



v1.5.1

# PRÁCTICAS Y PROBLEMAS

Ingeniería del Software I

Versión 1.5.1 (marzo 2010)

**F.Javier Zarazaga**



Departamento de Informática e Ingeniería de Sistemas  
Universidad de Zaragoza  
María de Luna 1, 50018, Zaragoza, España.





## DATOS DE CONTROL DEL DOCUMENTO

Título documento: PRÁCTICAS Y PROBLEMAS

Título proyecto/producto: Ingeniería del Software I

Código del documento: Versión: Revisión:

Nombre del fichero: practicas\_y\_ejercicios\_IS1.odt

Tamaño fichero: 986112 páginas: 21 palabras: 5500 caracteres: 32878

Fecha de Inicio: 23/2/02 17:53

## REGISTRO DE CAMBIOS EN EL DOCUMENTO

Rev.	Fecha	Desarrollador	Secciones Afectadas
1.0	23/02/02	F.J.Zarazaga	Todas (creación)
1.1	12/06/02	F.J.Zarazaga	Todas (creación)
1.2	12/12/02	F.J.Zarazaga	Todas (revisión y actualización)
1.2.1	16/02/04	J. Nogueras	Todas (revisión y actualización)
1.2.2	15/02/05	J. Nogueras	Todas (revisión y actualización)
1.2.3	28/02/06	F.J.Zarazaga	Todas (revisión y actualización)
1.3	23/02/08	F.J.Zarazaga	Todas (revisión, actualización y paso a Open Office)
1.4	24/02/09	F.J.Zarazaga	Todas (revisión y actualización)
1.5	10/02/10	J. Nogueras	Todas (revisión y actualización)
1.5.1	08/03/10	R. Béjar	Todas (actualización de instrucciones para el uso de Visio)

# INDICE

<b>CAPÍTULO 1. INTRODUCCIÓN.....</b>	<b>1</b>
<b>CAPÍTULO 2. PRÁCTICA 1.....</b>	<b>2</b>
2.1. EL REGALO S.A.....	2
2.1.1. Descripción de la aplicación proporcionada por el cliente.....	2
2.1.2. Tareas a realizar.....	3
2.1.3. Captura de requisitos.....	4
2.1.3.1. Lista de requisitos versión 0.....	4
2.1.3.1.1. Requisitos funcionales.....	4
2.1.3.1.2. Requisitos no funcionales.....	4
2.1.3.2. Lista de requisitos versión 1.....	5
2.1.3.2.1. Requisitos funcionales.....	5
2.1.3.2.2. Requisitos no funcionales.....	5
2.1.3.2.3. Falta de información.....	6
2.1.3.3. Lista de requisitos versión 2.....	8
2.1.3.3.1. Requisitos funcionales.....	8
2.1.3.3.2. Requisitos no funcionales.....	8
2.2. GESTIÓN DE UN APARCAMIENTO PÚBLICO.....	9
2.2.1. Descripción de la aplicación proporcionada por el cliente.....	9
2.2.2. Tareas a realizar.....	9
2.3. GESTIÓN DE UNA RED DE RADIOTELEFONÍA MÓVIL.....	10
2.3.1. Descripción de la aplicación proporcionada por el cliente.....	10
2.3.2. Tareas a realizar.....	11
<b>CAPÍTULO 3. PRÁCTICA 2.....</b>	<b>12</b>
3.1. EL REGALO S.A.....	12
<b>CAPÍTULO 4. PRÁCTICA 3.....</b>	<b>13</b>
4.1. DESCRIPCIÓN DE LA AMPLIACIÓN PROPUESTA.....	13
4.2. TAREAS A REALIZAR.....	13
4.3. PROPUESTA DE MODIFICACIÓN EN EL LISTADO DE REQUISITOS.....	14
4.3.1. Requisitos funcionales.....	14
4.3.2. Requisitos no funcionales.....	14
<b>CAPÍTULO 5. PRÁCTICA 4.....</b>	<b>15</b>
<b>CAPÍTULO 6. PRÁCTICA 5.....</b>	<b>16</b>



## Capítulo 1. Introducción

El presente documento tiene como objetivo actuar como guía para el desarrollo de las prácticas de la asignatura Ingeniería del Software I, del sexto cuatrimestre de la titulación de Ingeniería Informática de la Universidad de Zaragoza.

En los sucesivos apartados se detallan cada una de las cinco sesiones que integran el plan de prácticas de la asignatura. Para algunas de las sesiones se detallan ejercicios adicionales.

El material aquí presentado se complementa con ficheros de trabajo adicionales que estarán disponibles en las salas de prácticas del Departamento de Informática e Ingeniería de Sistemas denominadas L.0.03 y L.0.04. El acceso a la cuenta de trabajo se realiza utilizando “b-isi” como nombre de usuario y “lit52cor” como password.



## Capítulo 2. Práctica 1

Durante esta práctica se trabajará en la identificación y clasificación de requisitos utilizando diferentes técnicas.

El trabajo de laboratorio se centrará en el primero de los enunciados ("El Regalo s.a."), quedando la repetición del mismo tipo de trabajo sobre el resto de enunciados como un trabajo voluntario que será revisado en algunas de las sesiones de problemas programadas en el aula.

### 2.1. El Regalo S.A.

Seguidamente se da una posible descripción de un problema proporcionada por el cliente. En el punto 2.1.3 se da una primera propuesta de requisitos del sistema, acompañada con una serie de posibles preguntas a realizar y las correspondientes respuestas. A partir de estas respuestas se reconstruye la lista de requisitos del sistema.

Para poder sacar el máximo provecho a la práctica, se recomienda no acceder a las correspondientes secciones de este documento hasta que no se han resuelto (o al menos intentado resolver) por uno mismo. Para ello se recomienda utilizar el documento que se halla en el directorio de la práctica y en el que se incluye el enunciado y las tablas de requisitos vacías.

*ATENCIÓN: La identificación de requisitos no es una ciencia exacta para la que dado un problema existe una única solución. La definición y acotación de los requisitos está claramente influida por elementos subjetivos. La propuesta de solución que se hace es eso mismo, una propuesta subjetiva. Ante cualquier duda o no comprensión de algo (enunciado, proceso de trabajo, solución propuesta), acudir al profesor de la asignatura.*

#### 2.1.1. Descripción de la aplicación proporcionada por el cliente

La empresa "El Regalo S.A." es una cadena de tiendas de pequeños artículos de regalo. Actualmente, los empleados que trabajan en estas tiendas cada vez que efectúan una venta deben apuntar el código del producto que venden, así como una descripción del mismo para verificar que no se ha producido ningún error de transcripción. El cálculo del importe de una venta se hace utilizando una sencilla calculadora.

Semanalmente, se procede a unificar los listados de productos vendidos sólo en uno, en el que se van acumulando las ventas que se realizan en un mes. En este proceso de unificación se comprueban y solucionan los posibles errores de transcripción.

Una vez al mes, se procede a realizar un inventario en las tiendas con el objeto de verificar que las existencias reales corresponden con lo que debería haber en la tienda. Para ello, recopilan un listado de todos los productos que hay en las estanterías. Con este listado, el de productos vendidos y las facturas que les remiten los proveedores pueden determinar si se ha producido algún robo. Si detectan que en una tienda se producen muchos robos, incrementan el personal de la misma.

En los últimos meses han visto que el número de robos ha crecido en prácticamente todas las tiendas. La contratación de nuevo personal para realizar la vigilancia se plantea como una solución no viable desde el punto de vista económico, ya que haría que los gastos de personal no pudiesen ser cubiertos con los ingresos. Uno de los nuevos consejeros, Saturnino Regalo (sobrino del presidente de la empresa, don Angel Regalo), ha planteado que el problema reside en el gran número de tareas que deben realizar los empleados. Por ello propone que se proceda a automatizar en la medida de lo posible estas tareas mediante la adopción de soluciones informáticas. El resto del consejo no lo ve claro ya durante los 30 años de vida de la empresa se han hecho las cosas más o menos de la misma manera, y el negocio ha prosperado notablemente (la última revolución que se introdujo fue el uso de las calculadoras para calcular los importes en vez de realizar las sumas a mano). Finalmente, se llega a una solución de compromiso mediante la cual se decide que se va a abordar una primera fase de automatización en la cual se incluirá una sencilla gestión de *stocks* y de ventas, eso sí, el coste de instalación en cada una de las tiendas debe ser mínimo por lo que la aplicación a desarrollar deberá



poderse ejecutar en ordenadores de pequeña potencia, y sin necesidad de adquirir ningún tipo de licencia de ejecución (run-time), o cualquier otro software específico (motores de bases de datos, etc.). Se espera que sea necesario adquirir únicamente el ordenador y el sistema operativo.

A grandes rasgos, se espera que esta primera versión de la aplicación permita introducir en la base de datos las mercancías que se reciban en la tienda, tanto si son nuevos productos, como si ya se vendían anteriormente. Igualmente, se podrán realizar en la base de datos correcciones tanto de descripción del producto (cambio de modelo, distinto color, etc.), como del precio o el número de unidades. Éstas últimas suelen producirse generalmente por recepción de más unidades de un producto o por devolución de productos, pero también si se detectan desfases a la hora de hacer inventarios. De todos modos, el motivo no es relevante. Hay que tener cuidado ya que no se pueden dar de alta dos productos con el mismo código. También debe comprobarse cuidadosamente que el producto cuyas existencias se corrigen es el deseado.

A la hora de proceder a gestionar una venta, hay que proporcionar un sistema flexible que permita comprobar los productos que se está llevando el cliente y efectuar las oportunas correcciones. Cada vez que un producto se venda, se debe quitar de la base de datos. No se pueden vender más unidades de un producto de las que se tienen contabilizadas en la base de datos. Además, se puede tener la oportunidad de informar al cliente cuanto lleva gastado.

El sistema debe ser seguro y robusto a caídas. Si a mitad de una venta se cae el sistema, la recuperación debe permitir restaurar la situación justo anterior a dicha caída. Por motivos de seguridad se espera poder tener acceso a un fichero en el que se puedan consultar todas las ventas que se han hecho.

## 2.1.2. Tareas a realizar

- Identificar los requisitos (funcionales y no funcionales).
- Identificar falta y/o deficiencias en la información. Proponer preguntas destinadas a satisfacer estas deficiencias de datos.
- Derivar nuevos requisitos de las respuestas.
- Identificar los casos de uso. Utilizando una herramienta de dibujo (Visio 2000). Acceder al programa y construir los casos de uso . Visio 2000 se encuentra en Y:\Software\Ofimatica\Visio2000\Visio32.exe y un stencil para casos de uso que se puede usar en

Y:\Software\Ofimatica\Visio2002\3082\Soluciones\Software\Caso de uso de UML.vss.

Para cargar el stencil de casos de uso, abrir Visio 2000, cancelar el diálogo inicial, ir a File -> Stencils -> Open Stencil, abrir el que se indica en el párrafo anterior, y crear un nuevo dibujo para empezar a trabajar.



## 2.1.3. Captura de requisitos

### 2.1.3.1. Lista de requisitos versión 0

#### 2.1.3.1.1. Requisitos funcionales

Código del requisito	Descripción

#### 2.1.3.1.2. Requisitos no funcionales

Código del requisito	Descripción



## 2.1.3.2. Lista de requisitos versión 1

## 2.1.3.2.1. Requisitos funcionales

Código del requisito	Descripción
RF-1	Cada vez que se venda una unidad de un producto, hay que proceder a descontarla en la base de datos.
RF-2	No se pueden vender más unidades de un producto de las que se tienen contabilizadas en la base de datos.
RF-3	Se debe poder comprobar los productos que se está llevando un cliente.
RF-4	Se deben poder corregir los datos de una venta.
RF-5	Se debe poder informar al cliente en todo momento de cuanto lleva gastado.
RF-6	Se debe poder dar de alta nuevos productos en la base de datos.
RF-7	Se debe permitir que se realicen correcciones en el precio, la descripción o el número de unidades de un producto.
RF-8	Se deben apuntar las ventas que se hacen en un fichero.

## 2.1.3.2.2. Requisitos no funcionales

Código del requisito	Descripción
RNF-1	El sistema debe poder correr en un ordenador de bajas prestaciones.
RNF-2	No deben ser necesarias licencias de ejecución.
RNF-3	No se debe necesitar ningún motor de base de datos o cualquier otro software específico. El único software que resultará imprescindible adquirir será el sistema operativo.
RNF-4	La recuperación de una caída del sistema a mitad de una venta debe poder permitir la restauración de la situación anterior al inicio de dicha venta.
RNF-5	Costes de instalación en tienda mínimos
RNF-6	Un producto = código + descripción + precio + unidades
RNF-7	Los códigos de producto son únicos en la base de datos.





### 2.1.3.2.3. Falta de información

Es necesario requerir información sobre los siguientes aspectos:

- Volumen de datos a manejar: número de productos, número de ventas.
  - Cota superior del número de productos distintos.
  - Cota superior del número de ventas diarias.
- Tamaño de los datos: que definen a los productos y a las ventas.
  - Cota superior del tamaño de la descripción de un producto.
  - Descripción del código de un producto.
  - Descripción de las unidades y el precio.
  - Cota superior del precio de una unidad de un producto.
  - Cota superior del número de unidades de un mismo producto en almacén.
  - Cota superior del importe de una venta.
- Concurrencia de usuarios.
- Cada cuanto se deben hacer las copias de los datos a disco.
- Tipo de la interfaz con el usuario.
- Identificación de las ventas.
- Sistema operativo.
- Control de acceso a la aplicación.
- Facilidad de ampliación de la aplicación.
- Descripción del campo “descripción”.
- Conocimientos de los operarios.
- ¿Hay que considerar al cliente como una parte del sistema?
- ¿y al proveedor?
- Frecuencia de las compras.
- Devolución de productos resulta muy poco clara, ¿cómo se resuelven los desfases de inventario?
- ¿Cómo se actúa en la base de los productos ante la renovación de modelos?
- Clarificar la comprobación de artículo en la venta.
- ¿Qué se entiende por informar al cliente de lo que “lleva gastado”?
- ¿Se pueden borrar productos de la BD?



Si tienes más preguntas destinadas a clarificar aspectos del enunciado, acude a tu profesor para que te las responda actuando como cliente.

Posibles respuestas:

- El número de productos distintos no es mayor de 500.
- Suelen hacerse unas 25 ventas diarias.
- El código de un producto es un número entero.
- La descripción de un producto no ocupa más de 25 caracteres.
- Las unidades y el precio son valores enteros.
- El precio de cualquier producto no supera las 25.000 ptas.
- Nunca se tiene más de 25 unidades de un mismo producto.
- Nunca se ha hecho una venta de un importe superior a las 500.000 ptas.
- Cada tienda irá equipada con un único PC. No se conectarán entre ellos.
- Deberíamos tener seguridad de los datos después de cada operación.
- Interfaz gráfica de usuarios tipo Windows.
- S.O. Windows.
- Las ventas se identifican por el día y la fecha de su inicio.
- No control de acceso.
- No se necesitan facilidades para la ampliación del sistema.
- Campo descripción es una cadena de texto.
- Los operarios tienen unos conocimientos muy básicos de Windows.
- Ni el cliente ni el proveedor son parte del sistema.
- La frecuencia de compras es muy poco relevante, una vez al mes, a lo sumo dos.
- Cuando se devuelven unidades de un productos, se meten directamente en la base de datos sin preocuparse de otra cosa.
- Cuando un modelo se renueva, simplemente cambiamos la descripción en la BD sin preocuparnos de si nos quedan unidades o no del mismo.
- Cuando se hace una venta, el cajero debe poder ver la descripción asociada al código de un producto.
- En cualquier momento de la venta, el cliente nos puede preguntar sobre la cantidad que lleva gastada.
- Se deben poder dar de baja productos de la base de datos.



## 2.1.3.3. Lista de requisitos versión 2

## 2.1.3.3.1. Requisitos funcionales

Código del requisito	Descripción
<i>RF-1</i>	<i>Cada vez que se venda una unidad de un producto, hay que proceder a descontarla en la base de datos.</i>
<i>RF-2</i>	<i>No se pueden vender más unidades de un producto de las que se tienen contabilizadas en la base de datos.</i>
<i>RF-3</i>	<i>Se debe poder comprobar los productos que se está llevando un cliente.</i>
<i>RF-4</i>	<i>Se deben poder corregir los datos de una venta.</i>
<i>RF-5</i>	<i>Se debe poder informar al cliente en todo momento de cuanto lleva gastado.</i>
<i>RF-6</i>	<i>Se debe poder dar de alta nuevos productos en la base de datos.</i>
<i>RF-7</i>	<i>Se debe permitir que se realicen correcciones en el precio, la descripción o el número de unidades de un producto.</i>
<i>RF-8</i>	<i>Se deben apuntar las ventas que se hacen en un fichero.</i>

## 2.1.3.3.2. Requisitos no funcionales

Código del requisito	Descripción
<i>RNF-1</i>	<i>El sistema debe poder correr en un ordenador de bajas prestaciones.</i>
<i>RNF-2</i>	<i>No deben ser necesarias licencias de ejecución.</i>
<i>RNF-3</i>	<i>No se debe necesitar ningún motor de base de datos o cualquier otro software específico. El único software que resultará imprescindible adquirir será el sistema operativo.</i>
<i>RNF-4</i>	<i>La recuperación de una caída del sistema a mitad de una venta debe poder permitir la restauración de la situación anterior al inicio de dicha venta.</i>
<i>RNF-5</i>	<i>Costes de instalación en tienda mínimos</i>
<i>RNF-6</i>	<i>Un producto = código + descripción + precio + unidades</i>
<i>RNF-7</i>	<i>Los códigos de producto son únicos en la base de datos.</i>
RNF-8	El número de productos distintos no es mayor de 500.
RNF-9	Suelen hacerse unas 25 ventas diarias.
RNF-10	El código de un producto es un número entero.
RNF-11	La descripción de un producto no ocupa más de 25 caracteres.
RNF-12	Las unidades y el precio son valores enteros.
RNF-13	El precio de cualquier producto no supera las 25.000 ptas.
RNF-14	Nunca se tiene más de 25 unidades de un mismo producto.
RNF-15	Nunca se ha hecho una venta de un importe superior a las 500.000 ptas.
RNF-16	Cada tienda irá equipada con un único PC. No se conectarán entre ellos.

Ojo, la lista anterior no tiene porque estar completa.



## 2.2. Gestión de un aparcamiento público

### 2.2.1. Descripción de la aplicación proporcionada por el cliente

Un aparcamiento público admite tanto a clientes que cogen el ticket en la entrada, como a abonados que cuentan con una tarjeta que los identifica. Para ello, el aparcamiento cuenta con dos barreras, una a la entrada y otra a la salida, conectadas al ordenador de gestión. La barrera de entrada dispone de un lector de tarjeta de abonados y un expendedor de tickets, mientras que la barrera de salida cuenta también con un lector de tarjetas y con un lector de tickets.

Cuando un usuario llega a la barrera de entrada y retira el ticket de entrada, el emisor de tickets envía un mensaje al ordenador indicando el código del ticket y la fecha y hora de retirada del mismo, y abre automáticamente la barrera. El código del ticket es un número secuencial y cíclico que va generando el emisor entre 0 y 99.999. Si el usuario es abonado, entonces el lector de tarjeta de abonado envía un mensaje al ordenador de gestión con el número de abonado y se queda esperando a recibir confirmación de que puede abrir la barrera. El número de abonado está entre el 1 y el 999.

Un usuario no abonado debe pasar por caja antes de poder salir. El cajero dispone de un lector inteligente de tickets. Al introducir un ticket en el lector, éste solicita al ordenador al que está conectado que le remita la fecha y hora de entrada con el fin de poder calcular el importe. Una vez que el operario valida el pago de este importe, el lector envía un mensaje al ordenador para que apunte el ticket como disponible para salir. Cuando el usuario va a salir con el coche, introduce el ticket pagado en el lector que se encuentra en la barrera de salida. Este envía un mensaje al ordenador para que confirme que el ticket ha sido pagado y que autoriza la salida.

Cuando un usuario abonado desea sacar su coche del aparcamiento, introduce su tarjeta en el lector de la barrera de salida. Este lector envía un mensaje al ordenador confirmando que puede dejar pasar al abonado.

Pese a ser un único dispositivo físico, cada una de las barreras está integrada por tres dispositivos lógicos (la barrera en sí, el lector de tarjeta de abonado y el expendedor o el lector de tickets) que obedecen y se comunican por separado con el ordenador de gestión.

Un abonado debe haber pasado su tarjeta por el lector de salida antes de poder volver a utilizarla en el de entrada.

Un ticket que no ha sido pagado no debe permitir abrir la barrera de salida.

Se debe atender las peticiones de apertura de barrera, tanto de entrada como de salida, en menos de tres segundos.

Del abonado se debe guardar información referente a su nombre, su dirección y su número de teléfono.

### 2.2.2. Tareas a realizar

- Identificar los requisitos (funcionales y no funcionales).
- Identificar falta y/o deficiencias en la información. Proponer preguntas destinadas a satisfacer estas deficiencias de datos.
- Derivar nuevos requisitos de las respuestas.



### 2.3. Gestión de una red de radiotelefonía móvil

### 2.3.1. Descripción de la aplicación proporcionada por el cliente

La Figura 1 ilustra una organización típica de una red trunking que consta de un nodo central (NC) con responsabilidad del control de red, y una serie de estaciones base de zona (EBZ) conectadas al NC a través de enlaces de control. El sistema de gestión está conectado al NC a través del cual tiene acceso a las informaciones de la red y puede actuar sobre los elementos. El sistema de gestión es la interfaz entre la red trunking y el operador de la misma. Dicho operador es el encargado de realizar las labores de operación, mantenimiento y explotación.

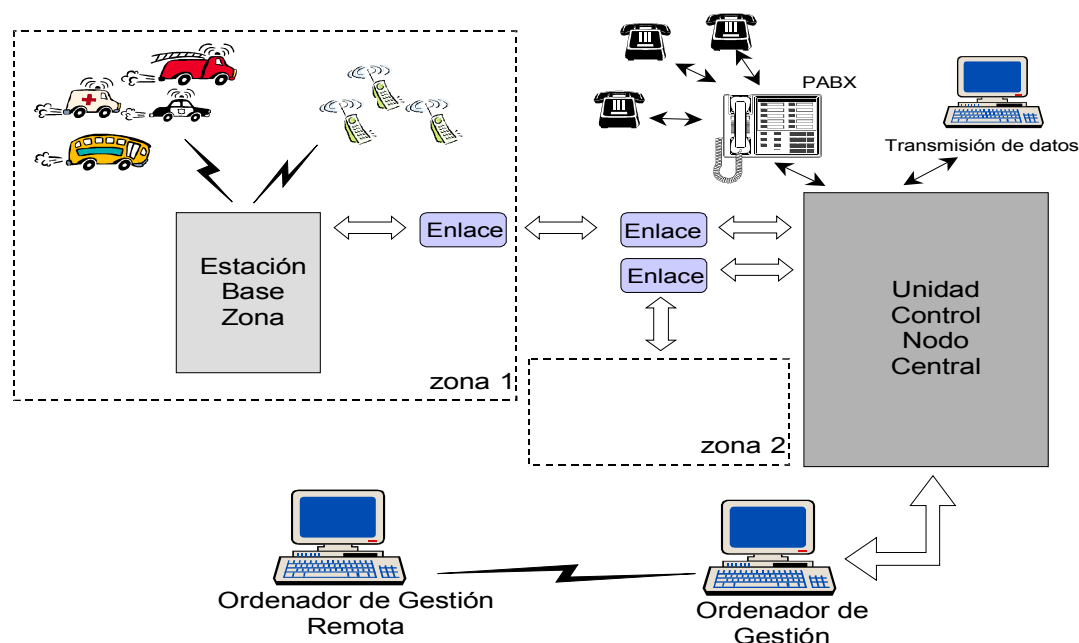


Figura 1. Esquema típico de una red trunking

El objetivo del sistema a construir es ampliar la capacidad del ordenador de gestión para ir guardando la información relativa a las llamadas que se realizan dentro de la red. Para ello, se creará un proceso que deberá funcionar continuamente recibiendo por el puerto serie la información sobre las llamadas y guardándola en la base de datos. La información que se recibe del Nodo Central viene codificada en varios bytes. Esta información hace referencia al equipo llamante, el equipo llamado, la fecha y hora de inicio de la llamada y su duración (ver Figura 2).

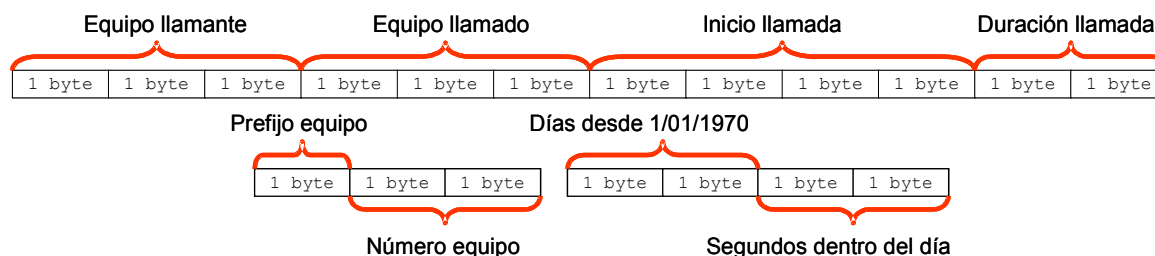


Figura 2. Codificación de la información

La información que sobre el equipo maneja el NC se ajusta a la norma 1327, mientras que en la base de datos habrá que guardarla siguiendo la norma 1343. En la primera de ellas al equipo se le



identifica por el prefijo (0..127) y el número de equipo con ese prefijo (0..6800). En la segunda a un equipo se le identifica por el número de flota (0..199) y número de equipo dentro de la flota (0..999). El sistema debe ser capaz de recibir y almacenar en base de datos llamadas con una frecuencia punta de 4 llamadas por segundo.

### 2.3.2. Tareas a realizar

- Identificar los requisitos (funcionales y no funcionales).
- Identificar falta y/o deficiencias en la información. Proponer preguntas destinadas a satisfacer estas deficiencias de datos.
- Derivar nuevos requisitos de las respuestas.



## Capítulo 3. Práctica 2

Durante esta práctica se trabajará en la construcción de los modelos que representa el análisis de la aplicación.

El trabajo de laboratorio se centrará en el primero de los enunciados ("El Regalo s.a."), quedando la realización del segundo trabajo como voluntario que será revisado en las horas de tutoría.

### 3.1. El Regalo S.A.

Partiendo de los requisitos identificados en la práctica 1, construir el DFD de nivel 0 y de nivel 1 de la aplicación. Para ello se utilizará la herramienta de dibujo (Visio).

La versión que se usará de Visio (Visio 2000) se encuentra en

Y:\Software\Ofimatica\Visio2000\Visio32.exe. Para dibujar DFD según la notación de Yourdon y Gane-Sarson hay que cargar el *template* y el *stencil* apropiados. Usaremos los que están en Y:\Software\Ofimatica\Visio2002\3082\Soluciones\Diagrama de flujo\Diagrama de flujo de datos.vst y Y:\Software\Ofimatica\Visio2002\3082\Soluciones\Diagrama de flujo\Formas para diagrama de flujo de datos.vss. Para ello, una vez lanzado Visio, iremos a File -> New -> Choose Drawing Type para cargar el primero (*template*) y luego a File -> Stencils -> Open Stencil para cargar el segundo (*stencil*). Una vez hemos cargado el *stencil*, veremos que hay disponible un conjunto de iconos con las formas que necesitamos para crear el DFD.

Para contrastar los resultados se puede acudir a los ficheros ".vsd" ubicados en el directorio de la práctica (NO MIRAR LOS FICHEROS .vsd HASTA QUE NO SE HAYA INTENTADO APORTAR UNA SOLUCIÓN):

- El fichero modelo0.vsd muestra una propuesta del diagrama de contexto
- El fichero modelo1.vsd muestra una propuesta del diagrama de nivel 1
- El fichero modelo2.vsd y modelo3.vsd muestra una propuesta de los diagramas de nivel 2

Crear el Diccionario de Datos donde se recojan las especificaciones de los elementos construidos. Este diccionario se puede construir utilizando un procesador de textos o una hoja de cálculo.



## Capítulo 4. Práctica 3

En esta práctica se propone llevar a cabo una ampliación del sistema “El Regalo s.a.” que se está construyendo.

### 4.1. Descripción de la ampliación propuesta

En la última cena familiar Saturnino Regalo logró convencer a su tío, don Angel Regalo, de la necesidad de que la aplicación que les están construyendo dispusiese al menos de algún recurso para poder visualizar las existencias existentes y su valoración. Para ello, comentaron que el sistema podría estar dotado con una ventana, del estilo de la esbozada en el siguiente dibujo (a ninguno de los dos se les da muy bien el dibujo), en la que poder presentar el listado de productos y unidades que hay en el almacén y su valoración. Esta nueva ventana iría vinculada a una nueva opción de menú.

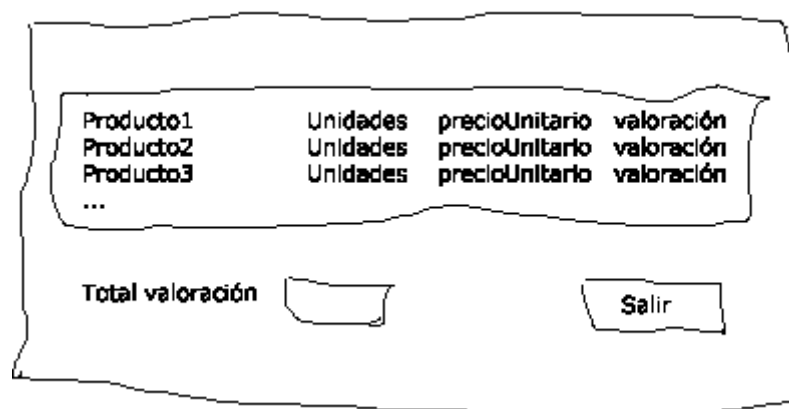


Figura 3. Esquema de la ventana de existencias

Como es lógico, esta modificación va a suponer algunas modificaciones en los requisitos y en el análisis realizado.

### 4.2. Tareas a realizar

- Modificar la lista de requisitos del sistema.
- Modificar los diagramas de flujo de datos.
- Actualizar los diagramas de navegación (fichero ventanas.odp dentro del directorio de las práctica).
- Realizar el prototipado simple de la nueva ventana (fichero ventanas.odp dentro del directorio de las práctica).
- Utilizando Visual C/C++ 6.0 realizar el prototipado avanzado de la nueva ventana y modificar el prototipo del menú. Dentro del directorio del proyecto, en el directorio Proyecto\_El\_Regalo\ElRegalo se encuentra el fichero del proyecto (ElRegalo.dsp) que permite compilar la primera versión de la aplicación incluyendo las ventanas prototipadas. En el directorio Proyecto\_El\_Regalo\Fuentes se encuentra el código fuente de esta primera versión. No te olvides copiar a un lugar seguro (tu cuenta, un disquete) los ficheros con las modificaciones de las ventanas (resource.h y resource.rc).

Copiar todo el directorio de la práctica al escritorio para poder realizar las modificaciones oportunas sin problemas.





## 4.3. Propuesta de modificación en el listado de requisitos

### 4.3.1. Requisitos funcionales

Código del requisito	Descripción
RF-1	Cada vez que se venda una unidad de un producto, hay que proceder a descontarla en la base de datos.
RF-2	No se pueden vender más unidades de un producto de las que se tienen contabilizadas en la base de datos.
RF-3	Se debe poder comprobar los productos que se está llevando un cliente.
RF-4	Se deben poder corregir los datos de una venta.
RF-5	Se debe poder informar al cliente en todo momento de cuanto lleva gastado.
RF-6	Se debe poder dar de alta nuevos productos en la base de datos.
RF-7	Los códigos de producto son únicos en la base de datos.
RF-8	Se debe permitir que se realicen correcciones en el precio, la descripción o el número de unidades de un producto.
RF-9	Se deben apuntar las ventas que se hacen en un fichero.

### 4.3.2. Requisitos no funcionales

Código del requisito	Descripción
RNF-1	El sistema debe poder correr en un ordenador de bajas prestaciones.
RNF-2	No deben ser necesarias licencias de ejecución.
RNF-3	No se debe necesitar ningún motor de base de datos o cualquier otro software específico. El único software que resultará imprescindible adquirir será el sistema operativo.
RNF-4	La recuperación de una caída del sistema a mitad de una venta debe poder permitir la restauración de la situación anterior al inicio de dicha venta.
RNF-5	El número de productos distintos no es mayor de 500.
RNF-6	Suelen hacerse unas 25 ventas diarias.
RNF-7	El código de un producto es un número entero.
RNF-8	La descripción de un producto no ocupa más de 25 caracteres.
RNF-9	Las unidades y el precio son valores enteros.
RNF-10	El precio de cualquier producto no supera las 25.000 ptas.
RNF-11	Nunca se tiene más de 25 unidades de un mismo producto.
RNF-12	Nunca se ha hecho una venta de un importe superior a las 500.000 ptas.
RNF-13	Cada tienda irá equipada con un único PC. No se conectarán entre ellos.
RNF-14	Copiar a HD después de cada operación.
RNF-15	Sistema operativo Windows 9x/NT
RNF-16	Interfaz gráfica típica de Windows
RNF-17	No control de acceso
RNF-18	Campo descripción es una cadena de texto
RNF-19	Los operadores tienen unos conocimientos básicos de Windows



## Capítulo 5. Práctica 4

Partiendo de la ampliación propuesta en la práctica 3, en esta práctica se deberá realizar las fases de diseño y codificación de dicha ampliación. Para ello habrá que llevar a cabo las siguientes tareas:

- Modificar el diseño para dar cabida a la ampliación propuesta (el fichero diseño.odp dentro del directorio de la práctica tiene el diseño de la versión actual).
- Utilizar Visual C/C++ 6.0 para codificar dicha ampliación modificando y añadiendo el software necesario. Dentro del directorio del proyecto, en el directorio Proyecto\_El\_Regalo\ElRegalo se encuentra el fichero del proyecto (ElRegalo.dsp) que permite compilar la primera versión de la aplicación incluyendo las ventanas prototipadas. En el directorio Proyecto\_El\_Regalo\Fuentes se encuentra el código fuente de esta primera versión.

Copiar todo el directorio de la practica al escritorio para poder realizar las modificaciones oportunas sin problemas.



## Capítulo 6. Práctica 5

En esta práctica se deben diseñar y construir los juegos de prueba de caja negra que permitan detectar errores en el software creado para la gestión de la empresa “El Regalo S.A.”

La empresa “El Regalo S.A.” es una cadena de tiendas de pequeños artículos de regalo. Antes de la creación del software a analizar en esta práctica, cada vez que los empleados que trabajaban en estas tiendas efectaban una venta debían apuntar el código del producto que venden, así como una descripción del mismo para verificar que no se había producido ningún error de transcripción. Pero finalmente la empresa decidió automatizar en la medida de lo posible estas tareas mediante la adopción de soluciones informáticas. El consejo de dirección llegó a una solución de compromiso mediante la cual se decidió que se iba a abordar una primera fase de automatización en la cual se incluiría una sencilla gestión de *stocks* y de *ventas*.

A grandes rasgos, la primera versión de la aplicación permite introducir en la base de datos las mercancías que se reciben en la tienda, tanto si son nuevos productos, como si ya se vendían anteriormente. Igualmente, se pueden realizar en la base de datos correcciones tanto de descripción del producto (cambio de modelo, distinto color, etc.), como del precio o el número de unidades. Estas últimas suelen producirse generalmente por recepción de más unidades de un producto o por devolución de productos, pero también si se detectan desfases a la hora de hacer inventarios. La aplicación no permite que se puedan dar de alta dos productos con el mismo código.

A la hora de proceder a gestionar una venta, la aplicación proporciona un sistema flexible que permite comprobar los productos que se está llevando el cliente y efectuar las oportunas correcciones. Cada vez que un producto se vende, se quita de la base de datos. Además la aplicación verifica que no se puedan vender más unidades de un producto de las que se tienen contabilizadas en la base de datos. Además, se informa al cliente cuanto lleva gastado.

Para comprender mejor las funcionalidades de la aplicación se puede ejecutar el fichero “*ElRegalo.exe*” que contiene la versión ejecutable de la aplicación “El Regalo”.

**El objetivo de esta práctica va a ser construir los juegos de pruebas de caja negra que permitan comprobar los requisitos que aparecen en la tabla que se muestra a continuación. Uno o más de estos requisitos no son satisfechos por el software desarrollado. Para diseñar los casos de prueba de caja negra deberás utilizar al menos la técnica de identificación de particiones de equivalencia.**

Nr	Descripción
1	El número de productos distintos no es mayor de 500.
2	Nunca se ha hecho una venta de un importe superior a las 500.000 ptas.
3	La recuperación de una caída del sistema a mitad de cualquier operación debe poder permitir la restauración de la situación anterior al inicio de dicha operación.

Para poder codificar los juegos de prueba que hayas diseñado y solventar los posibles problemas que hubiese, deberás tomar como punto de partida el código fuente de la aplicación “El regalo”.

Utilizar Visual C/C++ 6.0 para codificar dicha ampliación modificando y añadiendo el software necesario. El directorio *Proyecto\_El\_Regalo* dentro del directorio de la práctica contiene el software del proyecto Visual C/C++ 6.0 a modificar. En el subdirectorio *Proyecto\_El\_Regalo\Pruebas* se encuentra el fichero del proyecto (*Pruebas.dsp*) en el que ya se incluyen todos los ficheros de código fuente necesarios para las pruebas. Y el subdirectorio *Proyecto\_El\_Regalo\FUENTES* es el que contiene los ficheros fuente, siendo los ficheros *productDataBase.h* y *productDataBase.c*, y *ticketDataBase.h* y *ticketDataBase.c* los que implementan la gestión de las bases de datos de productos y ventas de la aplicación “El Regalo”. También dentro del subdirectorio *Proyecto\_El\_Regalo\FUENTES* se encuentra el fichero *main.c* que será en el que deberás construir el código que realice las pruebas.

Copiar todo el directorio de la práctica al escritorio (o a un directorio con permiso de escritura para el usuario) para poder realizar las modificaciones oportunas sin problemas.



