Introducción al Diseño de Sistemas de Información

Unidad Nº III: Diagramas Estructurados

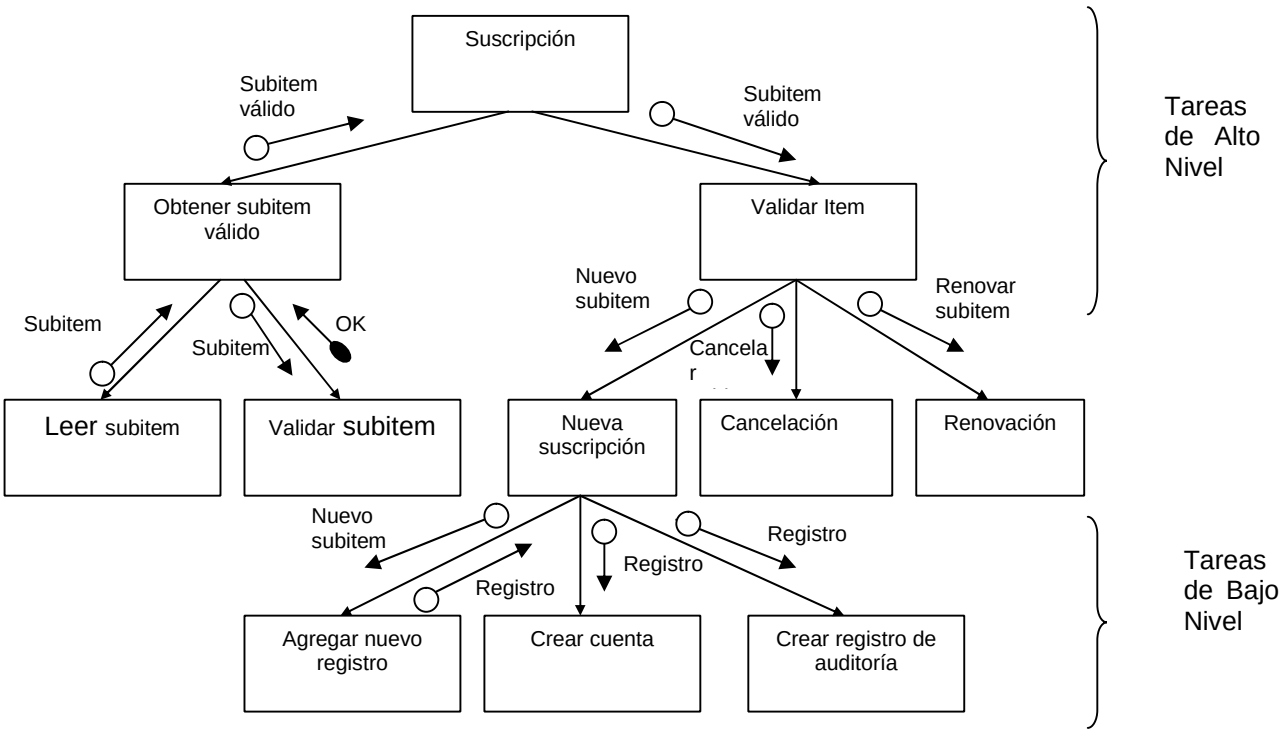
Facultad Regional Santa Fe Universidad Tecnológica Nacional

Diagramas Estructurados

Los Diagramas Estructurados (DE) se utilizan para la representación modular de las funciones del sistema. Representan el *diseño* de procesos, y se vinculan normalmente con un *programa*. El bloque de construcción básico de un programa es un *módulo*. Los programas estructurados están organizados en una *"jerarquía de módulos"*.

El diagrama estructurado es un árbol o diagrama jerárquico que define la arquitectura de un programa mostrando los módulos de un programa y sus interrelaciones.

La siguiente figura muestra el diagrama estructurado para un Sistema de Suscripción.



El programa es representado como un conjunto de módulos ordenados jerárquicamente. Los módulos que realizan *tareas de programa de alto nivel* están ubicados en la parte superior de la jerarquía, mientras que los módulos que realizan *tareas de bajo nivel* se encuentran en los niveles inferiores de la jerarquía.

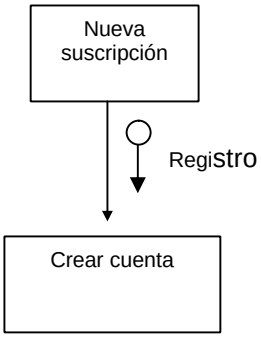
Los bloques de construcción básica de los diagramas estructurados son *rectángulos* y *flechas* que los conectan. Cada rectángulo en el diagrama representa un módulo.

Lógicamente, un módulo es una tarea que el programa ejecuta, tal como 'Agregar nuevo registro' o 'Crear cuenta'.

El nombre del módulo debe ser escrito dentro del rectángulo. El nombre debería ser descriptivo de la tarea que realiza el módulo.

Los módulos están interrelacionados por una *estructura de control*. El diagrama estructurado muestra las interrelaciones ordenando los módulos en niveles y conectándolos por líneas.

Una flecha entre dos módulos de niveles sucesivos significa que en tiempo de ejecución, el control del programa pasa de un módulo al segundo en la dirección de la flecha. Decimos que el primer módulo invoca o llama al segundo. Por ejemplo, en este caso:



El módulo 'Nueva Suscripción' invoca al módulo 'Crear cuenta'. Cuando el módulo 'Crear cuenta' termina su ejecución el control vuelve nuevamente al módulo 'Nueva Suscripción'.

Es posible que un módulo invoque o llame a varios módulos. Los diagramas estructurados no muestran estrictamente secuencia, con lo cual no sabemos en qué orden un módulo invoca a sus hijos, pero, como normal, se estable que la *secuencia* de ejecución de los módulos se muestra de arriba hacia abajo y de izquierda a derecha.

Un módulo que no tiene hijos se llama Hoja.

Reglas de control para un diagrama estructurado

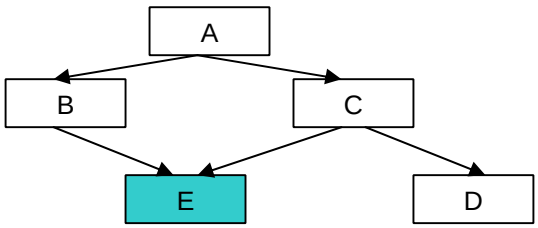
Hay uno y sólo un módulo al tope de la jerarquía (nivel 1) del DE. Este módulo el llamado raíz (root).

Desde la raíz el control es pasado hacia abajo nivel por nivel a los otros módulos.

El control siempre es devuelto al módulo invocante. Por esto, cuando la ejecución del programa finaliza, el control regresa al root.

Hay a lo sumo una relación de control entre dos módulos cualesquiera. Si el módulo A invoca a B, B no puede invocar a 'A'.

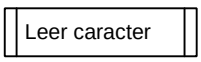
Módulos comunes



En este ejemplo, el módulo E es llamado *módulo común*.

Este diagrama ya no es mas un árbol, esto se hace para no repetir el módulo E dos veces.

Módulos librería



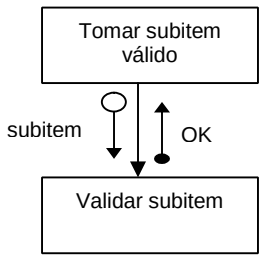
En algunos casos, el programa puede usar módulos de *librerías* predefinidas. Esto es indicado en el diagrama por un rectángulo con dos líneas verticales.

Normalmente, un módulo librería aparece como *hoja* en el DE.

Transferencia de datos

Cuando el control es transferido entre dos módulos, usualmente se transfieren también datos. Los datos pueden ser transferidos en *ambas direcciones* entre los módulos. La dirección de la

transferencia de los datos lo indican la *flechas pequeñas*. Los *nombres* de datos son escritos sobre las flechas.



Dos tipos básicos de información pueden comunicarse entre dos módulos:

Datos (acoplamiento de datos) ○→

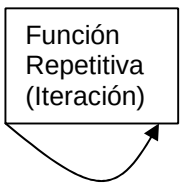
Control (acoplamiento de control) ●→

Los *datos* son información usada en el problema, tal como una 'suscripción de subitem', o un 'legajo' de empleado, o 'el detalle de una factura', etc.. Este tipo de información es identificada por una flecha con un círculo vacío en su origen
( ).
Control es información usada por el programa para *dirigir* el flujo de ejecución, tal como un indicación de que ha ocurrido un error o se ha llegado al fin de archivo. Es identificada por una flecha con un círculo pintado en uno de sus extremos ( )

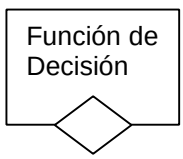
Secuencia, selección e iteración

Secuencia: orden en el cual los bloques son ejecutados.
Selección: uso de condiciones para controlar si o no un módulo es ejecutado o cuál de varios bloques será ejecutado.
Iteración: control de ciclos.

La *secuencia* de ejecución de los módulos se muestra de arriba hacia abajo y de izquierda a derecha.
Los DE *no* muestran ni iteración ni selección por defecto, pero pueden incorporarse simbología para estos casos.
Iteración:

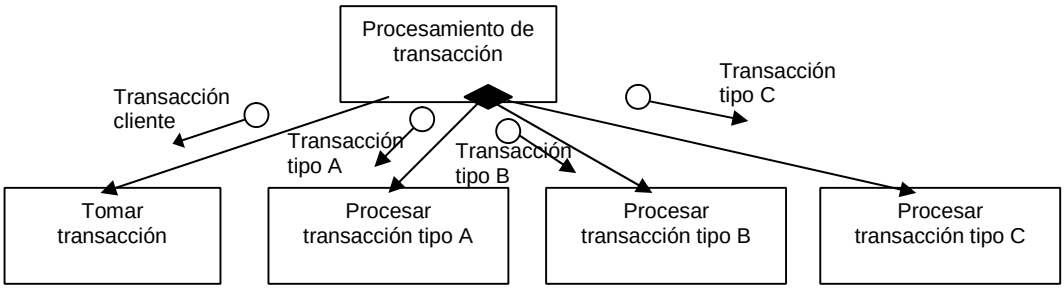


Selección / Decisión:



Centro de Transacción

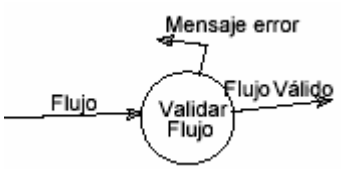
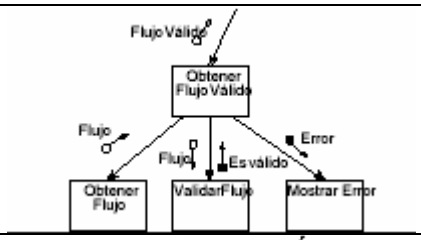
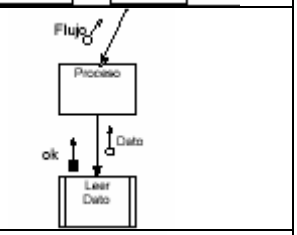
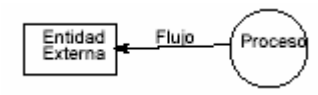
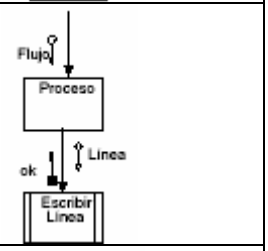
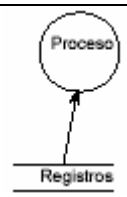
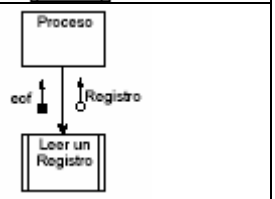
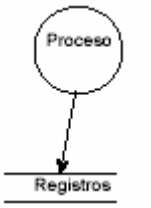
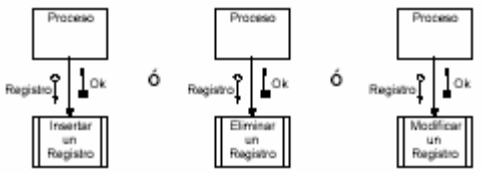
Cuando son procesadas varios *tipos de transacciones*, un módulo de programa separado puede ser usado para procesar cada tipo de transacción.



El diamante negro en el bloque superior es llamado *centro de transacción*. Para procesar cada tipo de transacción se usan módulos separados. El centro de transacción determina el tipo de una transacción y transfiere el control al módulo apropiado.

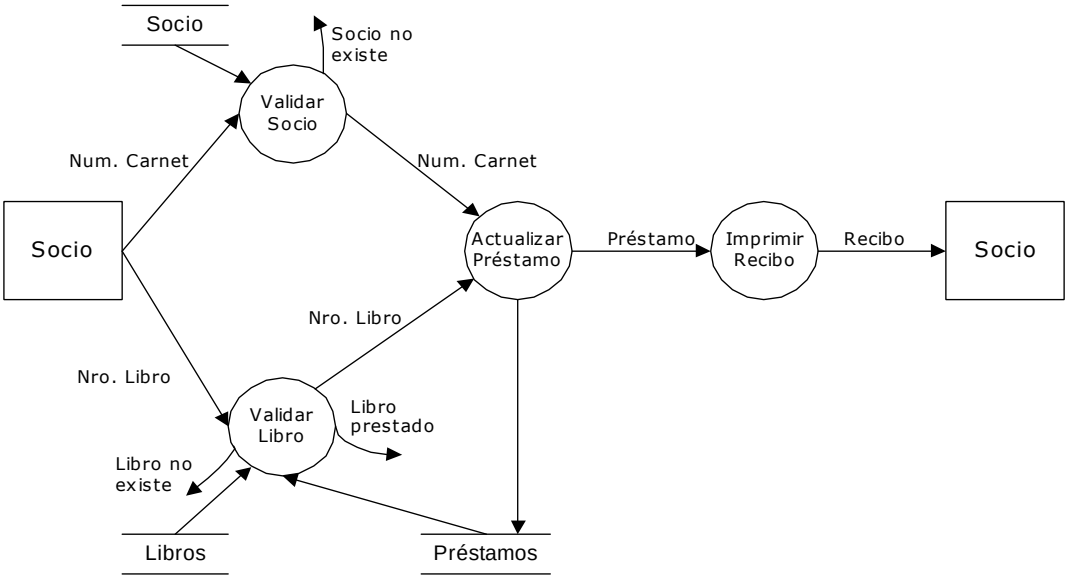
Guías de diseño

Algunas guías de diseño prácticas que se pueden aplicar para transformar DFDs en DE, son las siguientes:

Patrones de Diseño		
Validación Asociado a los procesos de validación de datos de entrada al sistema		
Lectura de datos desde entidades		
Emisión de datos a Entidades Externas		
Lectura desde almacenes de datos		
Escritura en almacenes de datos (inserción, eliminación o modificación de registros)		

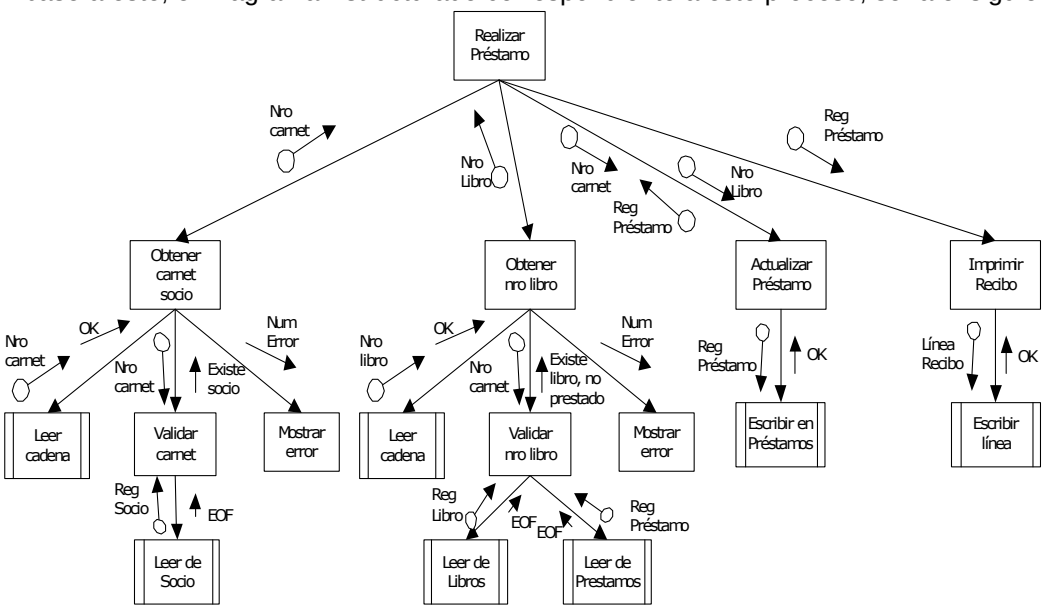
Ejemplo de DFD a DE transformacional

El siguiente DFD representa un proceso de 'Prestamos de Libros a Socios', siendo los datos requeridos para esto, el 'Número de Carnet' del Socio y el 'Número de Libro' requerido. Si los datos son correctos y aceptados, se actualiza los préstamos realizados, y se imprime el recibo correspondiente que se entrega al Socio.



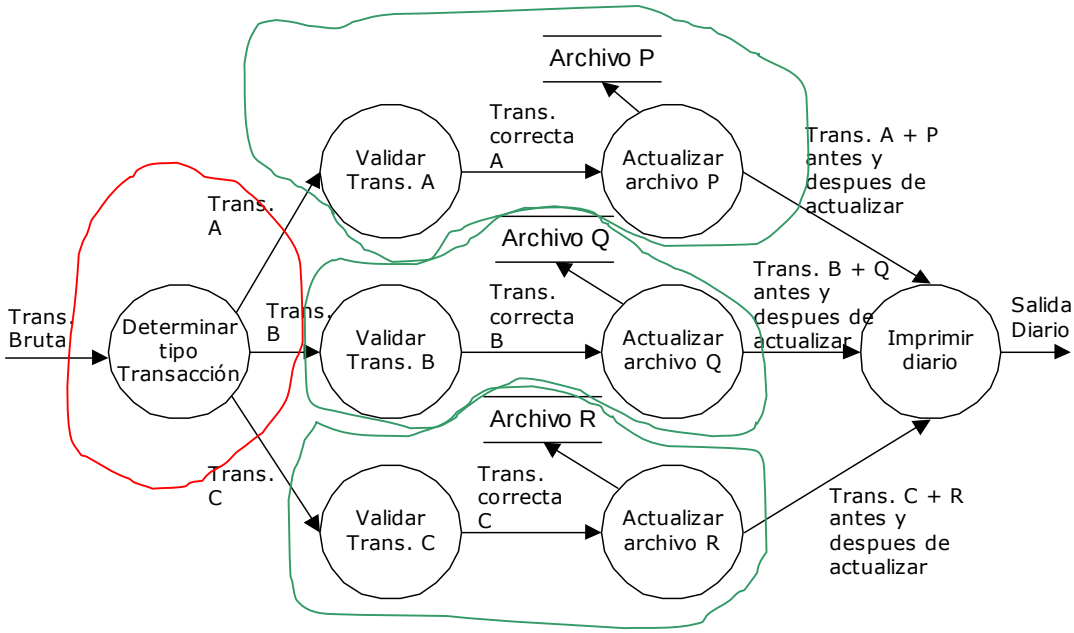
Analizando el DFD resultante, pueden establecerse:
Ramas aferentes (procesos que leen o validan los datos a la entrada del sistema): Validar Socio, Validar Libro.
Ramas eferentes (procesos que dan a los datos el formato adecuado para ser emitidos al exterior – visualizados, impresos, etc): Imprimir Recibo.
Centro de transformación (procesos de cálculo, procesamiento de datos, actualización de datos, etc): Actualizar Préstamo

En base a esto, el Diagrama Estructurado correspondiente a este proceso, sería el siguiente:



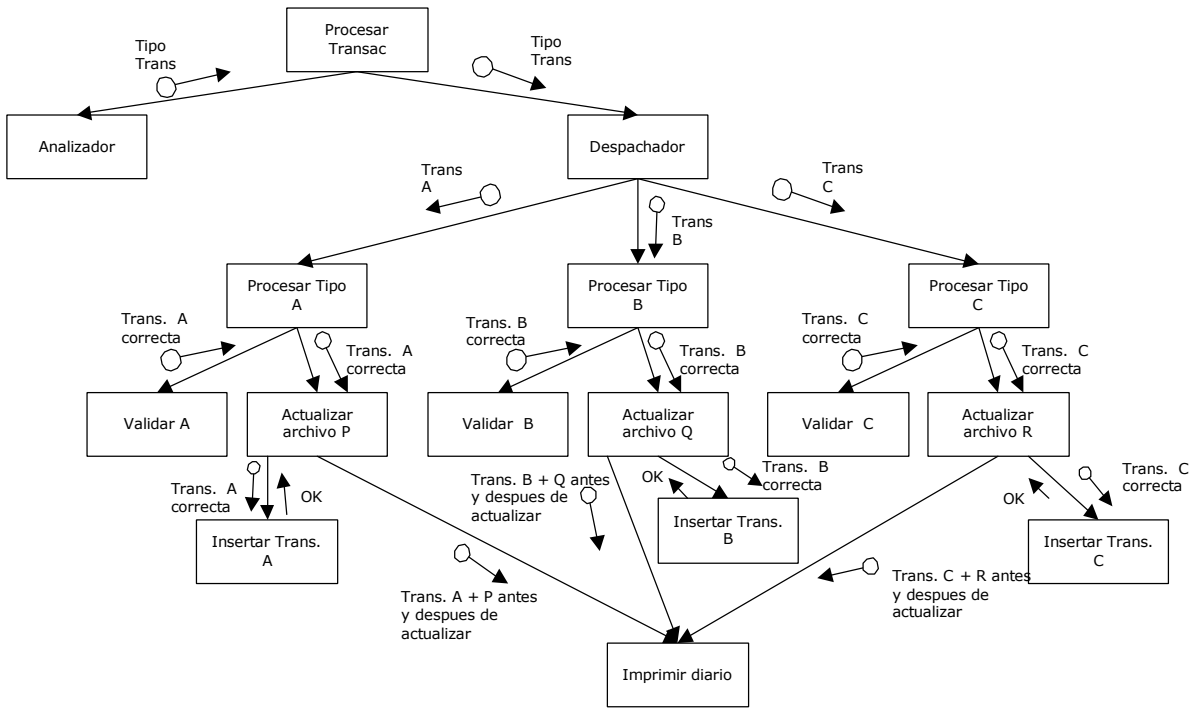
Ejemplo de DFD a DE transaccional

En el siguiente DFD muestra un caso genérico, en el que puede determinarse los siguientes elementos:



- Un centro de transacción
- Tres caminos de transacción

Con estos elementos podemos derivar en el siguiente Diagrama Estructurado:



Bibliografía
Structured Techniques – The Basis for CASE. James Martin, Carma McClure.
Información desde la Web.