

A solid red square located to the left of the title text.

Seminario 4

Modelado de Flujo de Datos

Material desarrollado por ...

Fernando Berzal

Ignacio J. Blanco

Francisco J. Cabrerizo

Jesús Campaña

Carlos Cruz

María José Martín

Daniel Sánchez

Contenido

- Diagramas de Flujo de Datos (DFDs)
- Primitivas de refinamiento y transformaciones
- Estrategias de diseño
- Análisis conjunto de datos y funciones orientado a las funciones

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.

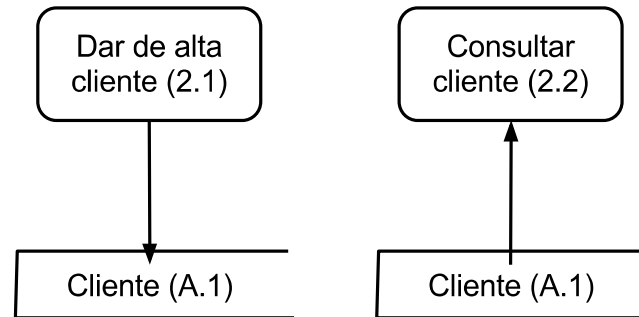


DFD: Almacén de datos

- 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).

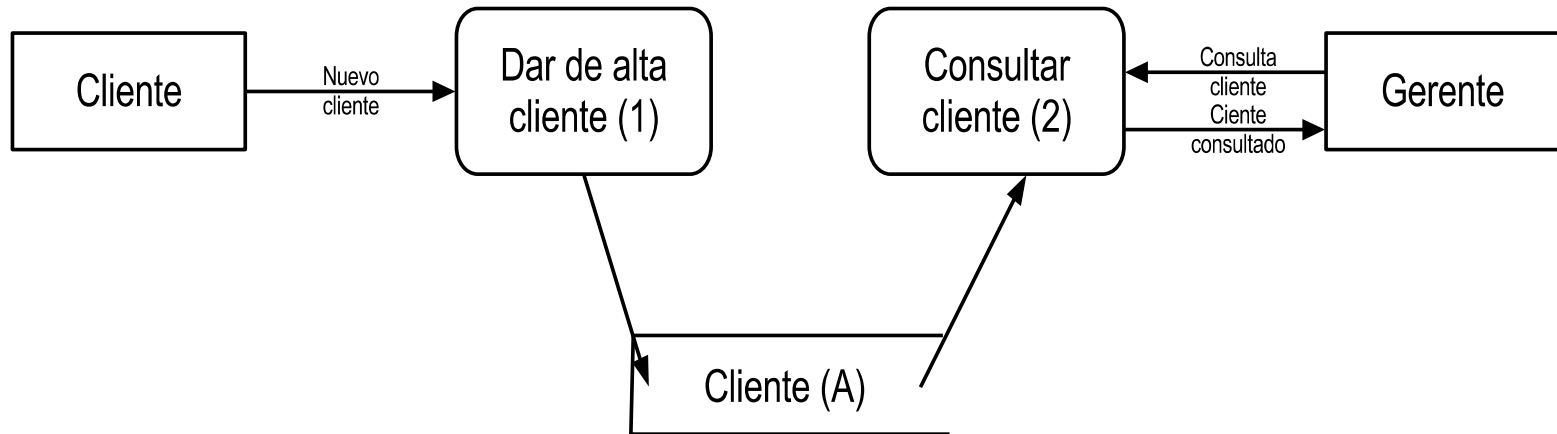
DFD: Interfaz

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

A rectangular box with a black border containing the word "Cliente" in black text, representing an external entity in a Data Flow Diagram.

Cliente

DFD: Un ejemplo

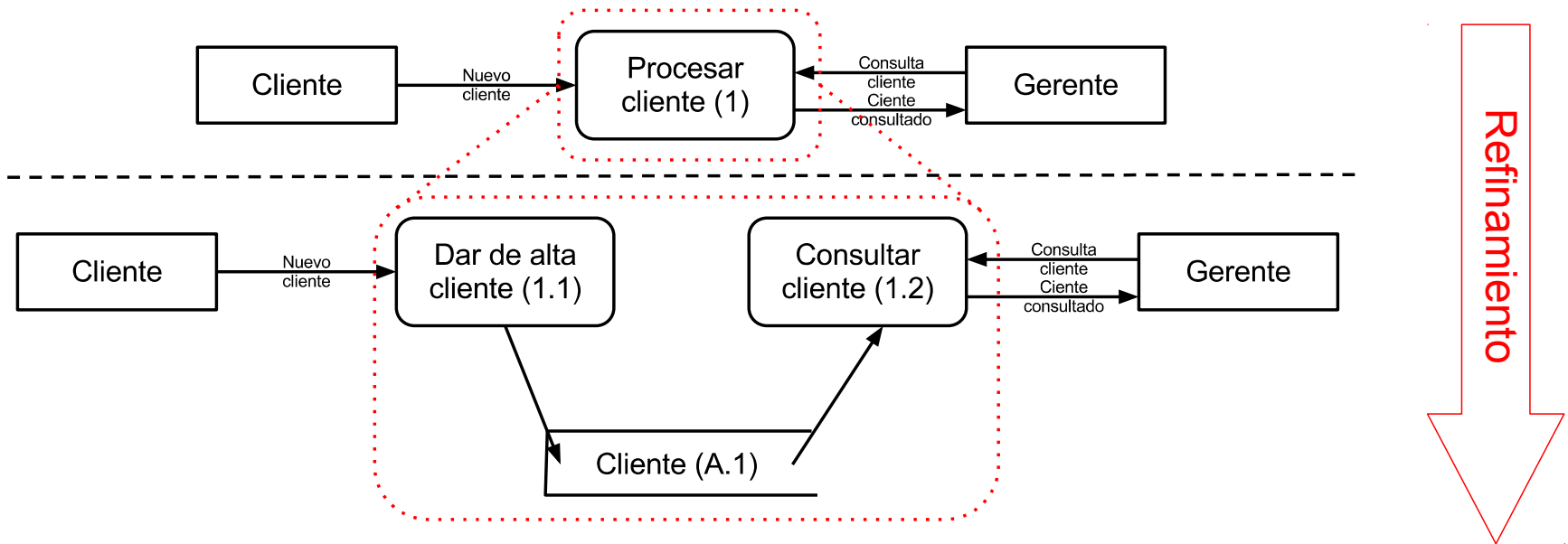


Contenido

- Diagramas de Flujo de Datos (DFDs)
- Primitivas de refinamiento y transformaciones
- Estrategias de diseño
- Análisis conjunto de datos y funciones orientado a las funciones

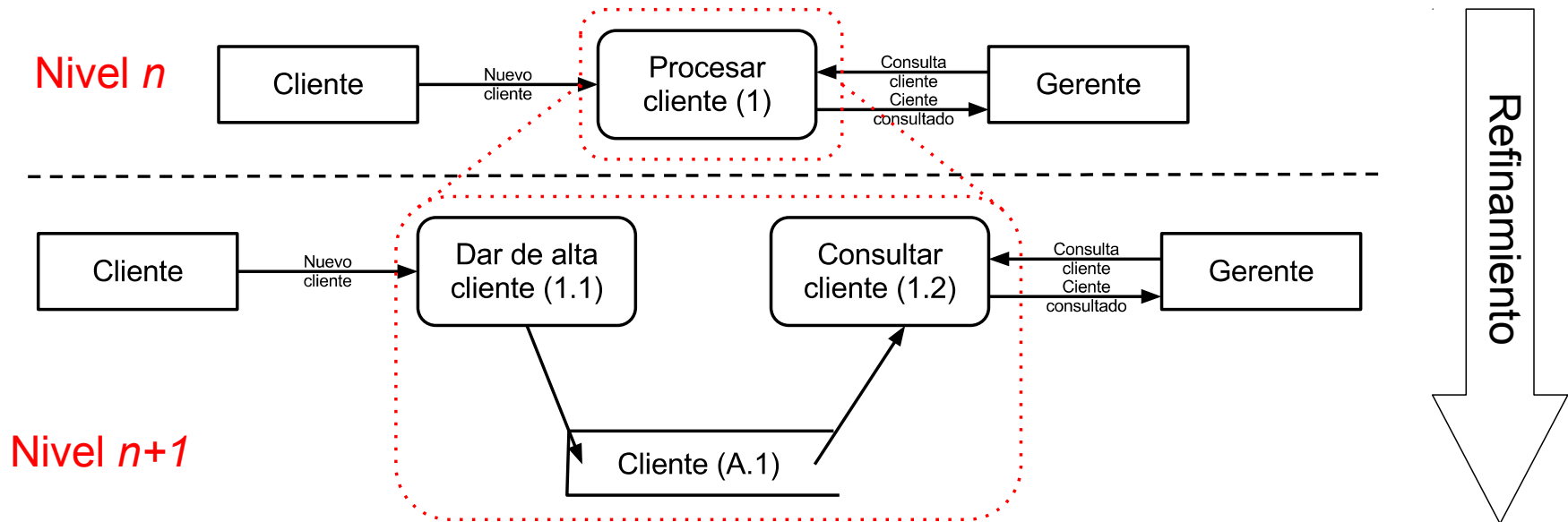
Primitivas para el diseño funcional: refinamiento

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



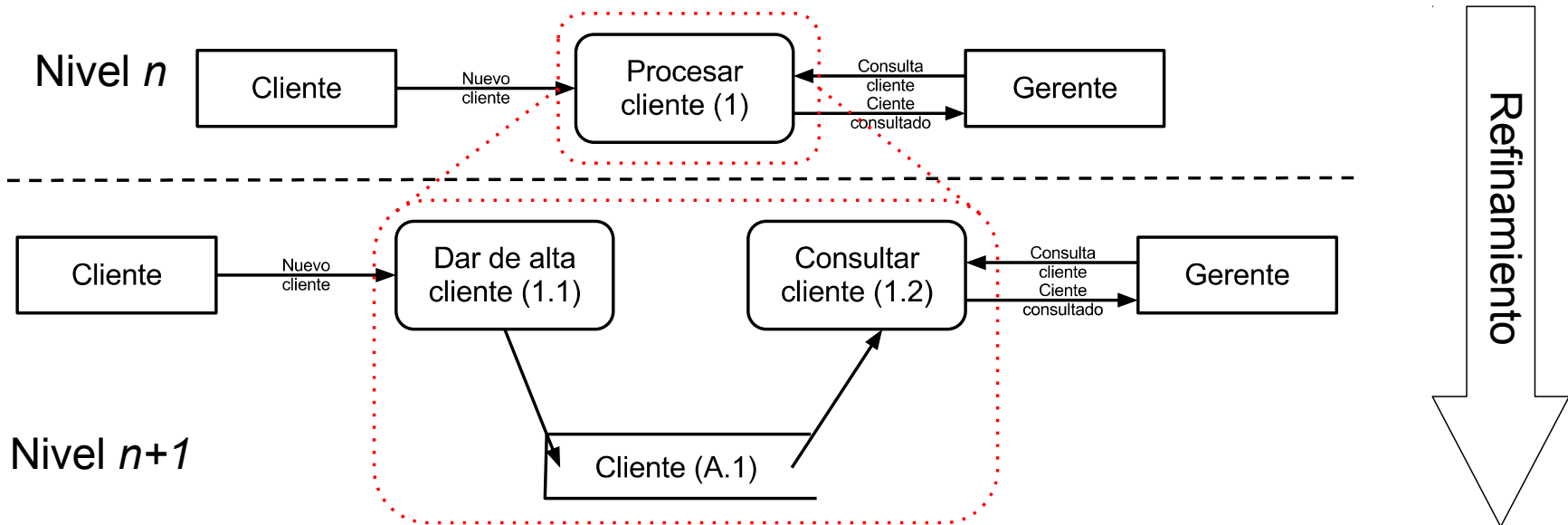
Primitivas para el diseño funcional: niveles de refinamiento

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



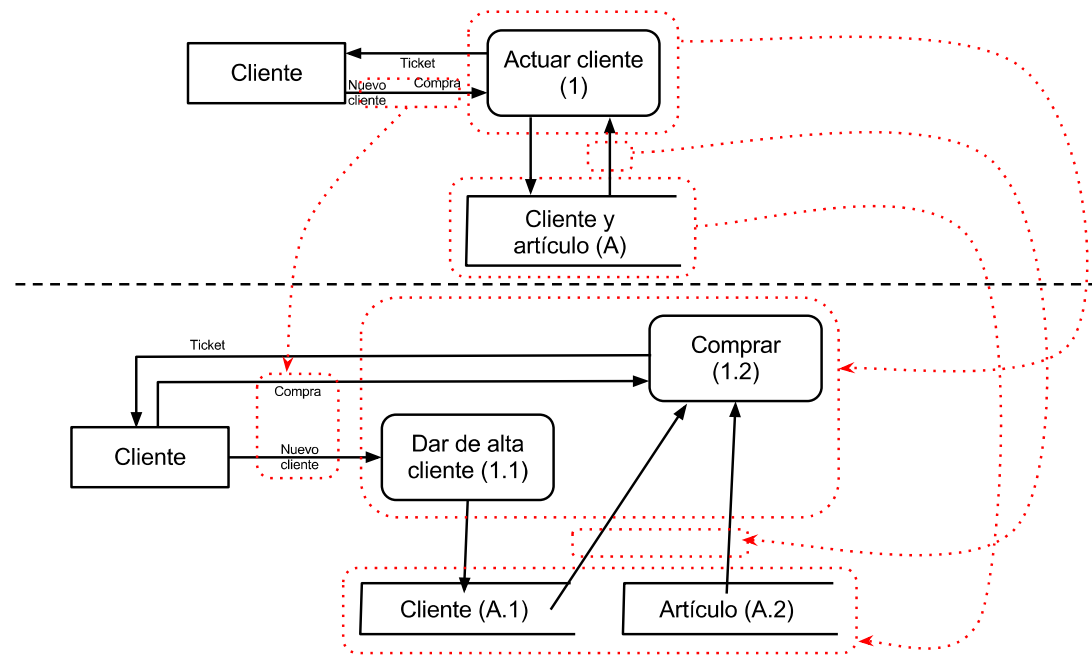
Primitivas para el diseño funcional: primitiva

- 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

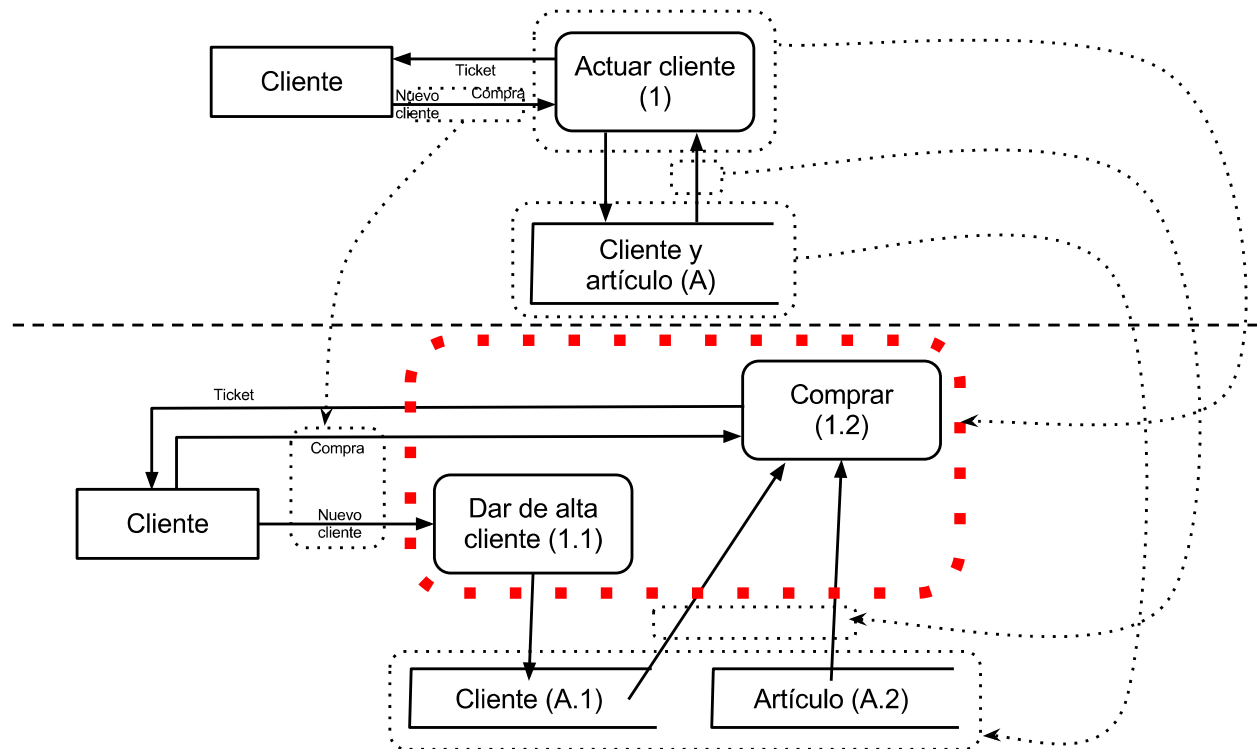
- Al esquema del nivel n en la transformación, se le denomina *esquema inicial*.

Primitivas para el diseño funcional: esquema resultante

- Al esquema final $n+1$ en la transformación, se le denomina *esquema resultante*.

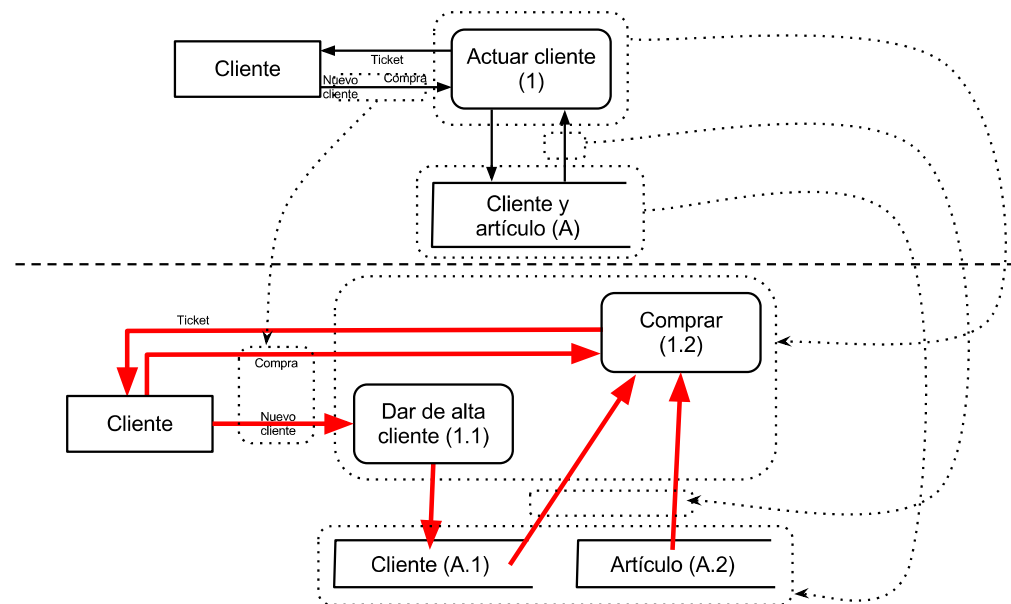
Primitivas para el diseño funcional: normas de transformación

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



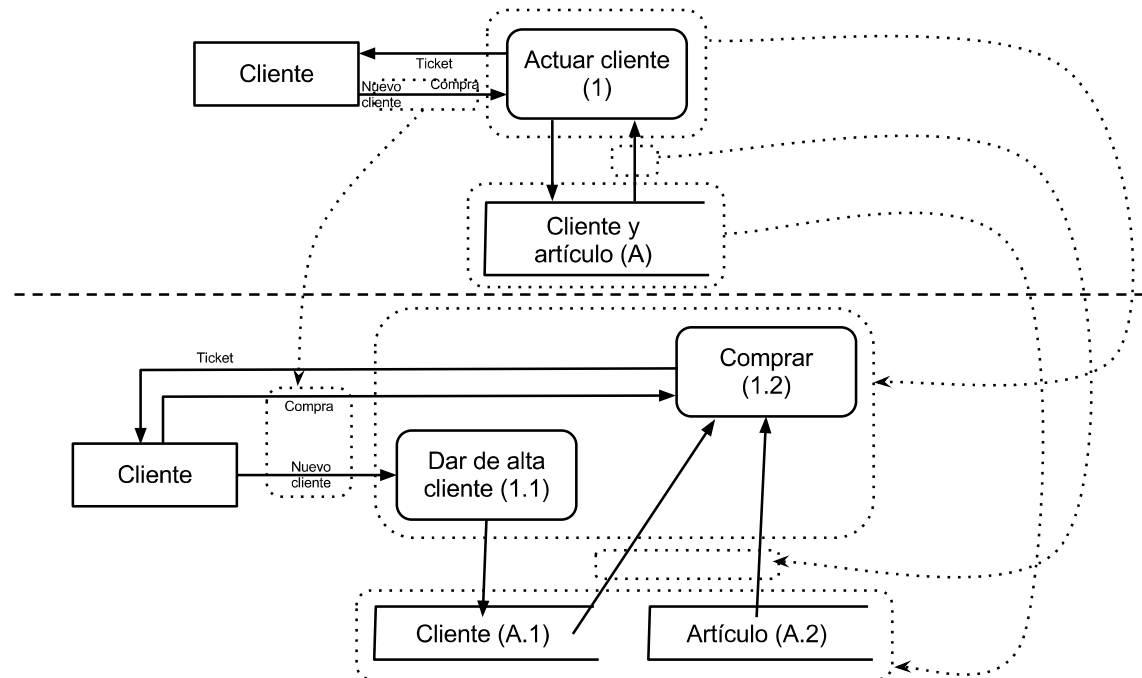
Primitivas para el diseño funcional: normas de transformación

- 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.



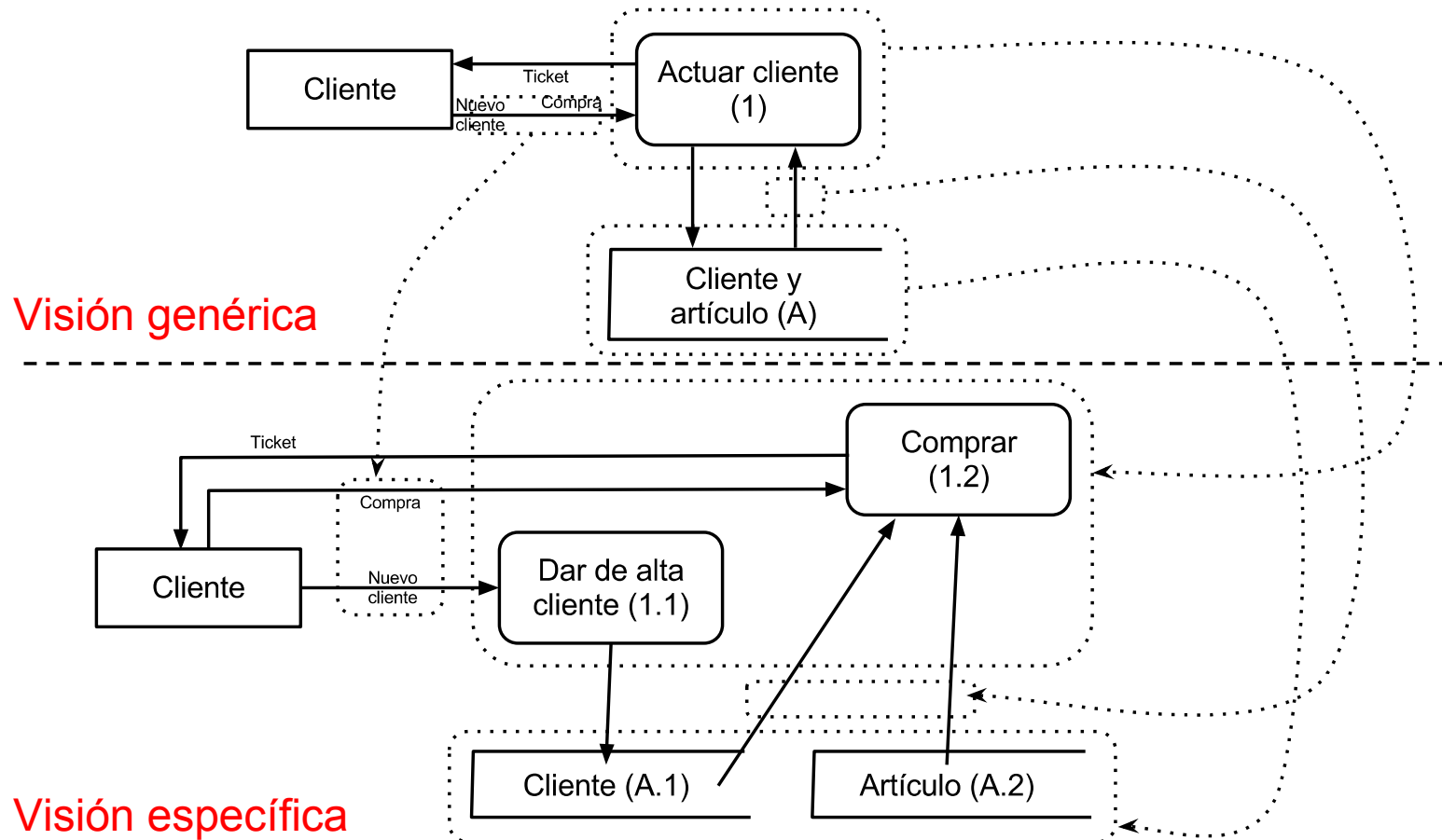
Primitivas para el diseño funcional: tipos

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

Primitivas para el diseño funcional: primitivas descendentes

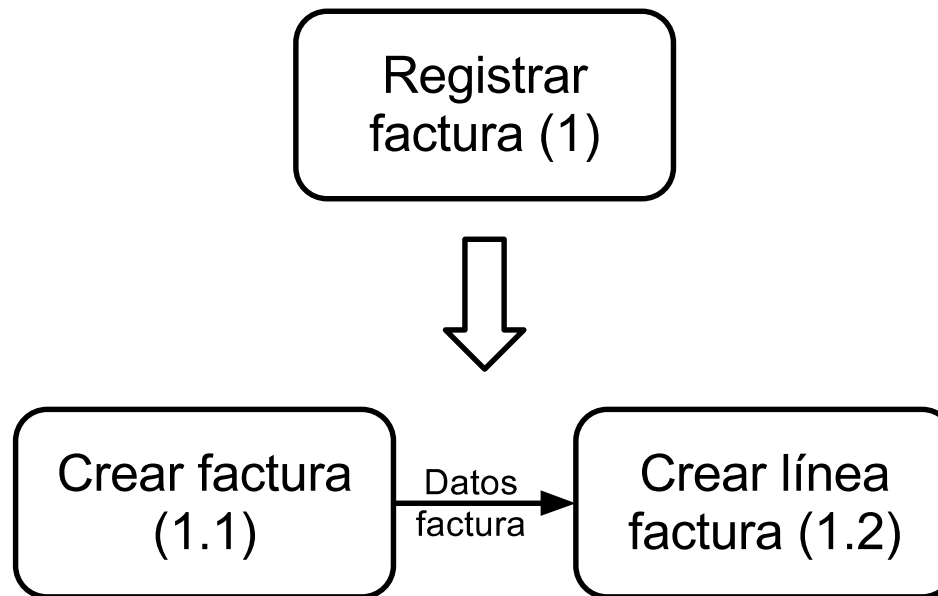
- Permiten llegar desde una visión genérica y abstracta de las funciones de un sistema hasta una versión concreta o específica (como se ha visto en los ejemplos anteriores)

Primitivas para el diseño funcional: primitivas descendentes



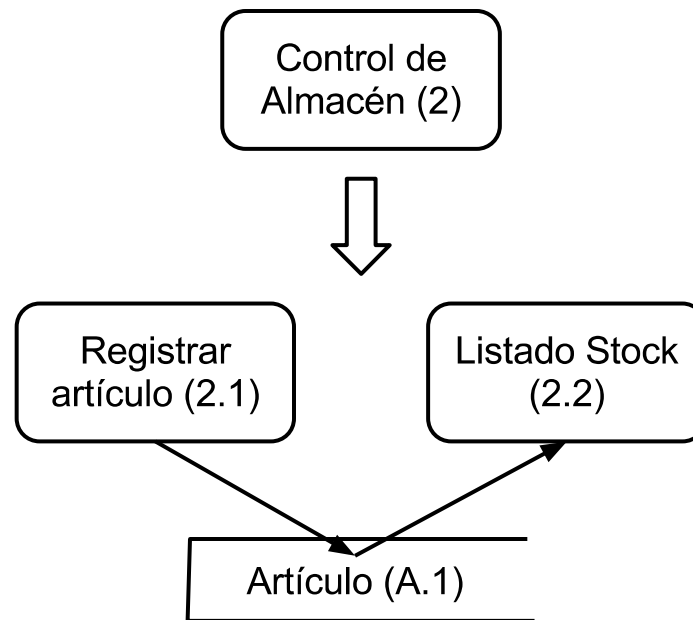
Primitivas para el diseño funcional: primitivas descendentes

- T1: descomposición de proceso en procesos con flujo intermedio



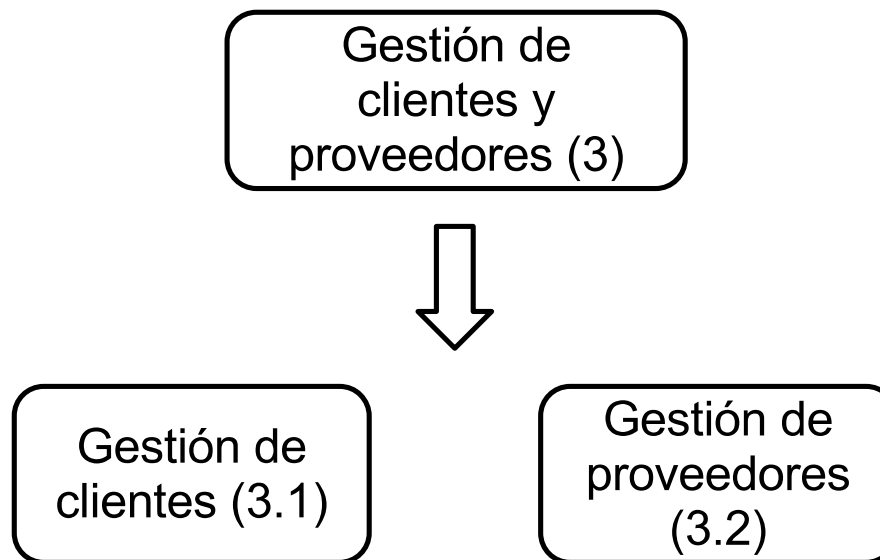
Primitivas para el diseño funcional: primitivas descendentes

- T2: descomposición de proceso en procesos con almacén intermedio



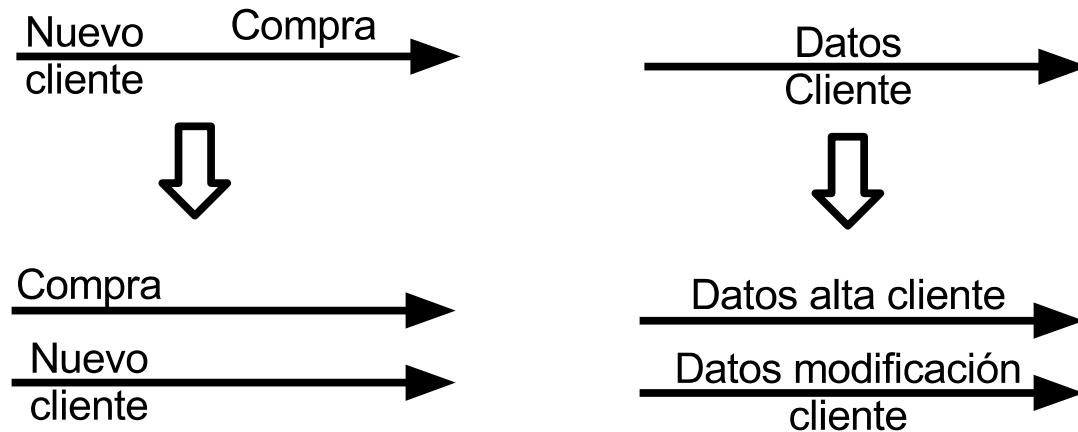
Primitivas para el diseño funcional: primitivas descendentes

- T3: descomposición de proceso en procesos sin conexiones



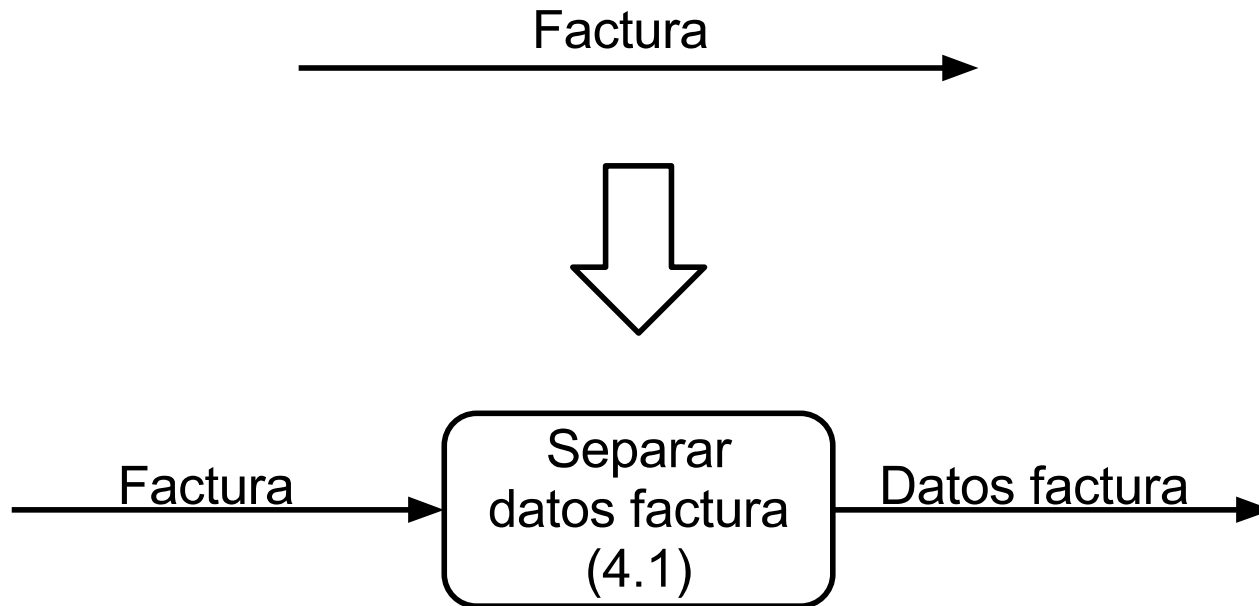
Primitivas para el diseño funcional: primitivas descendentes

- T4: descomposición de flujo



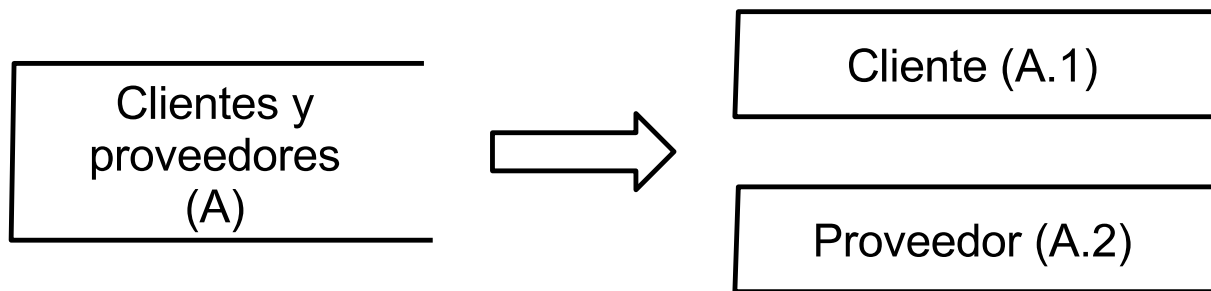
Primitivas para el diseño funcional: primitivas descendentes

- T5: refinamiento de flujo



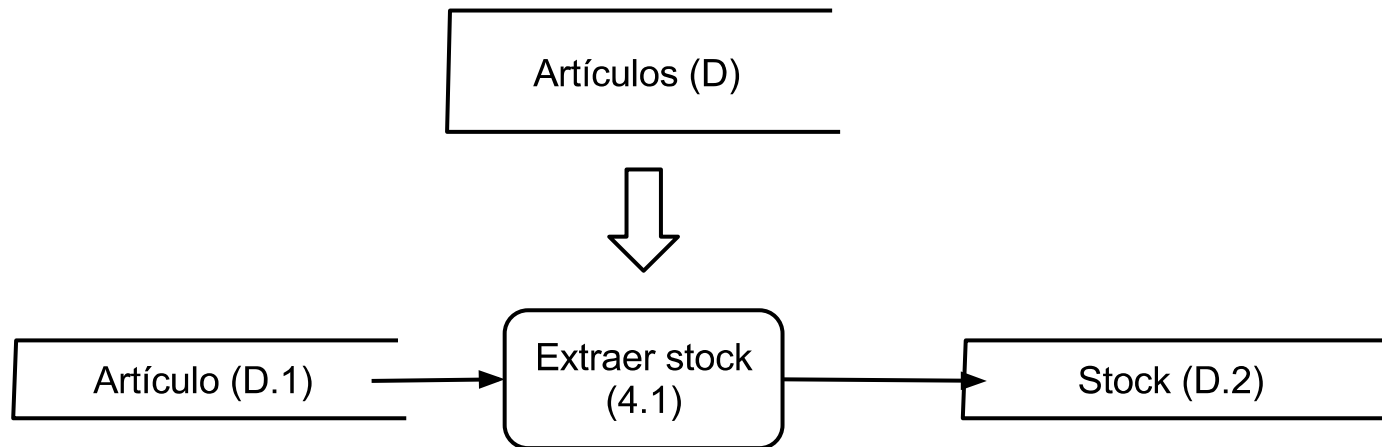
Primitivas para el diseño funcional: primitivas descendentes

- T6: descomposición de almacén



Primitivas para el diseño funcional: primitivas descendentes

- T7: creación de almacén

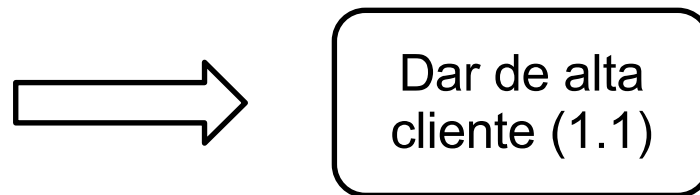


Primitivas para el diseño funcional: primitivas ascendentes

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

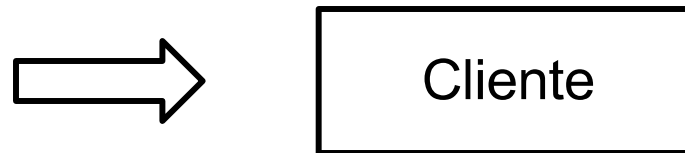
Primitivas para el diseño funcional: primitivas ascendentes

- B1: de generación de proceso



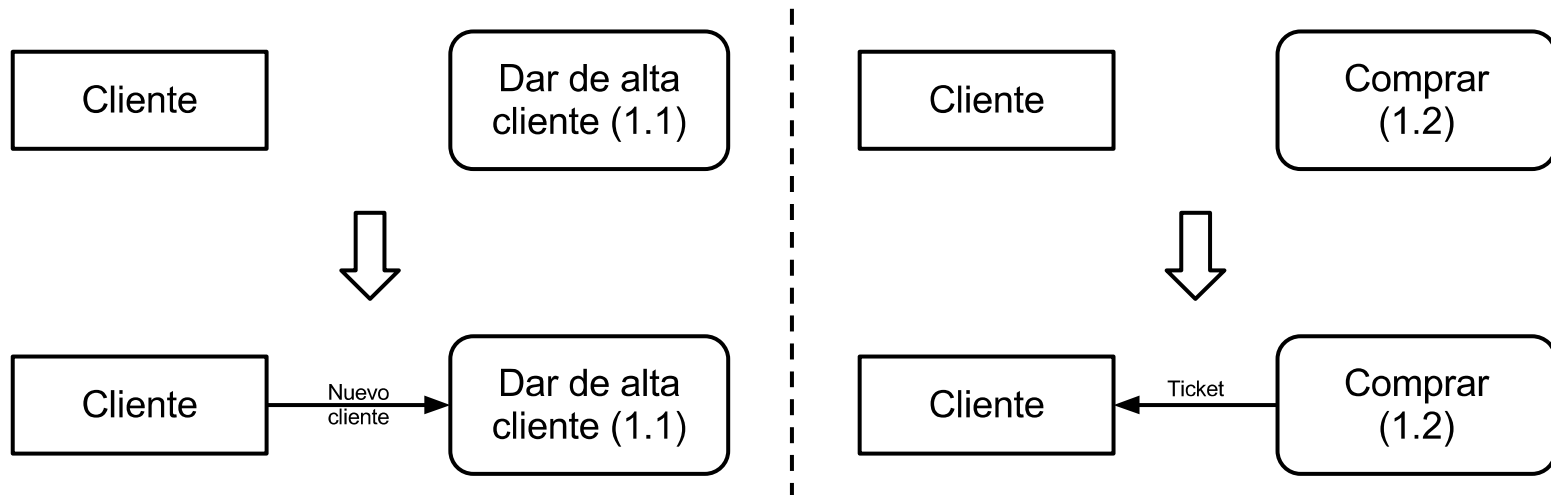
Primitivas para el diseño funcional: primitivas ascendentes

- B2: de generación de interfaz



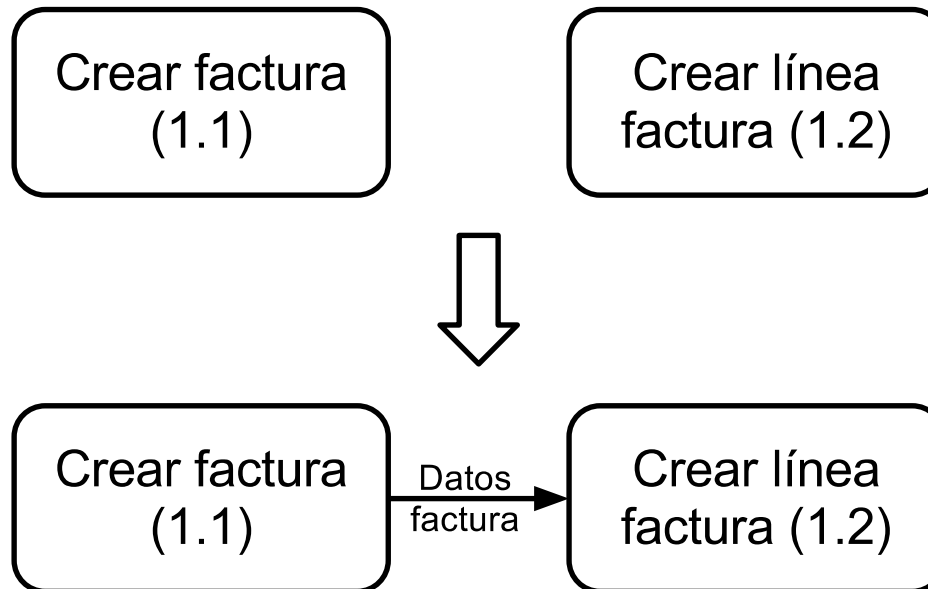
Primitivas para el diseño funcional: primitivas ascendentes

- B3: de generación de flujo entre interfaz y proceso



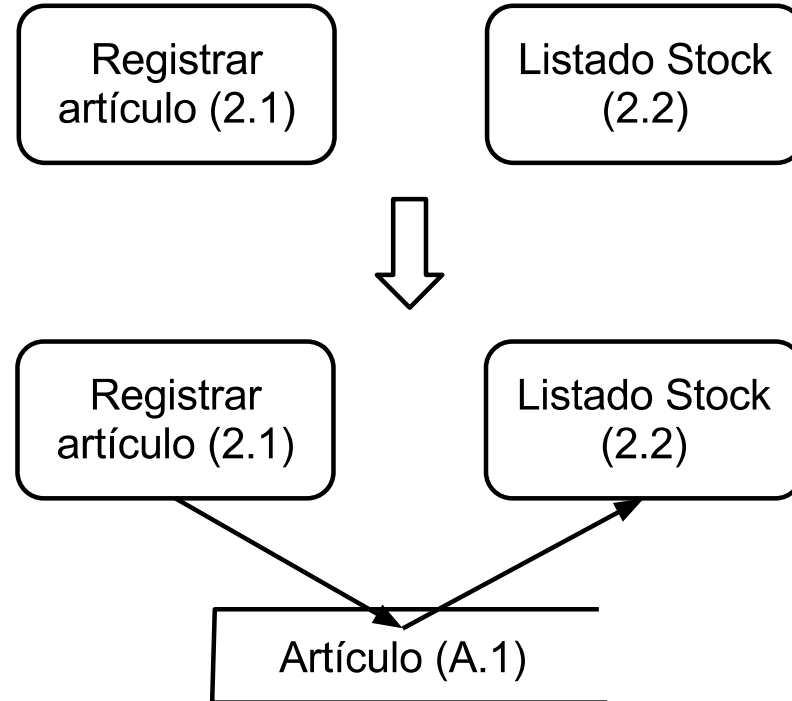
Primitivas para el diseño funcional: primitivas ascendentes

- B4: de generación de flujo entre procesos



Primitivas para el diseño funcional: primitivas ascendentes

- B5: de generación de almacén entre procesos



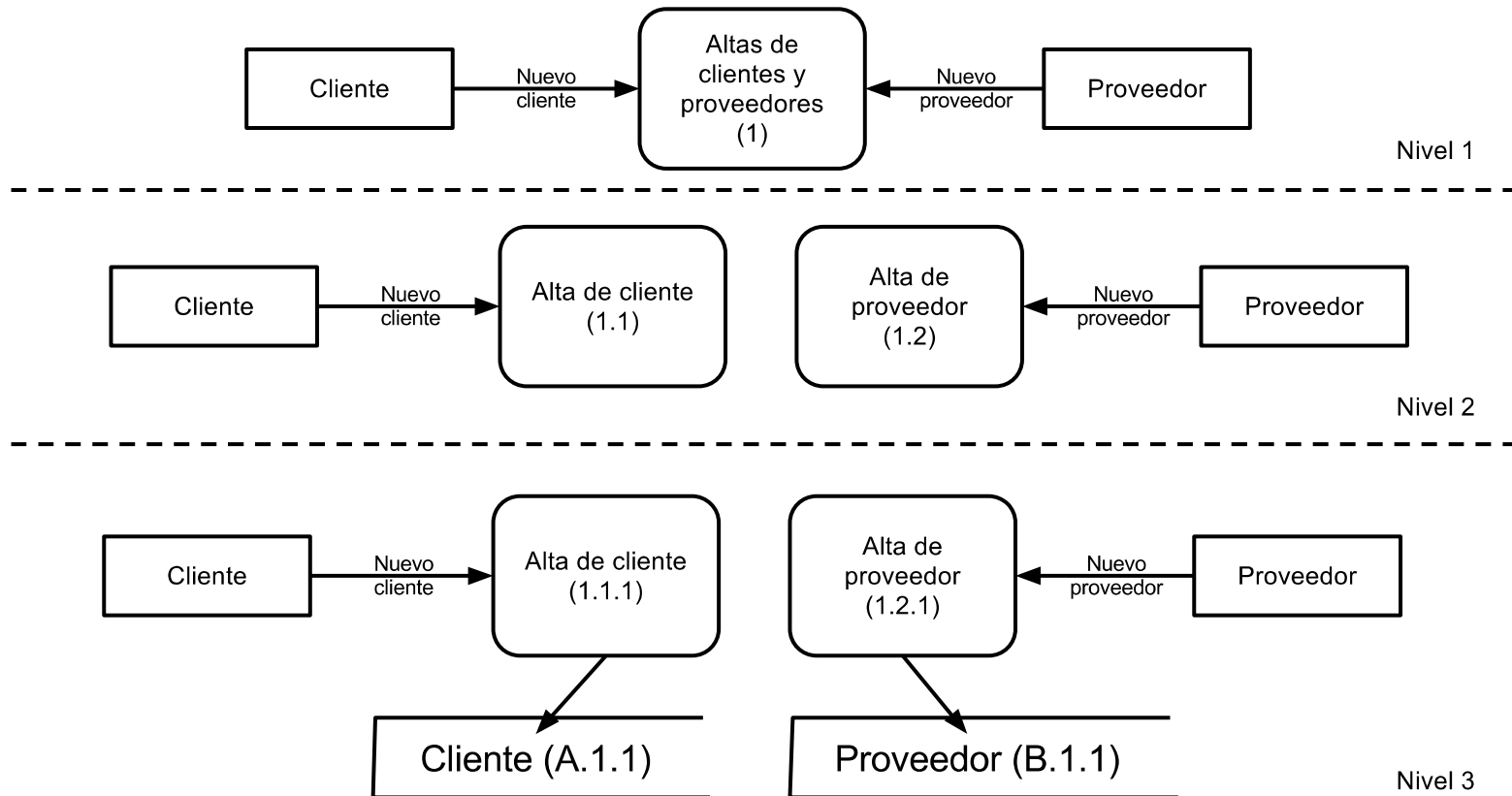
Contenido

- Diagramas de Flujo de Datos (DFDs)
- Primitivas de refinamiento y transformaciones
- Estrategias de diseño
- Análisis conjunto de datos y funciones orientado a las funciones

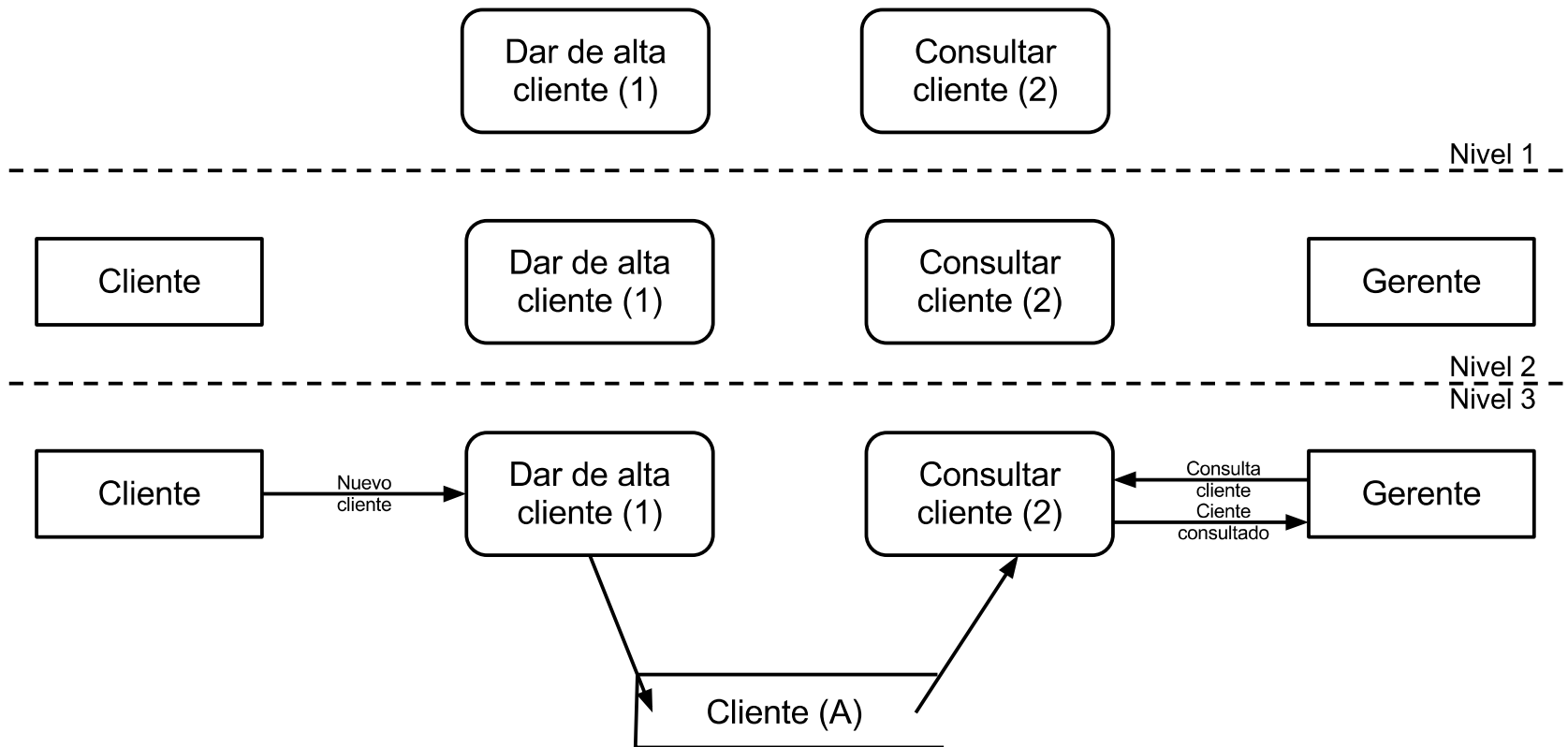
Primitivas para el diseño funcional: Estrategia de diseño

- 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



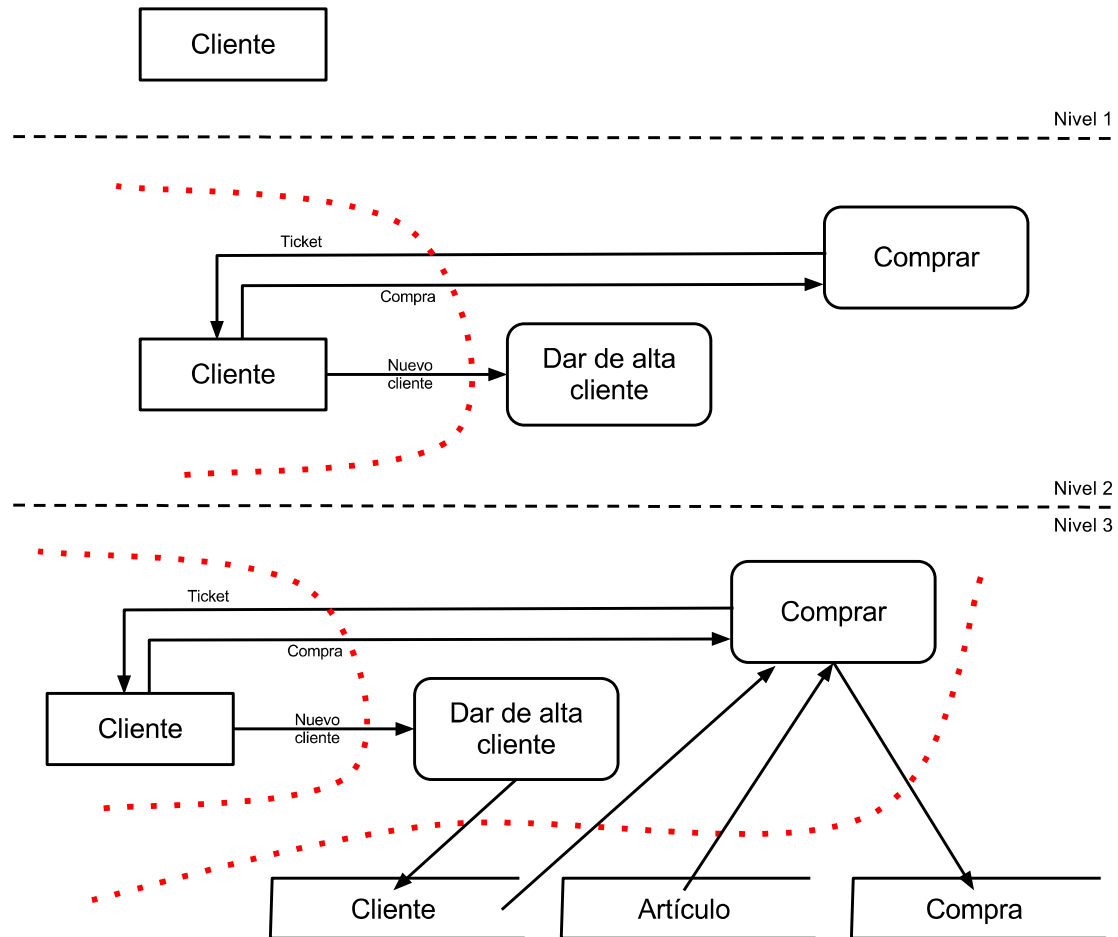
Primitivas para el diseño funcional: Estrategia de diseño ascendente



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.

Contenido

- Diagramas de Flujo de Datos (DFDs)
- Primitivas de refinamiento y transformaciones
- Estrategias de diseño
- Análisis conjunto de datos y funciones orientado a las funciones

Análisis conjunto de datos y funciones

- Refinamiento mutuo: análisis conjunto de datos y funciones
- Esquema externo
- Metodología para el análisis conjunto de datos y funciones
- Sugerencias para refinamientos mutuos
- Operaciones de datos
- Esquemas de operación
- Esquemas de navegación

Refinamiento mutuo: análisis conjunto de datos y funciones

- Se basa en el desarrollo simultáneo de la componente funcional y conceptual del Sistema de Información, desarrollando una a la vez que la otra.
- La forma de conectar ambos refinamientos se centra en el concepto de **esquema externo**.
- Al esquema funcional se le conoce como esquema F.
- Al esquema conceptual se le conoce como esquema D.

Análisis conjunto de datos y funciones

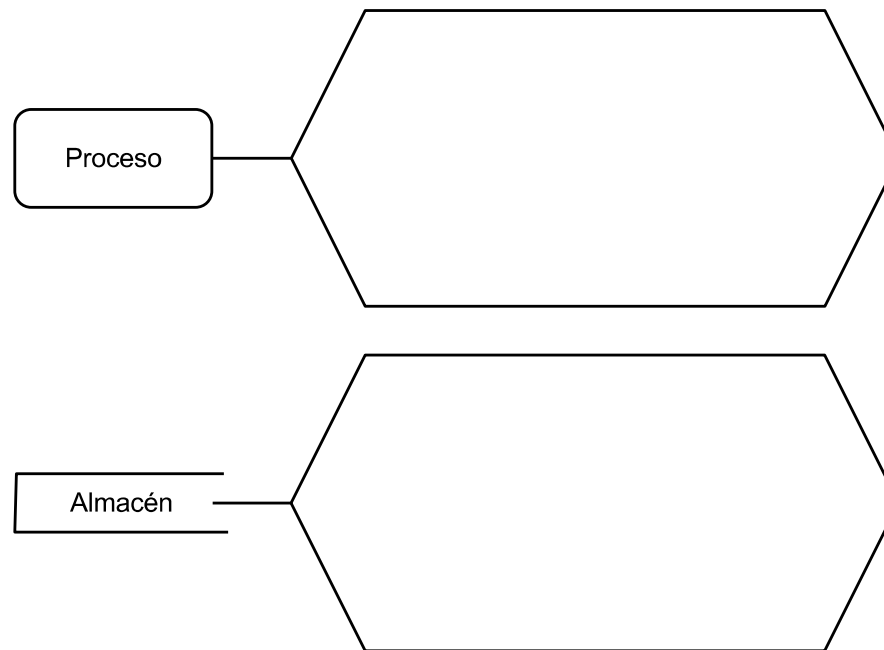
- Refinamiento mutuo: análisis conjunto de datos y funciones
- Esquema externo
- Metodología para el análisis conjunto de datos y funciones
- Sugerencias para refinamientos mutuos
- Operaciones de datos
- Esquemas de operación
- Esquemas de navegación

Esquema externo: concepto

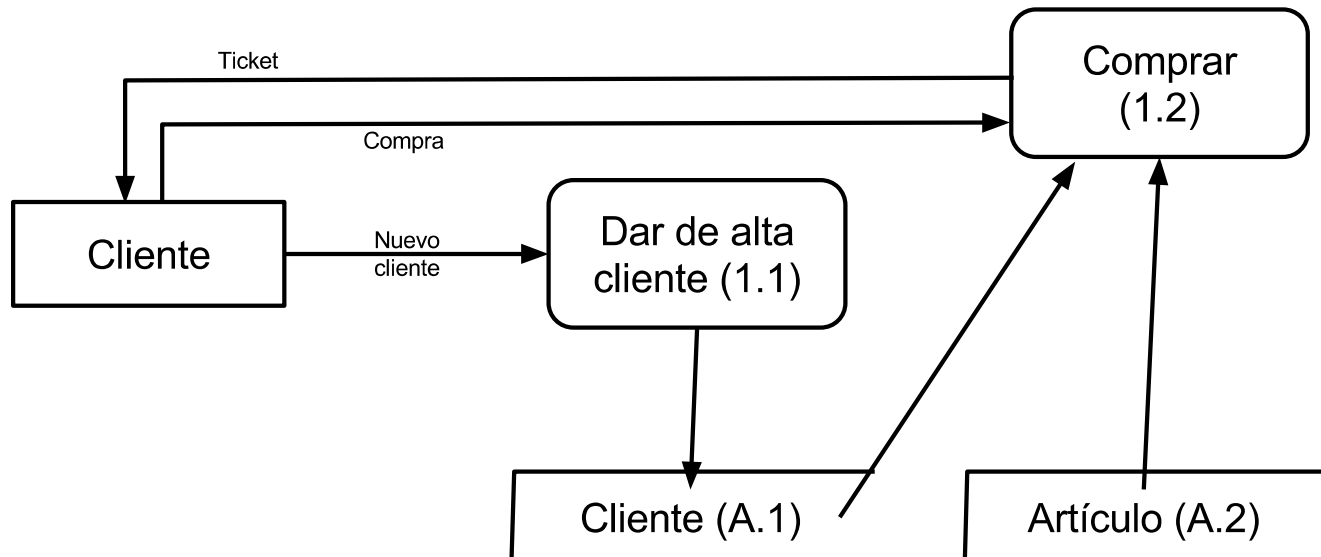
- Se llama *esquema externo* a una vista particular de una base de datos por parte de una aplicación.
- En el entorno de esta metodología, un *esquema externo* es:
 - La parte del esquema entidad-relación del sistema con el que trabaja un proceso, o
 - La parte del esquema entidad-relación del sistema que almacena un almacén.

Esquema externo: representación

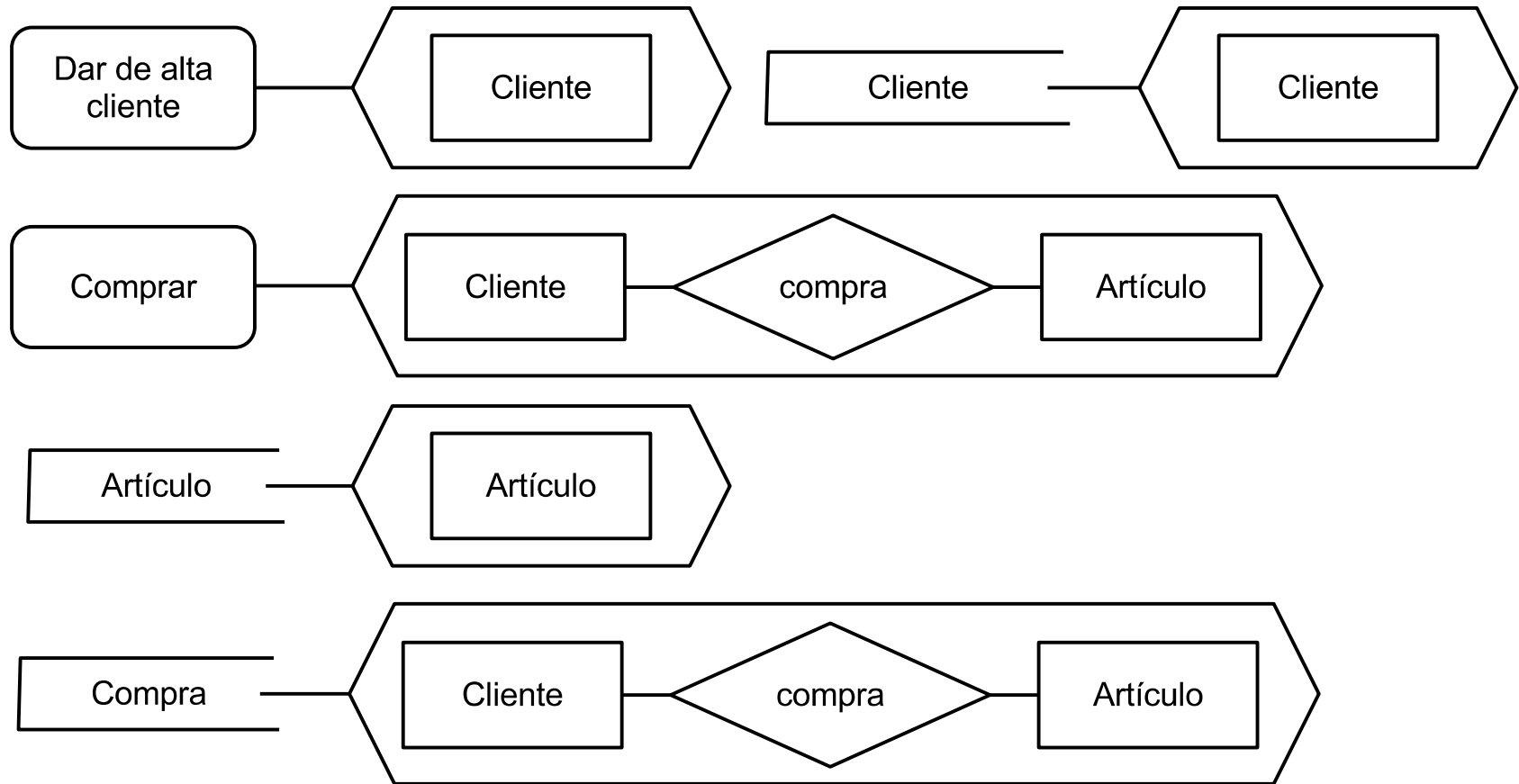
- Gráficamente, se representa dentro de un hexágono:



Esquema externo: un ejemplo



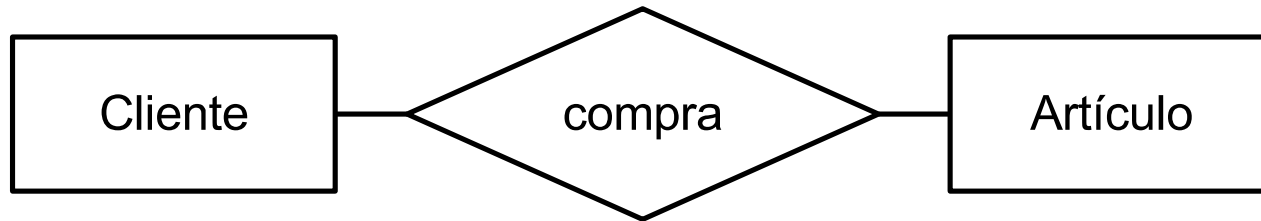
Esquema externo: un ejemplo



Esquema externo: ventajas

- Uniendo todos los esquemas externos (de procesos y almacenes), se obtiene el esquema conceptual del sistema.
- Ayuda a la verificación de la compleción del esquema conceptual.
- Permite definir de forma sencilla las operaciones de datos que aparecen en el diseño detallado funcional.

Esquema externo: ventajas



Metodología para el análisis conjunto de datos y funciones

- Usa las estrategias de refinamiento funcional y conceptual de forma conjunta para el refinamiento mutuo del sistema de información.

Metodología para el análisis conjunto de datos y funciones: variantes

- El análisis conjunto de datos y funciones **orientado a los datos** se centra en el refinamiento del esquema conceptual, añadiendo las funciones que trabajan con cada parte del mismo.
- El análisis conjunto de datos y funciones **orientado a las funciones** se centra en el refinamiento del esquema funcional, generando los datos con los que trabaja cada una de las mismas. Nos centraremos en este.

Metodología para el análisis conjunto orientado a las funciones: pasos

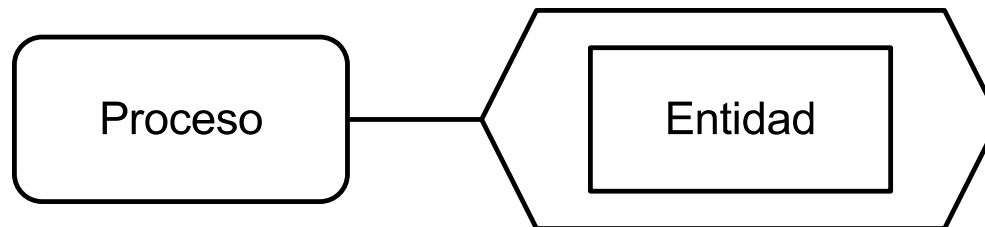
1. Esquema de caja negra
2. Esquema F almacén (DFD 0)
3. Esquemas externos para los elementos del DFD 0
4. Esquema D almacén (uniendo los esquemas externos)
5. Comprobar corrección de esquemas F y D
6. Si todos los requisitos no están representados:
 - a) Refinamientos parciales del DFD n
 - b) Unir los refinamientos parciales en el plano de refinamiento F DFD $n+1$
 - c) Esquemas externos para los elementos del DFD $n+1$
 - d) Plano de refinamiento D $n+1$ (uniendo los esquemas externos)
 - e) Comprobar corrección de esquemas F y D
 - f) Volver al paso 6

Metodología para el análisis conjunto orientado a las funciones: corrección de esquemas

- Los diagramas DFD deben ser independientes funcionalmente, es decir, cada tarea sólo es realizada por un proceso.
- Los diagramas E/R tienen que ser minimales, es decir, cada elemento sólo puede estar en una entidad, relación o atributo.

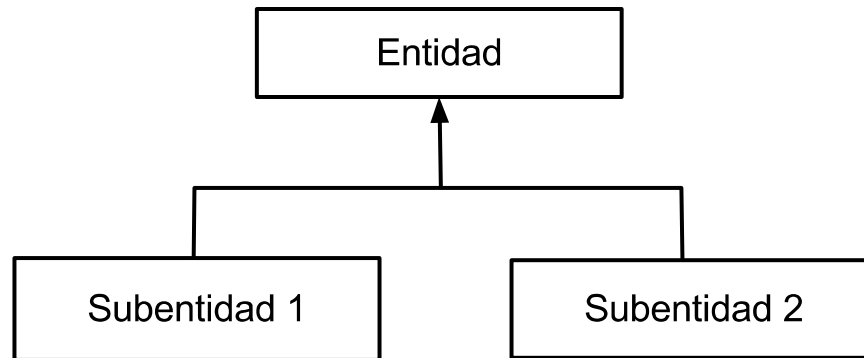
Sugerencias para refinamientos mutuos

Supongamos que tenemos un proceso con un esquema externo tal que:



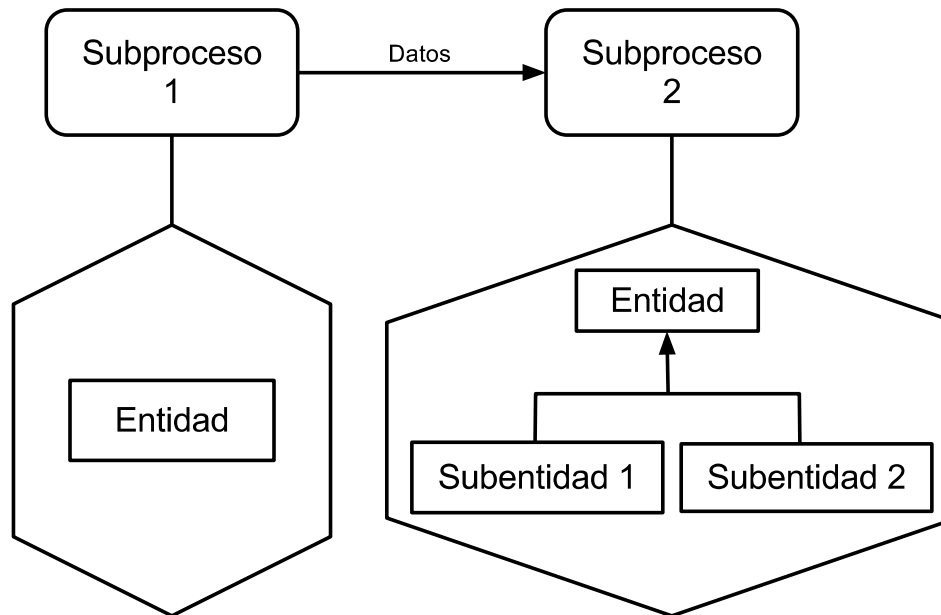
Sugerencias para refinamientos mutuos

... en la que la entidad tiene que ser refinada como sigue:



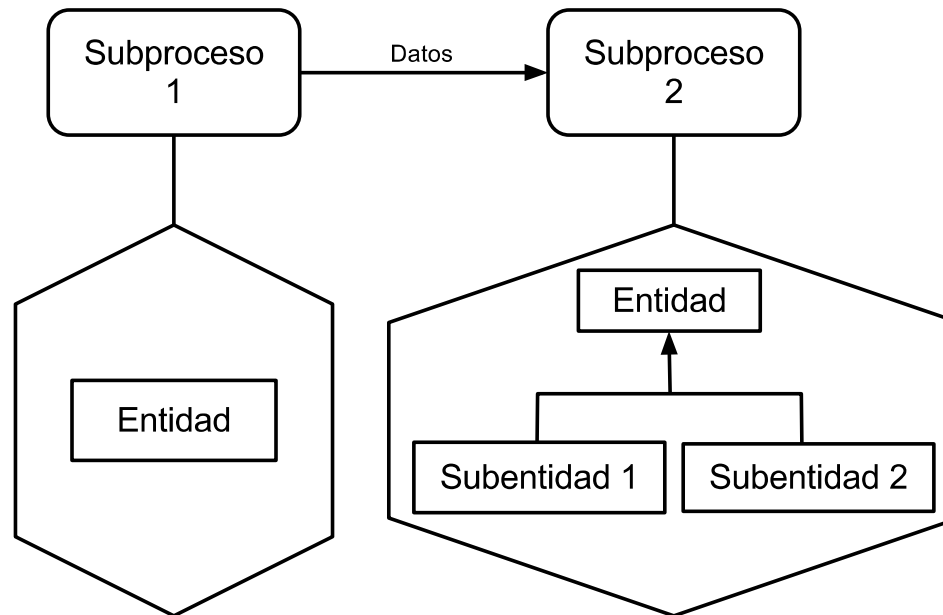
Sugerencias para refinamientos mutuos

Podríamos pensar que el refinamiento funcional consecuente sería:



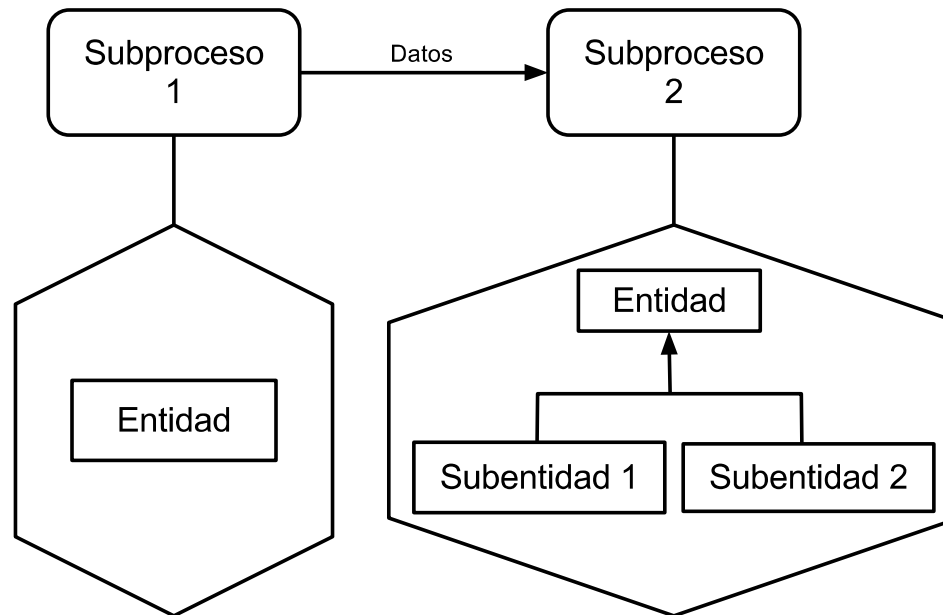
Sugerencias para refinamientos mutuos

El subproceso 1 se encarga de la entidad generalizada...



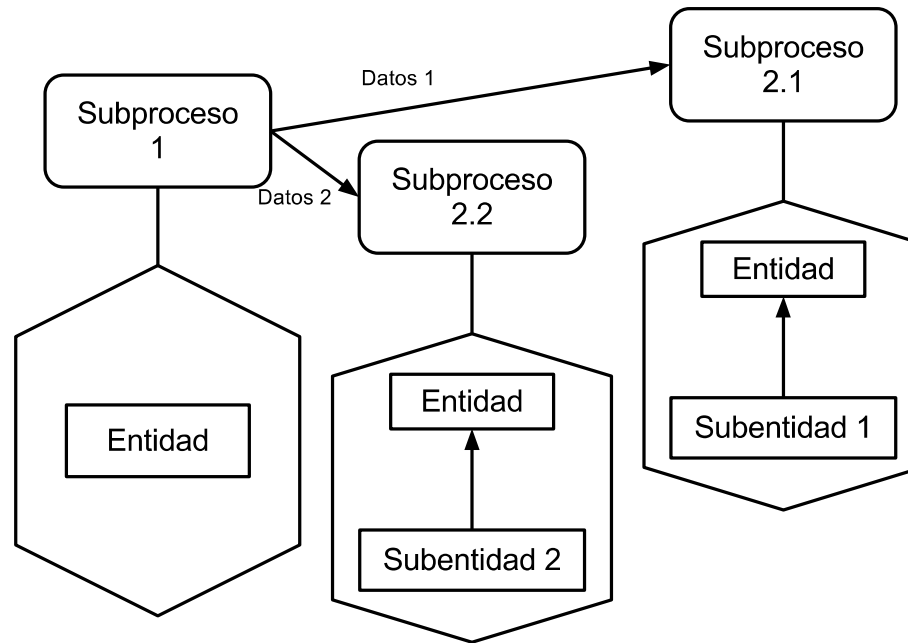
Sugerencias para refinamientos mutuos

... y el subproceso 2 se encarga de las entidades especializadas, pero hace dos cosas...

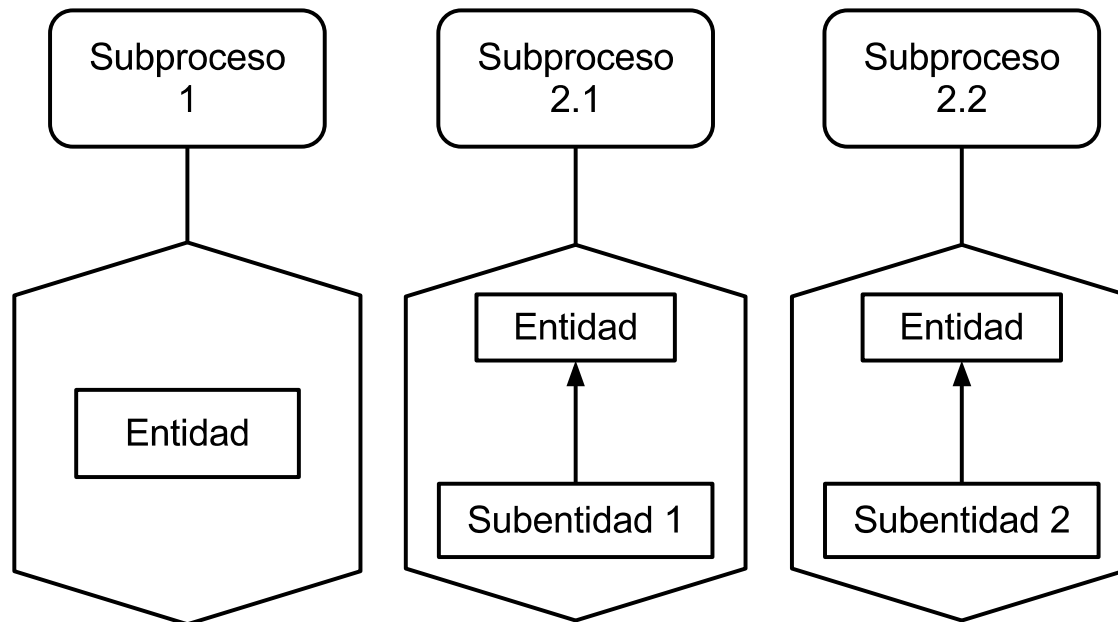


Sugerencias para refinamientos mutuos

... por lo que necesitará un refinamiento posterior:



Sugerencias para refinamientos mutuos



Cada subproceso se encarga de una única entidad.

Operaciones de datos: concepto

- Una vez finalizado el diseño funcional (alcanzado el DFD final), es necesario describir procedimentalmente cada proceso incluido en dicho esquema funcional.
- Las sentencias de esas descripciones procedimentales que acceden a los almacenes se conocen como **operaciones de datos**

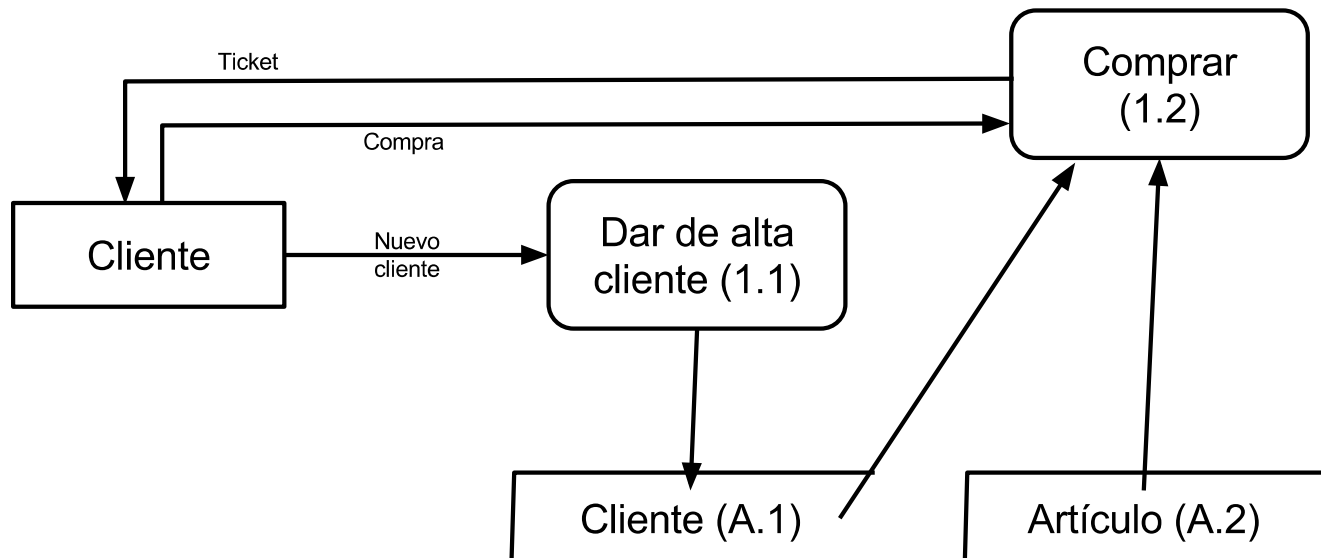
Operaciones de datos: lista de operaciones de datos

- Informalmente hablando, por cada proceso del DFD final que tenga un flujo con un almacén, hay una o más operaciones de datos de ese proceso con ese almacén.
- Teniendo esto en cuenta, es necesario realizar una lista de las operaciones de datos que incluye un sistema de información.

Operaciones de datos: lista de operaciones de datos

- Cada operación de datos tiene que estar completamente descrita, indicando qué operación se realiza sobre qué entidad, entidades, relación o relaciones, a partir de qué atributos y con qué valores de atributos.
- En el caso de una base de datos, una operación de datos tiene que tener todo lo necesario para la construcción de una consulta.

Operaciones de datos: un ejemplo



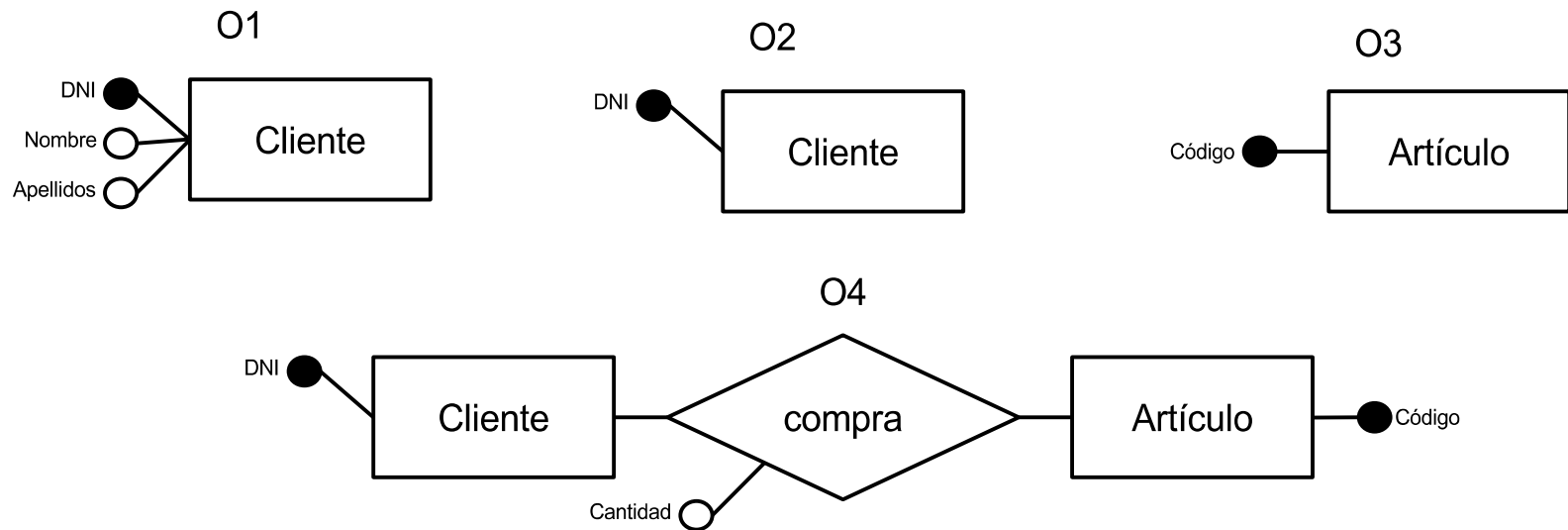
Operaciones de datos: un ejemplo

- Lista de operaciones:
 - O1: Insertar un nuevo cliente a partir de su DNI, nombre y apellido.
 - O2: Buscar cliente por su DNI
 - O3: Buscar artículo por su código de artículo
 - O4: Insertar una nueva compra a partir del DNI del cliente, el código del artículo y la cantidad comprada.

Esquemas de operación: concepto

- Un **esquema de operación** es la parte del esquema externo del proceso que realiza la operación al que afecta dicha operación. Debe incluir las entidades, relaciones y atributos especificadas en dicha operación de datos y ninguno más.

Esquemas de operación: un ejemplo



Esquemas de navegación: concepto

- El **esquema de navegación** para una operación de datos consiste en marcar su esquema de operación, indicando cómo se realiza la operación en cuestión.
- Es necesario marcar entidades, relaciones, atributos y arcos que conectan relaciones y entidades.

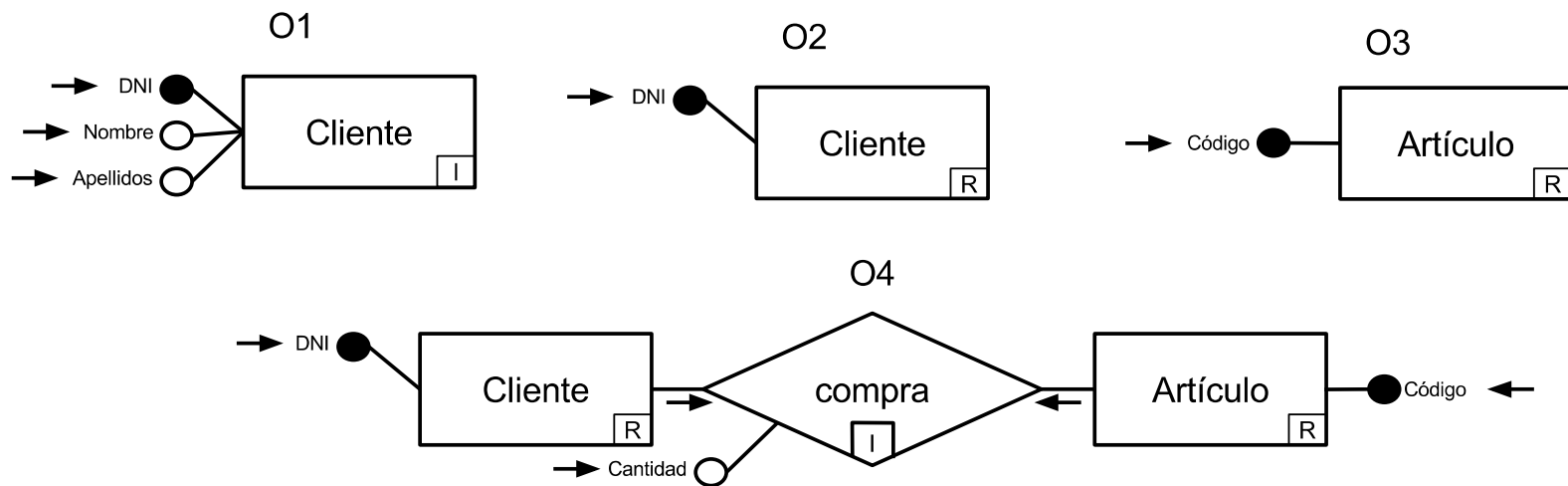
Esquemas de navegación: concepto

- **Entidades y relaciones:** dependiendo de la operación que se realice sobre la entidad, esta puede estar marcada con una *I* (inserción), una *D* (borrado), una *U* (actualización) o una *R* (consulta).
- **Atributos:** un *atributo de entrada* (se proporciona un valor para él) se marca con una flecha; un *atributo de salida* (se obtiene su valor) es aquel que no tiene flecha.

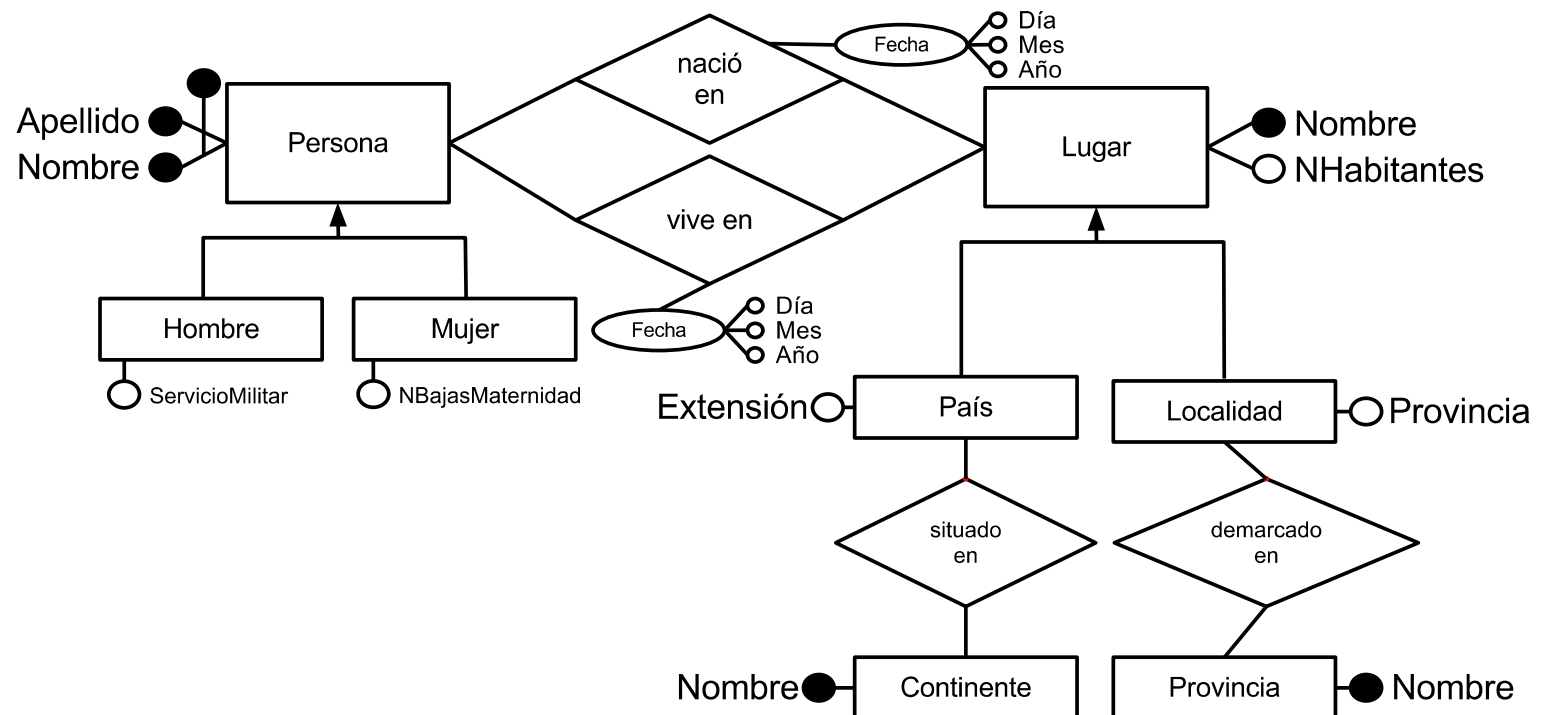
Esquemas de navegación: concepto

- **Conexiones entre entidades y relaciones:** se marcan con una flecha para indicar el sentido en el que se recorren.

Esquemas de navegación: un ejemplo



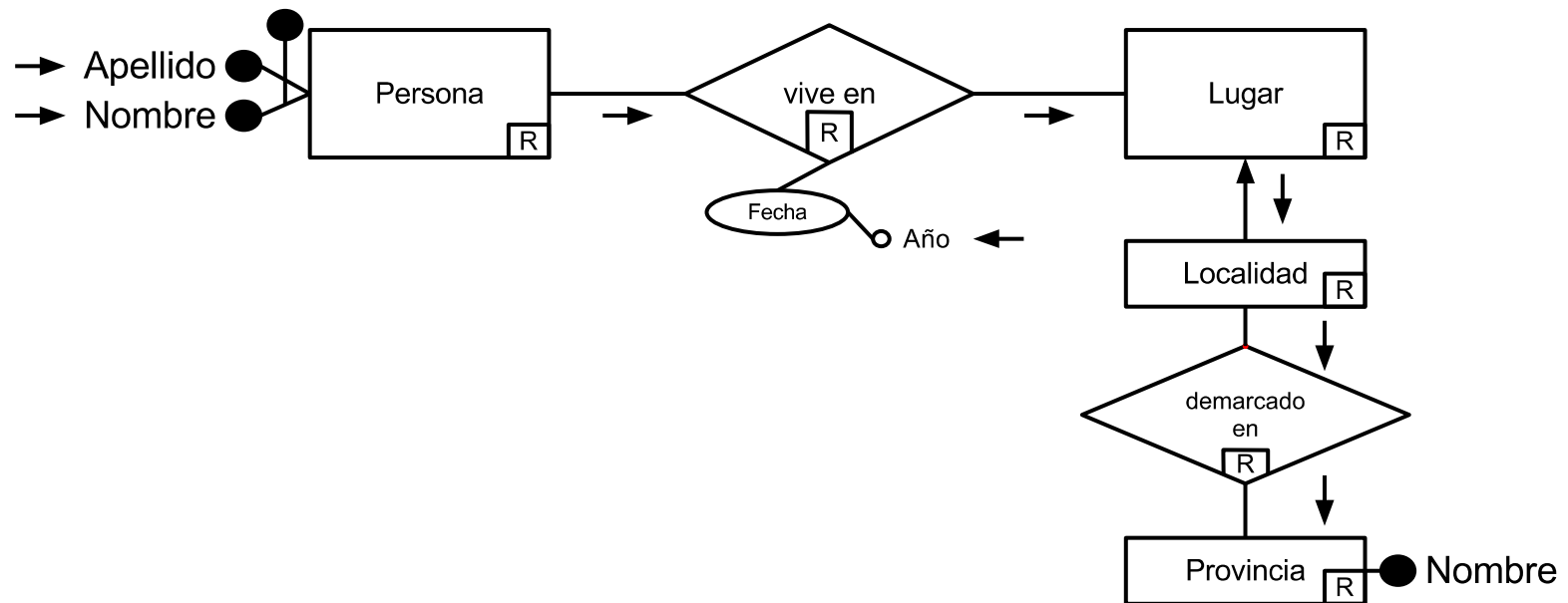
Esquemas de navegación: otro ejemplo



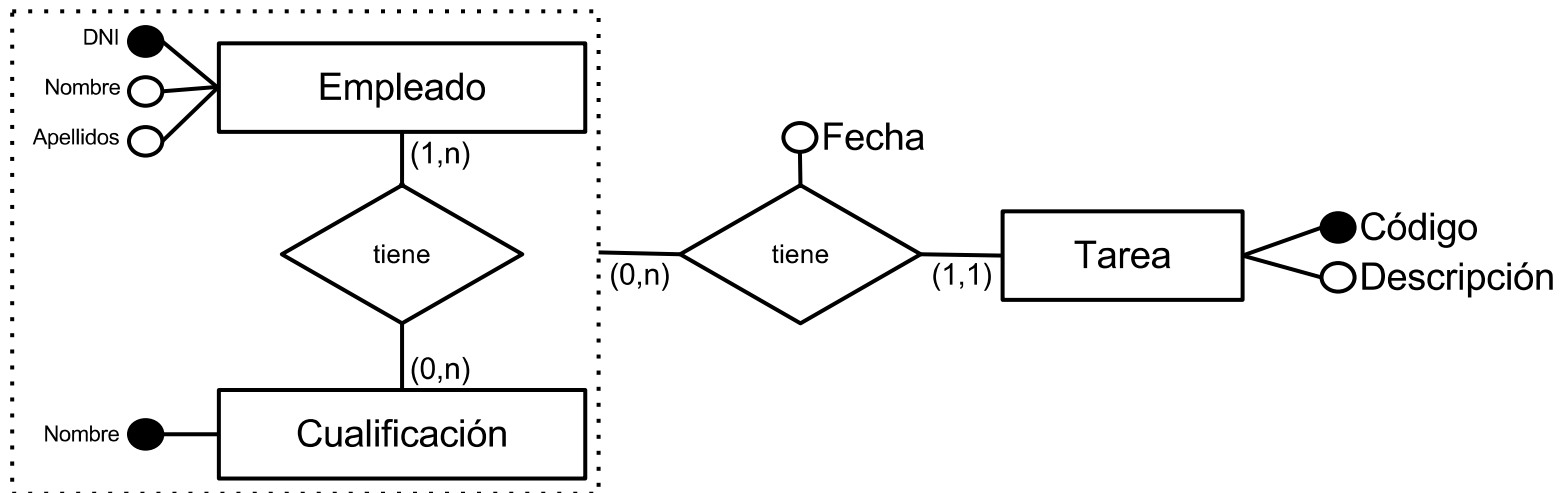
Esquemas de navegación: otro ejemplo

- Operación de datos:
 - Encontrar el nombre de las provincias en las que vive o ha vivido la persona de nombre “Juan Martínez” en el año 2010

Esquemas de navegación: otro ejemplo



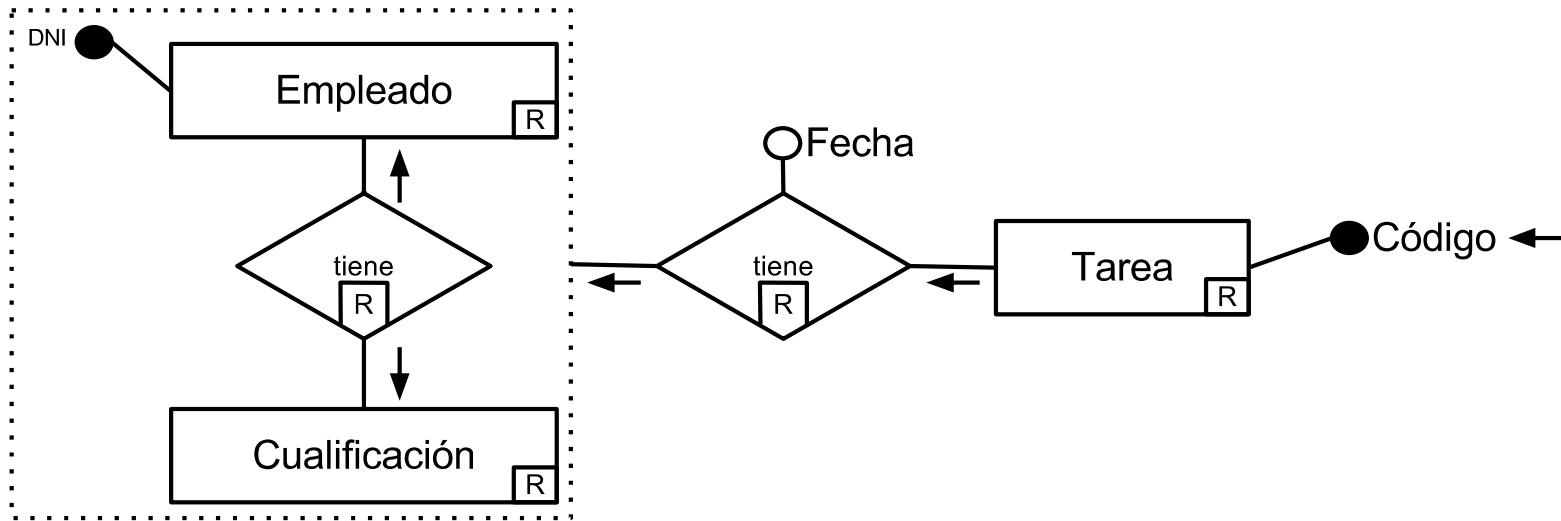
Esquemas de navegación: otro ejemplo más



Esquemas de navegación: otro ejemplo más

- Operación de datos:
 - Encontrar los DNIs de los empleados a los que se ha asignado una tarea de código dada y las fechas.

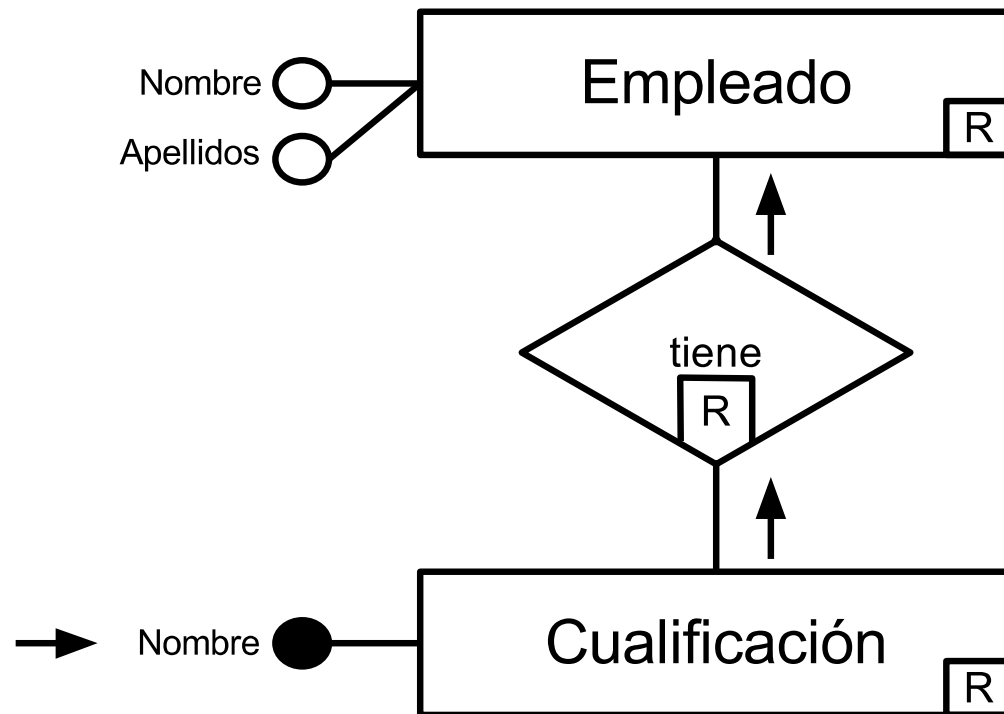
Esquemas de navegación: otro ejemplo más



Esquemas de navegación: otro ejemplo más

- Operación de datos:
 - Encontrar los nombres y apellidos de los empleados que tengan una cualificación dada.

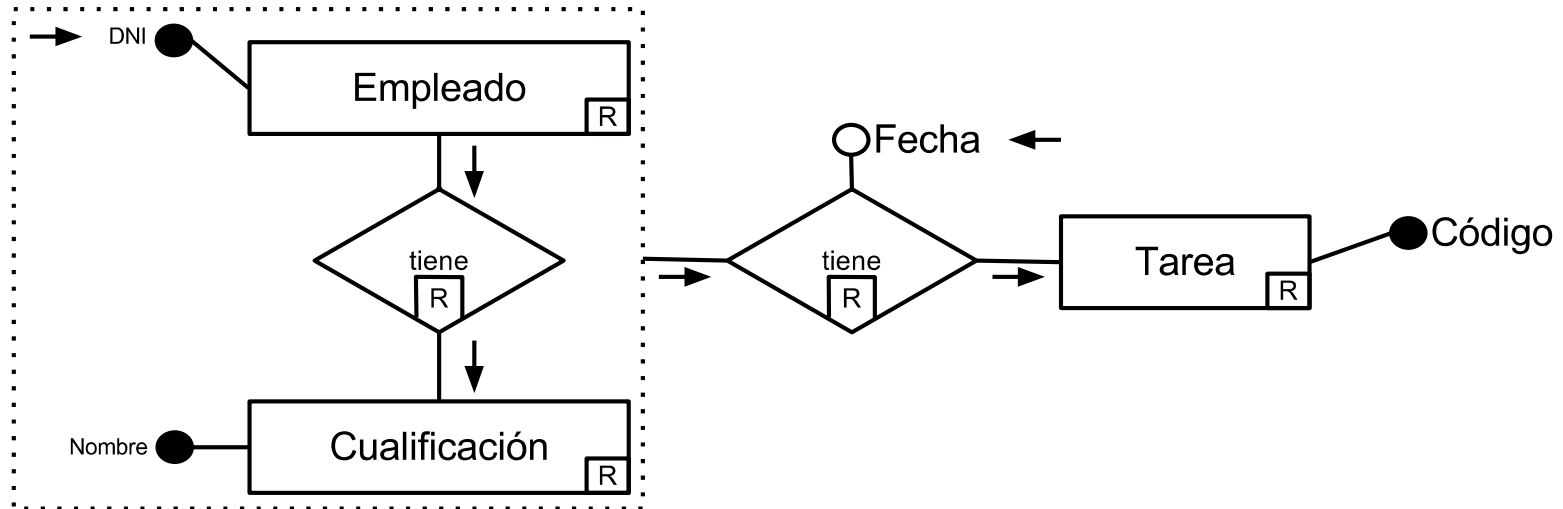
Esquemas de navegación: otro ejemplo más



Esquemas de navegación: otro ejemplo más

- Operación de datos:
 - Encontrar los nombres de cualificación y códigos de tarea que ha tenido un empleado con un DNI determinado antes del 1/1/2012

Esquemas de navegación: otro ejemplo más



Esquemas de navegación: algunos errores

- Un esquema de navegación puede tener todas las operaciones de consulta (R) que sean necesarias.
- Un esquema de navegación no puede tener más de una operación de inserción (I), modificación (U) ni borrado (D), salvo en el caso de *entidades débiles*, en las que la entidad y la relación de debilidad tienen que realizar inserciones (I) y borrados (D) a la vez.