



UNIVERSIDAD
DE GRANADA



Diseño y Desarrollo de Sistemas de Información

Grado en Ingeniería Informática

Descripción del sistema, análisis y especificación de requisitos

©I. J. Blanco, F. J. Cabrerizo, C. Cruz, M. J. Martín, M.J. Rodríguez, D. Sánchez

Este documento está protegido por la Ley de Propiedad Intelectual (Real Decreto Ley 1/1996 de 12 de abril).

Queda expresamente prohibido su uso o distribución sin autorización del autor.

Departamento de Ciencias de la
Computación e Inteligencia Artificial
<http://decsai.ugr.es>

- ❑ La parte práctica de la asignatura se realizará en subgrupos, y consistirá en el **diseño y desarrollo de un Sistema de Información sobre una temática a elegir por el subgrupo con la aprobación del profesorado**. El número de personas de cada subgrupo lo determinará el profesorado de prácticas. Todas las personas de cada subgrupo deben pertenecer al mismo grupo de prácticas.
- ❑ Cada subgrupo deberá definir un **espacio común de trabajo colaborativo**. Este espacio incluirá:
 - **Una carpeta compartida en Google Drive**. Será creada por una de las personas del subgrupo con su cuenta go.ugr.es y comunicada al resto del subgrupo y a el/la profesor/a de prácticas, que os indicará su cuenta go.ugr.es para ello. Contendrá los ficheros del subgrupo. Ayuda sobre cómo hacerlo en el siguiente video:

https://drive.google.com/file/d/1X_vcmrufuwMR1yuBxF9rJpruH2Zv2qAt/view?usp=sharing

- **Una sala de Google Meet** para las reuniones del subgrupo. Deberá crearse un evento periódico semanal en Google Calendar por la duración del semestre, compartiéndolo tan solo con el subgrupo y el/la profesor/a. En ese evento se incluirá el enlace a la sala. Ayuda sobre cómo hacerlo en el siguiente video:

<https://drive.google.com/file/d/1JYqzA6Xo0yWcxO64c7t73dHETqszGWEh/view?usp=sharing>

- ❑ Los vídeos anteriores estarán también disponibles en Prado.
- ❑ Para el trabajo grupal será necesario el uso de dispositivos para la realización de videoconferencia en las clases presenciales (portátil, tableta ó teléfono móvil), incluso si todo el grupo asiste presencialmente. También será imprescindible el uso de auriculares.
- ❑ **¡Aprovechad el tiempo de clase!** Para consulta de dudas con el profesorado, avanzar en la realización de las prácticas y seminarios ... el tiempo es ajustado y se puede acumular mucha tarea al final.

- ❑ Formación de los subgrupos de prácticas y creación del espacio común de trabajo. Estos subgrupos serán los mismos para todos los trabajos grupales de la asignatura (prácticas, seminarios, etc.).
- ❑ Elegir una temática para el sistema de información. Debe informarse al profesorado de prácticas para su aprobación.
- ❑ Escribir una descripción lo más detallada posible del sistema elegido. Condiciones:
 - Debe haber 1 subsistema por persona, numerados. Cada subsistema debe contener al menos 5 requisitos funcionales también numerados (ahora veremos en qué consisten).
 - Tanto los subsistemas como los requisitos deben ser aprobados por el profesorado de prácticas.
- ❑ Llevar a cabo un análisis de requisitos según la metodología que detallaremos en este documento.

- ❑ Se utilizará como metodología el “Análisis conjunto de datos y funciones guiado por las funciones” que veremos.
- ❑ A partir de la descripción del sistema realizaremos un proceso de análisis que nos dará un listado de requisitos funcionales, requisitos de datos y restricciones semánticas, con el formato que veremos. Se asumen como hechas, y por tanto no haremos, otras tareas propias de la Ingeniería del Software (casos de uso, requisitos no funcionales, bocetos de interfaces de usuario, etc.)
- ❑ El resultado del análisis debe ser coherente con la descripción del sistema. Cuando sea necesario se modificará o completará la descripción realizada previamente.
- ❑ Toda la documentación generada se incluirá en un único fichero tipo .docx, .ods o .tex, a partir del cual se obtendrá un fichero .pdf que será la entrega final. Debe indicarse en el documento la persona responsable de cada subsistema.

El objetivo de esta metodología es determinar el conjunto de datos y las restricciones sobre los mismos necesarios para el sistema de información.

En este apartado veremos:

- ☐ Qué es un requisito funcional
- ☐ Qué es un requisito de datos
- ☐ Qué es una restricción semántica
- ☐ Cómo reflejarlos en el documento de la práctica
- ☐ Algunos ejemplos



La Ingeniería del Software proporciona diversas formas de especificar todos los requisitos funcionales, que ya conocéis.

En el contexto de esta asignatura nos interesa centrarnos en analizar con más detalle las funcionalidades que interactúan con la Base de Datos, que son las que cumplen lo siguiente:

Requisito funcional: Una funcionalidad concreta del sistema que requiere acceso a la base de datos, para lo cual necesita:

- ☐ Una acción de algún agente externo al SI que activa la funcionalidad y que puede opcionalmente aportar **datos de entrada (E)**.
- ☐ Unos **datos de la base de datos** que hay que consultar (**lectura, R**), insertar, modificar y/o borrar (**escritura, W**).
- ☐ Unas acciones de salida que pueden incluir opcionalmente **datos de salida (S)** que se transmiten a algún agente externo.

Los datos de entrada, de la base de datos y de salida se especifican mediante **requisitos de datos**, conteniendo nombre y tipo de los datos.

Ejemplo de descripción del sistema:

- Deseamos crear un sistema de información para un único usuario que registre los contactos de una agenda. De cada contacto, almacenaremos su nombre (en una serie de hasta 20 caracteres), su apellido (en una serie de hasta 40 caracteres) y un número de teléfono (en una serie de hasta 20 caracteres que pueden ser todos numéricos menos el primero que puede ser numérico o un signo +). Para **dar de alta** un nuevo contacto, el usuario deberá proporcionar el nombre del mismo, su apellido y su número de teléfono, datos que el sistema almacenará, confirmando la inserción o dando un error. Para **dar de baja** un contacto, el usuario deberá proporcionar el número de teléfono, confirmando el borrado o dando un error. El sistema también permitirá **mostrar un listado** de contactos con todos sus datos a petición del usuario.

RF1: Dar de alta contacto.

Entrada: Agente externo: usuario. Acción: solicitar inserción.
Requisito de datos de entrada **RDE1**.

BD: Requisito de datos de escritura **RDW1**.

Salida: Agente externo: usuario. Acción: confirmación resultado.
Requisito de datos de salida: ninguno.

RDE1: Datos de entrada de alta de contacto
Nombre: Cadena de caracteres (20)
Apellidos: Cadena de caracteres (40)
Teléfono: Cadena de caracteres (20)

RDW1: Datos almacenados de contacto.
Los mismos datos que RDE1.



RF2: Dar de baja contacto.

Entrada: Agente externo: usuario. Acción: solicitar borrado.
Requisito de datos de entrada **RDE2**.

BD: Requisito de datos de escritura **RDW2**.

Salida: Agente externo: usuario. Acción: confirmación resultado.
Requisito de datos de salida: ninguno.

RDE2: Datos de entrada de baja de contacto
Teléfono: cadena de caracteres (20).

RDW2: Datos almacenados de contacto.
Los mismos datos que RDW1.

RF3: Mostrar listado de contactos.

Entrada: Agente externo: usuario. Acción: solicitar listado.
Requisito de datos de entrada: ninguno.

BD: Requisito de datos de lectura **RDR3**.

Salida: Agente externo: usuario. Acción: confirmación resultado.
Requisito de datos de salida: **RDS3**.

RDR3: Datos de contacto almacenado.
Los mismos datos que RDW1.

RDS3: Listado de registros, cada uno de ellos con los mismos datos de RDR3.

La especificación de requisitos funcionales debe expresar, con el detalle que sea necesario, pero en lenguaje natural:

- qué efecto tiene la función requerida,
- qué datos requiere para producirlo,
- qué efecto produce en el interior del sistema y
- qué salida comunica el resultado de dicha función.

Los ejemplos que acabamos de ver son muy simples y se han descrito de forma muy frugal, porque nos interesan sobre todo los datos y agentes externos involucrados.

RF1.1: Primera funcionalidad del subsistema 1.

Entrada: Agente externo: ... Acción:

Requisito de datos de entrada: **RDE1.1** (opcional).

BD: Requisito de datos de lectura **RDR1.1** y/o escritura **RDW1.1**
(al menos uno de los dos, o ambos, pudiendo un mismo dato aparecer en los dos).

Salida: Agente externo: ... Acción: ...

Requisito de datos de salida: **RDS1.1** (opcional).

+ Descripción de los distintos requisitos de datos.

- El resultado será un listado de requisitos funcionales para cada subsistema, con sus requisitos de datos asociados.
Cada persona del grupo hará la especificación de los requisitos de su subsistema.
- Es necesario leer repetidamente la descripción del sistema para garantizar que todos los requisitos funcionales y de datos han quedado reflejados.
- El orden en que se numeren es irrelevante.
- Si la descripción del sistema va acompañada de listados, formularios o *pantallazos*, es necesario analizar y extraer los requisitos correspondientes a cada uno de ellos.

- Una restricción semántica está asociada a un requisito funcional y uno o varios requisitos de datos.
- Describe cambios en el comportamiento del **requisito funcional cuando se da una configuración específica en los requisitos de datos**. Altera la realización de una función concreta del sistema por alguna configuración de datos.

Ejemplos:

- *Es necesario tener un mínimo de 18 años para registrarse en youtube.* **RF:** registrarse. **RD:** RDE. “Si la edad es menor de 18, no se insertan los datos de nuevo usuario y se devuelve un error.”
- *Un usuario no puede tener prestado más de dos libros en la biblioteca.* **RF:** solicitar préstamo libro. **RD:** RDE, RDR. “Si ya hay dos libros prestados al usuario en la BD, no se realiza la inserción y se devuelve aviso.”

En relación a nuestro ejemplo de agenda, entre otros:

- El formato del número de teléfono debe ser el correcto. Si el formato del teléfono no es correcto, no se realiza la inserción y se pide que se corrija.
- Nombre, apellidos y teléfono no pueden ser vacíos. Si nombre, apellido o teléfono son vacíos, no se realiza la inserción y se pide que se corrija.
- Un teléfono corresponde a un único contacto. Si ya había un contacto con el mismo teléfono, no se inserta el nuevo contacto y se devuelve un error.

RS1: El formato del número de teléfono debe ser el correcto.

RF: RF1

RD(s): RDE1

Descripción: “Si el formato del teléfono no es correcto, no se realiza la inserción y se pide que se corrija.”

RS2: Nombre, apellidos y teléfono no pueden ser vacíos.

RF: RF1

RD(s): RDE1

Descripción: “Si nombre, apellido o teléfono son vacíos, no se realiza la inserción y se pide que se corrija.”

RS3: Un teléfono corresponde a un único contacto.

RF: RF1

RD(s): RDW1 (si lo comprueba el SGBD, si no RDE1 + RDR1 que habría que añadir)

Descripción: “Si ya había un contacto con el mismo teléfono, no se inserta el nuevo contacto y se devuelve un error.”

RS1.1: Primera restricción semántica del subsistema 1.

RF: ...

RD(s): ...

Descripción: debe incluir condiciones de los datos y los cambios correspondientes en la acción del sistema.

RS1.2: Segunda restricción semántica del subsistema 1.

RF: ...

RD(s): ...

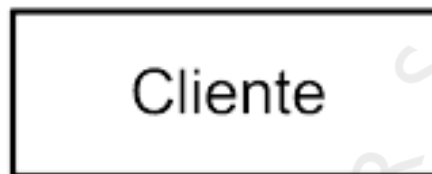
Descripción: debe incluir condiciones de los datos y los cambios correspondientes en la acción del sistema.

...

- Es necesario leer repetidamente la descripción del sistema para garantizar que todas las restricciones semánticas han quedado reflejados en la lista de restricciones semánticas. De nuevo, el orden en que se listen las restricciones no importa.
- El listado de requisitos de lectura y escritura en BD, más el listado de restricciones semánticas que afecten a dichos requisitos de BD, serán la base para el diseño conceptual de la base de datos de nuestro sistema de información (diagrama E/R).
- La metodología de análisis conjunto garantiza que la base de datos contiene todos los datos necesarios para cumplir con las funcionalidades requeridas de nuestro SI, ni más ni menos datos.

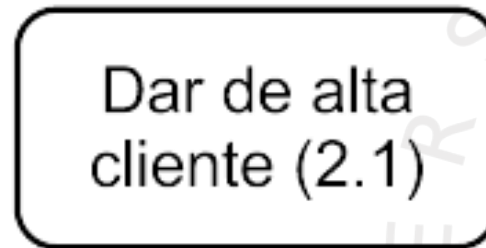
- Para completar el análisis vamos a realizar **Diagramas de Flujo de Datos** (DFDs), que son una representación gráfica de la evolución de la información dentro de nuestro SI.
- Suelen obtenerse como resultado del proceso de análisis, y permiten interactuar con el cliente y dentro del equipo de desarrollo, así como realizar labores de diseño del sistema centradas en los flujos de información entre componentes funcionales del sistema, base de datos y agentes externos.
- En estos diagramas no se representan flujos de control, sino flujos de datos entre elementos de almacenamiento o procesamiento de datos.
- En UML pueden representarse mediante **Diagramas de Actividades**, aunque éstos son mucho más ricos semánticamente, ya que permiten integrar flujos de datos y de control en el mismo diagrama.

- Un **agente externo** es una **persona, SI u otro sistema activo** que se comunica con nuestro SI enviando información y/o recibiendo información.
- Los agentes externos no forman parte de nuestro SI, pero se comunican con él.
- Se representan mediante un rectángulo etiquetado con un nombre identificativo del agente externo.



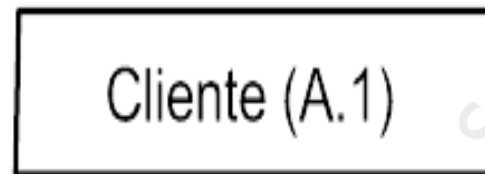
- **IMPORTANTE:** Un **DFD NO es un diagrama E/R**, aunque use elementos gráficos similares. Un DFD representa flujos de datos, no es una descripción de datos y sus restricciones. **Un agente externo NO es una entidad** de un conjunto de datos.

- Un **proceso** es una **actividad** dentro de un sistema, cuyo objetivo es el de **generar, usar, manipular o destruir información**.
- Se representan mediante rectángulos de esquinas redondeadas, o mediante círculos.
- Se etiquetan con un nombre representativo de la actividad y un código para identificarlos y describirlos en la documentación.



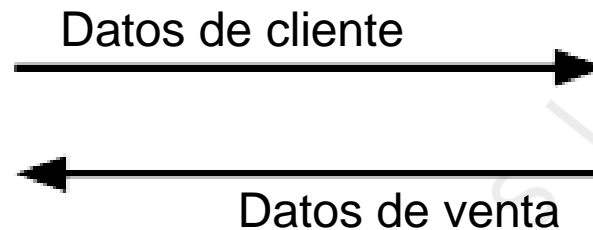
- En nuestra práctica, **aquellos procesos que coincidan con un requisito funcional** pueden etiquetarse utilizando el código **RF** correspondiente.

- Un **almacén de datos** representa un depósito de información dentro del SI, en nuestro caso una **base de datos**.
- Se representan mediante un rectángulo abierto por la derecha, y se etiquetan con un nombre representativo de los datos que contiene y un código para la descripción de los mismos en la documentación.



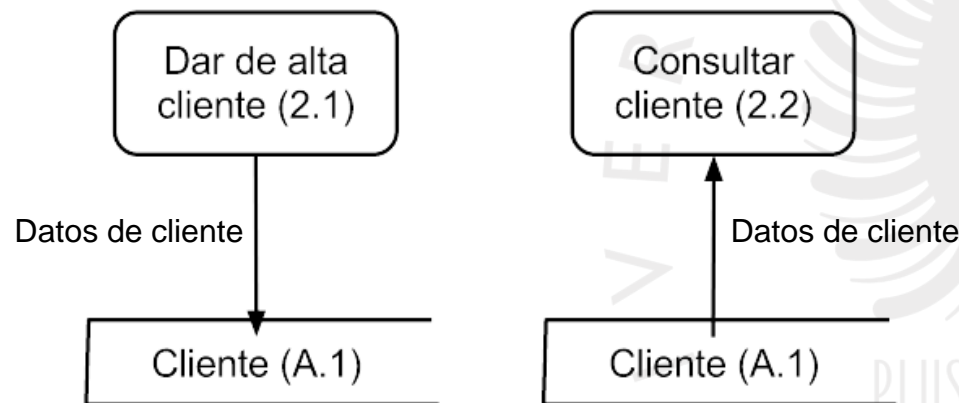
- El SI puede contener varios almacenes de datos.

- Los **flujos de datos** representan un **intercambio de información** entre procesos y otros elementos del DFD.
- Se representan mediante flechas, y suelen ir etiquetados para identificar los datos que fluyen.

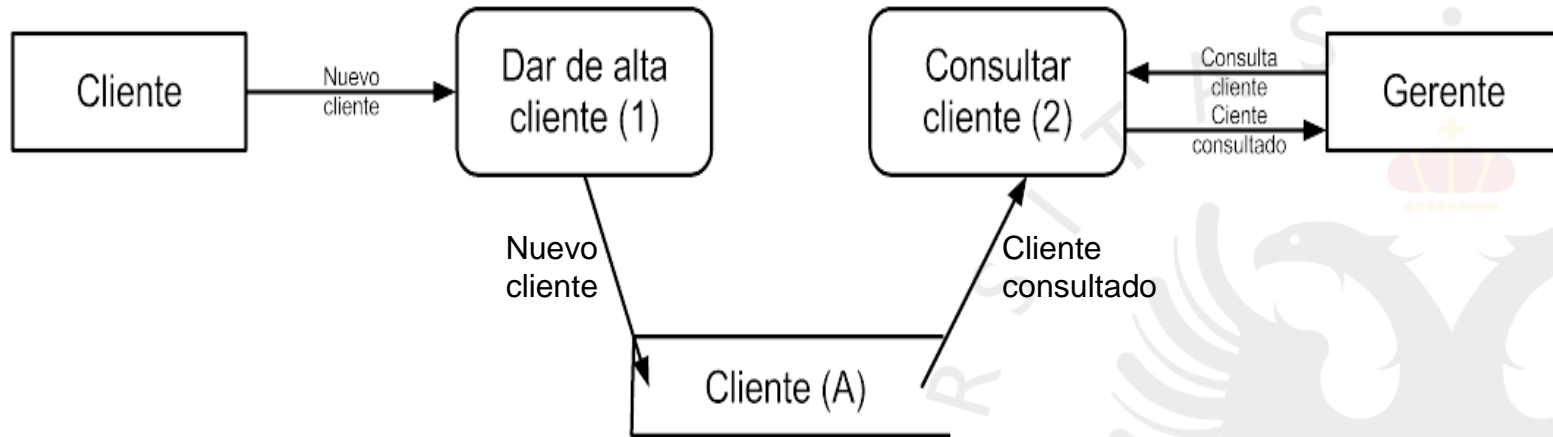


- En nuestra práctica, podemos usar los códigos que hemos empleado en el análisis (**RDE**, **RDW**, **RDR**, **RDS**) para etiquetar los flujos de los datos en dichos requisitos.

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



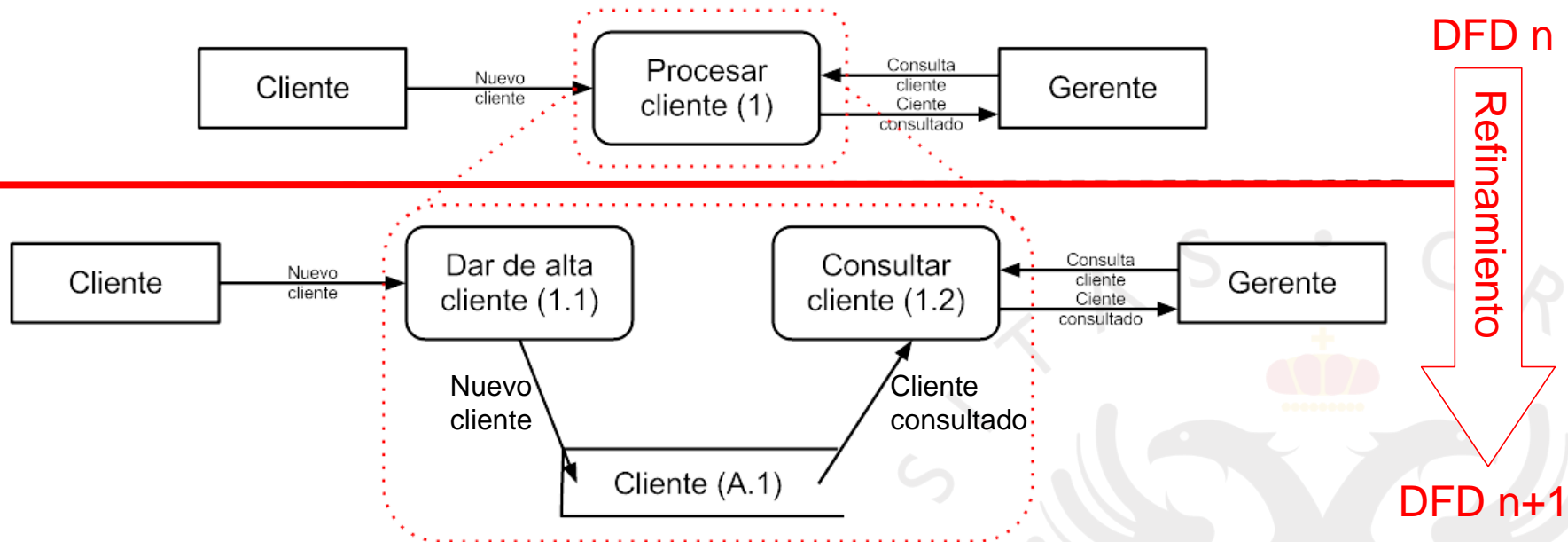
- Un ejemplo:



- Nótese que **no hay especificación de flujo de control**.

- Podemos utilizar distintos **DFDs** para describir nuestro SI a distintos niveles de detalle. Para ello se emplea una **jerarquía de DFDs**.
- En el nivel de mayor abstracción el SI se describe mediante lo que se denomina **Esquema de Caja Negra**, donde el SI se representa como un solo proceso, y se indican todos los agentes externos y flujos de datos.
- En el siguiente nivel, llamado **Esquema Armazón o DFD0**, se representa cada subsistema mediante un único proceso, y se incluye el almacén correspondiente a la BD (o varios si es necesario), indicando los correspondientes flujos de datos. Este esquema supone un **refinamiento (descripción más detallada)** del proceso que representa al SI completo en el esquema de caja negra.
- En el siguiente nivel, **DFD1**, se refina cada subsistema dividiéndolo en procesos. En nuestra práctica, habitualmente éstos coincidirán con los requisitos funcionales.

- Esta jerarquía puede continuar (DFD2, DFD3, ...) dependiendo de la complejidad del sistema, refinándose sucesivamente los procesos del nivel anterior. En las prácticas **nosotros no vamos a pasar del DFD1**.
- A partir de un cierto nivel de detalle es posible tener flujos de datos entre procesos, que pueden ya representar una descomposición modular de las funcionalidades del sistema. Nosotros no los vamos a usar en general.
- Existen mecanismos formales para verificar la coherencia de los refinamientos entre DFDs de distintos niveles. **Deben cumplirse ciertas propiedades:**
 - Los flujos de datos entre i) los procesos resultantes de refinar un proceso X del nivel anterior, y ii) agentes externos y almacenes del sistema, deben contener los mismos datos que los flujos entre X y esos agentes y almacenes.
 - El refinamiento de un proceso debe tener la misma semántica y funcionalidad que el proceso original. Proceso y refinamiento sólo pueden diferir en el nivel de detalle con el que están descritos.



- Se muestra el refinamiento del proceso “Procesar Cliente”.
- Pueden comprobarse las propiedades de coherencia en flujos y semántica.

- Un único documento .pdf que contendrá:
 - Nombre del sistema
 - Nombre de los subsistemas y personas responsables de cada subsistema.
 - Descripción del sistema en lenguaje natural, 2-3 páginas aprox
 - Listado de requisitos funcionales, requisitos de datos y restricciones semánticas.
 - DFD Esquema de Caja Negra.
 - DFD Armazón (DFD0).
 - Un DFD por cada subsistema mostrando el refinamiento del proceso correspondiente del esquema armazón, y que incluirá agentes externos, almacenes y los correspondientes flujos de datos.
 - Deberán comprobarse las propiedades de consistencia de los refinamientos.