Universidad Nacional del Litoral

Facultad de Ingeniería y Ciencias Hídricas

Departamento de Informática

Ingeniería de Software I

Realización del Diagrama de flujo de datos

Msc. Lic. Hugo Minni

2020

Realización del Diagrama de flujo de datos

El Diagrama de Flujo de Datos es uno de los tipos de diagramas que utilizamos para representar la realidad.

Dejan ver claramente los **procesos** de un sistema y los **flujos de información** entre ellos.

Algunas de sus características:

A. Es una herramienta de análisis, no de diseño, aunque permite descubrir y tal vez corregir errores de funcionamiento de una organización.

B. Estamos modelizando SISTEMAS DE INFORMACIÓN. **No sistemas físicos.** Aún cuando la implementación final no sea informática, se trata igualmente de un sistema de información.

C. Es fácilmente comprensible por el usuario final.

D. Es una herramienta que nos permite enfrentar nuestras limitaciones de abstracción.

E. A pesar de su nombre, se concentra en el flujo de información, no de datos.

F. No representa:
- Tiempo
- Operaciones simultáneas y/o concurrentes.
- Bifurcaciones lógicas o estructuras repetitivas.

En adelante usaremos la sigla **DFD** para hacer referencia a Diagrama de Flujo de Datos.

1. Procesos

En teoría de los sistemas, un proceso es una "actividad" que transforma determinado flujo de información de entrada en un flujo de información de salida.

Para que el proceso sea necesario, o tenga sentido, el proceso debe llevara cabo algún tipo de **transformación** entre el flujo de entrada y el flujo de salida.

DFD

En la simbología de [Yourdon], representamos un proceso a través de un círculo:

Todo proceso debe tener un nombre.

Como un proceso es una actividad, su nombre debe ser una frase que describa de la manera más exacta posible la actividad que lleva a cabo.

2. Límite del sistema

En teoría de los sistemas, todo sistema posee un límite. Este concepto es crucial.
La realidad es compleja, y, en nuestro trabajo, siempre estaremos abordando una porción asequible de ella. Esto es, por sentido común nos daremos cuenta de si nos es posible o no abordar una porción de la realidad.

La dimensión de lo que abordemos está determinada por el lugar donde pongamos los límites del sistema. Todo lo que quede adentro, será nuestra entera responsabilidad. Lo que quede afuera, será responsabilidad de otros sistemas.
Muchas veces se habla de lo que es "relevante" para el sistema. Si es relevante, quedará dentro. Si no es relevante, quedará fuera.

Ejemplo 1

Por ejemplo, un comercio ***A*** se relaciona, por ejemplo con el *cliente1*, el *cliente2*, el *cliente3*, el *proveedor1*, el *proveedor2* y la AFIP ((*Administración Federal de Ingresos Públicos*). Si debemos modelizar la realidad de ese comercio, está bastante claro que no modelizaremos la realidad de *cliente1*, *cliente2*, *cliente3*. O sea, no nos interesa, por ejemplo, como manejan sus finanzas.

Tampoco nos interesará modelar la realidad del *proveedor1* o el *proveedor2*. Por ejemplo, no nos interesará saber si poseen sus impuestos en regla.

Tampoco nos interesará modelar la realidad de la AFIP. Es un ente gubernamental y simplemente nos comunicaremos con él para proporcionar o recibir información (siempre que sea necesario o la legislación nos lo requiera).

La definición de los límites determina qué componentes de mi diagrama serán procesos y que otras pasarán a ser **entidades externas**.

3. Entidades externas

Una entidad externa es, como su nombre lo indica, una entidad, un objeto de la realidad, que se vincula de alguna manera con mi sistema pero está fuera de él.
Sin embargo, nos vinculamos con esa entidad de alguna manera, ya sea enviándole o recibiendo de ella información, y es por eso que debemos tenerla en cuenta.

Una entidad externa está claramente fuera del sistema a modelar. No nos interesa como funciona, ni qué hace con la información que le enviamos ni como produce la información que nos proporciona.

DFD

En la simbología de Yourdon, representamos una entidad externa a través de un rectángulo:

3.1. Límites del sistema y entidades externas

La decisión de donde ubicar los límites de un sistema define sistemas diferentes. Veremos un ejemplo:

Ejemplo 2

Consideremos un sistema que modeliza una fábrica de autopartes. La distribución de las mismas una vez fabricadas se realiza de manera terrestre y es responsabilidad de una entidad externa, a la que llamamos Flota de distribución:

Podría darse el caso de que la empresa crezca y vea conveniente hacerse cargo también de la distribución terrestre de sus productos. Si este fuese el caso, el sistema sería otro:

4. Flujos de información

Como dijimos antes, un DFD representa flujos de información, no de datos.
¿Cuál es la diferencia?

4.1. Ítems de datos

Un ítem de dato es, por ejemplo, **apellido**.
Es un dato atómico.

4.2. Información

Información es un **conjunto de ítems de datos** que tienen una significación semántica para que determinado **proceso** pueda llevar a cabo su función.

Por ejemplo: "*Información para inscripción a carrera*"

Esta información está constituida por un **conjunto de ítems de datos**, pero los mismos están *preparados*, *tienen sentido* y *son necesarios* como entrada a un proceso llamado *Inscripción a carrera*.

Esto concepto es crucial en Teoría de los Sistemas.

Un sistema hace uso de los datos agrupándolos de acuerdo a sus necesidades en **conjuntos de información que nutren a sus procesos**.

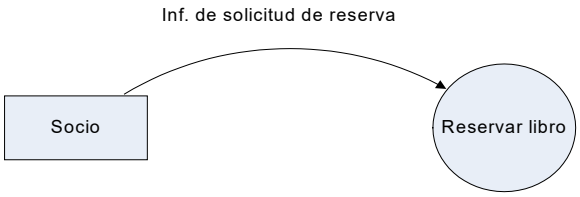
De manera análoga, los procesos de nuestro sistema darán como resultados también flujos de información que serán útiles para alguien u algo (otros procesos o entidades externas).

Por ejemplo, Resultados definitivos es un flujo de información que puede ser útil a una entidad externa.

DFD

Flujo de información

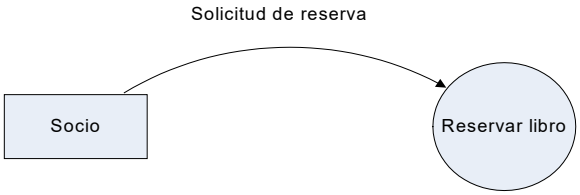
Los flujos de información se representan por un segmento dirigido, identificado por un nombre:



Algunos autores permiten flujos de información bidireccionales. En la práctica de DFD **no utilizaremos flujos de información bidireccionales**.

Al principio dijimos que en los DFD representaban sistemas **de información** (no "físicos").
Sin embargo, en la práctica es normal confundirse y creer que estas líneas pueden representar el tráfico de algo físico, o –peor aún- representar alguna acción o "control".

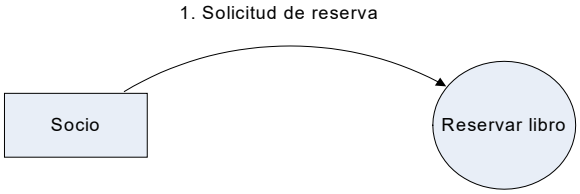
En el siguiente ejemplo "*Solicitud de reserva*" no significa una acción, sino el flujo –entre la entidad externa *Socio* y el proceso *Reservar Libro*, de un flujo de información llamado *Solicitud de reserva*[1].



DFD

Durante la realización de la práctica, y para mantener más claros los gráficos, también proporcionaremos un número a cada flujo de dato.

Así, cuando tengamos que redibujar una parte del diagrama bastará con que volvamos a escribir su número. Por ejemplo:



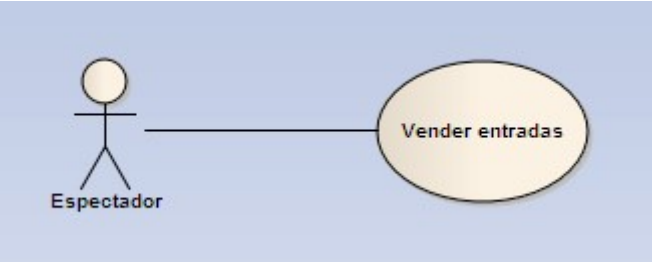
[1] Para que esto quede claro a veces es conveniente agregar como parte del nombre del flujo de datos las palabras *"Información de..."* .

5. Soportes de información

Dijimos que un flujo de datos no es el tráfico de algo físico. Sin embargo hay elementos físicos que sirven de soporte a la información.

Supongamos que tenemos que modelizar la venta de entradas en un teatro:

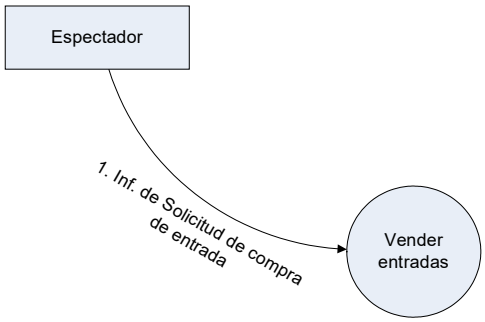
Tenemos un *caso de uso* llamado **Vender entradas**, con un Actor llamado *Espectador*:



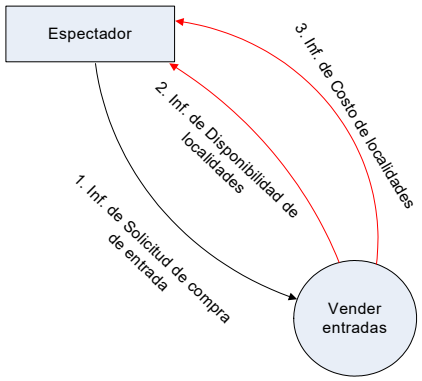
Supongamos ahora que tenemos que confeccionar un DFD que represente esta venta de entradas en un teatro.
Seguramente tendremos una narrativa, que describe que un espectador se presenta a comprar una entrada.

En el DFD, su sola presentación o requerimiento en la boletería disparará el funcionamiento del sistema.

O sea, su requerimiento **ya es información**.
Sería "*Información de solicitud de compra de entrada*".



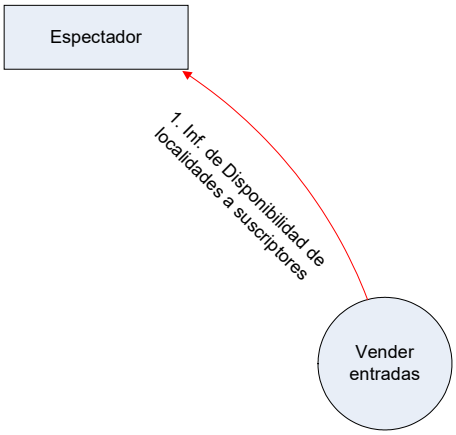
Seguramente el vendedor le indicará verbalmente la disponibilidad o no de entradas y el precio de la misma. Esto **también es información**.



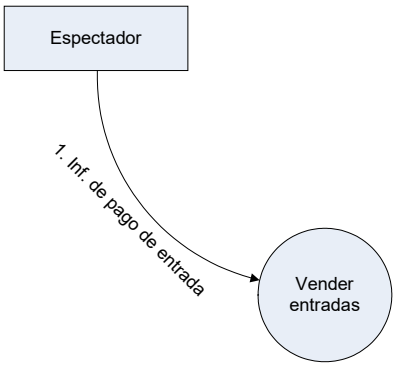
Esta información es verbal. No se guarda ni se transporta en ningún soporte físico. Sería algo así como "*Información de Disponibilidad de localidades*" e "*Información de Costo de localidades*".

A veces vale la pena representar este nivel de detalle en el intercambio de la información. A veces no. Se trata de un balance entre nivel de detalle y claridad del diagrama. En este caso no parece relevante.

Sin embargo, si el teatro quisiera enviar por correo la disponibilidad de localidades a un evento a clientes suscriptos, sí sería relevante representar este flujo de información.



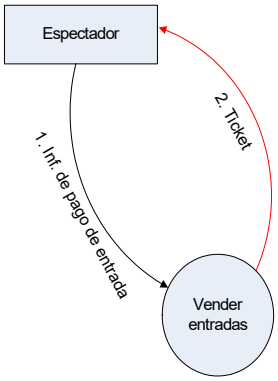
Supongamos que el Espectador desea adquirir la entrada. Debe abonar por la misma. Esto **también es información**.



En este caso, hay un flujo físico entre el Espectador y el vendedor, pero **no es eso lo que representamos**.
Lo que representamos es la información que ingresa al sistema de que está ingresando determinada cantidad de dinero.

Como contrapartida, el espectador recibe un ticket o factura:

Esto también es información. Y en este caso, viaja en un formato físico (papel).



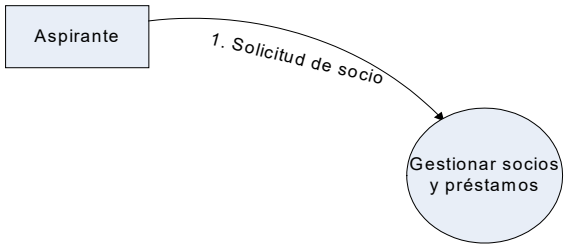
Cuando elaboramos un DFD, otros ejemplos de flujo de información son llamadas telefónicas, emails, formularios, etc.

6. Origen y destino de un flujo de información

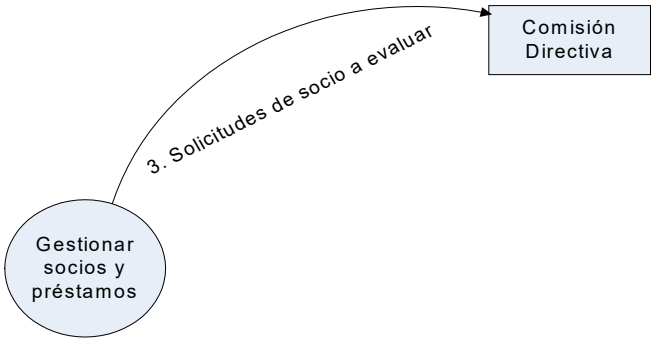
El único Nivel de DFD que posee almacenamientos es el último.

En todos los demás niveles, los orígenes y destinos que tienen sentido para un flujo de información[2], son los siguientes:

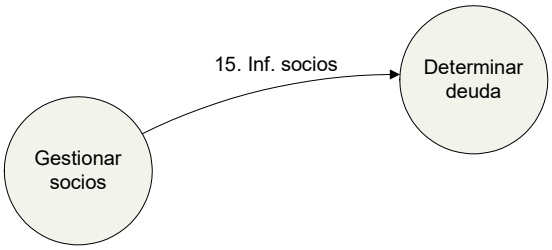
Un flujo de información puede tener como origen una entidad externa y como destino un proceso:



...o como origen un proceso y como destino una entidad externa:



...o como origen un proceso y como destino un proceso:



[2] *Usaremos "flujo de datos" o "flujo de información" de manera intercambiable. Sin embargo, debemos tener en claro que lo que representamos circulando siempre es información.*

7. Nivel 0

Como vimos, un DFD es una herramienta a través de la cual vamos haciendo una especie de zoom sobre la realidad. Cada "grado de zoom" se denomina Nivel.

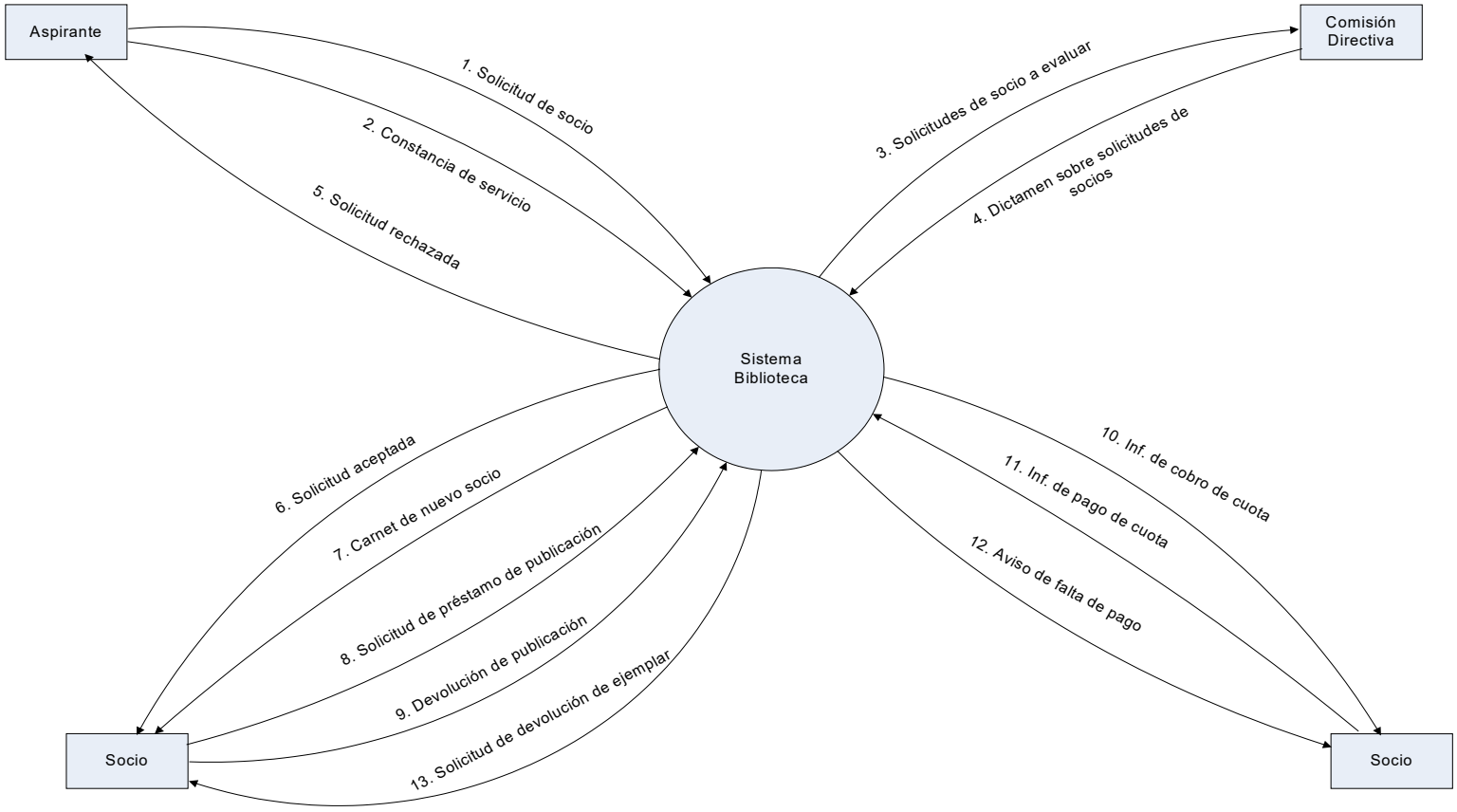
Al nivel 0 se le llama Nivel Inicial. Aquí el sistema se ve como una "caja negra". Vemos que entra y que sale, pero no se vislumbra cómo funciona.

En este nivel, en el diagrama se representa:

- Un único proceso (nuestro sistema)
- Las entidad externas
- El intercambio de información entre nuestro sistema y esas entidades externas:

Excepcionalmente en este nivel cero aceptamos darle como nombre a nuestro sistema una frase no verbal, que represente la función del sistema en si.
Podría ser "Sistema Biblioteca" o algo similar.

Nivel 0


Aspirante
Comisión Directiva
Sistema Biblioteca
Socio
Socio
1. Solicitud de socio
2. Constancia de servicio
3. Solicitudes de socio a evaluar
4. Dictamen sobre solicitudes de socios
5. Solicitud rechazada
6. Solicitud aceptada
7. Carnet de nuevo socio
8. Solicitud de préstamo de publicación
9. Devolución de publicación
10. Inf. de cobro de cuota
11. Inf. de pago de cuota
12. Aviso de falta de pago
13. Solicitud de devolución de ejemplar

8. Desagregación

Una vez que tenemos el Nivel 0 listo, se lleva a cabo un proceso llamado **desagregación**.

Lo que hacemos es "explotar" el proceso inicial en subprocesos.

Si bien no hay una regla, en la realización de la práctica, no generaremos más de 2 o 3 procesos a partir de un proceso origen.

Como resultado de esta desagregación suceden varias cosas:

A. Se mantienen sin cambios las entidades externas

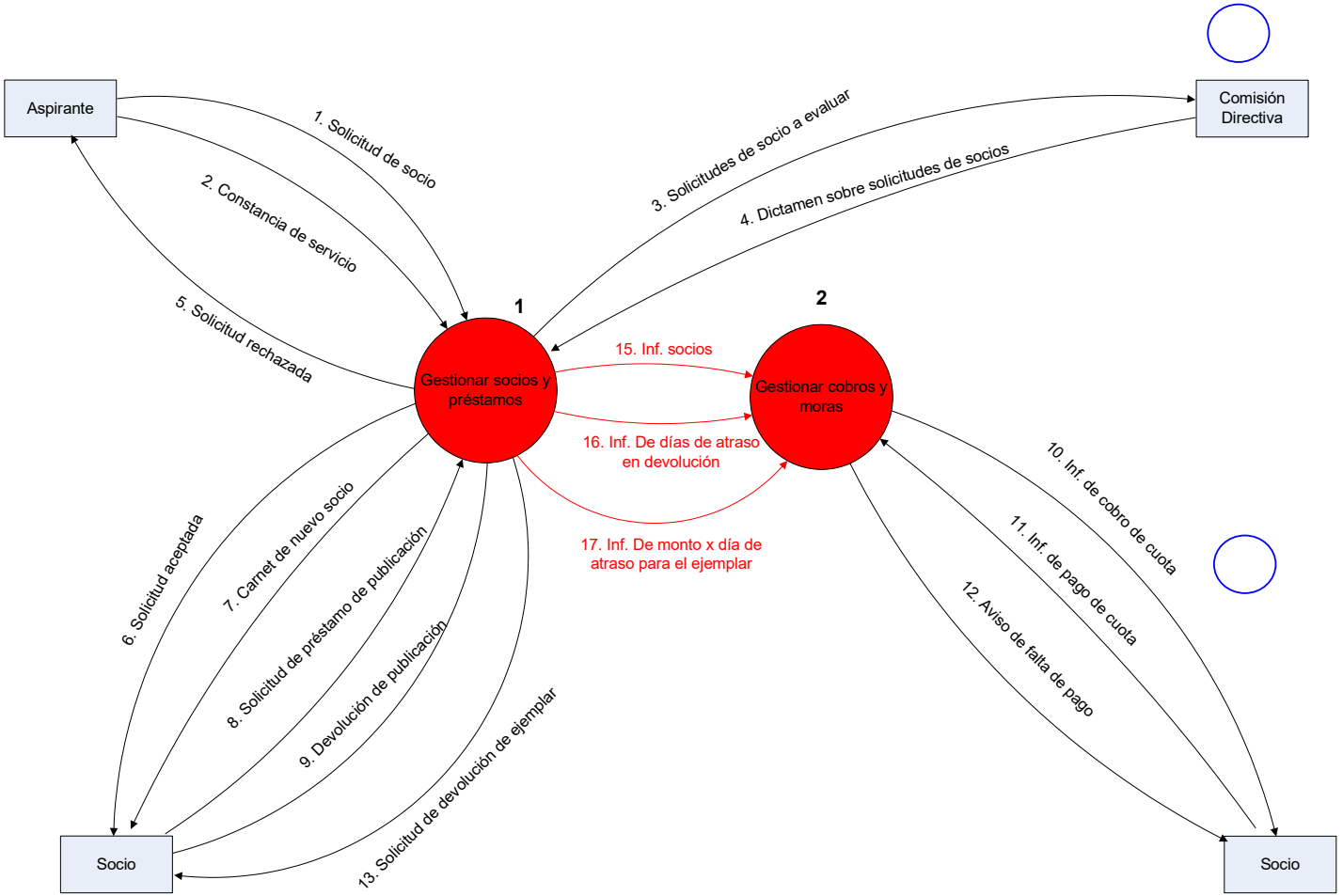
B. Se mantienen sin cambios los flujos de información entre nuestro sistema y las entidades externas.

C. Se agregan los subprocesos que obtenemos (y desaparecen los originales).

D. Se agregan flujos de datos que intercomunican a estos nuevos procesos.

Así, por ejemplo, obtenemos un **Nivel 1**:

Nivel 1

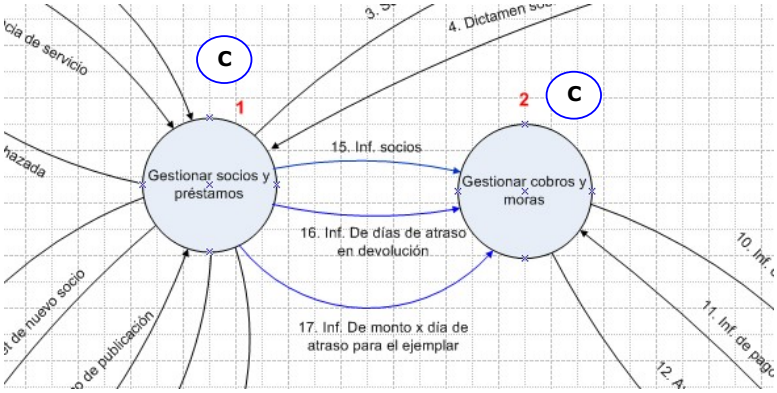

Aspirante
Comisión Directiva
Socio
Socio
1
2
Gestionar socios y préstamos
Gestionar cobros y moras
1. Solicitud de socio
2. Constancia de servicio
3. Solicitudes de socio a evaluar
4. Dictamen sobre solicitudes de socios
5. Solicitud rechazada
6. Solicitud aceptada
7. Carnet de nuevo socio
8. Solicitud de préstamo de publicación
9. Devolución de publicación
10. Inf. de cobro de cuota
11. Inf. de pago de cuota
12. Aviso de falta de pago
13. Solicitud de devolución de ejemplar
15. Inf. socios
16. Inf. De días de atraso en devolución
17. Inf. De monto x día de atraso para el ejemplar

Como vemos:

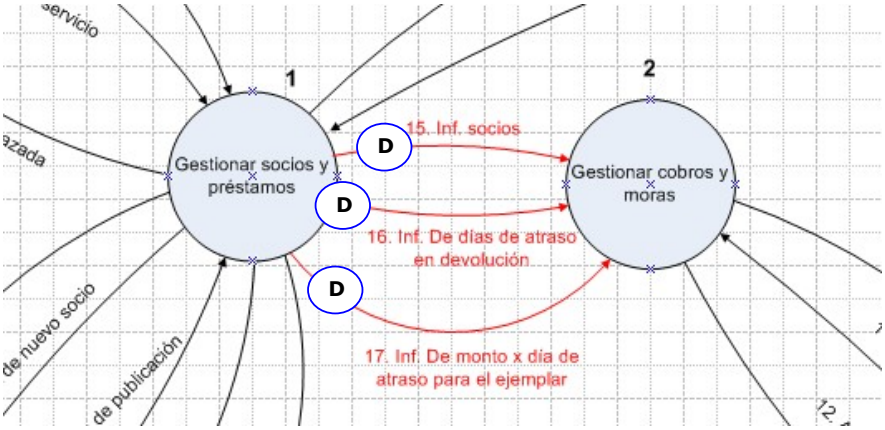
- Se mantienen sin cambios las entidades externas (**A**)

- Se mantienen sin cambios los flujos de información entre nuestro sistema y las entidades externas. (**B**)

- Se agregan nuevos procesos (**C**), a los que le asignamos números:



- Se agregan flujos de datos que intercomunican a estos nuevos procesos (**D**)



¿Cómo se desagrega?

La desagregación es un proceso que requiere de oficio por parte del analista. Hay mucho de sentido común y algunas reglas básicas.

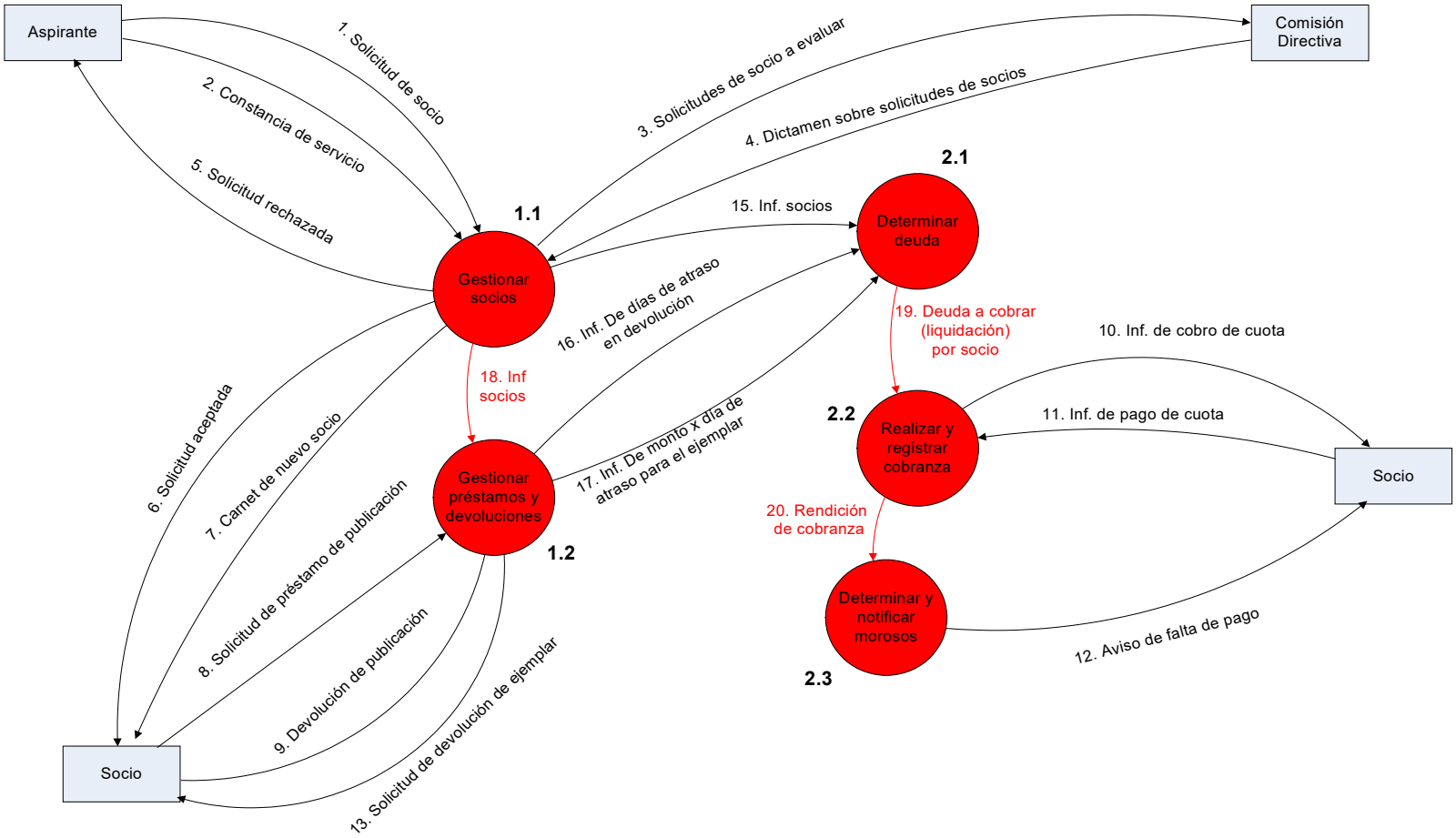
A veces ayuda imaginarse físicamente el lugar donde se lleva a cabo el proceso (por ejemplo, una oficina) y "dividir" el proceso mayor en las subtareas que se realizan en ese lugar.

Una regla básica es que hay "actividades" o "procesos" que por naturaleza están bien separados. Por ejemplo, si estamos modelizando un Consultorio odontológico sabemos que los ámbitos donde se realiza la práctica odontológica y el ámbito donde se cobra el servicio están claramente separados. Esta realidad se transfiere de manera directa a la modelización en el diagrama.

A continuación volvemos a llevar a cabo el proceso de **desagregación**. Nuevamente, no generaremos más de dos o tres procesos a partir de un proceso origen.

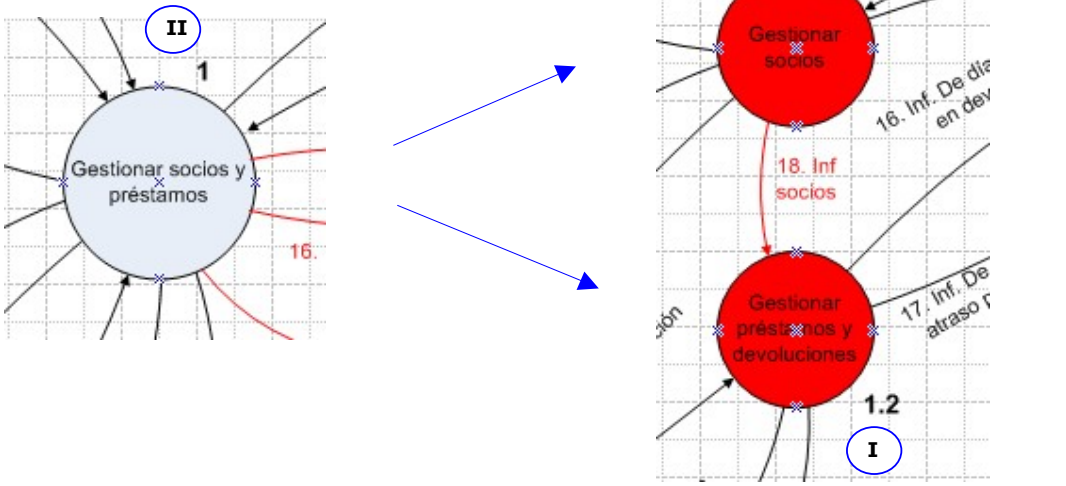
Arribamos al **Nivel 2**.

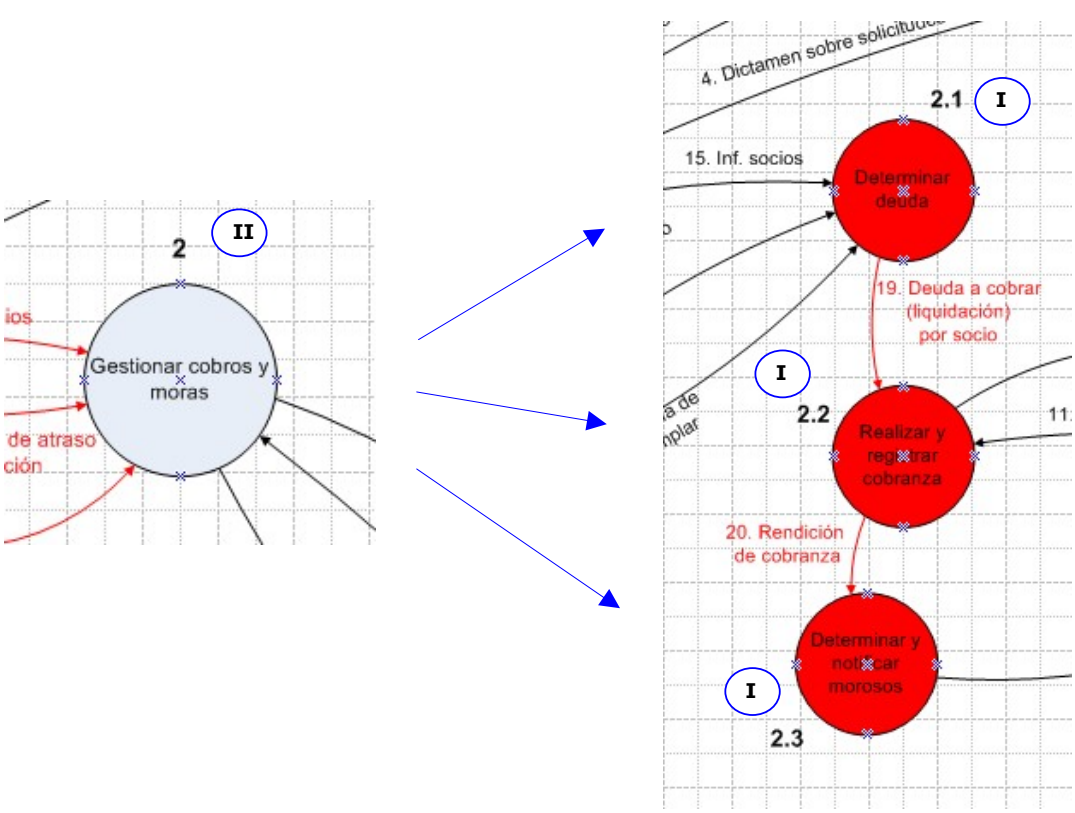
Nivel 2


Aspirante
Comisión Directiva
Socio
Socio
1. Solicitud de socio
2. Constancia de servicio
3. Solicitudes de socio a evaluar
4. Dictamen sobre solicitudes de socios
5. Solicitud rechazada
6. Solicitud aceptada
7. Carnet de nuevo socio
8. Solicitud de préstamo de publicación
9. Devolución de publicación
10. Inf. de cobro de cuota
11. Inf. de pago de cuota
12. Aviso de falta de pago
13. Solicitud de devolución de ejemplar
15. Inf. socios
16. Inf. De días de atraso en devolución
17. Inf. De monto x día de atraso para el ejemplar
18. Inf socios
19. Deuda a cobrar (liquidación) por socio
20. Rendición de cobranza
1.1
Gestionar socios
1.2
Gestionar préstamos y devoluciones
2.1
Determinar deuda
2.2
Realizar y registrar cobranza
2.3
Determinar y notificar morosos

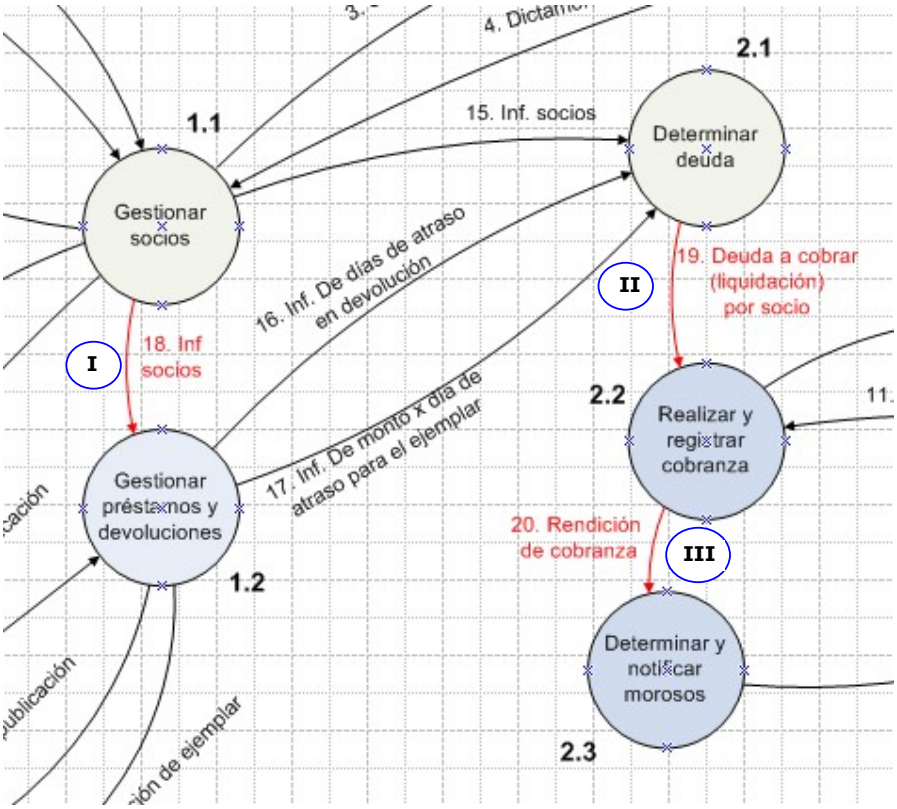
Nuevamente:

A. Se agregan nuevos procesos. Los identificamos con una numeración **item.subitem (I)** respecto al número del proceso en el nivel anterior (**II**):





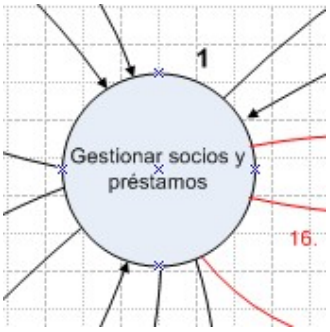
B. Se agregan flujos de datos que intercomunican a estos nuevos procesos (**I**, **II** y **III**)



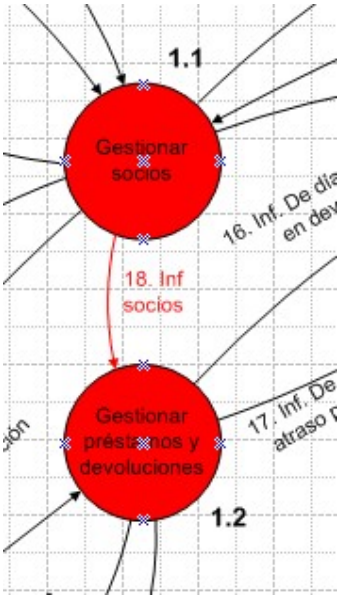
Consistencia de nombres de procesos entre niveles

Los nombres de los procesos no solo deben describir de manera los más exacta posible su función o comportamiento sino que también deben ser consistentes entre niveles.

Por ejemplo, el proceso *Gestionar préstamos y socios* en el Nivel 1...



...se desagrega en los procesos *Gestionar Socios* y *Gestionar préstamos y devoluciones* en el Nivel 2:



La "suma" de las funciones de los subprocesos (y en consecuencia sus nombres) debe ser consistente con la función del proceso del cual se desagregaron (y con su nombre).

9. Último nivel de desagregación

El proceso de desagregación debería finalizar cuando vemos que cada uno de los procesos representa o realiza una tarea lo suficientemente atómica como para ser totalmente comprensible. Observamos que no tiene sentido seguir subdividiendo cada proceso al que hemos arribado.
A veces a este proceso se le llama *proceso primitivo*.

Si bien como dijimos nuestro DFD no está necesariamente destinado a una solución informática, puede ser útil imaginar la realidad representada en un ámbito computacional. En tal caso debemos percibir que cada proceso es comprensible y que por tanto sería viable traducirlo o codificarlo como una módulo (procedimiento, función o método de objeto) computacional con un buen grado de cohesión[3].

[3] Cohesión es una medida de calidad de un módulo o procedimiento,. Un buen grado de cohesión significa que el módulo realiza una única tarea específica.

10. Consistencia de entidades externas y flujos de datos entre niveles.

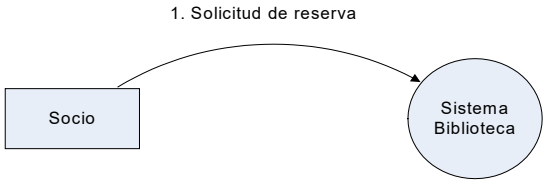
Los límites del sistema, nombres de flujos de datos, cantidad y calidad de entidades externas, etc., no deben modificarse porque el DFD esté dividido en niveles.

Un diagrama que permita este tipo de situaciones es inconsistente, no representa la realidad y no es un DFD.

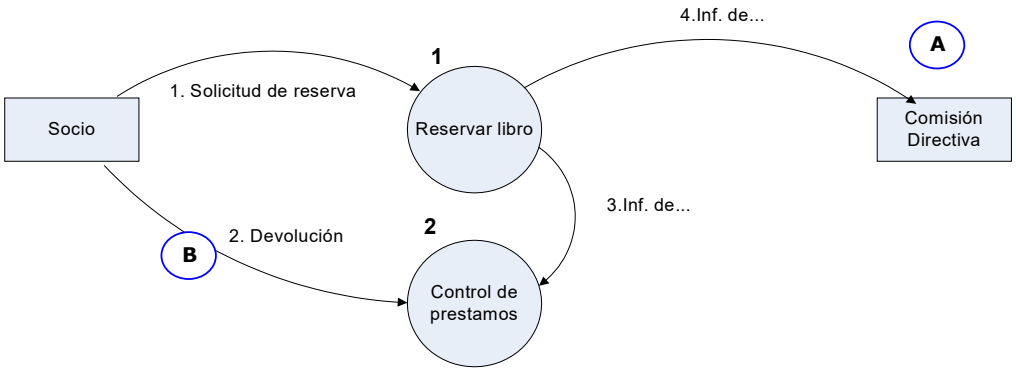
Ejemplo 1

Supongamos los siguientes dos niveles de un DFD que modeliza el funcionamiento de una biblioteca:

Nivel 0



Nivel 1



Este diagrama no es un DFD por las dos siguientes razones:

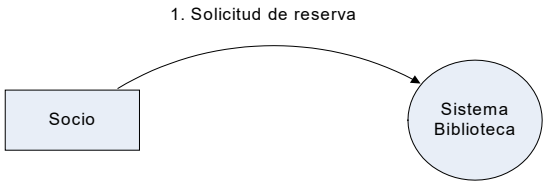
En el nivel 1 se agrega una entidad externa (**A**) que no existía en el nivel 0. En otras palabras, estamos modificando los límites del sistema, lo cual convierte al sistema original en otro sistema.

En el nivel 2 aparece un flujo de datos (**B**) que no existía en el nivel 0. En este caso, estamos modificando el comportamiento de nuestro sistema (sus entradas y salidas) respecto a las del nivel 0. Esto lo torna inconsistente, y por tanto no representa la realidad.

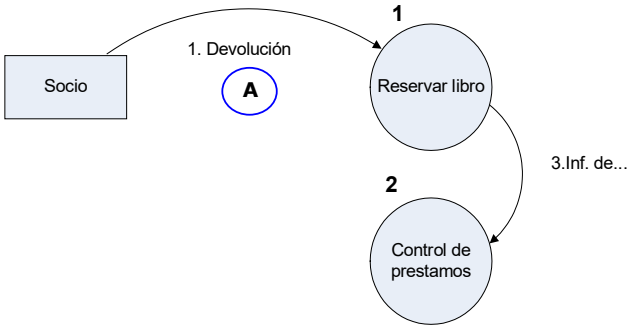
Ejemplo 2

Supongamos los siguientes dos niveles de un DFD que modeliza el funcionamiento de una biblioteca:

Nivel 0



Nivel 1



Este diagrama no es un DFD por la siguiente razón:

En el nivel 2 aparece un flujo de datos (**A**) con el mismo número pero diferente información que el del nivel 0. Haciendo la analogía con un mapa carretero, es como si hubiésemos hecho zoom sobre un barrio y los accesos a la ciudad hubiesen cambiado de numeración o nombre.
Esto torna al diagrama inconsistente, y por tanto no representa la realidad, no es un DFD.

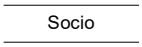
11. Almacenamientos

En los sistemas de la realidad la información se termina almacenando en algún tipo de soporte.

Una biblioteca que da de alta un nuevo socio deberá guardar ese registro en algún lugar, una ficha de cartón, un cuaderno o una base de datos.

DFD

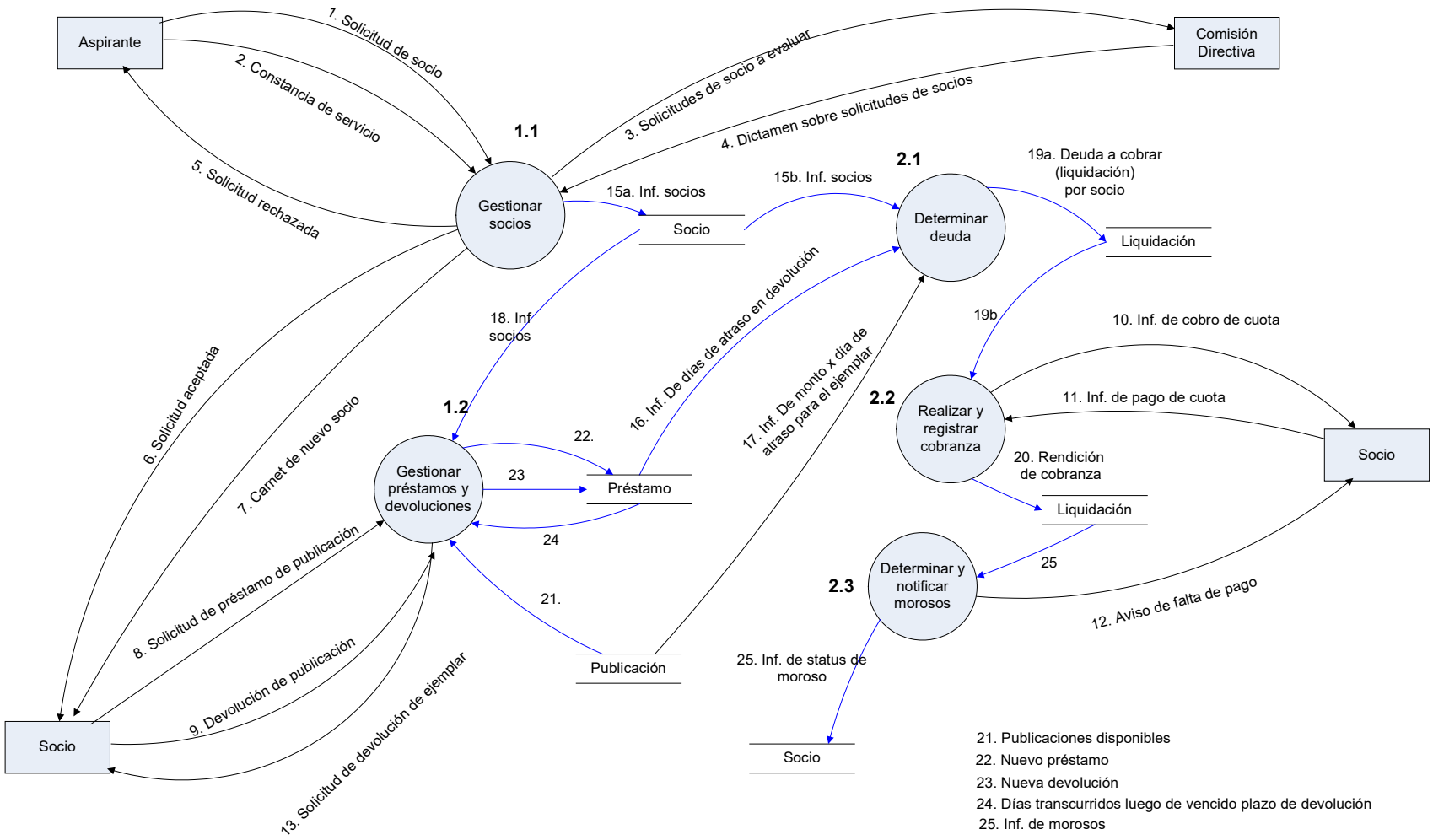
En la simbología de [Yourdon], representamos un almacenamiento de la siguiente manera:



12. Último nivel, con almacenamientos.

Cuando llegamos al nivel de detalle en el cual consideramos que no necesitamos seguir desagregando nos detenemos y transformamos el último nivel, en último nivel con almacenamientos.

Nivel 2 con almacenes



12.1. Transformación de flujos de información.

Cuando transformamos el último nivel de desagregación a un último nivel con almacenamientos debemos interrumpir muchos de los flujos de datos con almacenes:

Los flujos de datos de salida de un almacén son **lecturas** que el sistema hace del almacén. (llevado a un entorno computacional, podríamos ver estos flujos como sentencias SQL `SELECT` sobre una base de datos)

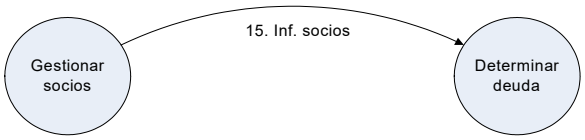
Los flujos de datos de entrada a un almacén son escrituras que el sistema hace sobre el almacén. (llevado a un entorno computacional, podríamos ver estos flujos como sentencias SQL `INSERT`, `UPDATE` o `DELETE` sobre una base de datos)

Este aspecto en particular se verá más claro cuando resolvamos el Ejercicio 1 – Sistema Olimpíadas.

12.1.1. Reglas para la nomenclatura de flujos de información.

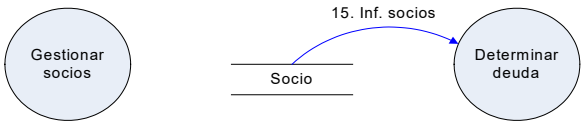
Caso 1

Debemos interrumpir un flujo de información con un almacén. Por ejemplo:

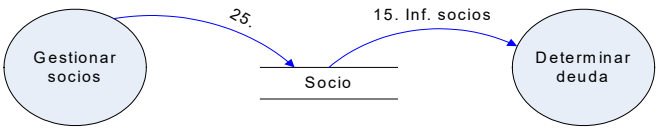


1. Agregamos el almacén:

2. Mantenemos el flujo existente como flujo **de lectura** sobre el almacén, **sin cambios en su nomenclatura**.

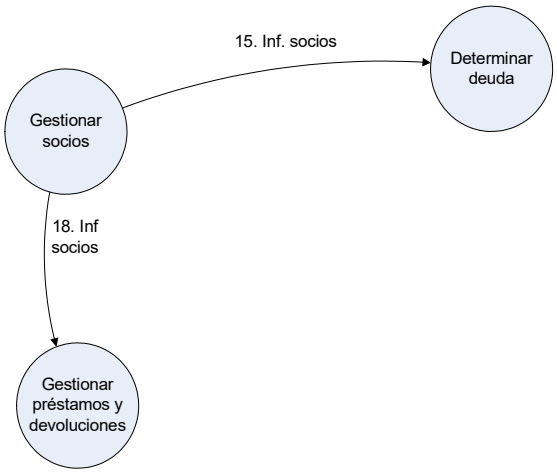


3. Agregamos un flujo NUEVO de escritura sobre el almacén. A este flujo simplemente lo identificamos con un número:

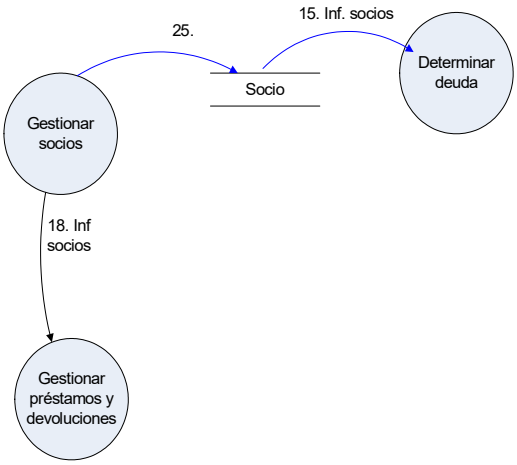


Caso 2

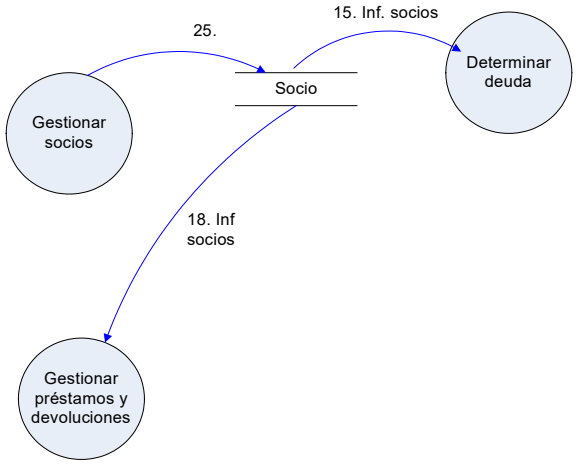
Más de un flujo de datos debe leer de un almacén. Por ejemplo:



1. Trabajamos primero sobre el Flujo 15: agregamos el almacén, mantenemos el flujo existente como flujo **de lectura** sobre el almacén, **sin cambios en su nomenclatura** y agregamos un flujo NUEVO de escritura sobre el almacén:



2. Trabajamos luego sobre el Flujo 18.
Supongamos que el este flujo debe leer también Información de Socios.
Entonces simplemente cambiamos el Origen del flujo existente.
El flujo 18, que antes iba de "*Gestionar socios*" a "*Gestionar préstamos y devoluciones*" ahora CAMBIA: va desde el almacén "*Socio*" a "*Gestionar préstamos y devoluciones*":



Observemos que NO agregamos un nuevo flujo de escritura sobre el almacén, ya que el Flujo 25 actualiza la información del almacén dejándolo listo para los flujos de lectura 15 y 18.

Lo que estamos haciendo es hacer **persistente** o **perdurable en el tiempo** un tráfico de información que antes se realizaba entre dos procesos.
Ahora este tráfico es INDIRECTO
El proceso "*Gestionar socios*" deja en el almacén *Socio* (a través del flujo 25) la información que los procesos "*Determinar deuda*" y "*Gestionar préstamos y devoluciones*" (y tal vez otros) necesitan.

En otro momento del tiempo (cuando la necesiten), los procesos "*Determinar deuda*" y "*Gestionar préstamos y devoluciones*" pueden recuperarla.

13. Otras consideraciones

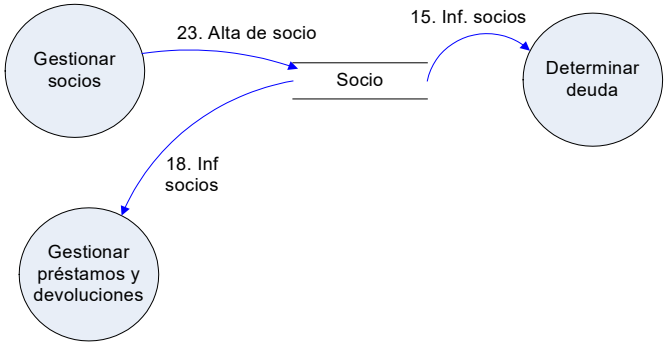
13.1. Ítems de Datos

Hasta que no llegamos al último nivel de desagregación no nos preocuparemos por los ítems de datos que componen cada flujo de información.

Podemos encontrarnos con flujo de información que se llaman igual pero que sabemos que son ligeramente diferentes. En este caso, les podemos poner el mismo nombre pero diferente numeración.

En un mismo sistema algunos procesos necesitarán obtener datos detallados de un proceso o almacén, mientras que otros necesitarán un subconjunto de los mismos. Por ejemplo, un subproceso que envíe newsletter a sus clientes en base a sus preferencias y compras anteriores deberá "leer" muchos más datos que un subproceso que necesite facturarle al cliente. En este último caso solo necesitará el Número de cuenta del cliente y los montos de sus consumos.

En el siguiente ejemplo, los flujos 15 y 18 se llaman *Inf. socios*, aunque seguramente la calidad de los datos que los respectivos procesos requieren son diferentes:



13.2. Varias representaciones de una entidad externa

A veces, a los fines de evitar el entrecruzamiento de flujos de datos es preferible dibujar dos o más veces una entidad externa. En la nomenclatura de [Gane Sarson] se dibujan con una marca en la esquina inferior derecha:

13.3. Almacenamientos actualizados por terceros

En un sistema de información bajo una implementación computacional tendremos almacenes de datos con información estructural (por ejemplo, en el ejemplo de la biblioteca, el stock de publicaciones, el padrón de socios, las reglas o montos a pagar por moras, etc.) y otros con información transaccional (préstamos realizados, devoluciones realizadas, pagos realizados, etc.).

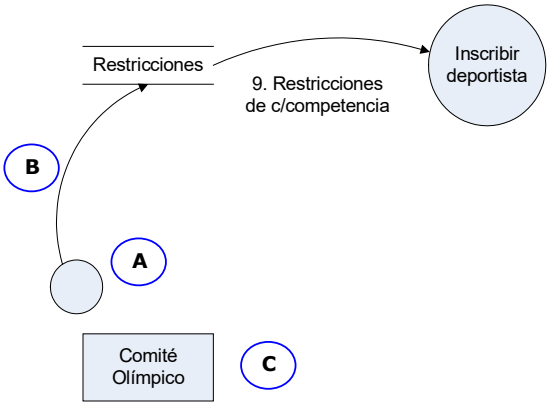
La responsabilidad de mantenimiento de los almacenes de datos transaccionales es siempre responsabilidad del sistema. Sin embargo, a veces el sistema hace uso de información de tipo estructural cuyo mantenimiento tiene sentido sea llevado a cabo por terceros. Algunos ejemplos son:

- Vademécum farmacológico de determinada provincia.
- Padrón de CUITs de un país.
- Padrón de CUILS de un país.
- Padrón de médicos de una localidad

Cuando un sistema necesita de un almacén actualizado por otra institución, generalmente obtiene una copia periódica de la misma.
Por ejemplo, una farmacia puede obtener una copia actualizada del vademécum de medicamentos del Colegio de Farmacéuticos, o una obra social puede obtener el padrón de médicos del Círculo Médico Provincial.

Con el advenimiento de las aplicaciones Web, y los avances en el ámbito del Gobierno electrónico, una entidad gubernamental podría averiguar –consultando un Web Service-, por ejemplo, si una determinada persona está inscripta en AFIP como monotributista, o si una persona está en el Veraz por su condición de moroso. Todos estos son casos de almacenes actualizados por terceras partes.

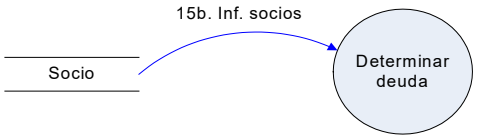
Cuando este es el caso, dibujamos la representación del almacén, pero sin ninguna operación de escritura sobre él.
Dibujamos un "proceso anónimo" (**A**) y un flujo de datos –también anónimo- de escritura (**B**) sobre dicho almacén. Todo esto cercano a la entidad que posee la responsabilidad de la actualización (**C**).
En el siguiente ejemplo la entidad Comité Olímpico es la encargada de actualizar el almacén de restricciones que se imponen para la admisión de un deportista en cada deporte olímpico:



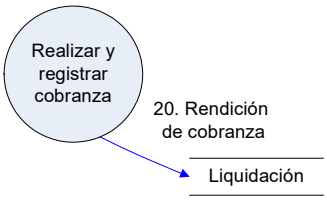
13.4. Origen y destino de un flujo de datos en niveles con almacenamientos

A las relaciones consignadas anteriormente se agregan:

Un flujo de datos puede tener como origen un almacén y como destino un proceso:



...o como origen un proceso y como destino un almacén:



No pueden existir flujos de datos desde una entidad externa a un almacén.

No pueden existir flujos de datos desde un almacén a una entidad externa.

Referencias

[Yourdon] Yourdon Edward. Análisis estructurado moderno. Prentice-Hall Hispanoamericana.

[Gane Sarson]. 1988. Análisis Estructurado de Sistemas. Librería "El ateneo" Editorial.