

**ESTE EJERCICIO ES DE TIPO MIXTO.**

**ES IRRELEVANTE SI CONTESTA A LA PREGUNTA DE TEST O NO. SIN EMBARGO, SE DEBE ESCANEAR DICHA HOJA JUNTO CON EL RESTO DE LA CONTESTACIÓN DEL EXAMEN. EL ALUMNO PUEDE QUEDARSE CON EL ENUNCIADO.**

Todas las preguntas de este ejercicio son eliminatorias en el sentido de que debe obtener una nota mínima en cada una de ellas. En cada pregunta teórica, que se valora con 2'5 puntos, la nota mínima es 1 punto; en la segunda parte (ejercicio de teoría aplicada que se valora con 5 puntos) la nota mínima que debe obtener es de 2 puntos.

Conteste a las preguntas teóricas, en cualquier orden, en hojas diferentes a las que utilice para la contestación de la segunda parte. En cada parte, la cantidad MÁXIMA de papel (de examen, timbrado) que puede emplear ESTÁ LIMITADA al equivalente a DOS (2) HOJAS de tamaño A4 (210 x 297 mm)

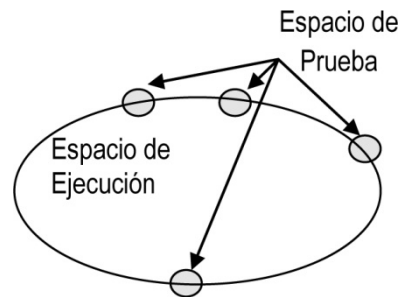
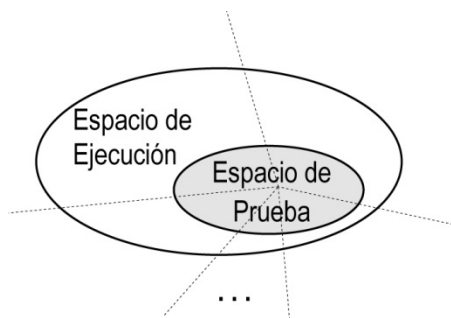
**PRIMERA PARTE. PREGUNTAS TEÓRICAS (2'5 PUNTOS CADA UNA)**

1. ¿Qué es el Análisis de Dominio y para qué se usa?

**Solución**

Epígrafes 2.1.2.5, en página 40, y 2.2.2 en la página 46.

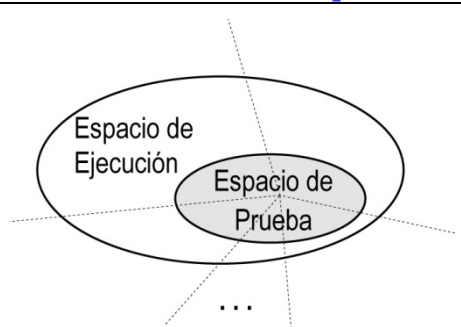
2. Las siguientes figuras representan dos métodos de prueba de caja negra: el análisis de valores límite y la partición en clases de equivalencia.
  - Estos métodos limitan el *espacio de prueba* a una parte del *espacio de ejecución*, ¿por qué?
  - Identifique qué método corresponde a cada figura y resuma brevemente en qué consiste cada método.

**Solución**

- a) En general, probar completamente un programa es inabordable y además no resulta rentable ni práctico. Por esta razón, se emplean métodos que limitan las pruebas a ciertas regiones del espacio de ejecución. Con las pruebas sólo se explora una parte de todas las posibilidades del programa. Se trata de alcanzar un compromiso para que con el menor esfuerzo posible se puedan detectar el máximo número de defectos y, sobre todo, aquellos que puedan provocar las consecuencias más graves.

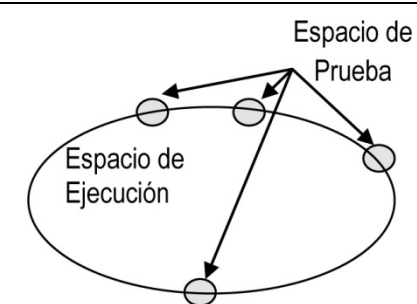
b)

### Partición en clases de equivalencia



Este método divide el espacio de ejecución de un programa en varios subespacios o clases equivalentes. Cada clase (delimitada por un par de líneas punteadas en la figura) agrupa a todos aquellos datos de entrada al programa que producen resultados equivalentes.

### Análisis de valores límite



Muchos programas se construyen codificando primero un tratamiento general, y retocando luego el código para cubrir casos especiales. Por esta y otras razones es bastante normal que los errores tengan cierta tendencia a aparecer precisamente al operar en las fronteras o valores límite de los datos normales (representados por círculos sombreados en la figura). Este método se basa en la identificación y prueba de los valores límite.

## SEGUNDA PARTE. PREGUNTA DE TEORÍA APLICADA (MÁXIMO 5 PUNTOS)

3. Un sistema informático de punto de venta (**PDV**) se suele utilizar en muchas tiendas para registrar ventas y realizar pagos. Tiene componentes hardware, como un terminal y un lector para los identificadores de los artículos. El software interactúa con módulos de terceras partes, que dan servicios de captura de facturación y cálculo de impuestos, de control del inventario y de cálculo de comisiones y márgenes de beneficio para el vendedor en el sistema de contabilidad. Una operación típica de este PDV es “**Procesar Venta**”, cuyo *escenario principal de éxito* (o **flujo básico**) se puede expresar, en *formato breve*:

**Procesar Venta:** Un cliente llega a la caja de una tienda para comprar varios artículos. El cajero utiliza el sistema PDV para registrar los artículos. El sistema presenta una suma parcial y detalles de cada línea de venta. El cliente introduce los datos del pago y el sistema los valida y registra. El sistema actualiza el inventario. El cliente recibe un recibo del sistema y se va con sus artículos.

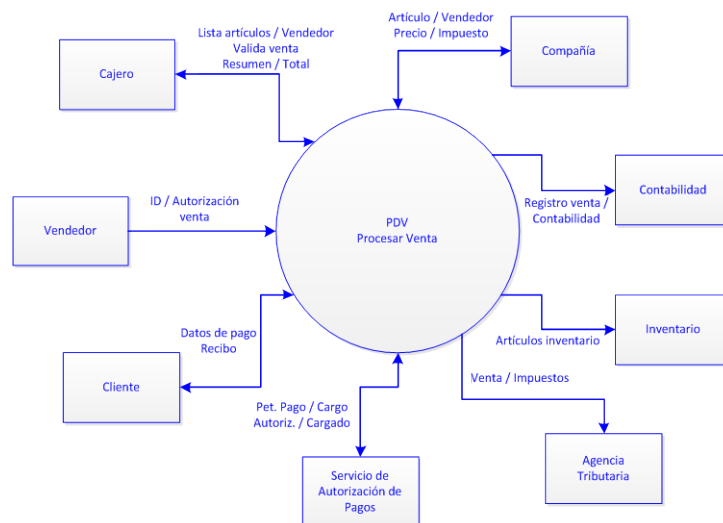
Reflexione sobre los intereses (necesidades) de cada uno de los actores involucrados en la operación:

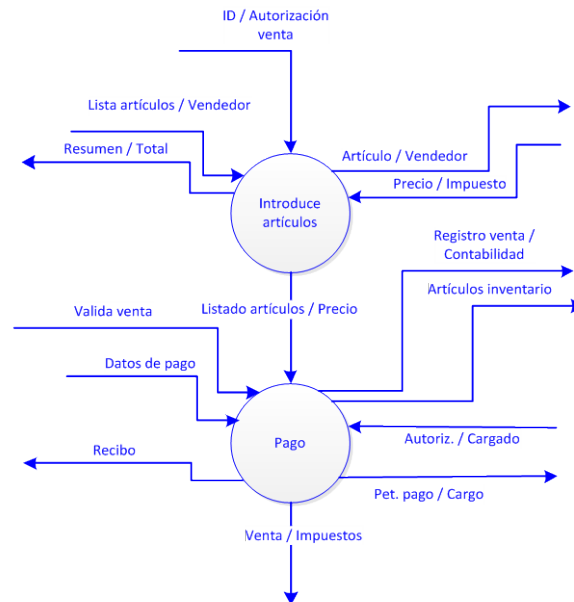
- **Compañía:** Quiere registrar las transacciones con precisión y satisfacer los intereses de los clientes. Quiere asegurar que se registran los pagos aceptados por el Servicio de Autorización de Pagos. Quiere actualización automática y rápida de la contabilidad y el inventario.
- **Cajero:** Quiere entradas precisas, rápidas y sin errores de pago, pues las pérdidas se deducen de su salario.
- **Cliente:** Quiere hacer el pago sin problemas y obtener el recibo correcto.
- **Vendedor:** Quiere actualizadas las comisiones de las ventas.
- **Agencia Tributaria:** Quiere recopilar los impuestos de cada venta.
- **Servicio de Autorización de Pagos:** Quiere recibir peticiones de autorización digital con el formato y protocolo correctos. Quiere registrar, de manera precisa, las cuentas por cobrar de la tienda.

- a. Construya un modelo de comportamiento para la operación **Procesar Venta**, descrita en su flujo básico, mediante DFDs (contexto y nivel 1). Explíquelo.

### Solución

Considerando la mayoría de los intereses de los actores involucrados en la operación, los DFD de nivel 0 y 1 son:



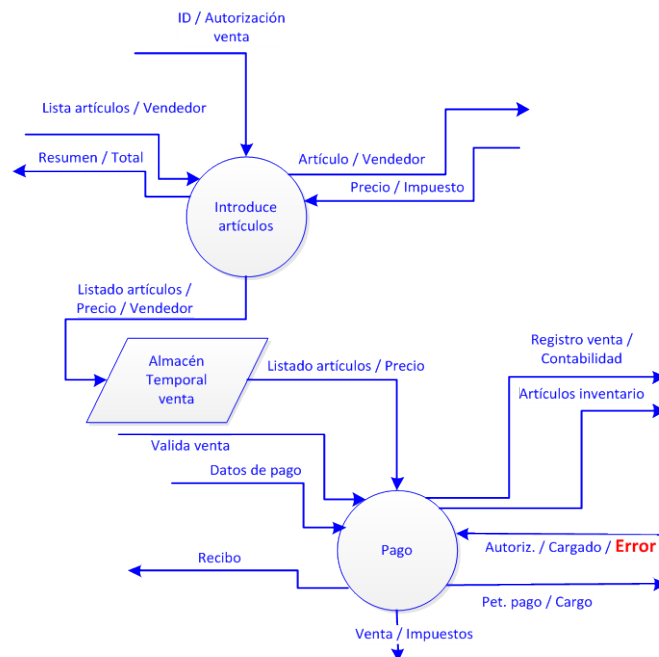


b. Qué solución aportaría al modelo anterior, si las necesidades de la compañía se modifican de la siguiente manera:

- **Compañía:** Quiere registrar las transacciones con precisión y satisfacer los intereses de los clientes. Quiere asegurar que se registran los pagos aceptados por el Servicio de Autorización de Pagos. Quiere cierta tolerancia a fallos que permita capturar las ventas, incluso si algún componente del servidor (ej. la validación remota del crédito) no están disponibles. Quiere actualización automática y rápida de la contabilidad y el inventario.

Vuelva a representar el DFD 0 anterior, pero con este nuevo requisito. Explíquelo.

### Solución



**ESTE EJERCICIO ES DE TIPO MIXTO.**

**ES IRRELEVANTE SI CONTESTA A LA PREGUNTA DE TEST O NO. SIN EMBARGO, SE DEBE ESCANEAR DICHA HOJA JUNTO CON EL RESTO DE LA CONTESTACIÓN DEL EXAMEN. EL ALUMNO PUEDE QUEDARSE CON EL ENUNCIADO.**

Todas las preguntas de este ejercicio son eliminatorias en el sentido de que debe obtener una nota mínima en cada una de ellas. En cada pregunta teórica, que se valora con 2'5 puntos, la nota mínima es 1 punto; en la segunda parte (ejercicio de teoría aplicada que se valora con 5 puntos) la nota mínima que debe obtener es de 2 puntos.

Conteste a las preguntas teóricas, en cualquier orden, en hojas diferentes a las que utilice para la contestación de la segunda parte. En cada parte, la cantidad MÁXIMA de papel (de examen, timbrado) que puede emplear ESTÁ LIMITADA al equivalente a DOS (2) HOJAS de tamaño A4 (210 x 297 mm)

**PRIMERA PARTE. PREGUNTAS TEÓRICAS (2'5 PUNTOS CADA UNA)**

1. ¿En qué consiste la actividad de reingeniería y cuándo es necesaria?

**Solución**

Consiste en generar un sistema bien organizado y documentado a partir de un producto software que no fue desarrollado siguiendo técnicas de ingeniería de software.

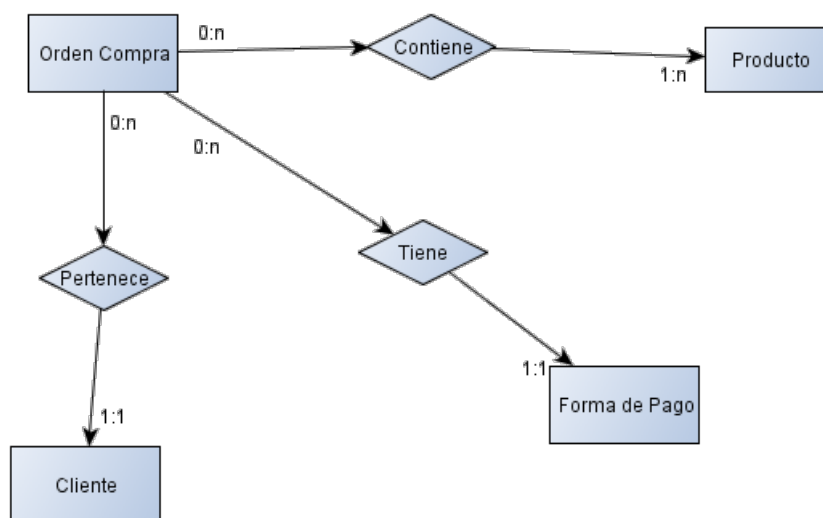
En ocasiones es necesaria para el mantenimiento de dichos productos software. (Pág. 26)

2. El sistema de ventas por Internet de una tienda funciona de la siguiente manera: para que el cliente formalice la compra debe estar previamente registrado. El formulario de compra consiste básicamente en tres partes: datos del cliente, forma de pago y la lista de los productos seleccionados. Cuando se formalice la compra el sistema guarda dicha operación con: un identificador (orden de compra), el cliente y la lista de productos.

Realice un diagrama de modelos de datos Entidad - Relación de la compra. Describa los datos más relevantes mediante el *diccionario de datos*.

**Solución**

El diagrama E – R para el modelo de datos de la 'compra':



Y el diccionario de datos para los elementos más relevantes:

Nombre: **Orden de compra**

Estructura: **Identificador + Cliente + Forma de Pago + {Producto}**  
**Identificador = {CaracterAlfanumérico}<sup>N</sup>**

Nombre: **Producto**

Estructura: **Nombre + Identificador + Precio**

Nombre: **Forma de pago**

Estructura: **[ Contrareembolso | Tarjeta | Transferencia ]**

Nombre: **Cliente**

Estructura: **Nombre + Apellidos + IdDNI + Usuario + Clave + Dirección**

**Nombre = {CaracterAlfanumérico}<sup>10</sup> /ristra de 10 caracteres /**

**Apellidos = {CaracterAlfanumérico}<sup>30</sup> /ristra de 30 caracteres/**

**IdDNI = {Dígito}<sup>8</sup>**

**Usuario = {CaracterAlfanumérico}<sup>10</sup>**

**Clave = {CaracterAlfanumérico}<sup>10</sup>**

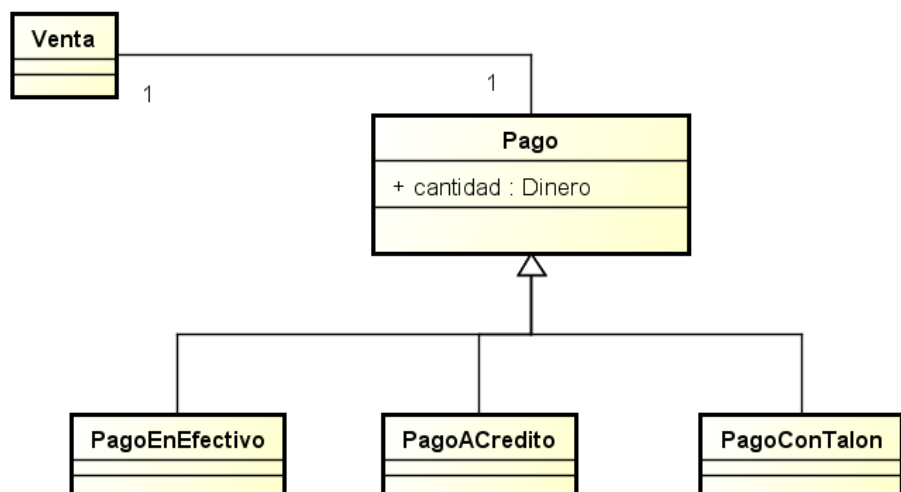
**Dirección = {CaracterAlfanumérico}<sup>50</sup>**

## SEGUNDA PARTE. PREGUNTA DE TEORÍA APLICADA (MÁXIMO 5 PUNTOS)

3. Un sistema informático de punto de venta (**PDV**) se suele utilizar en muchas tiendas para registrar ventas y realizar pagos. Una operación típica de este PDV es “**Procesar Venta**” en la que, tras registrar los artículos y calcular el importe, el proceso de **Venta** invoca la ejecución del **Pago**. Ahora bien, el **Pago** puede ser **EnEfectivo**, **ACredito** o **ConCheque**. Al hacer el diseño, se ha creado una clase **Venta** asociada a la clase **Pago**.

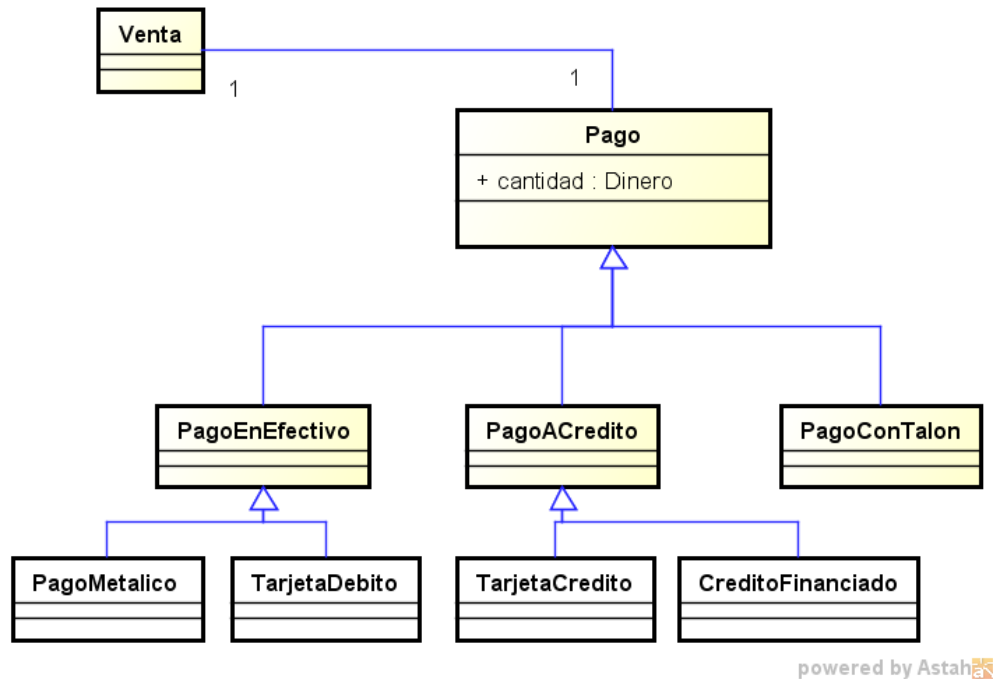
- a. Construya un modelo de objetos para las modalidades de pago.

### Solución



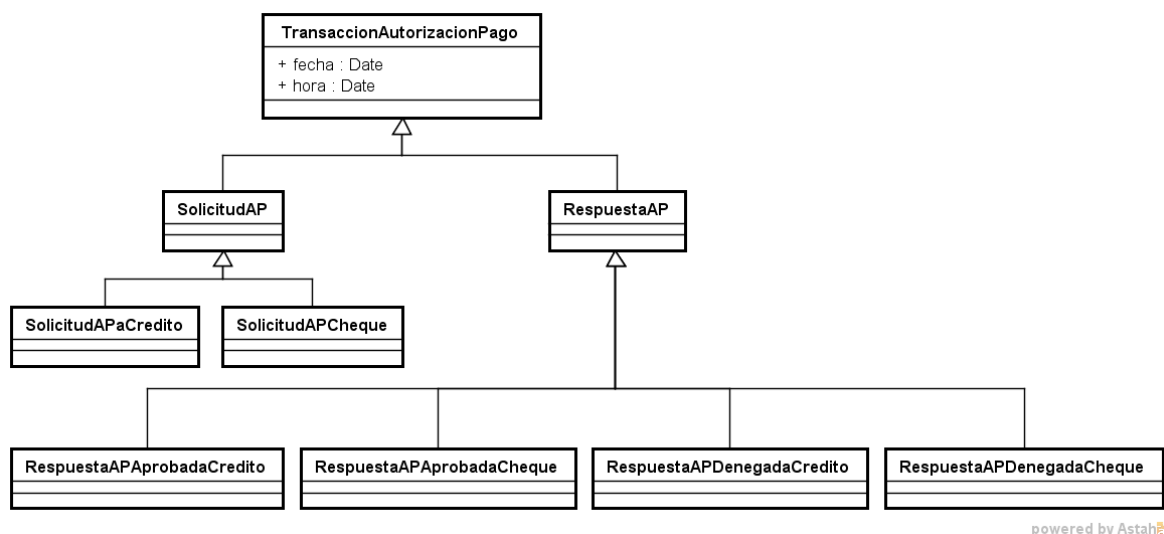
- b. Suponga que también se quiere considerar el pago en metálico, con tarjeta de débito, a crédito con tarjeta de crédito y crédito financiado. Construya el nuevo diagrama.

### Solución



- c. En alguna de estas modalidades de pago, su ejecución implica Transacciones de Autorización del Pago. Una TransaccionAutorizacionPago es una SolicitudAutorizacionPago o bien una RespuestaAutorizacionPago. A su vez, la respuesta puede ser RespuestaAprobadaPago o RespuestaDenegadaPago. Es obvio que cada transacción se comporta de distinta forma con cada modalidad de pago. Para las modalidades de Pago ACredito y ConCheque, construya la jerarquía de clases partiendo de TransaccionAutorizacionPago; de manera que tenga la máxima simplicidad y flexibilidad. ¿Cuántas clases hay que añadir si incluimos la comprobación de saldo en una tarjeta de débito?

### Solución



**ESTE EJERCICIO ES DE TIPO MIXTO.**

**ES IRRELEVANTE SI CONTESTA A LA PREGUNTA DE TEST O NO. SIN EMBARGO, SE DEBE ESCANEAR DICHA HOJA JUNTO CON EL RESTO DE LA CONTESTACIÓN DEL EXAMEN. EL ALUMNO PUEDE QUEDARSE CON EL ENUNCIADO.**

Todas las preguntas de este ejercicio son eliminatorias en el sentido de que debe obtener una nota mínima en cada una de ellas. En cada pregunta teórica, que se valora con 2'5 puntos, la nota mínima es 1 punto; en la segunda parte (ejercicio de teoría aplicada que se valora con 5 puntos) la nota mínima que debe obtener es de 2 puntos.

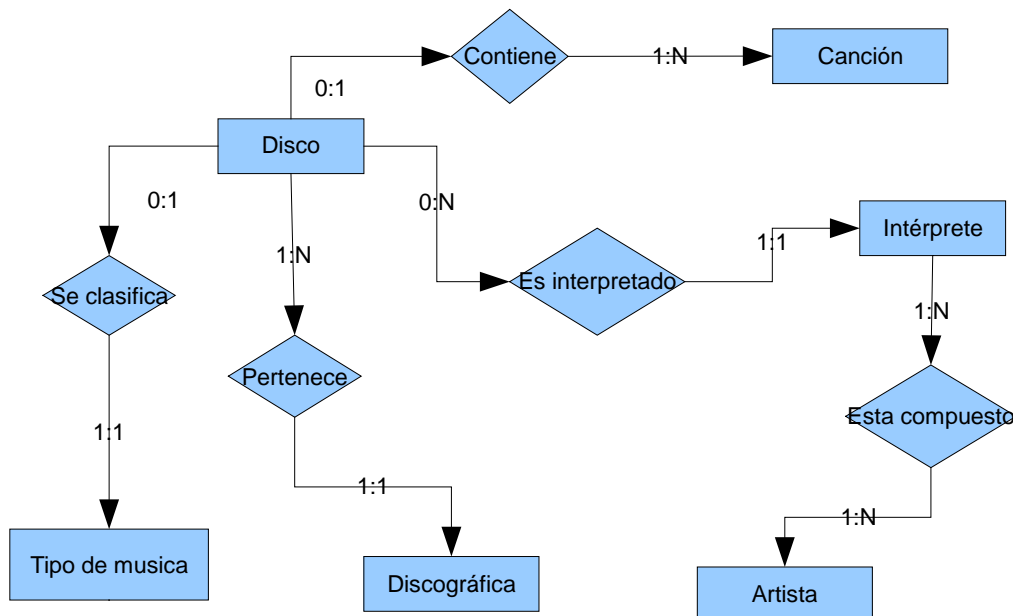
Conteste a las preguntas teóricas, en cualquier orden, en hojas diferentes a las que utilice para la contestación de la segunda parte. En cada parte, la cantidad MÁXIMA de papel (de examen, timbrado) que puede emplear ESTÁ LIMITADA al equivalente a DOS (2) HOJAS de tamaño A4 (210 x 297 mm)

**PRIMERA PARTE. PREGUNTAS TEÓRICAS (2'5 PUNTOS CADA UNA)**

- La gestión del catálogo de discos de una biblioteca se debe basar en la siguiente especificación:

*Un disco tiene un título y un intérprete. El intérprete puede ser un artista sólo o un grupo (conjunto de artistas). El disco pertenece a una discográfica y contiene una serie de canciones. Así mismo, cada disco puede ser clasificado según un tipo de música.*

Modele el enunciado anterior mediante un **Diagrama Entidad-Relación**. Describa los principales datos utilizando la notación del **Diccionario de Datos**.

**Solución**



Diccionario de datos:

DISCO = Nombre + INTERPRETE + TIPO-DE-MÚSICA + DISCOGRÁFICA + LISTA-DE-CANCIONES + año

INTERPRETE = ARTISTA

ARTISTA = {nombre}

DISCOGRÁFICA = nombre


TIPO-DE-MÚSICA = [jazz | clásica | pop | rock | heavy | dance ]

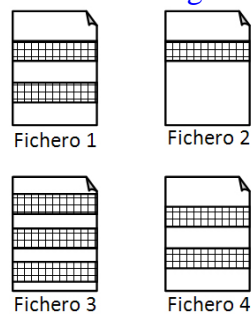
CANCIÓN = nombre + ARTISTA



LISTA-DE-CANCIONES = {CANCIÓN}

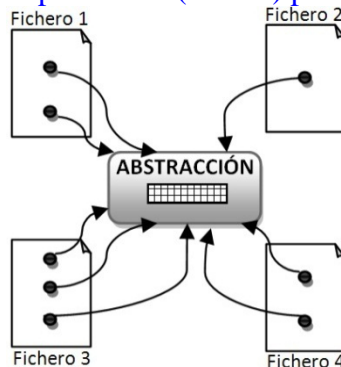
2. Una práctica muy extendida para reutilizar software es “cortar y pegar” fragmentos de código. Esto puede llevar a que un trozo de código se encuentre repetido muchas veces en un programa. ¿Cómo afecta esto al mantenimiento del programa? ¿Cómo podrían solventarse los efectos negativos del código duplicado?

### Solución

La duplicidad de código perjudica la adaptabilidad de un programa. Por ejemplo, la siguiente figura representa un programa compuesto por 4 ficheros que contienen cierto código repetido (representado por ). Si durante el mantenimiento del programa se decide modificar dicho código repetido, habrá que realizar el cambio en todos los lugares donde se encuentra el código (en el ejemplo, habría que repetir el cambio en 8 puntos). Este trabajo no sólo es tedioso, si no que puede llevar a inconsistencias cuando los cambios no se aplican a todos los lugares donde el código está repetido.



Como indica la siguiente figura, estos problemas podrían evitarse encapsulando el código repetido en algún tipo de abstracción y sustituyendo las repeticiones () por invocaciones () a la nueva abstracción.



## SEGUNDA PARTE. PREGUNTA DE TEORÍA APLICADA (MÁXIMO 5 PUNTOS)

3. Se ha codificado en Modula-2 el siguiente subprograma que distingue, por el tamaño de sus lados si un triángulo es isósceles (dos lados iguales), equilátero (todos los lados iguales) o escaleno (ningún lado igual).

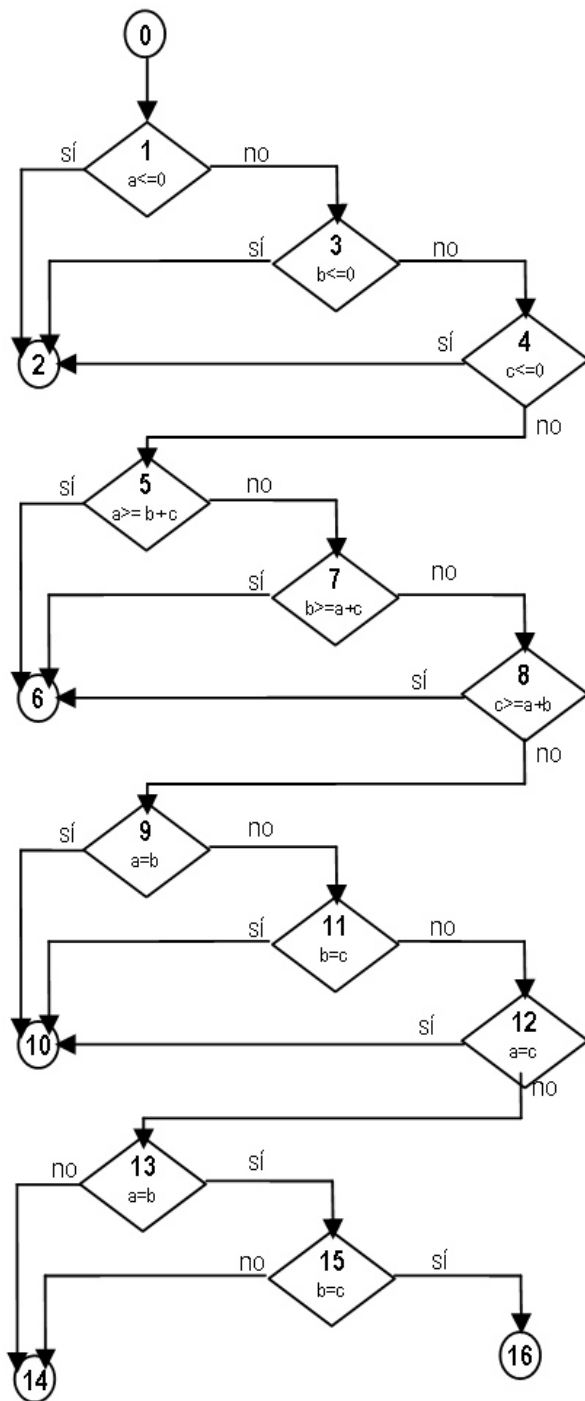
```
PROCEDURE TipoDeTriangulo(a, b, c: INTEGER);
BEGIN
  IF (a<=0) OR (b<=0) OR (c<=0) THEN
    WriteString("Error: alguno de los parámetros no es un número natural");
  ELSIF (a>=b+c) OR (b>=a+c) OR (c>=a+b) THEN
    WriteString("Error: no es posible cerrar el triángulo");
  ELSIF (a=b) OR (b=c) OR (a=c) THEN
    WriteString("El triángulo es isósceles");
  ELSIF (a=b) AND (b=c) THEN
    WriteString("El triángulo es equilátero");
  ELSE
    WriteString("El triángulo es escaleno");
  END;
END TipoDeTriangulo;
```

*Verifique el subprograma con pruebas de caja transparente, realizando el cubrimiento lógico correspondiente.*

### Solución

Para la verificación del subprograma se elaborará un conjunto de casos de prueba que consigan que se transite por todos los posibles caminos de ejecución y que pongan en juego todos los elementos del código.

En primer lugar, el código del subprograma se transformará en el siguiente diagrama de flujo, donde cada rombo representa un predicado lógico simple:



② Parámetros erróneos: valores no naturales

⑥ Parámetros erróneos: imposible cerrar el triángulo

⑩ Triángulo isósceles

⑭ Triángulo escaleno

⑮ Triángulo equilátero

A continuación, se calculará el nº de caminos básicos para recorrer todas las líneas de flujo del diagrama al menos una vez:

Nº de predicados = 11

Nº máximo de caminos = Nº de predicados + 1 = 12

A continuación se determinarán los caminos y los casos de prueba que forzarán su recorrido.

Como indica la siguiente figura, es imposible escribir un juego de prueba para los caminos 0-1-3-4-5-7-8-9-12-13-15-14 y 0-1-3-4-5-7-8-9-12-13-15-16. Es decir, **el subprograma es incorrecto** por que contiene dos caminos que no pueden recorrerse. Concretamente, la imposibilidad de transitar por el camino 0-1-3-4-5-7-8-9-12-13-15-16 hace que el subprograma clasifique erróneamente los triángulos equiláteros como isósceles.

Camino		Resultado esperado	Juego de prueba
1	0-1-2	Parámetros erróneos: valores no naturales	a=0, b=1, c=1
2	0-1-3-2		a=1, b=-3, c=1
3	0-1-3-4-2		a=3, b=1, c=0
4	0-1-3-4-5-6	Parámetros erróneos: imposible cerrar el triángulo	a=5, b=1, c=1
5	0-1-3-4-5-7-6		a=2, b=4, c=2
6	0-1-3-4-5-7-8-6		a=3, b=1, c=5
7	0-1-3-4-5-7-8-9-10	Triángulo isósceles	a=2, b=2, c=1
8	0-1-3-4-5-7-8-9-11-10		a=2, b=3, c=3
9	0-1-3-4-5-7-8-9-12-10		a=4, b=3, c=4
10	0-1-3-4-5-7-8-9-12-13-14	Triángulo escaleno	a=3, b=4, c=5
11	0-1-3-4-5-7-8-9-12-13-15-14		<b>Es imposible crear un juego de prueba</b>
12	0-1-3-4-5-7-8-9-12-13-15-16	Triángulo equilátero	

**ESTE EJERCICIO ES DE TIPO MIXTO.**

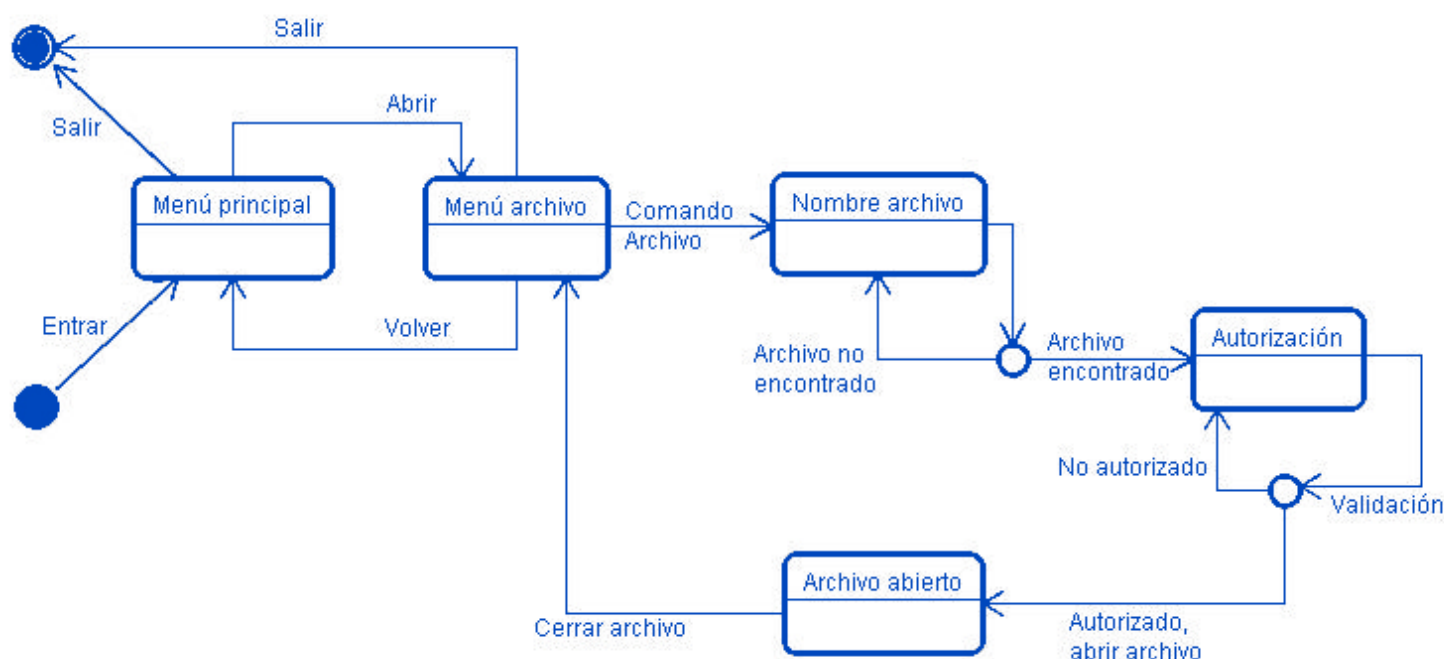
**ES IRRELEVANTE SI CONTESTA A LA PREGUNTA DE TEST O NO. SIN EMBARGO, SE DEBE ESCANEAR DICHA HOJA JUNTO CON EL RESTO DE LA CONTESTACIÓN DEL EXAMEN. EL ALUMNO PUEDE QUEDARSE CON EL ENUNCIADO.**

Todas las preguntas de este ejercicio son eliminatorias en el sentido de que debe obtener una nota mínima en cada una de ellas. En cada pregunta teórica, que se valora con 2'5 puntos, la nota mínima es 1 punto; en la segunda parte (ejercicio de teoría aplicada que se valora con 5 puntos) la nota mínima que debe obtener es de 2 puntos.

Conteste a las preguntas teóricas, en cualquier orden, en hojas diferentes a las que utilice para la contestación de la segunda parte. En cada parte, la cantidad MÁXIMA de papel (de examen, timbrado) que puede emplear ESTÁ LIMITADA al equivalente a DOS (2) HOJAS de tamaño A4 (210 x 297 mm)

**PRIMERA PARTE. PREGUNTAS TEÓRICAS (2'5 PUNTOS CADA UNA)**

1. Una interfaz gráfica tiene un “*Menú principal*” con un submenú de “*Archivo*” que, a su vez, tiene el comando de “*Abrir archivo*”. Cada menú tiene una opción para “*Salir*”. El comando “*Abrir archivo*” tiene un *cuadro de texto* en el que se puede escribir el nombre (y opcionalmente la ruta) del archivo que se quiere abrir. Supóngase que sólo se puede abrir un archivo cada vez. Si el nombre del archivo es correcto, se pedirá que se escriba una clave para autorizar su apertura y, si es incorrecta, la interfaz vuelve al submenú “*Archivo*”. *Construya un modelo del comportamiento (para el análisis) utilizando un diagrama de transición de estados.*

**Solución**

2. Varios módulos de una aplicación utilizan la información de un fichero que está en un disco. Para usarlo, los módulos invocan los servicios que provee la controladora del dispositivo –los cuales trabajan directamente con las características físicas del disco—. Por ejemplo, para escribir en el fichero, los módulos invocan la instrucción imaginaria del microcontrolador del disco: `WRITE_STRING(DiskID, Cyl, Sector, Pos, NBytes, "String")`. ¿Qué tipo de acoplamiento presenta este caso? ¿Qué consecuencias tiene este uso respecto a la reusabilidad y a la mantenibilidad? ¿Cómo cambiaría esta situación?

### Solución

Página 152 de bibliografía básica.

Éste es un claro ejemplo de acoplamiento externo, en el que varios módulos comparten información que reside en un dispositivo externo a ellos y a la que acceden directamente a través del ‘hardware’ que controla el dispositivo.

Esta práctica obliga a realizar todas las operaciones a través de las rutinas de servicio del propio dispositivo. Al ser, dichas rutinas, específicas del dispositivo; la posibilidad de reutilización del código –reusabilidad— en cualquier otro dispositivo que no sea idéntico, está –obviamente— disminuido o anulado. En cuanto a la mantenibilidad –o facilidad y posibilidad de hacer modificaciones en el código—, también se ve gravemente afectada; puesto que, si se cambia algo del dispositivo externo, obliga a cambiar todas las partes del código en las que se utilice dicho dispositivo.

La manera de evitar todo esto es mediante la abstracción y el encapsulamiento de todas estas operaciones en un módulo o paquete que se constituya como un controlador abstracto y genérico del dispositivo –disco— y que contenga, bajo invocaciones genéricas a los antiguos servicios específicos del dispositivo, las rutinas concretas de operación del dispositivo. Así, si se produce algún cambio, sólo será necesario hacer las modificaciones en el ‘controlador abstracto’.

## SEGUNDA PARTE. PREGUNTA DE TEORÍA APLICADA (MÁXIMO 5 PUNTOS)

3. Un fichero secuencial almacena información de entradas y salidas de stocks de un almacén. Estas transacciones están agrupadas en lotes. Cada lote consiste en una serie de transacciones referentes a un mismo producto. Los lotes comienzan con una marca de cabecera, un conjunto de registros con las entradas o salidas del producto y terminan con otra marca de fin.

***Diseñe un programa que cree un informe con los movimientos netos de cada producto. Utilice la metodología de Jackson para las estructuras de datos de entrada y salida así como para la estructura del programa.***

### Solución

