tener, condiciones o acciones. E1 estado inicial puede considerarse como el comportamiento del sistema antes de que haya ocurrido ninguna transición. En un DTE puede haber uno o varios estados finales.

- La **transición**. que obliga al paso de un estado a otro (o bien al mismo estado) si se cumple una condición. Determina la ejecución de una o más acciones sencillas, asumiendo que se producen simultáneamente. Se representa por una flecha entre dos estados.

Una aplicación muy interesante de los diagramas de transición de estados es su utilización para especificar transformaciones de datos y de control. Las **transformaciones** son elementos básicos utilizados en metodologías de modelado de sistemas en tiempo real. Su trabajo es describir, de una forma gráfica y sencilla, el comportamiento de un sistema.

Una **transformación** de datos simboliza, mediante un círculo, la función que se realiza para generar unos flujos de datos de salida (flechas que parten del círculo) a partir de unos flujos de datos de entrada (flechas que apuntan hacia el círculo). E1 flujo de salida de una transformación de datos puede convertirse en el de entrada para otra transformación de datos.

Una **transformación de control** es aquélla que sólo admite flujos de control como entradas y salidas (por ejemplo, una orden de abrir un dispositivo). La transformación de control sólo genera unas señales a partir de otras, no transforma datos. La notación utilizada son **círculos** (para las transformaciones de control) y **flechas discontinuas** (para los flujos de control). La salida de una transformación de control puede convertirse en entrada para otra, ya sea de datos o de control.



Éste es un ejemplo sencillo, muy simplificado, en el que vemos la especificación de una transformación de control mediante un diagrama de transición de estados. Se trata de representar el comportamiento de un controlador de paso a nivel. El acercamiento o alejamiento de los trenes se detecta por unos sensores situados a una cierta distancia. Cuando llega un tren se cierra la barrera de cada lado de la calzada. Cuando se detecta la salida de todos los trenes, se abre la barrera. Además, hay que accionar una alarma cada vez que se abra o cierre la barrera para alertar a los automovilistas.

Para saber más

Para saber más sobre DTE´s:

Más sobre DTE's [Versión en caché]

Para finalizar te presentamos una serie de ejercicios con su solución para que practiques con los DFD's:

- **Biblioteca**
- **Concesionario**
- **Hogar Seguro**
- **Omniremote**