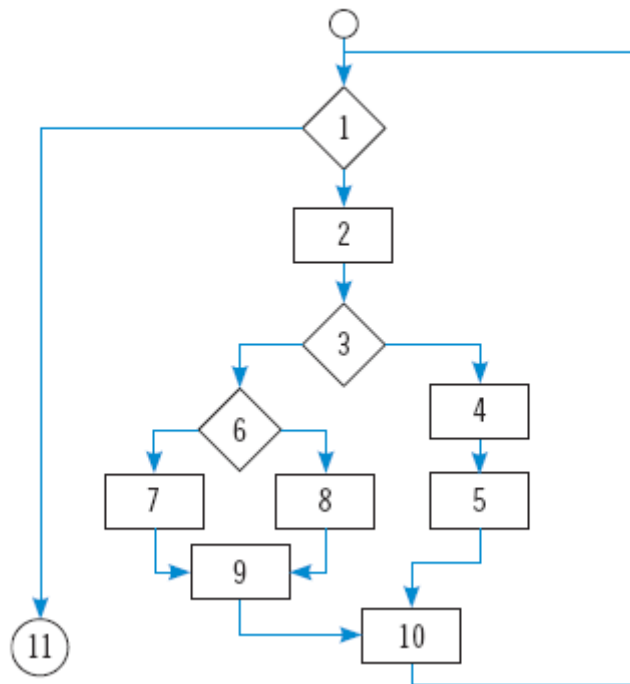


Aplicación práctica sobre el cálculo de la complejidad ciclomática

Calcular la complejidad ciclomática del diagrama de flujo presentado en la siguiente imagen. Obtener también las rutas independientes.

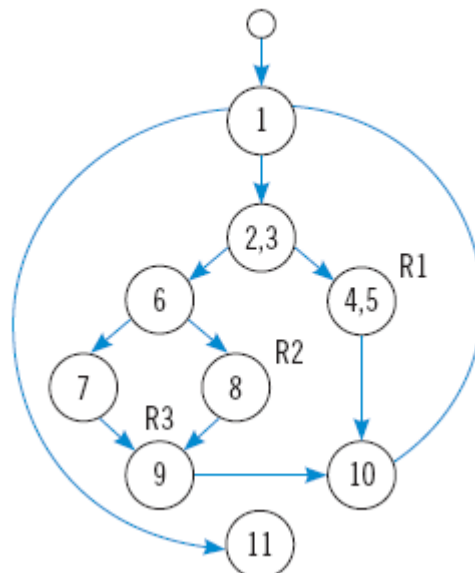
Flujo del programa



Solución A partir del diagrama se obtiene el gráfico de flujo. Este gráfico es una simplificación del diagrama, mostrando las instrucciones secuencias contenidas dentro del mismo nodo. Así, 4 y 5 van dentro de un nodo, pero 10 no se incluye, ya que, pese a ser secuencial con las anteriores, también es compartida por 9. El resto de las instrucciones tienen su propio nodo.

En el gráfico resultante se observa que hay cuatro regiones (tres cerradas más la exterior). Con esto, ya se dispone de la complejidad ciclomática (que sería 4), pero para contrastar este dato se aplicará también el cálculo mediante la fórmula.

Grafo de flujo



Sabiendo que hay once aristas y nueve nodos, utilizando la fórmula previamente descrita, se confirma que el grafo tiene una complejidad de 4.

Ahora quedaría obtener las rutas independientes, que se sabe que son 4, puesto que es el valor de la complejidad ciclomática. De la apreciación del gráfico se deducen las siguientes rutas:

- **Ruta 1:** 1 – 11
- **Ruta 2:** 1 – 2 – 3 – 4 – 5 – 10 – 1 – 11
- **Ruta 3:** 1 – 2 – 3 – 6 – 8 – 9 – 10 – 1 – 11
- **Ruta 4:** 1 – 2 – 3 – 6 – 7 – 9 – 10 – 1 – 11

Diseño de pruebas para caja negra: Prueba de arreglo ortogonal

Estas pruebas tienen aplicación cuando ***se dispone de varias entradas con valores claramente definidos***. Un ejemplo sería un programa con tres puntos de entrada que pueden tomar un valor del rango (1, 2, 3). Si se quiere realizar una prueba exhaustiva, habría que probar todas las combinaciones, que son 27 casos de prueba.

Combinaciones de entrada							
PRUEBA	ENTRADA 1	ENTRADA 2	ENTRADA 3	PRUEBA	ENTRADA 1	ENTRADA 2	ENTRADA 3
Caso 1	1	1	1	Caso 15	2	2	3
Caso 2	1	1	2	Caso 16	2	3	1
Caso 3	1	1	3	Caso 17	2	3	2
Caso 4	1	2	1	Caso 18	2	3	3
Caso 5	1	2	2	Caso 19	3	1	1
Caso 6	1	2	3	Caso 20	3	1	2
Caso 7	1	3	1	Caso 21	3	1	3
Caso 8	1	3	2	Caso 22	3	2	1
Caso 9	1	3	3	Caso 23	3	2	2
Caso 10	2	1	1	Caso 24	3	2	3
Caso 11	2	1	2	Caso 25	3	3	1
Caso 12	2	1	3	Caso 26	3	3	2
Caso 13	2	2	1	Caso 27	3	3	3
Caso 14	2	2	2				

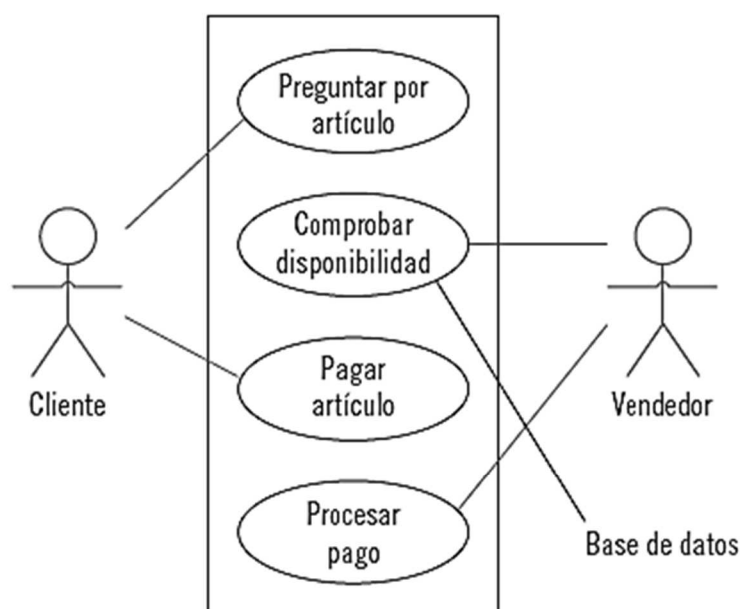
El arreglo ortogonal contiene un número reducido de combinaciones, además de disponer de una ***propiedad de equilibrio*** (que en cada columna los valores tengan la misma frecuencia de aparición) que provoca que ***los casos se dispersen de manera uniforme***. La siguiente imagen establece un arreglo de tipo L9 (donde 9 es el número de casos de prueba).

Arreglo ortogonal L9			
PRUEBA	ENTRADA 1	ENTRADA 2	ENTRADA 3
Caso 1	1	1	1
Caso 2	1	2	2
Caso 3	1	3	3
Caso 4	2	1	2
Caso 5	2	2	3
Caso 6	2	3	1
Caso 7	3	1	3
Caso 8	3	2	1
Caso 9	3	3	2



Diagrama de caso de uso "Compra de artículo"

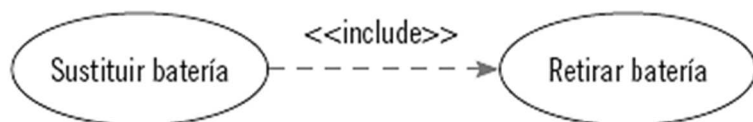
Ejemplo de diagrama de casos de uso



En un mismo diagrama pueden estar presentes varios casos de uso. En el supuesto de que dos casos estén ligados entre sí, se puede establecer una relación de dependencia entre ambos. Se permiten tres tipos de relación:

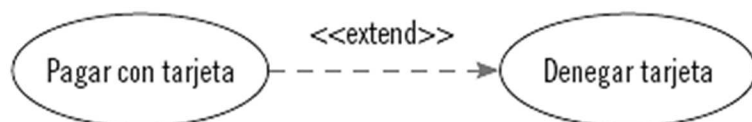
- **<<include>>**: un caso de uso está incluido dentro de un caso base, de tal manera que la realización de uno implique necesariamente la realización del otro. Por ejemplo: "Sustituir batería" incluye "Retirar batería antigua" o, lo que es lo mismo, "Retirar batería antigua" está incluido en "Sustituir batería".

Ejemplo de <<include>>



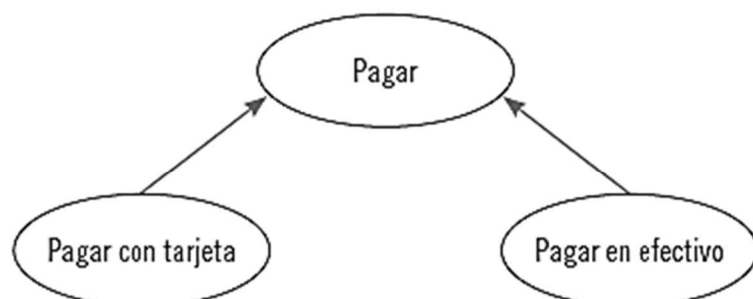
- **<<extend>>**: un caso de uso puede extender de un caso base bajo ciertas circunstancias. Mientras que <<include>> implicaba la ejecución del caso de uso incluido, ahora esa ejecución depende de una condición. Por ejemplo: el caso de uso "Denegar tarjeta" extenderá de "Pagar con tarjeta" cuando se haya producido un error al efectuar el pago.

Ejemplo de <<extend>>



- **Generalización:** un caso de uso es una especialización de un caso base (o un caso base es una generalización de un determinado caso de uso). Es parecido a la idea de orientación a objetos, que se tratará en capítulos posteriores. Como ejemplo se pueden exponer los casos de uso “Pagar con tarjeta” y “Pagar en efectivo”, cuya generalización es el caso “Pagar”.

Ejemplo de generalización



Recuerde

En el <<include>> hay un comportamiento común, siendo un caso de uso consecuencia o implicación de un caso base. En el <<extend>>, una circunstancia o condición puede implicar que se dispare (o no) el caso de uso que extiende del caso base.

Ejemplo

Especificación de caso de uso "Compra de artículo".

Especificación de Caso de Uso					
Caso de Uso	Compra de Artículo				CU-1
Actores	Cliente (iniciador), Vendedor, Base de Datos del Sistema				
Tipo	Primario				
Referencias	-----	-----			
Precondición	-----				
Postcondición	La venta queda registrada en la Base de Datos del Sistema				
Autor	RGL	Fecha	07/09/2013	Versión	1
Propósito					
Registrar la venta de un artículo					
Resumen					
El Cliente llega a la tienda. Pregunta por un artículo. El Vendedor consulta disponibilidad y le trae el artículo. El Cliente paga, el Vendedor registra la Venta y el Cliente se marcha con su producto.					
Flujo normal					
1	El Cliente llega al mostrador				
2	El Cliente solicita un artículo al Vendedor.				
3	El Vendedor se identifica en el Sistema.				
4	El Vendedor comprueba disponibilidad.				
5	El Vendedor va al almacén por el artículo solicitado.				
6	El Vendedor informa del precio al Cliente.				
7	El Cliente paga (en efectivo o tarjeta).				
8	El Vendedor marca la venta del artículo en el Sistema.				
9	El Vendedor genera el ticket.				
10	El Cliente toma el ticket y su artículo.				

Flujos alternos	
4a	El artículo no tiene disponibilidad actualmente. Sugerir su encargo y volver al punto 1.
6a	El Cliente considera que el artículo es muy caro. Cancelar.
7a	El Cliente paga con tarjeta y hay un error durante el proceso. Sugerir pagar en efectivo o Cancelar.

Aplicación práctica sobre diagrama de flujo

En una entrevista de trabajo le proponen realizar un diagrama de flujo de nivel 0. Se le informa que las entidades externas son Impresora, Usuario y Pantalla, mientras que existe un almacén externo que es Base de Datos.

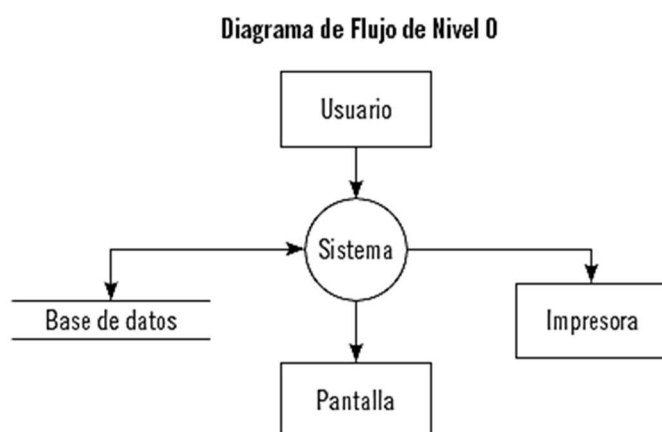
Realizar también el diagrama de flujo de nivel 1 sabiendo que el sistema presenta las siguientes opciones:

- Imprimir (accede a la entidad base de datos).
- Mostrar por pantalla (accede a la base de datos y muestra información por pantalla).
- Realizar altas y bajas (accede/modifica a la base de datos).

Solución

En el Diagrama de Nivel 0 (también llamado Diagrama de Contexto) se representa el sistema junto con el resto de entidades. Del texto se obtiene que hay tres entidades (usuario, impresora y pantalla), junto con un almacén de base de datos.

Únicamente hay que situar los elementos en el diagrama, prestando atención a la doble flecha con la base de datos (ya que recibe información para almacenar y permite consulta de datos).

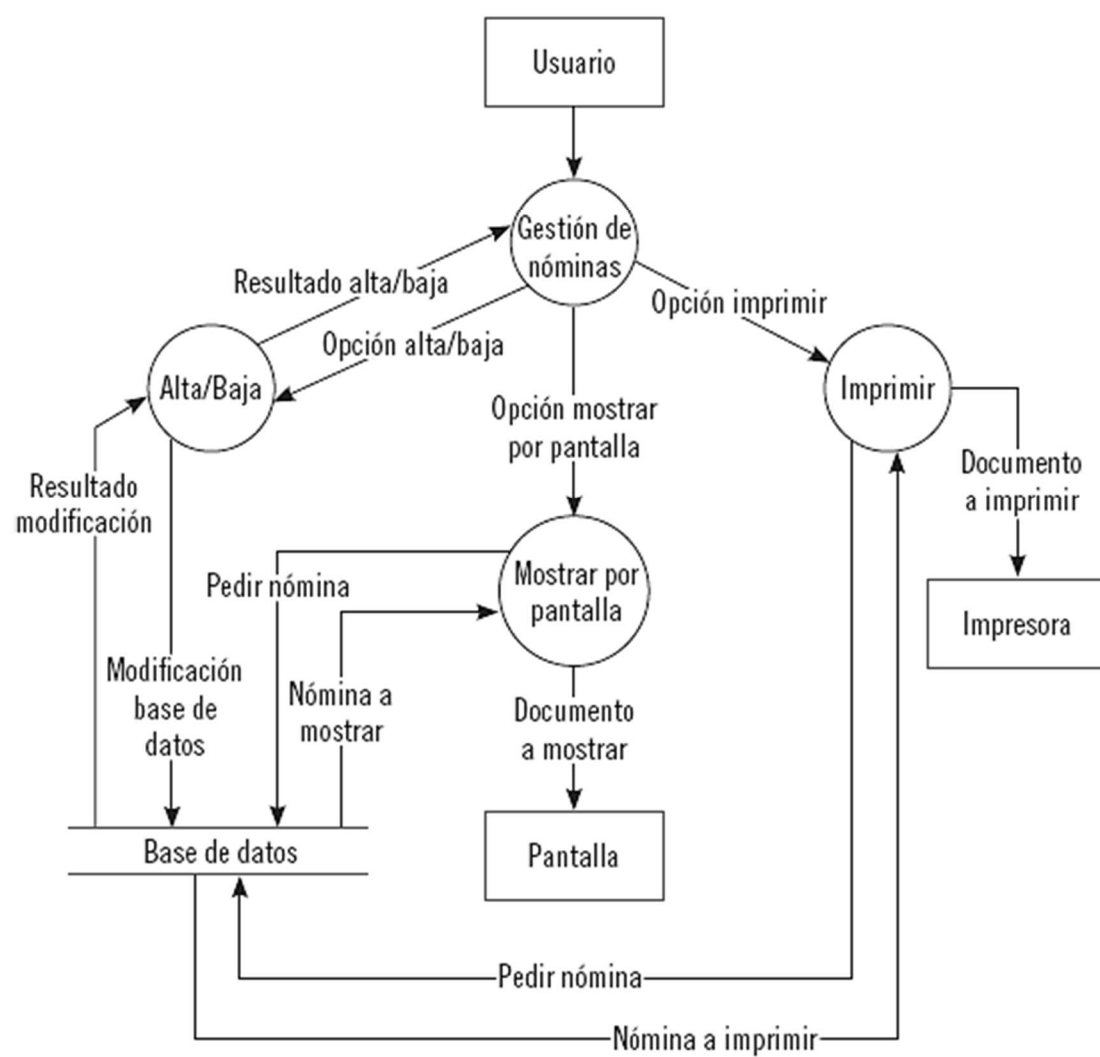


Para el Diagrama de Nivel 1, se considera un proceso central (*Gestión de nóminas*) que coordinará el resto de procesos. A partir de *Gestión de nóminas* se accederá a las opciones propuestas (Imprimir, Mostrar por pantalla y Alta/baja).

En cada una de las interacciones se añade una etiqueta con información complementaria en lenguaje natural. A continuación se proponen las siguientes etiquetas, todas ellas en concordancia con lo demandado en el enunciado:

- Pedir nómina: Acceso a base de datos para obtener una nómina.
- Modificación base de datos: Modificación de la base de datos de las nóminas.
- Opción alta/baja: Se selecciona la opción "Alta/baja" en el menú.
- Opción mostrar por pantalla: Se selecciona la opción "Mostrar por pantalla" en el menú.
- Opción imprimir: Se selecciona la opción "Imprimir" en el menú.
- Nómina a mostrar: Nómina para mostrar por pantalla.
- Nómina a imprimir: Nómina para imprimir.
- Documento a imprimir: Documento ya formateado para imprimir.
- Resultado modificación: Resultado de la modificación en la base de datos, para informar al subproceso "Alta/baja".
- Resultado alta/baja: Resultado de la opción "Alta/baja" para informar al proceso principal "Gestión de nóminas".

Diagrama de Flujo de Nivel 1



Diagramas de Diseño: Diagrama de Secuencia

Aplicación práctica sobre diagramas de secuencia

Modele un diagrama de secuencia para una actividad de compra online. La secuencia del proceso descrita en el caso de uso es la siguiente:

- El cliente realiza el pedido de un artículo en la tienda online.
- El encargado de la tienda prepara el paquete para envío y espera la confirmación de pago.
- El cliente recibe la notificación de que el pedido está listo y paga el importe.
- El banco confirma el pago a la tienda online y al cliente.
- El encargado envía el paquete.

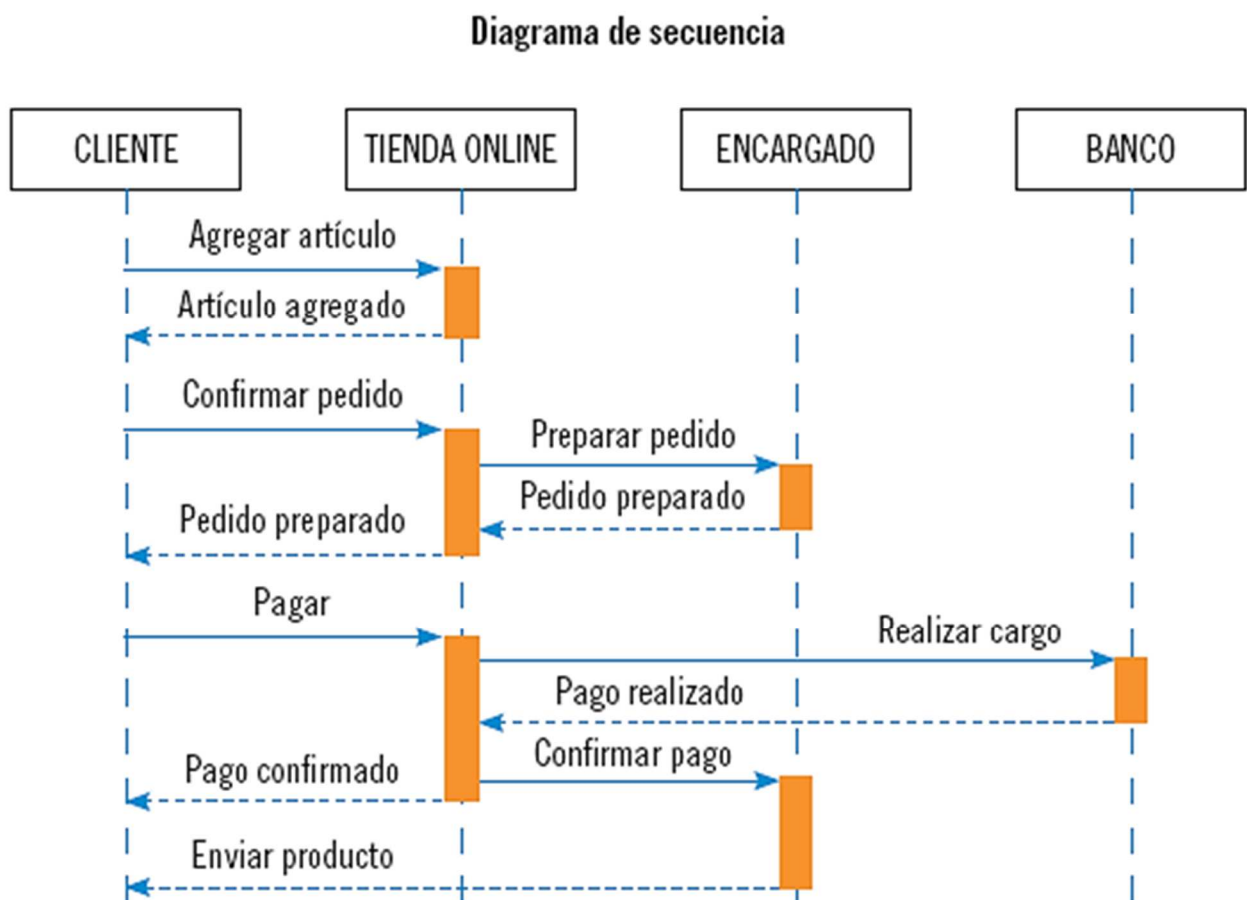
Solución

Del enunciado se deduce que hay cuatro participantes: **cliente**, **tienda online**, **encargado** y **banco**. El encargado y la tienda online no se engloban dentro de la misma figura, ya que, mientras que el encargado realiza la labor física de preparar el paquete y mandarlo, la plataforma online de la tienda realiza toda la interacción con el cliente y con la plataforma de pago del banco. A continuación, se desglosan las interacciones:

- El cliente realiza el pedido de un artículo en la tienda online:
 1. El cliente selecciona un artículo, así que se traza una flecha continua hacia la tienda online (ya que se invocará un supuesto método que permita agregar un artículo).
 2. La plataforma agrega el artículo al carrito de la compra y, a continuación, manda un mensaje de confirmación al cliente (flecha discontinua).
 3. El cliente confirma el pedido llamando al método correspondiente de la plataforma online de la tienda (flecha continua).
 4. La plataforma online avisa al encargado de que hay que preparar un envío. Se debe dibujar una flecha continua, esta vez hacia el encargado, con el fin de que prepare el envío.
- El encargado de la tienda prepara el paquete para el envío y espera la confirmación de pago:
 1. El encargado prepara el envío. Una vez preparado, realiza una confirmación sobre la plataforma, indicando que el pedido está listo para enviar (flecha discontinua).
- El cliente recibe la notificación de que el pedido está listo y paga el importe:
 1. La plataforma avisa al cliente de que el pedido está listo a través de un mensaje de confirmación (flecha discontinua).
 2. El cliente debe realizar el pago para que el pedido le sea enviado. Se trazará una flecha continua hacia la tienda online.
 3. La tienda contacta la plataforma de pago del banco. Nueva flecha continua, ahora hacia el banco.

- El banco confirma el pago a la tienda online y al cliente.
 1. El banco manda un mensaje de confirmación (flecha discontinua) a la tienda, comunicando que el pago ha sido aceptado.
 2. La plataforma online de la tienda avisa al encargado para que envíe el pedido (flecha continua).
 3. Mensaje de confirmación hacia el cliente (flecha discontinua) indicando que el pago ha sido aceptado.
- El encargado envía el paquete.
 1. Mensaje de confirmación hacia el cliente (flecha discontinua) indicando que el paquete ha sido enviado.

La representación gráfica de todo el proceso queda expuesta en el siguiente diagrama de secuencia:

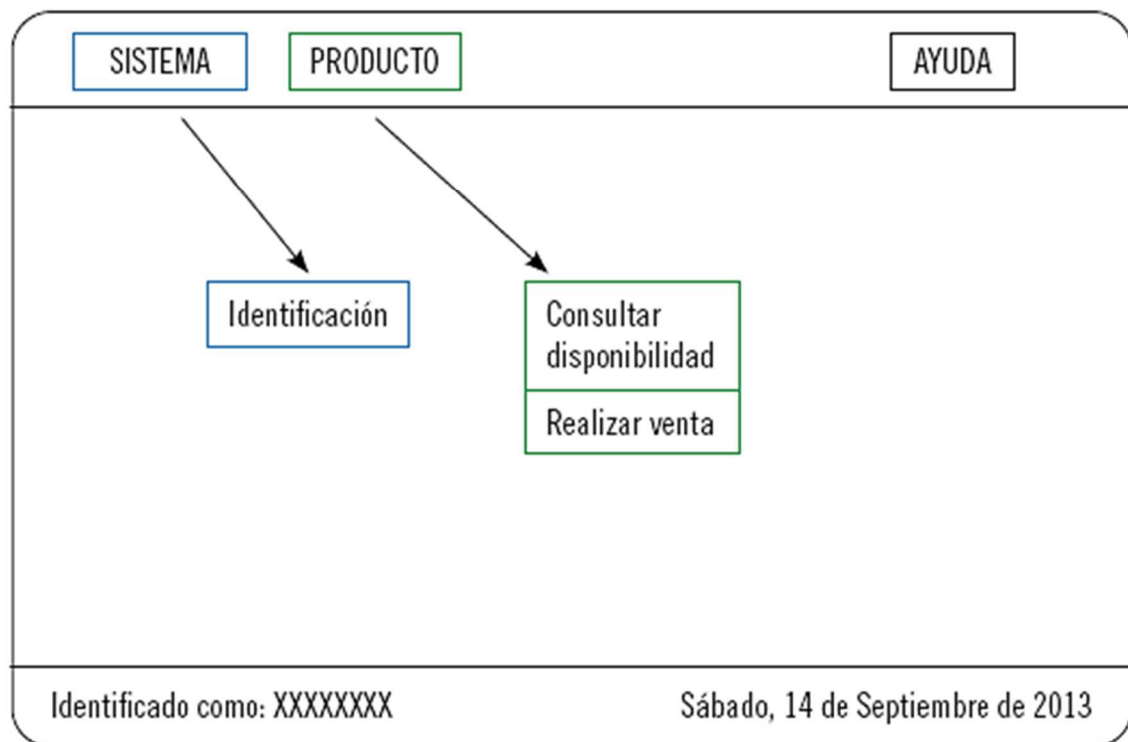


Modelos para el diseño de sistemas: Modelos de Interfaz

Ejemplo Definición de interfaz para la especificación de caso de uso usada como ejemplo anteriormente. El caso de uso incluía la siguiente secuencia de acciones:

1. El cliente llega al mostrador.
2. El cliente solicita un artículo al vendedor.
3. El vendedor se identifica en el sistema.
4. El vendedor comprueba disponibilidad.
5. El vendedor va al almacén a por el artículo.
6. El vendedor informa del precio al cliente.
7. El cliente paga (en efectivo o con tarjeta).
8. El vendedor marca la venta y genera el ticket.
9. El cliente toma el ticket y su artículo.

Ejemplo de diseño de interfaz

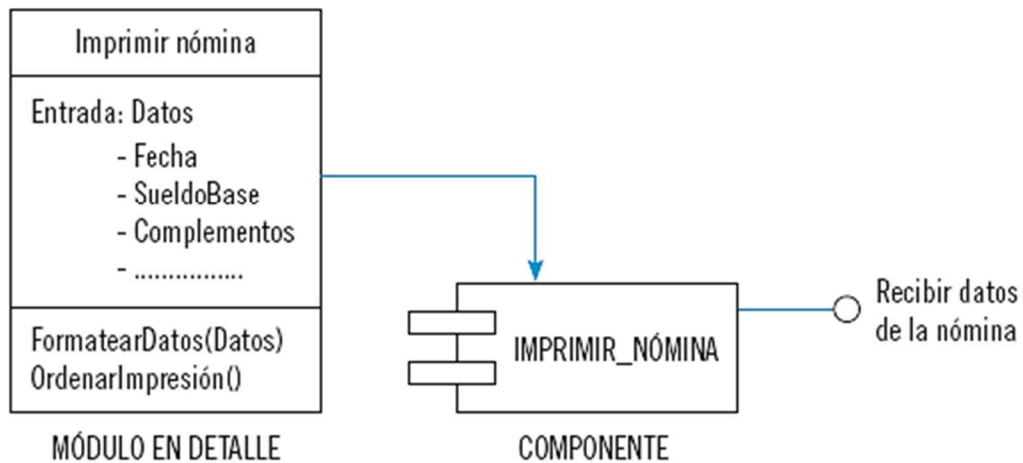


Modelos para el diseño de sistemas: Modelos de componentes

Visión Tradicional

Ejemplo Componente para la funcionalidad ImprimirNómina. Los datos requeridos, es decir, la información relativa a la nómina, se reciben a través de una interfaz.

Componente tradicional

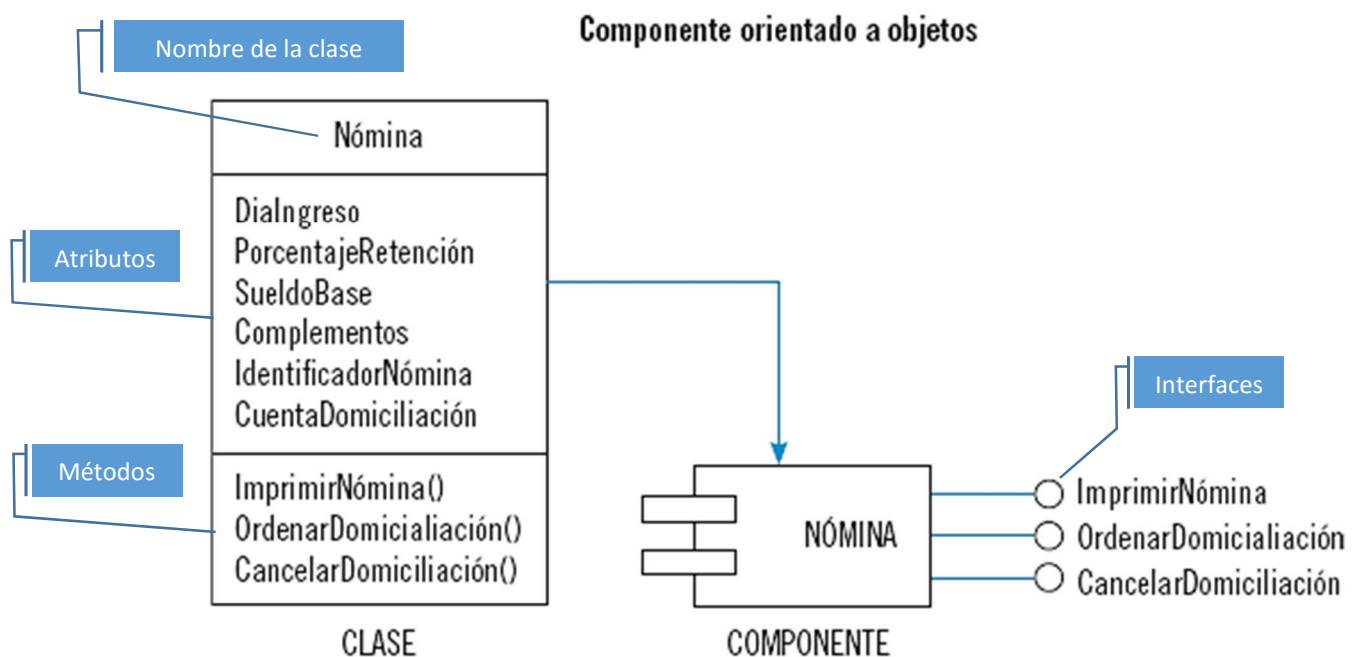


Modelos para el diseño de sistemas: Modelos de componentes

Visión Orientada a Objetos

Ejemplo Crear el componente para una clase Nómina con la siguiente información:

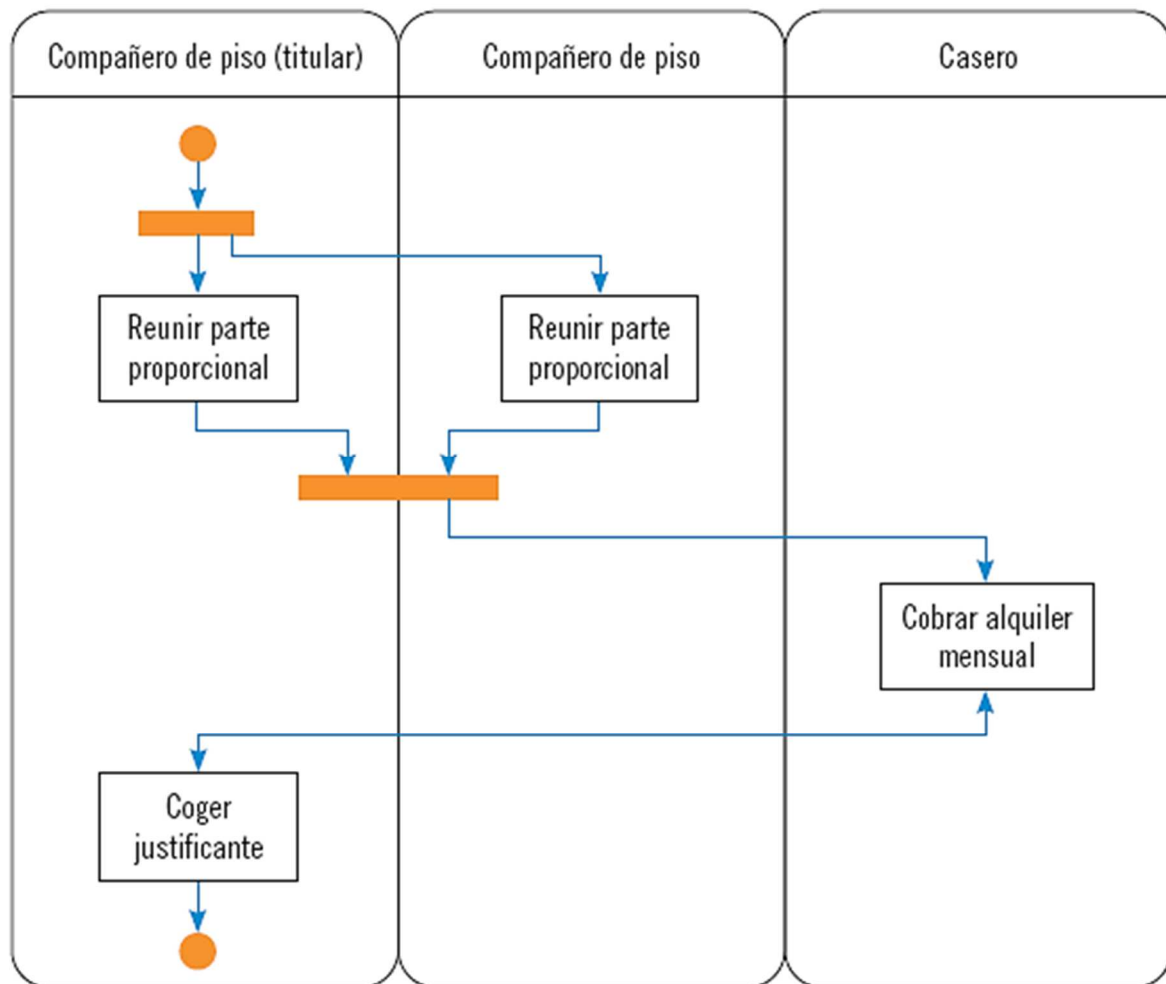
- Atributos:
 - DíaIngreso.
 - PorcentajeRetención
 - SueldoBase
 - Complementos.
 - IdentificadorNómina.
 - CuentaDomiciliación.
- Interfaces:
 - OrdenarDomiciliación
 - CancelarDomiciliación
 - ImprimirNómina.



Diagramas de Diseño: Diagrama de Actividad

Ejemplo: Modelar una actividad referida a pagar la mensualidad del alquiler del inmueble. Hay dos compañeros de piso, siendo uno el titular que recibirá el justificante de ingreso.

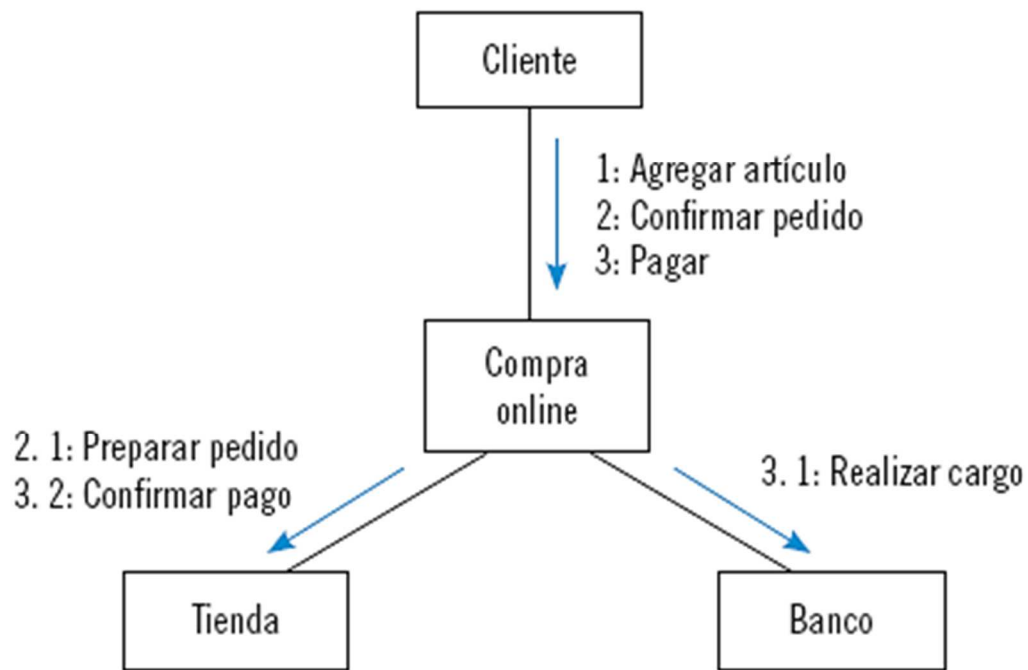
Diagrama de actividad



Diagramas de Diseño: Diagrama de Comunicación

Ejemplo: Realizar el diagrama de comunicación para la actividad propuesta en la aplicación anterior **“Compra OnLine”**.

Diagrama de comunicación



Diagramas de Diseño: Diagrama de Estado

Ejemplo: Diagrama de estados para una alarma. Se considerarán dos estados: “activa” y “no activa”, siendo el evento disparador la propia alarma.

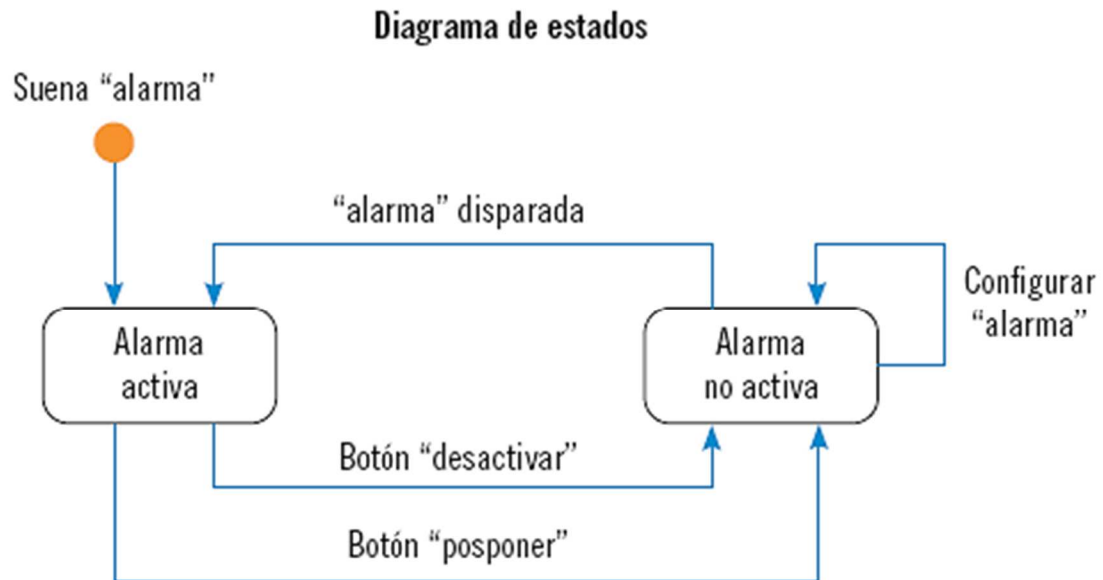
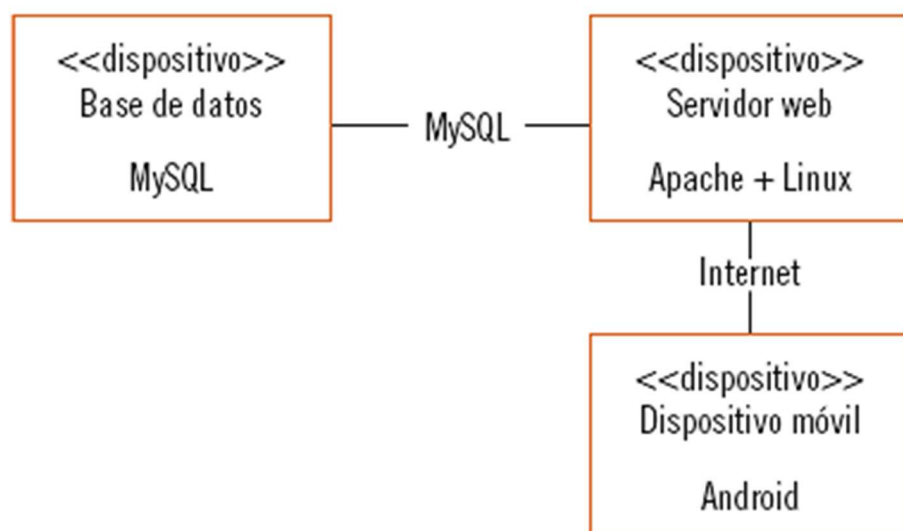


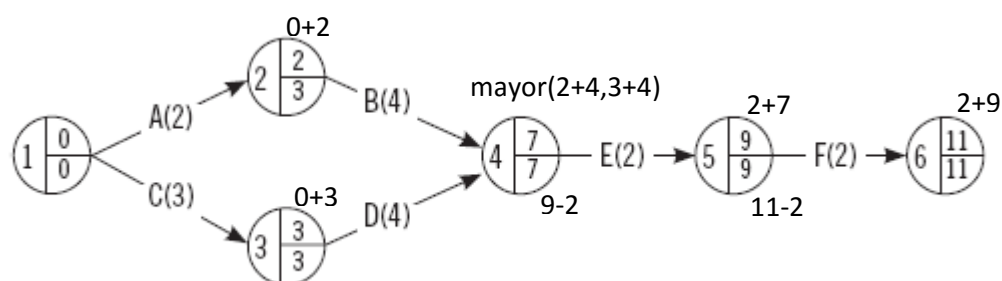
Diagrama de implementación



Ejemplo: Diagrama de Pert y camino mínimo para la siguientes tareas.

Diagrama de Pert

TAREA	PRECEDENTE	DURACIÓN
A	-	2
B	A	4
C	-	3
D	C	4
E	D, B	2
F	E	1



Camino crítico

Margen = Fin máximo – Fin

TAREA	PRECEDENTE	DURACIÓN	INICIO MIN	FIN MIN	INICIO MAX	FIN MAX	MARGEN
A	-	2	0	2	1	3	1
B	A	4	2	6	3	7	1
C	-	3	0	3	0	3	0
D	C	4	3	7	3	7	0
E	D, B	2	7	9	7	9	0
F	E	1	9	10	10	11	1

Fin mínimo = Inicio mínimo + duración

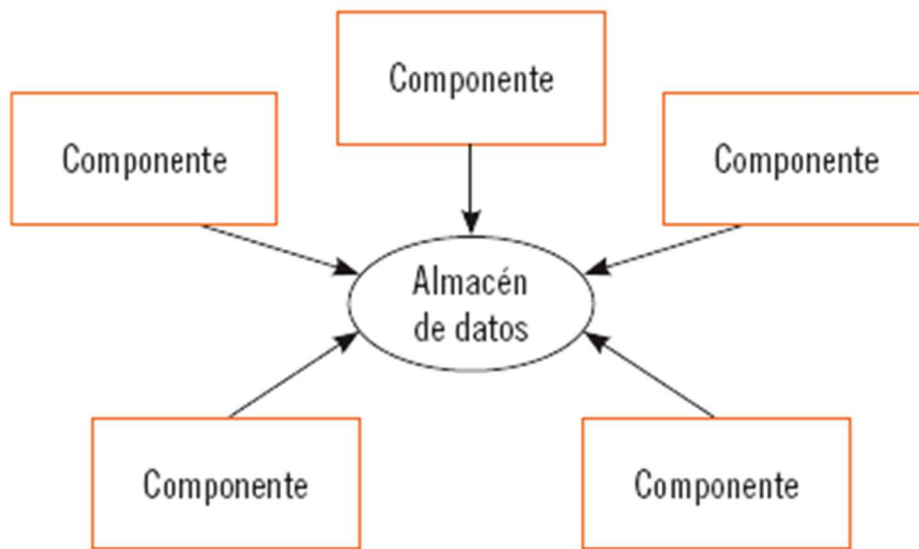
Inicio máximo = Fin máximo – duración

Introduzca dos nuevas actividades.

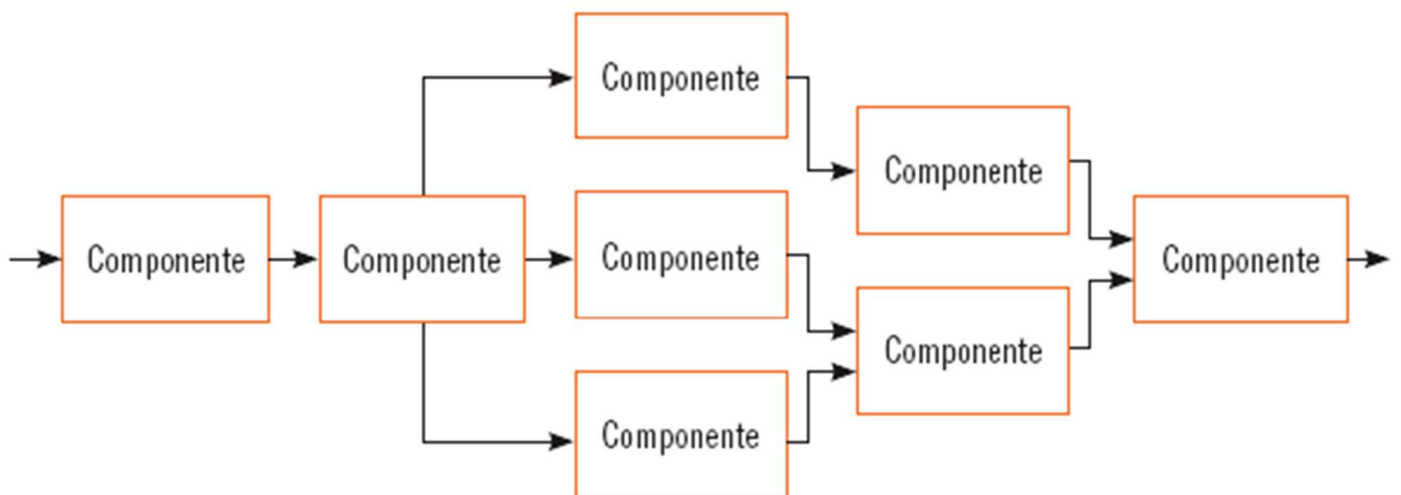
- A1 se integra entre A y B, teniendo una duración de 5.
- La actividad E1 hace lo propio entre E y F, con una duración de 3.
- Calcule el camino crítico.

Modelos para el diseño de sistemas: Modelos de arquitectura

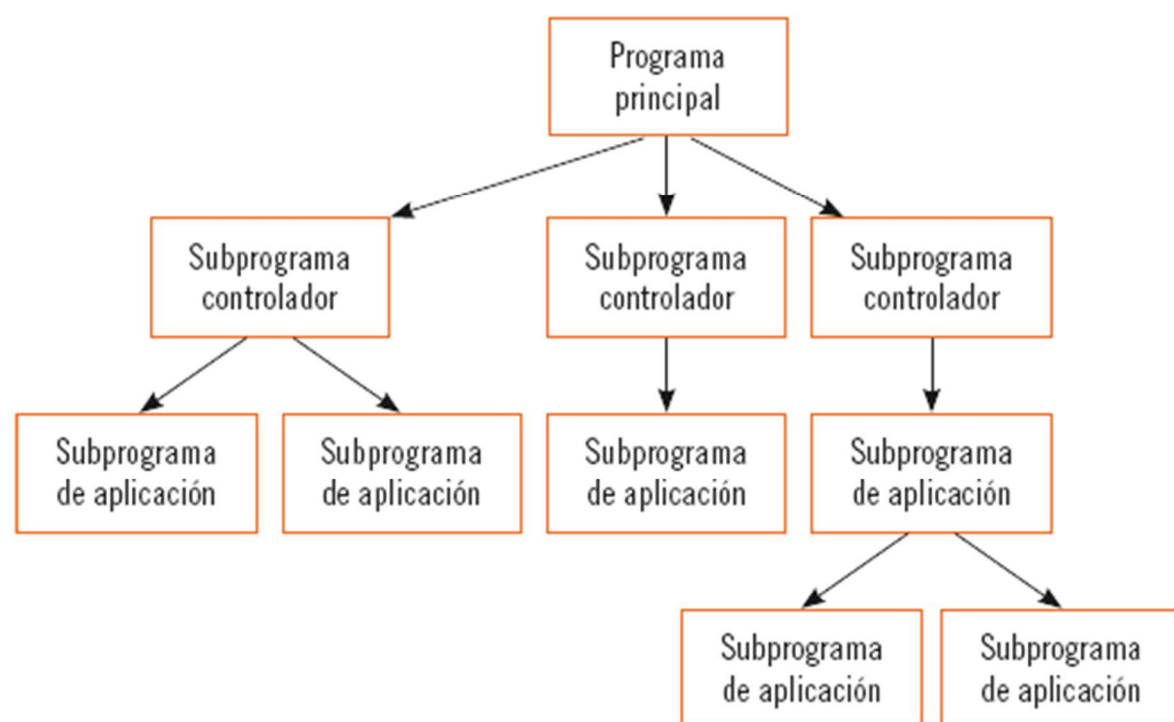
Arquitectura centrada en datos



Arquitectura de flujo de datos



Arquitectura de programa principal/subprograma



Arquitectura orientada a objetos

