

Modelado de Flujo de Datos

- Diagramas de Flujo de Datos (DFDs)
- Primitivas de refinamiento y transformaciones
- Estrategias de diseño



Diagrama de Flujo de Datos (DFD)

- Representación gráfica de la evolución de la información dentro de un *Sistema de Información*.
- Desde que la información ingresa a un S. I., va sufriendo sucesivas transformaciones, hasta que se almacena definitivamente en él o sale transformada.



DFD: Proceso

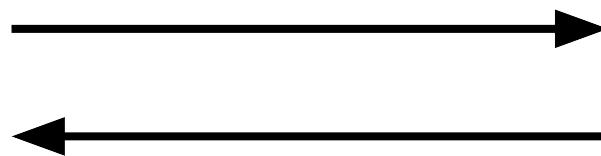
- Actividad dentro de un sistema, cuyo objetivo es el de generar, usar, manipular o destruir información.

Dar de alta
cliente (2.1)



DFD: Flujo de datos

- Representa un intercambio de información entre dos procesos o entre una interfaz y un proceso.



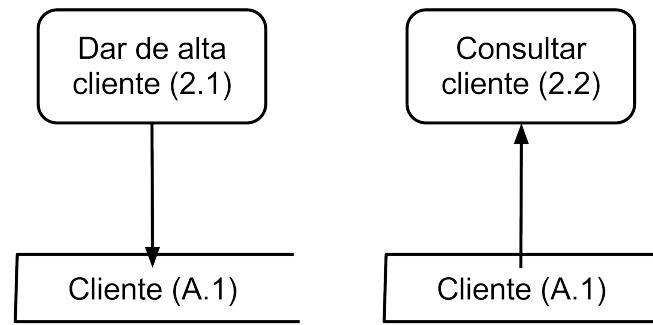
DFD: Almacén de datos

- Representa un depósito de información dentro del sistema y, en el mundo real, puede representar un archivo, tablas de consulta, formularios en papel, electrónicos, ...

Cliente (A.1)



DFD: Almacén de datos



- Un flujo que va de un proceso a un almacén representa que el proceso introduce, actualiza o elimina información del almacén (modifica el contenido).
- Un flujo que va de un almacén a un proceso representa que el proceso consulta información del almacén (no modifica el contenido).

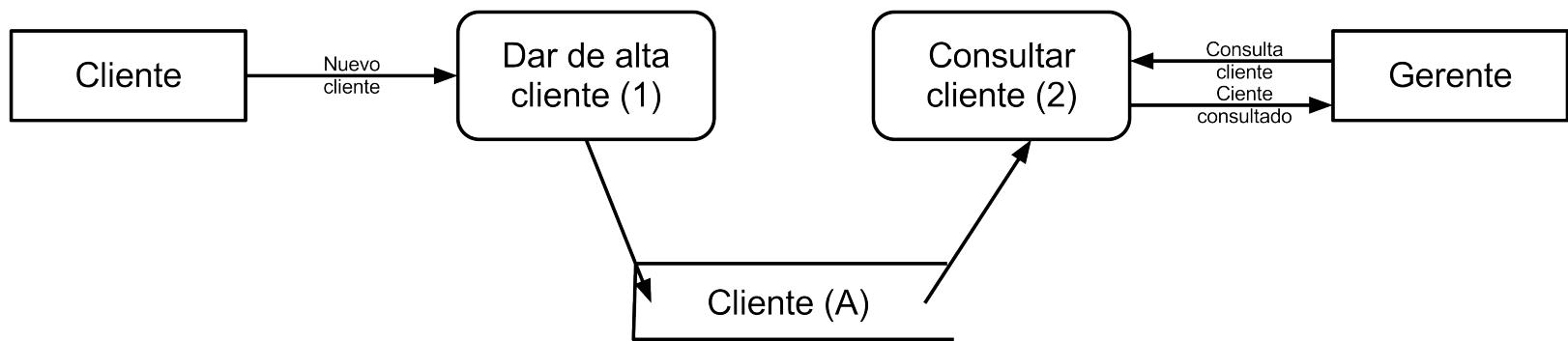
DFD: Interfaz

- Usuario o entidad activa del mundo real que se comunica con el sistema enviando información y recibiendo información.

Cliente

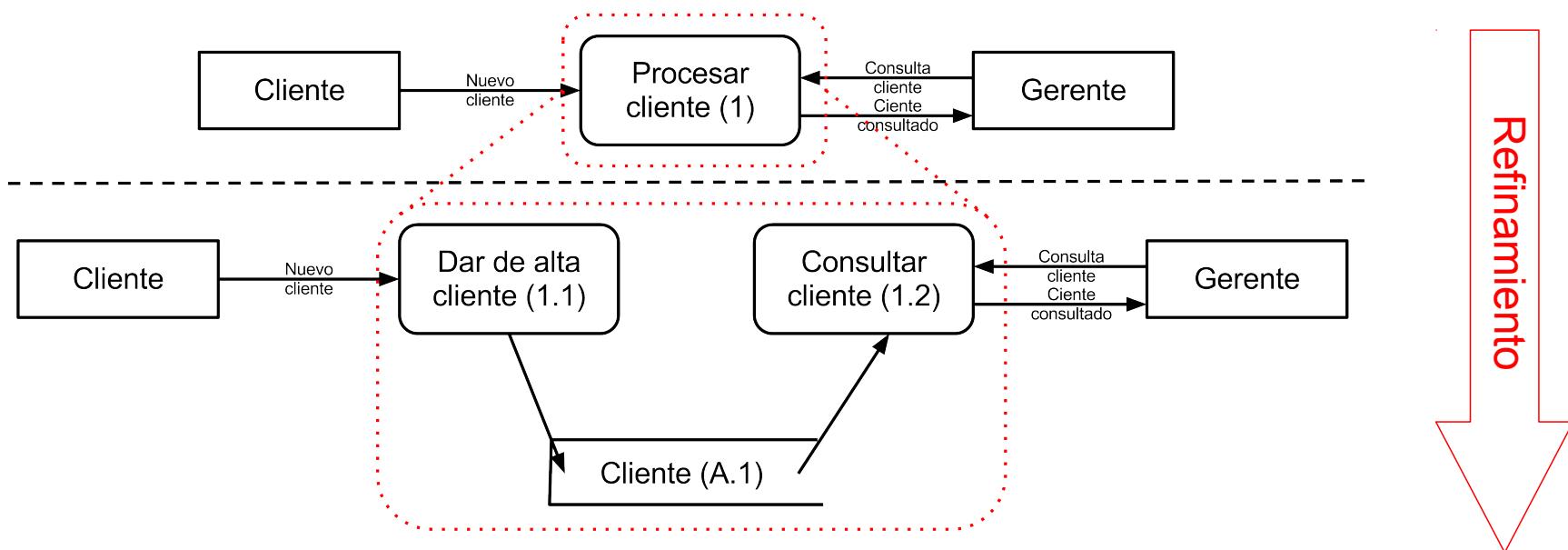


DFD: Un ejemplo



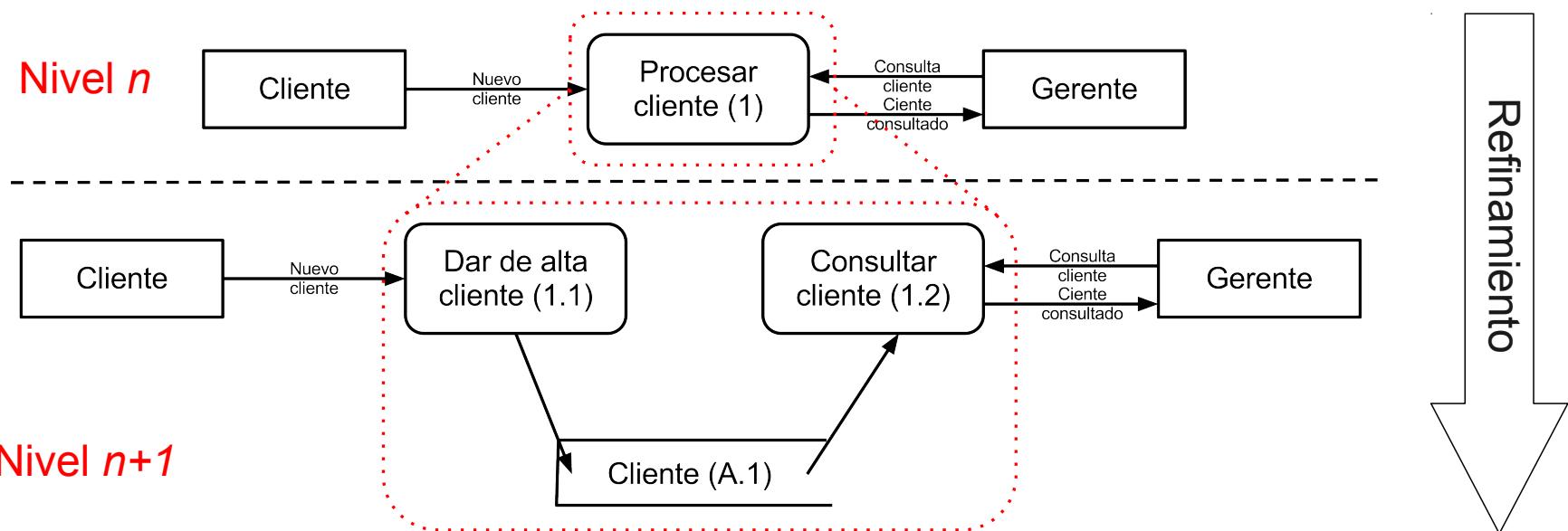
Primitivas para el diseño funcional: refinamiento

- Al proceso de pasar de un DFD a otro relacionado con él siguiendo ciertas normas, se le conoce como *refinamiento*



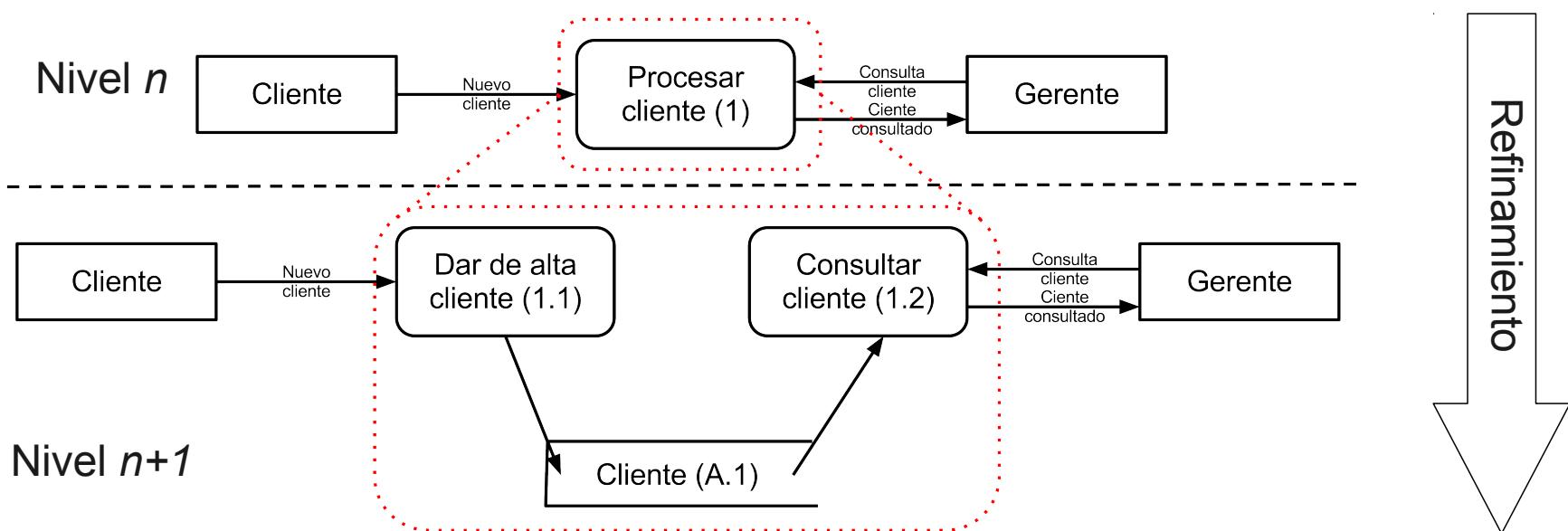
Primitivas para el diseño funcional: niveles de refinamiento

- A dos vistas distintas en un proceso de refinamiento, se les conoce como *niveles de refinamiento*



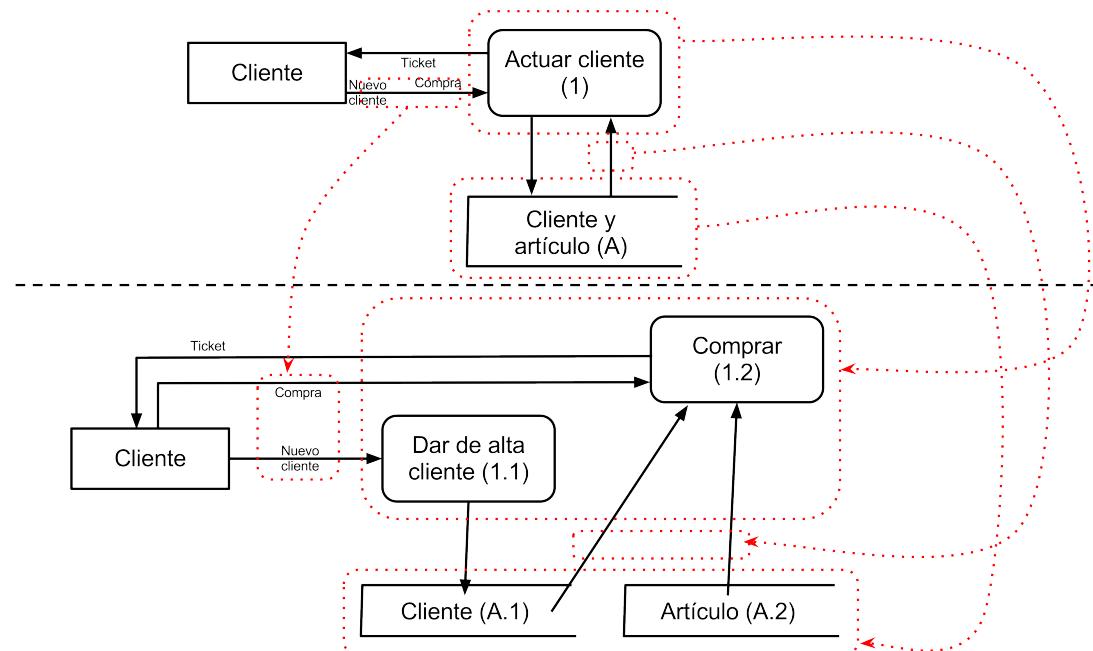
Primitivas para el diseño funcional: primitiva

- A la transformación que permite refinar una parte de un DFD, se le conoce como *primitiva de refinamiento*



Primitivas para el diseño funcional: transformación

- Llamamos *transformación* al conjunto de primitivas aplicadas a la vez sobre elementos del mismo esquema.



Primitivas para el diseño funcional: esquema inicial

- Al esquema del nivel n en la transformación, se le denomina *esquema inicial*.



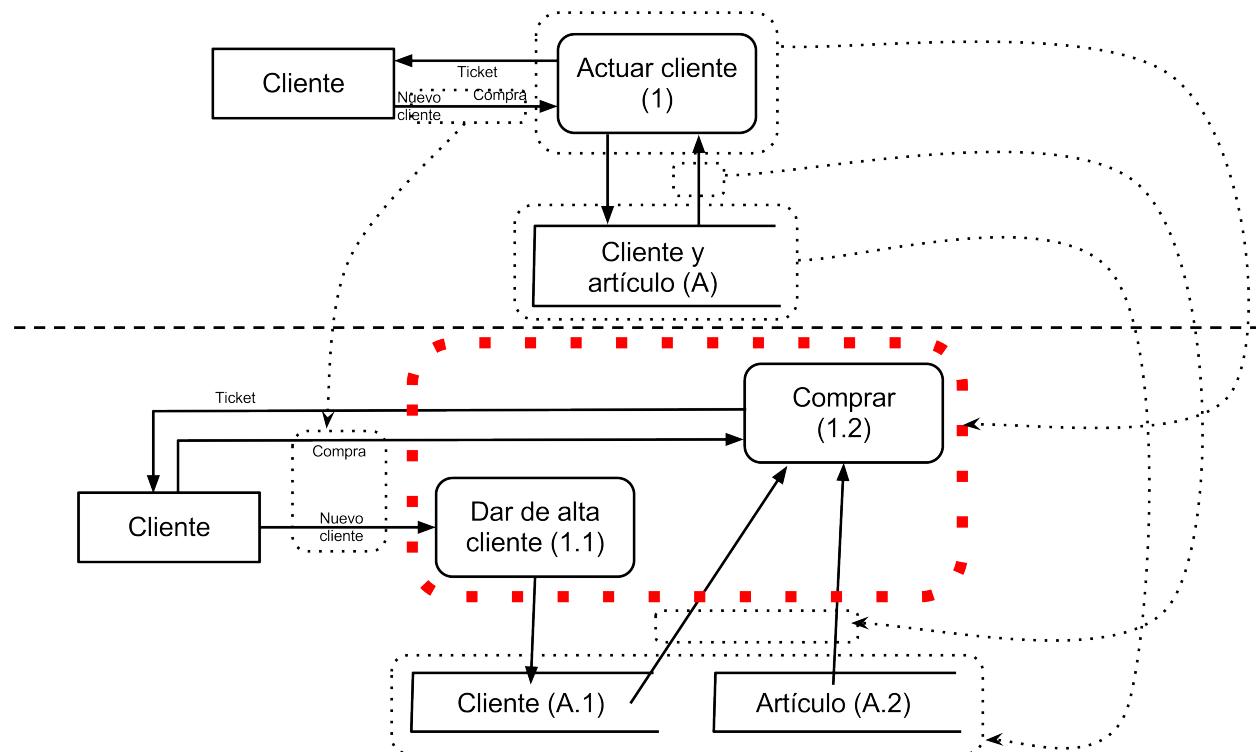
Primitivas para el diseño funcional: esquema resultante

- Al esquema final en la transformación, se le denomina *esquema resultante*.



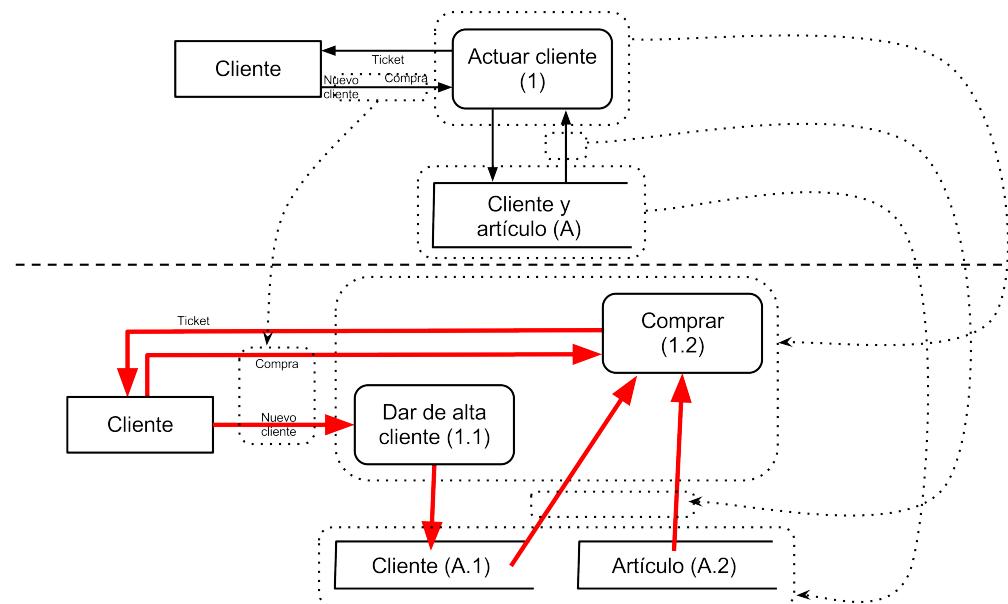
Primitivas para el diseño funcional: normas de transformación

- A la línea discontinua que envuelve al refinamiento de un elemento mediante primitiva, se le llama *frontera*.



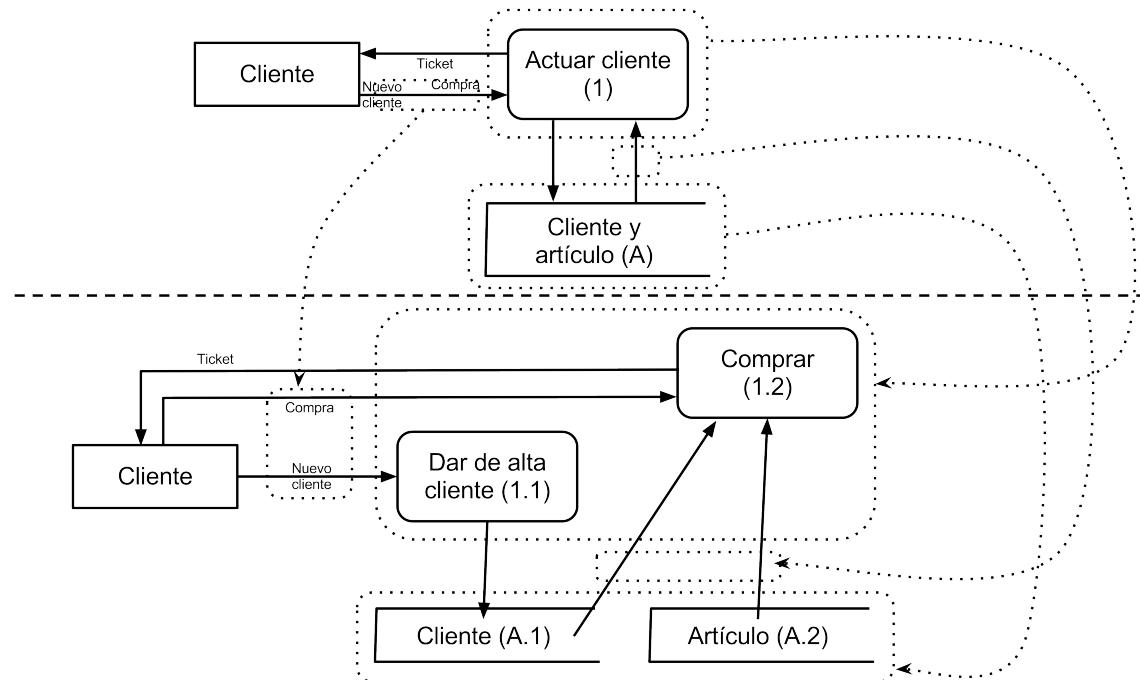
Primitivas para el diseño funcional: normas de transformación

- Si dos elementos en el esquema inicial están conectados entre sí, sus transformados tienen que estar conectados entre sí.



Primitivas para el diseño funcional: normas de transformación

- *Preservación semántica*: la semántica de un elemento sin refinar y la de su refinamiento, difieren únicamente en el grado de abstracción.



Primitivas para el diseño funcional: tipos

- Según si permiten un diseño analítico o sintético, existen dos tipos básicos:
 - **Primitivas descendentes**
 - **Primitivas ascendentes**

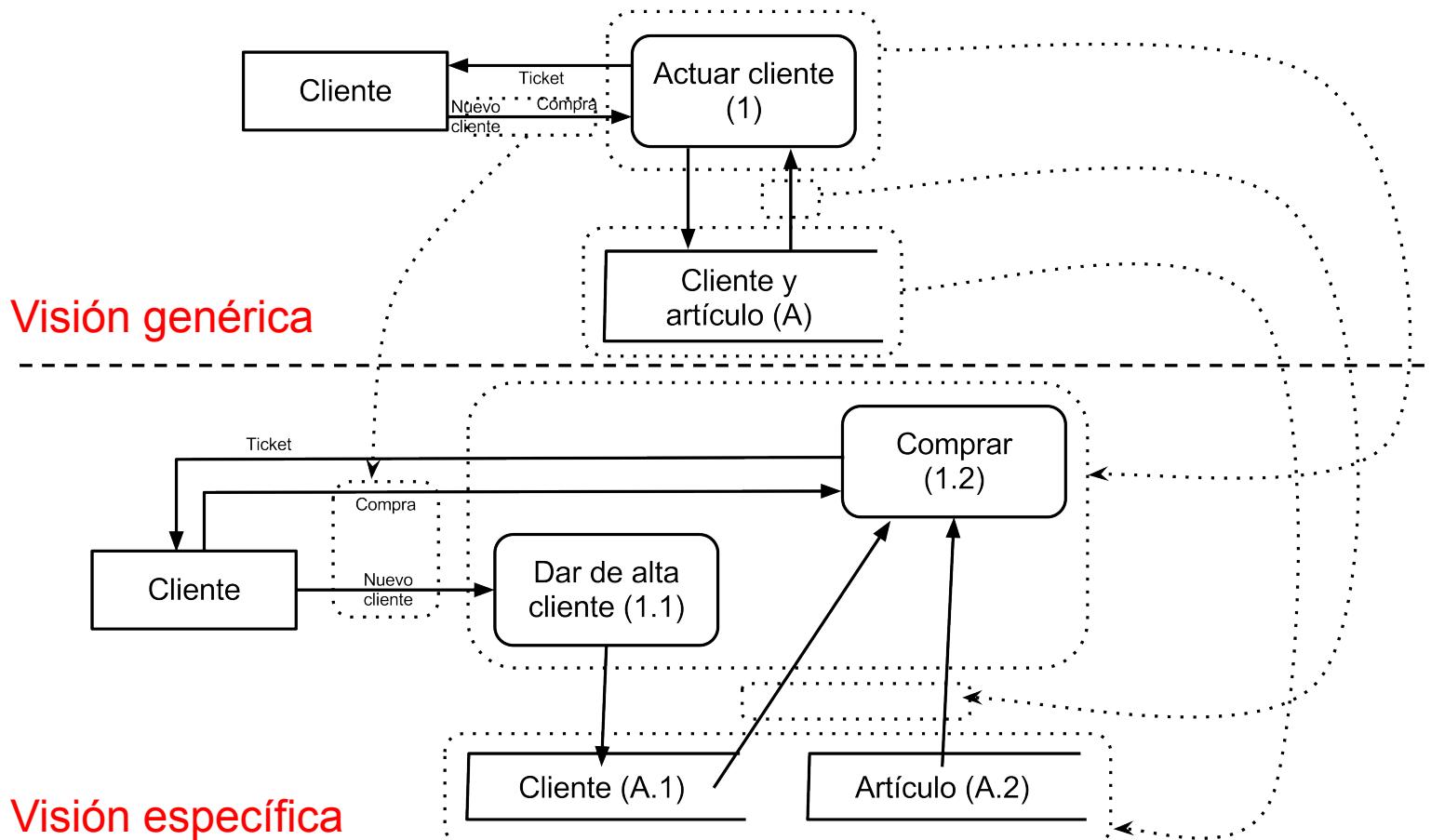


Primitivas para el diseño funcional: primitivas descendentes

- Permiten llegar desde una visión genérica y abstracta de las funciones de un sistema hasta una versión concreta o específica (como se ha visto en los ejemplos anteriores)

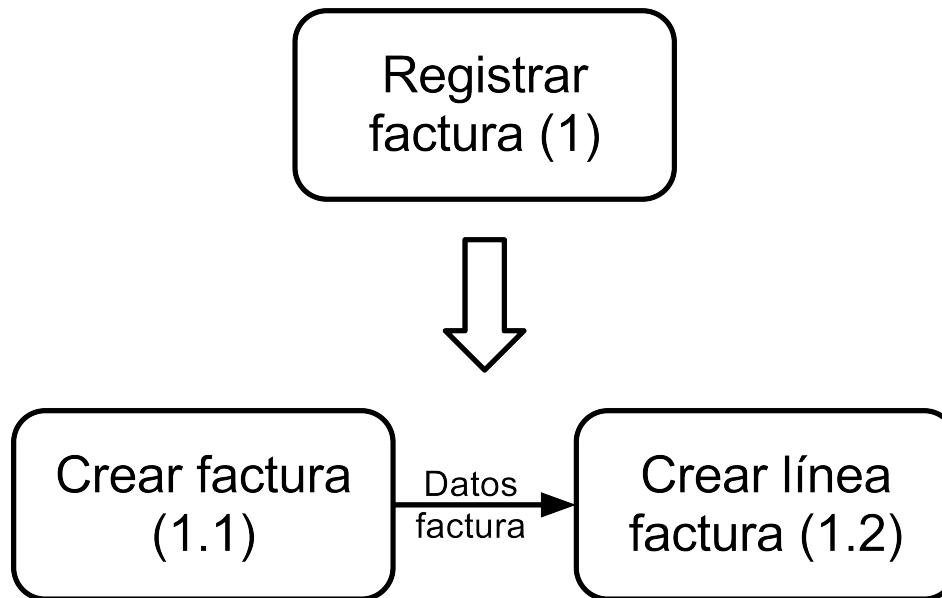


Primitivas para el diseño funcional: primitivas descendentes



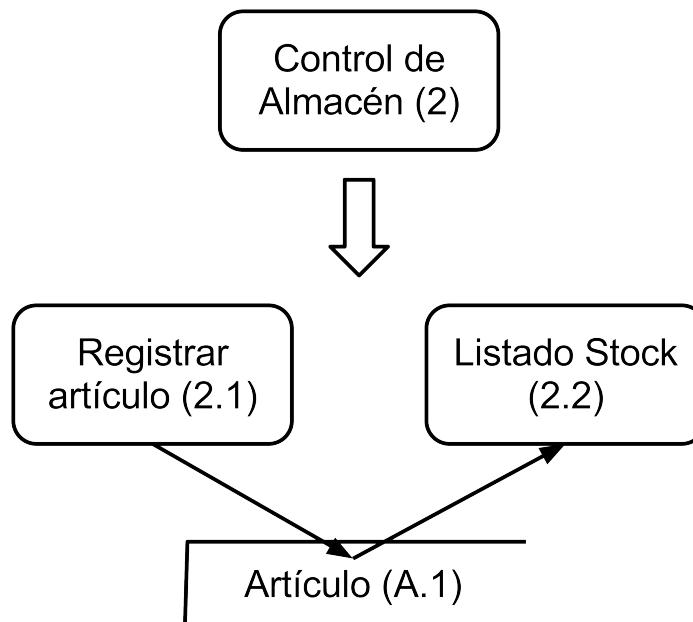
Primitivas para el diseño funcional: primitivas descendentes

- T1: descomposición de proceso en procesos con flujo intermedio



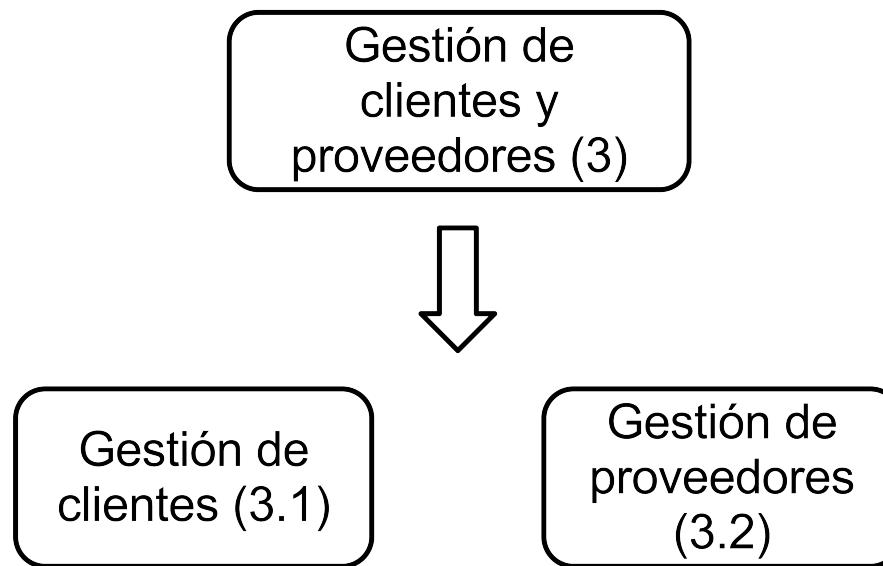
Primitivas para el diseño funcional: primitivas descendentes

- T2: descomposición de proceso en procesos con almacén intermedio



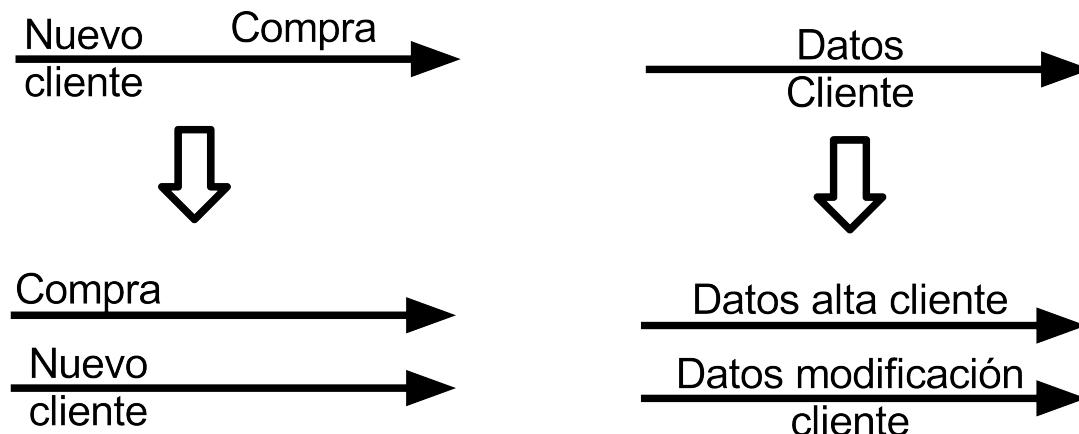
Primitivas para el diseño funcional: primitivas descendentes

- T3: descomposición de proceso en procesos sin conexiones



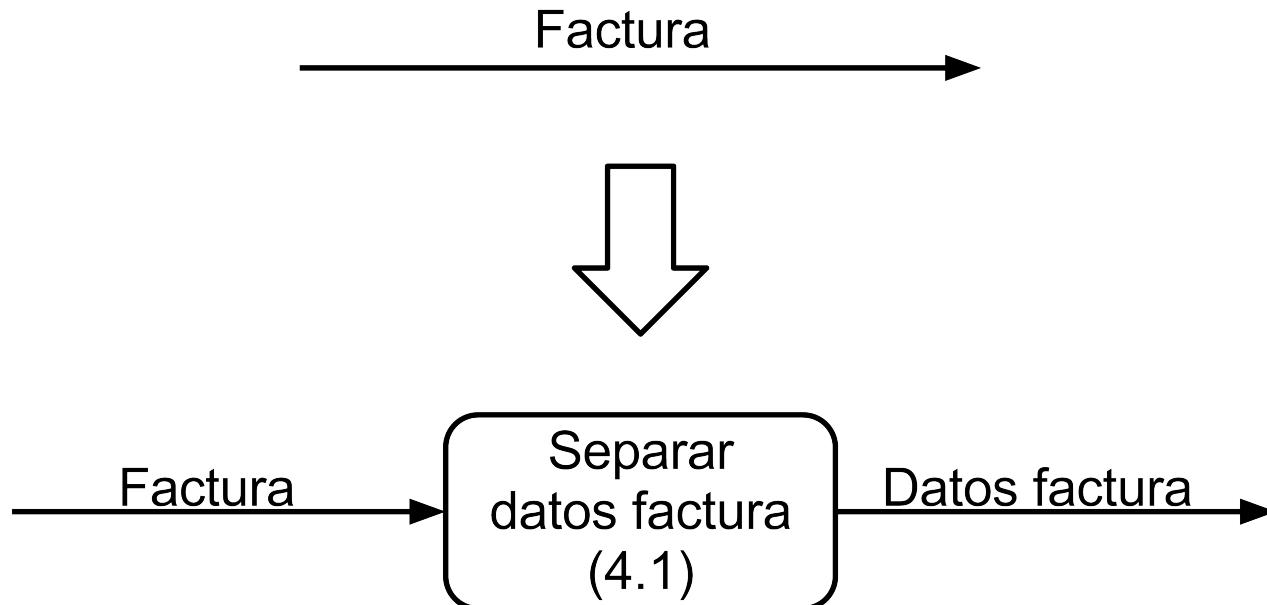
Primitivas para el diseño funcional: primitivas descendentes

- T4: descomposición de flujo



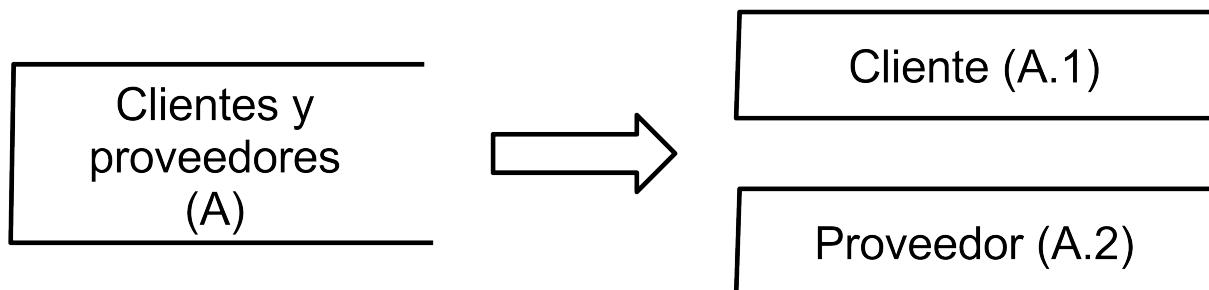
Primitivas para el diseño funcional: primitivas descendentes

- T5: refinamiento de flujo



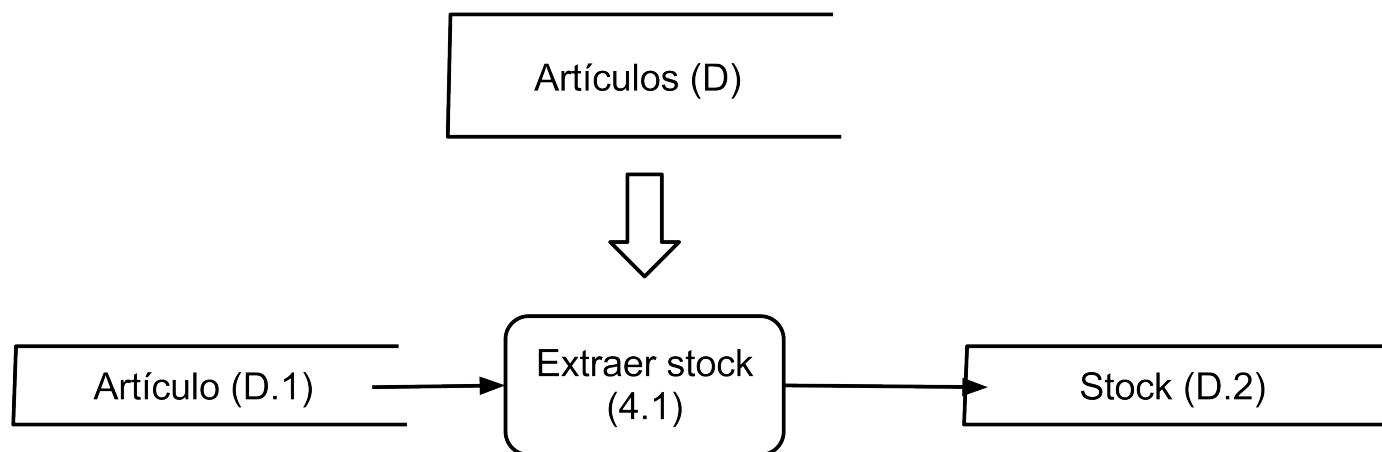
Primitivas para el diseño funcional: primitivas descendentes

- T6: descomposición de almacén



Primitivas para el diseño funcional: primitivas descendentes

- T7: creación de almacén



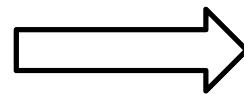
Primitivas para el diseño funcional: primitivas ascendentes

- Permiten llegar desde una visión concreta o específica de las funciones de un sistema hasta una versión conectada del mismo.



Primitivas para el diseño funcional: primitivas ascendentes

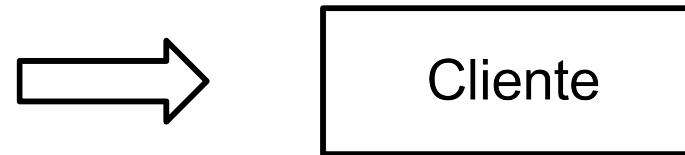
- B1: de generación de proceso



Dar de alta
cliente (1.1)

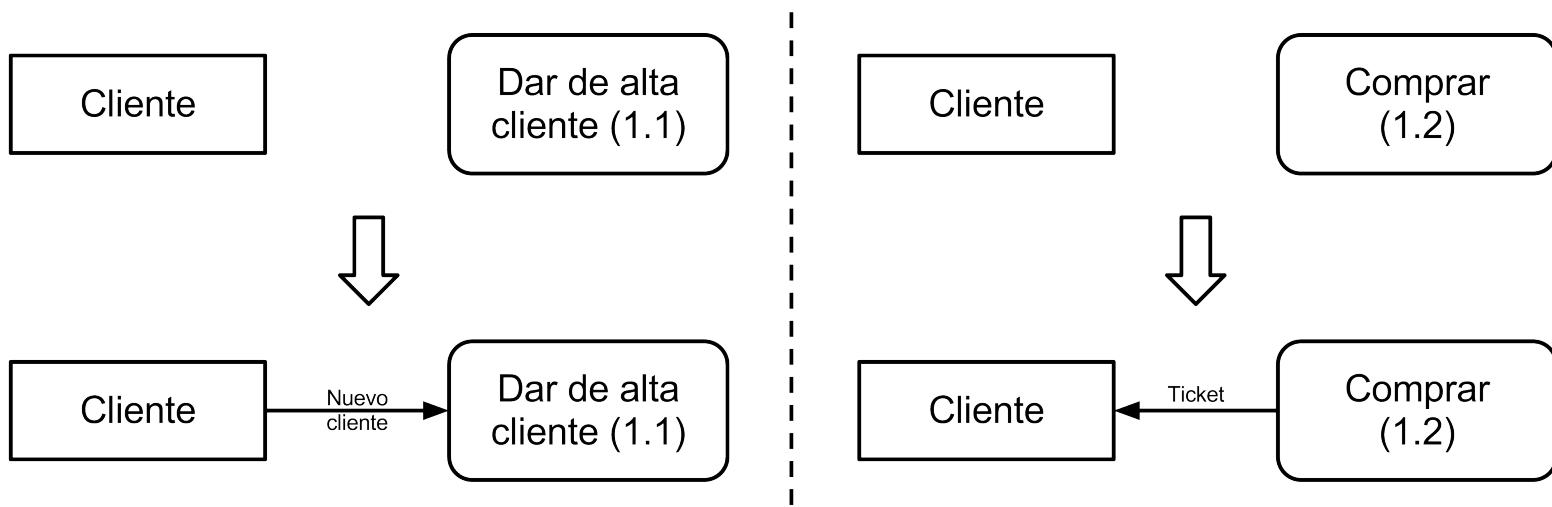
Primitivas para el diseño funcional: primitivas ascendentes

- B2: de generación de interfaz



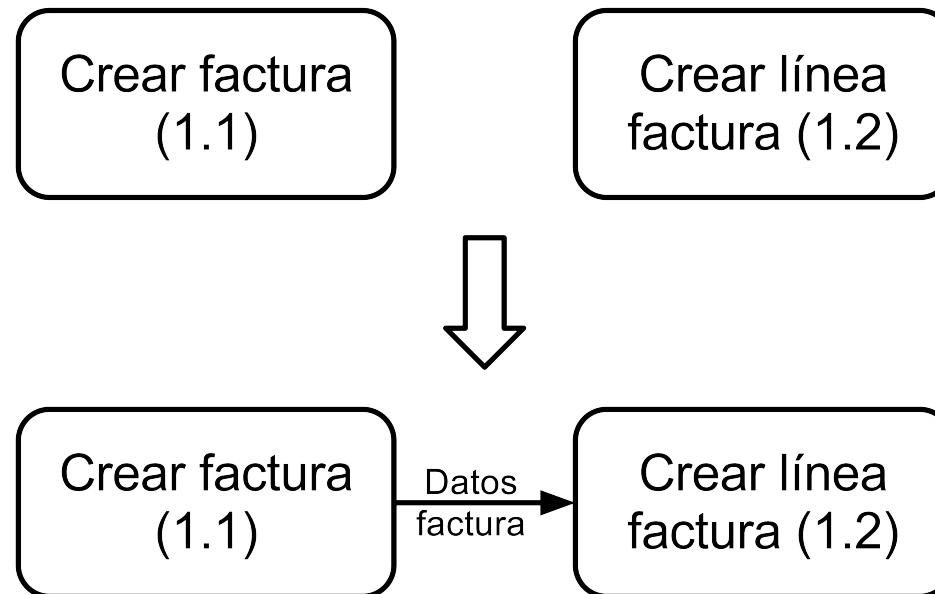
Primitivas para el diseño funcional: primitivas ascendentes

- B3: de generación de flujo entre interfaz y proceso



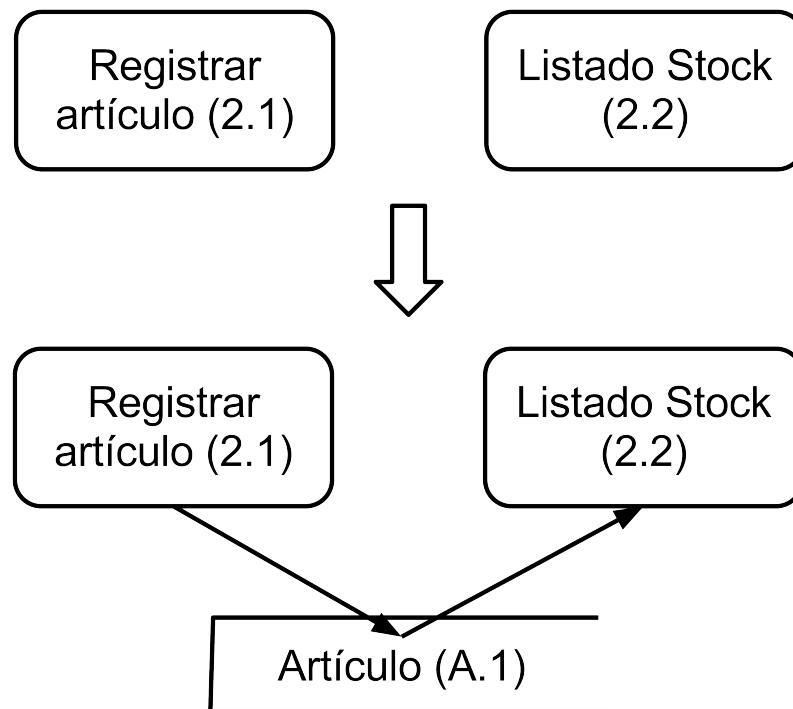
Primitivas para el diseño funcional: primitivas ascendentes

- B4: de generación de flujo entre procesos



Primitivas para el diseño funcional: primitivas ascendentes

- B5: de generación de almacén entre procesos

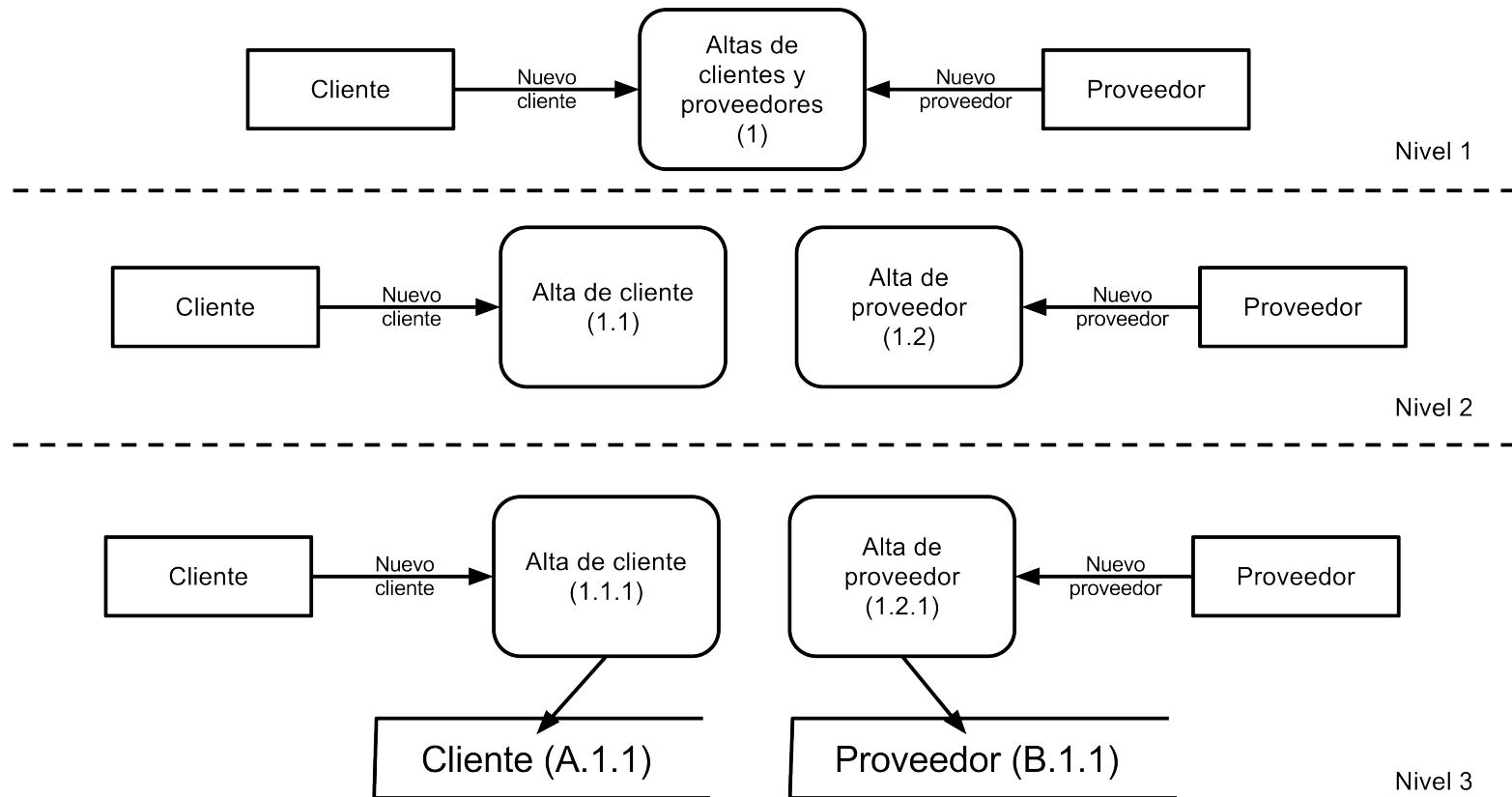


Primitivas para el diseño funcional: Estrategia de diseño

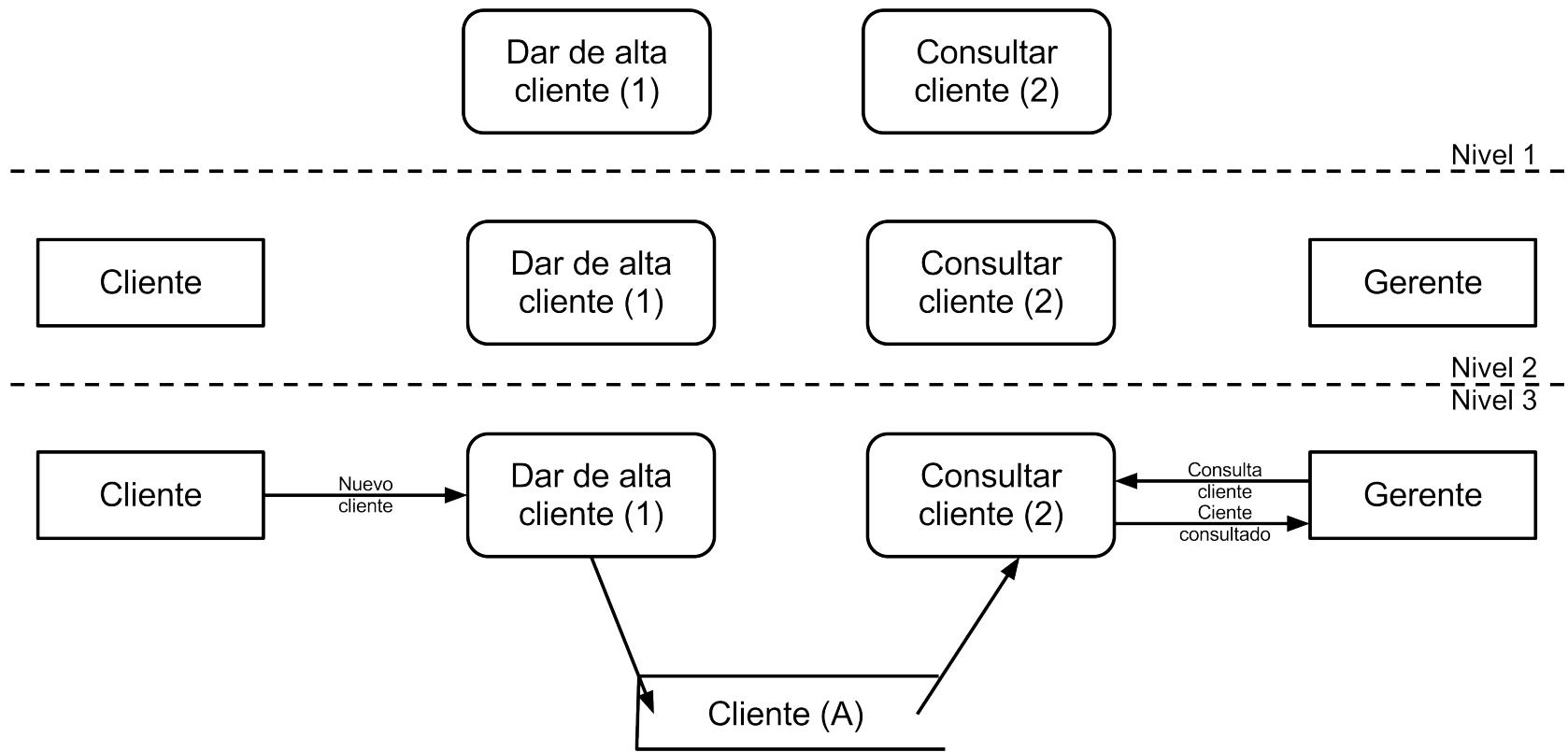
- Son de dos tipos:
 - **Descendente**: consiste en aplicar las primitivas descendentes a todos los elementos de un refinamiento funcional (a un *nivel de refinamiento*), y aplicarlo a cada refinamiento hasta que todos los requisitos queden representados explícitamente.
 - **Ascendente**: consiste en aplicar las primitivas ascendentes a todos los elementos de un refinamiento funcional (a un *nivel de refinamiento*), y aplicarlo a cada refinamiento hasta que todos los requisitos queden conectados.



Primitivas para el diseño funcional: Estrategia de diseño descendente



Primitivas para el diseño funcional: Estrategia de diseño ascendente

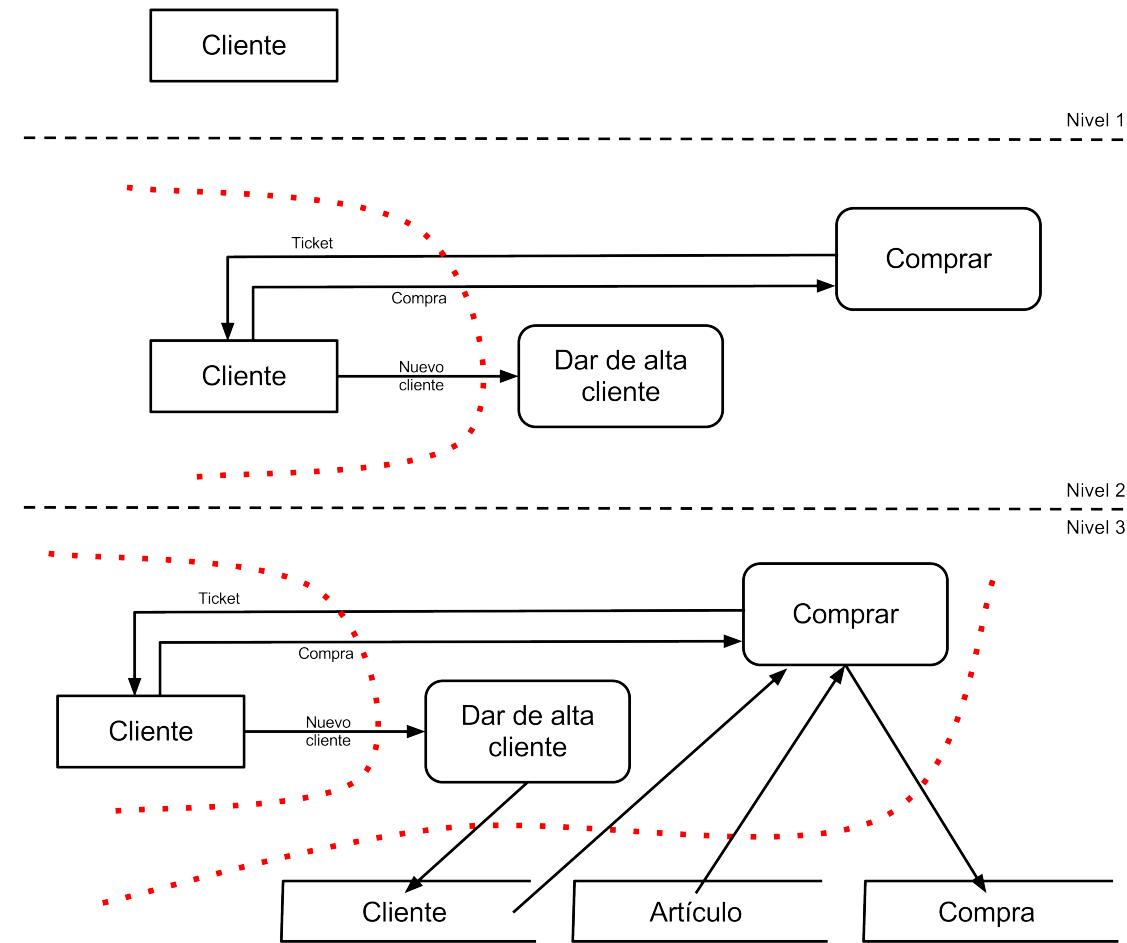


Primitivas para el diseño funcional: Estrategia de diseño centrífuga

- Modificación de la estrategia de diseño ascendente que permite centrarse en una serie de funciones totalmente detalladas, modelarlas y pasar al siguiente grupo de funciones conectadas con las primeras.



Primitivas para el diseño funcional: Estrategia de diseño centrífuga



Primitivas para el diseño funcional: Estrategia de diseño centrífuga

- Comienza con los **interfases** y determina progresivamente los procesos que participan en los flujos
- Estrategia orientada **hacia la entrada**
 - Parte de los **interfases** que proporcionan **información**
- Estrategia orientada **hacia la salidas**
 - Parte de los **interfases** que **reciben información** mostrada como salida del sistema



Primitivas para el diseño funcional: Estrategia de diseño mixta

- Paso 1: Esquema armazón (ascendente):
 - Se dividen los requisitos funcionales en varios conjuntos (pocos) según algún criterio lógico.
 - Se representa cada conjunto de requisitos mediante un proceso (*subsistema*). Cada proceso se identificará por su nombre y un número.
 - Se añaden las interfaces y se interconectan con los procesos.
 - Se interconectan los procesos según sus necesidades de transmitirse información (flujo) o de compartirla (almacén). Cada flujo entre procesos deberá ir identificado por un nombre que represente la información que transmite. Cada almacén se identificará por su nombre y una letra.



Primitivas para el diseño funcional: Estrategia de diseño mixta

- Paso 2: División del esquema inicial (armazón o cualquier refinamiento funcional que necesite más refinamiento):
 - Se separa el esquema funcional (DFD) obtenido en todas las partes que lo integran.



Primitivas para el diseño funcional: Estrategia de diseño mixta

- Paso 3: Refinamiento parcial (descendente):
 - Cada elemento separado que necesite refinamiento, se refinará aplicando las primitivas descendentes, sin tener en cuenta las conexiones que pueda tener con los demás elementos (técnica *divide y vencerás*).
 - Cada refinamiento parcial se rodea de una frontera.
 - Los procesos que provengan del refinamiento de un proceso, tendrán su propio nombre y su identificación será la del proceso del que provienen seguido de un punto y un número de orden dentro del refinamiento parcial.



Primitivas para el diseño funcional: Estrategia de diseño mixta

- Paso 3: Refinamiento parcial (descendente):
 - Los almacenes que provenga del refinamiento de un proceso, tendrán su propio nombre y su identificación será una nueva letra no usada, seguida de tantos .1 como el nivel de refinamiento en el que aparece el almacén.
 - Los almacenes que provengan del refinamiento de un almacén tendrán su propio nombre y su identificación será la del almacén del que provienen seguido de un punto y un número de orden dentro del refinamiento parcial.



Primitivas para el diseño funcional: Estrategia de diseño mixta

- Paso 4: Reconstrucción del esquema funcional resultante (ascendente):
 - Se colocan todos los refinamientos parciales de elementos (y los elementos no refinados, en su caso), y se conectan las fronteras (conectando con elementos de dentro de ellas) y elementos no refinados aplicando primitivas ascendentes para regenerar las conexiones, siguiendo el patrón del esquema inicial.



Diseño de un esquema funcional

1. Esquema de caja negra:

- Se representa el sistema mediante un sólo proceso, se establecen las interfaces y los flujos de entrada y salida del sistema.

2. Esquema funcional armazón (DFD 0)

3. Si todos los requisitos no están representados:

- a) Refinamientos parciales del DFD n
- b) Unir los refinamientos parciales en el plano de refinamiento funcional DFD $n+1$
- c) Volver al paso 3

4. Comprobar las cualidades del esquema



Cualidades de un esquema funcional

- Independencia funcional: realiza las funciones independientemente
 - ▶ Separable
 - ▶ Fácilmente integrable
 - ▶ Flexible
- Completo: representa todos los detalles procedimentales sin entrar en rasgos procedimentales.
- Correcto: usa las herramientas correctamente.



Cualidades de un esquema funcional

- Legible
- Minimal:
 - ▶ Los almacenes de datos no tienen partes en común.
 - ▶ Cada tarea está en un sólo proceso.

