

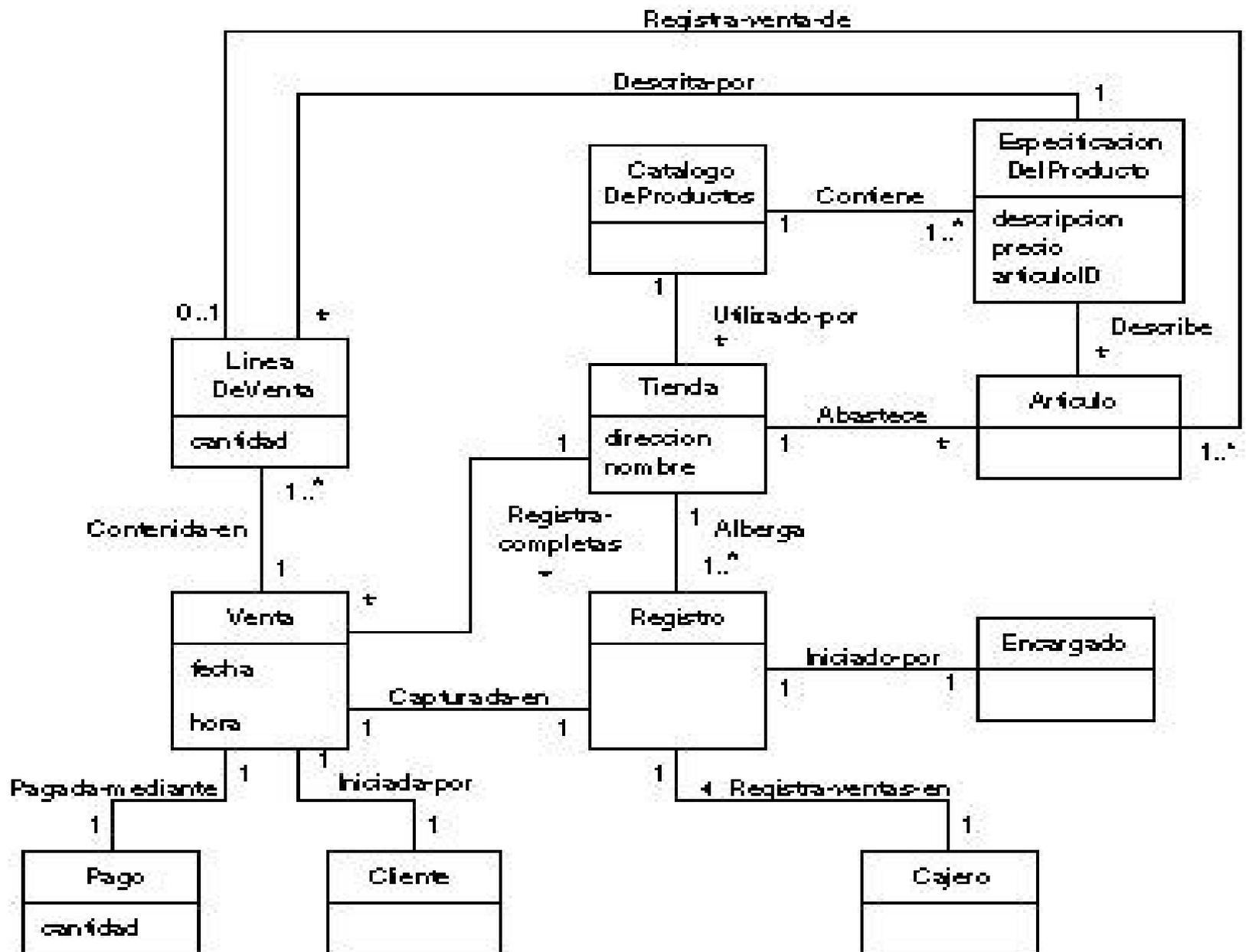
Diseño Dirigido por Responsabilidades con los patrones GRASP

Experto (en información)

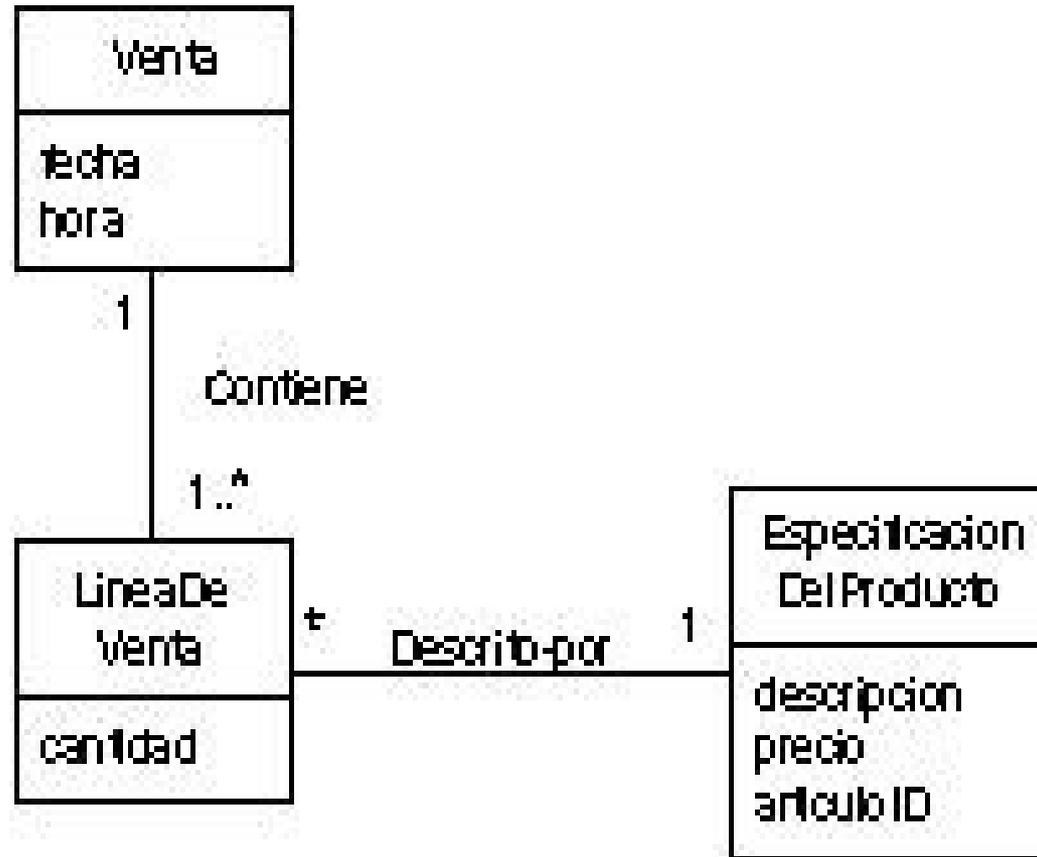
- ¿Cómo asignar responsabilidades?
- Asignar una responsabilidad al objeto que tiene la información necesaria para realizarla:
 - “El objeto que tiene la información realiza el trabajo”.
 - Por ejemplo, ¿qué objeto de software calcula el impuesto sobre las ventas?
 - ¿Qué tipo de información se necesita para llevar a cabo esta tarea?
 - ¿Qué objeto u objetos tienen la mayor parte de esta información?

Experto (en información)

- En la aplicación del terminal de punto de venta (TPV) hay una responsabilidad que debemos asignar:
 - Calcular el importe total de una venta
 - ¿A qué objeto vamos a asignar esta responsabilidad?
- El patrón Experto nos sugiere asignar la responsabilidad al objeto que posea la información necesaria para desempeñarla.
 - ¿Qué objeto tiene la información necesaria para --Calcular el importe total de una venta--?

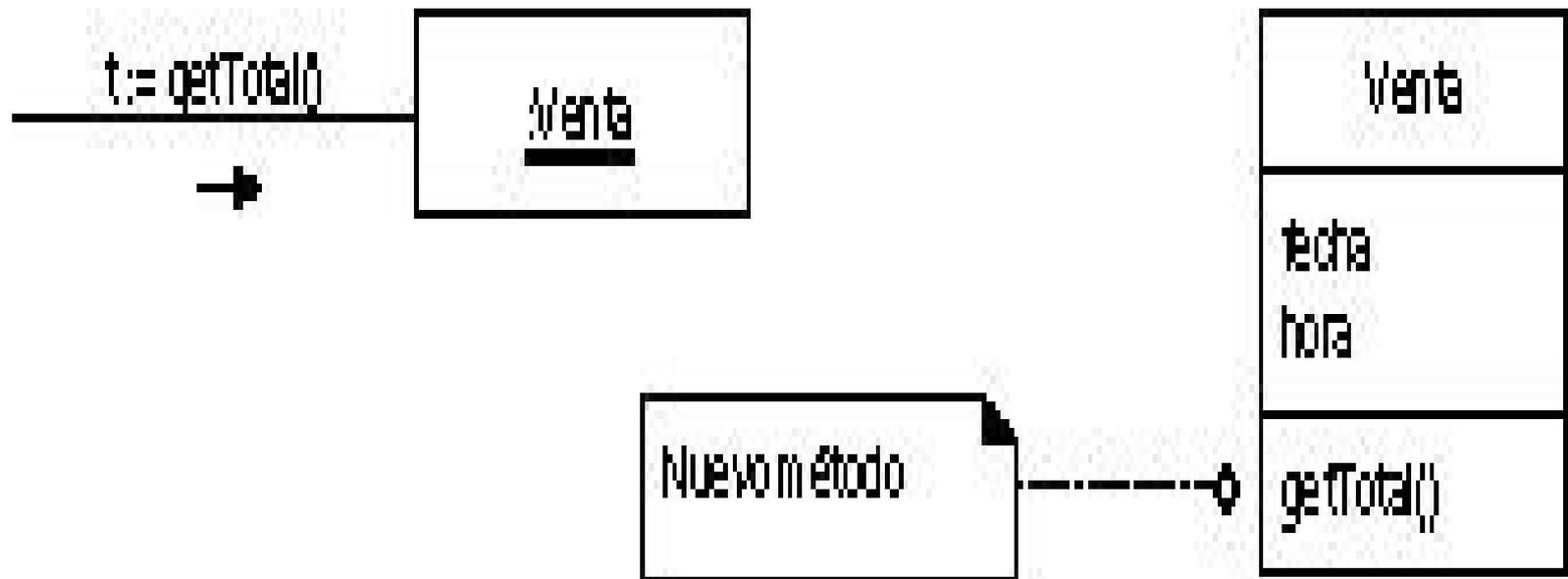


Centrémonos en la parte que más nos interesa



• ¿Qué información se necesita para calcular el importe total?

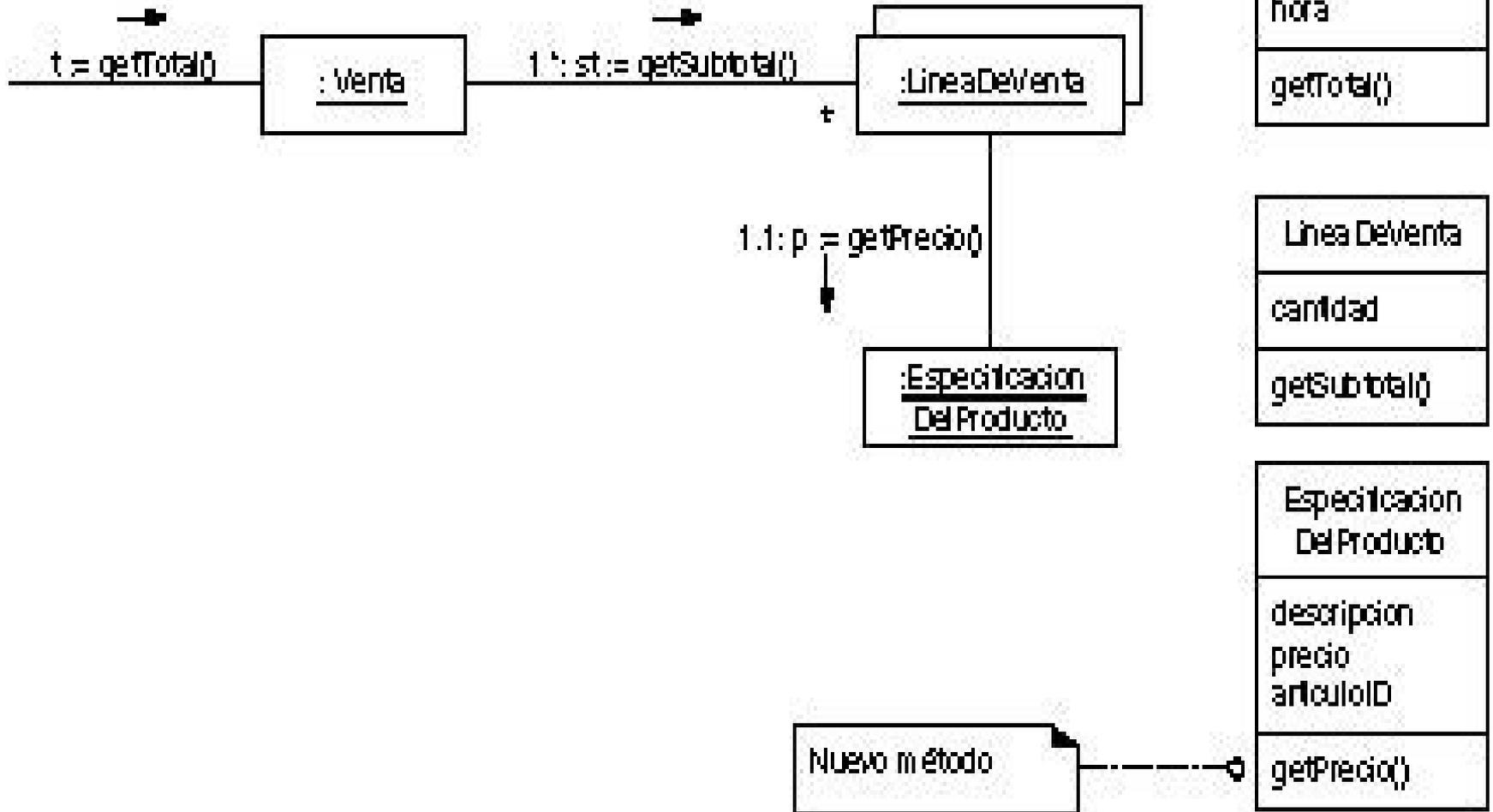
- Todas las instancias de LineaDeVenta, y La suma de sus subtotales
- Ya que Venta contiene las instancias de LineaDeVenta, podría ser el *experto* adecuado



Experto (en información)

- ¿Qué información se necesita para calcular la suma de sus subtotales?
 - LineaDeVenta.cantidad, y
 - EspecificacionDelProducto.precio
- Ya que LineaDeVenta conoce la cantidad y la EspecificacionDelProducto correspondiente, podría ser el *experto* adecuado
- Mientras que EspecificacionDelProducto se responsabilizaría de conocer el precio del producto

Experto (en información)



Experto (en información)

<i>Clase</i>	<i>Responsabilidad</i>
Venta	Calcular el importe total de una venta
LineaDeVenta	Calcular el subtotal de la línea de venta
EspecificaciónDelProducto	Conocer el precio del artículo

Experto (en información)

<i>Patrón</i>	
Nombre	Experto
Problema	¿A qué objeto asignar una responsabilidad?
Solución	Asignarla a la clase que posea la información necesaria para desempeñarla
Ejemplo	Calcular el importe total de una venta en la aplicación TPDV

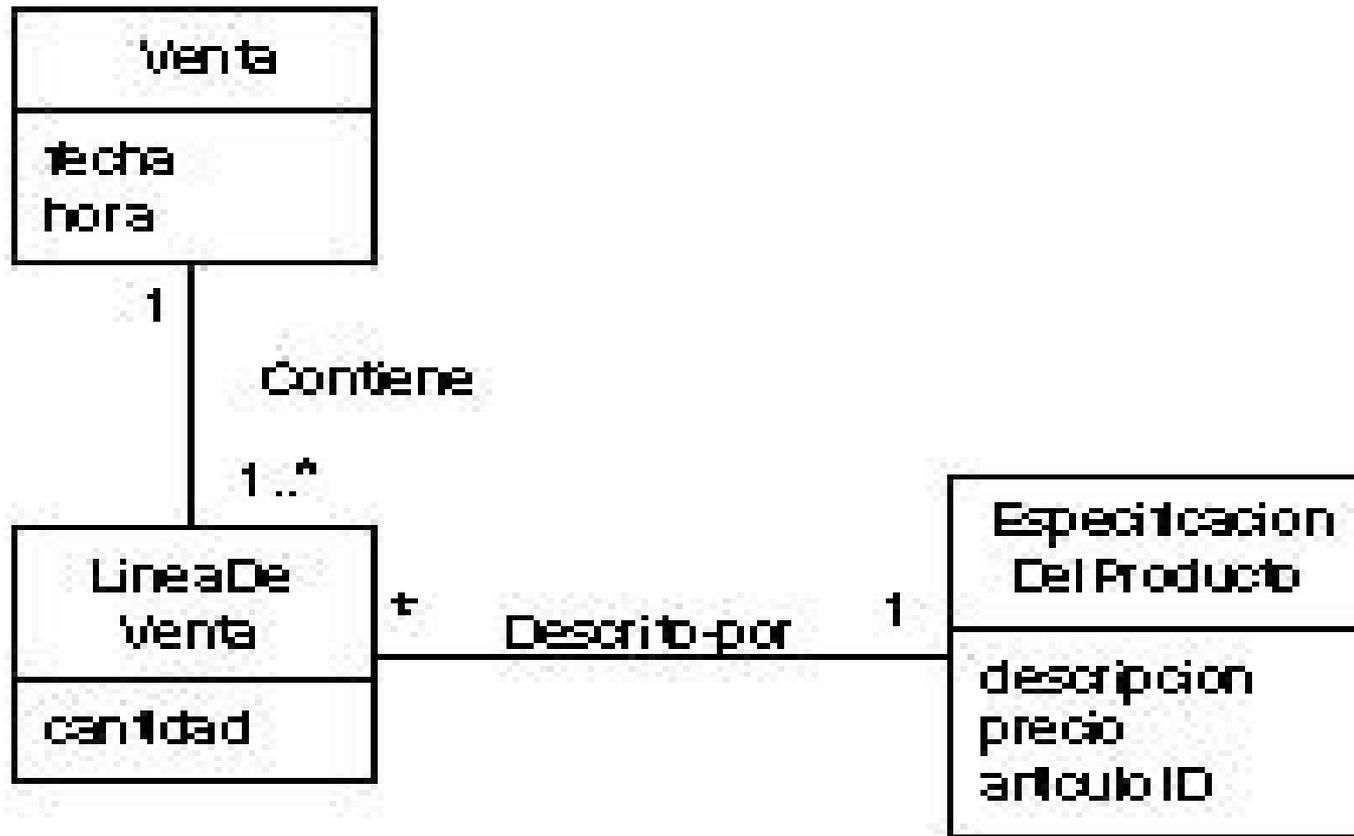
Creador

- ¿Qué objeto debería crear X?
 - Elegir un objeto C, de manera que:
 - C contenga X,
 - C agregue X,
 - C registre X,
 - C utilice más estrechamente X,
 - C tenga los datos de inicialización para X.
- Cuantas más opciones se apliquen, mejor.

Creador

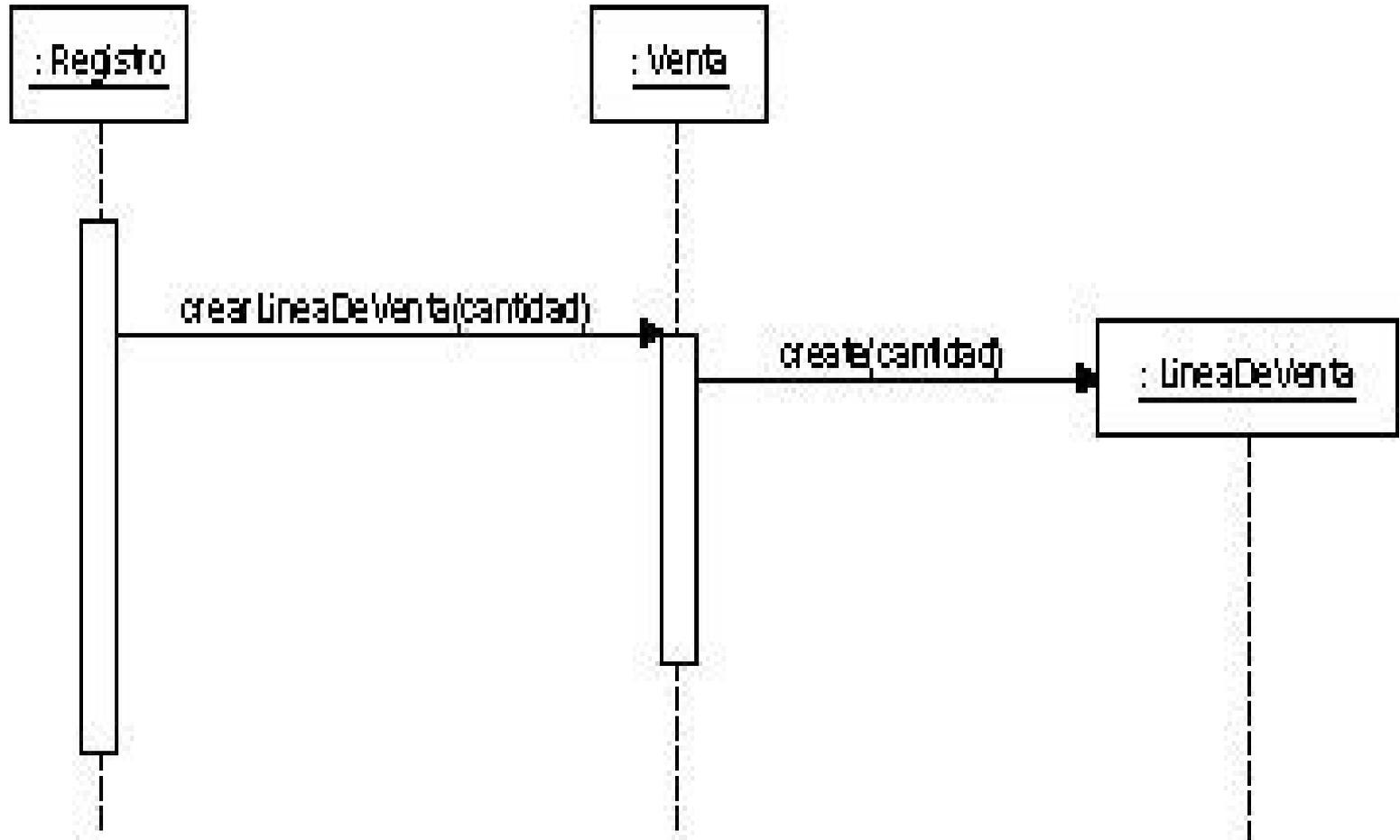
- En la aplicación TPV debemos asignar la responsabilidad de:
 - Crear una lineaDeVenta
 - ¿A qué objeto vamos a asignar esta responsabilidad?
- El patrón Creador nos sugiere asignar la responsabilidad a un objeto que agregue, contenga, ... instancias de lineaDeVenta.

Creador



- Los objetos Venta parecen buenos candidatos

- Debemos, pues, definir un método crearLineaDeVenta() en la clase Venta



Creador

<i>Patrón</i>	
Nombre	Creador
Problema	¿A qué objeto asignar una responsabilidad de creación de instancias de una clase?
Solución	Elegir los objetos que agregen, contengan, registren, utilicen más o posea los datos de inicialización de las instancias,
Ejemplo	Creación de instancias de LineaDeVenta en la aplicación TPDV

Bajo acoplamiento

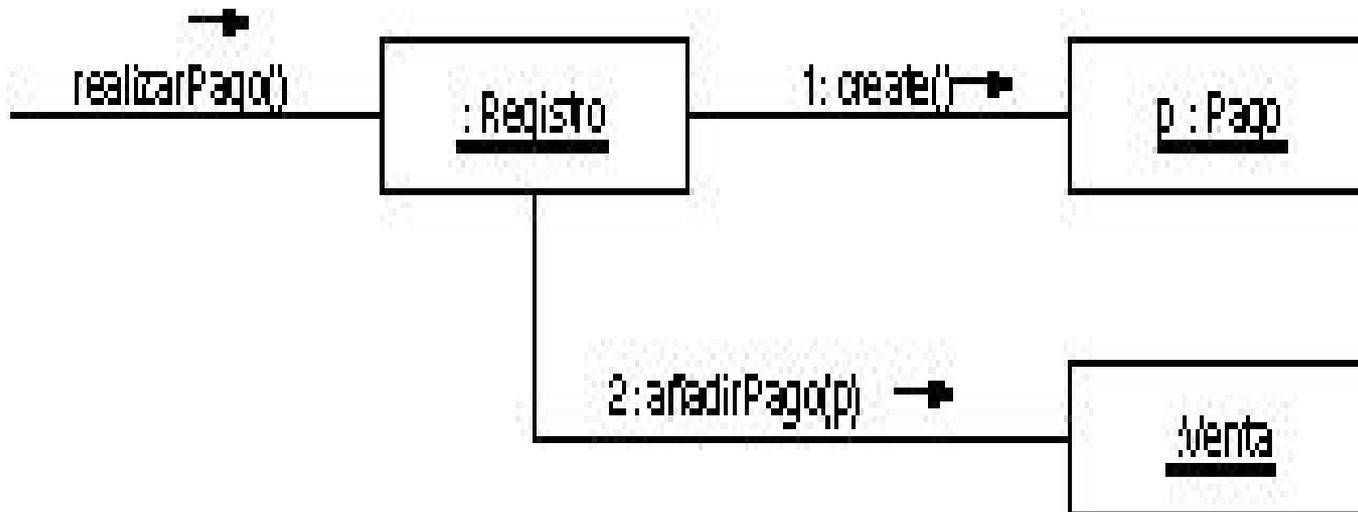
- Asignar responsabilidades de manera que el acoplamiento permanezca bajo.
- ¿Qué significa bajo acoplamiento?
- Las clases con acoplamiento alto:
 - Deben modificarse cuando cambian las clases de las que dependen
 - Son difíciles de entender por sí solas
 - Son más difíciles de reutilizar

Pago

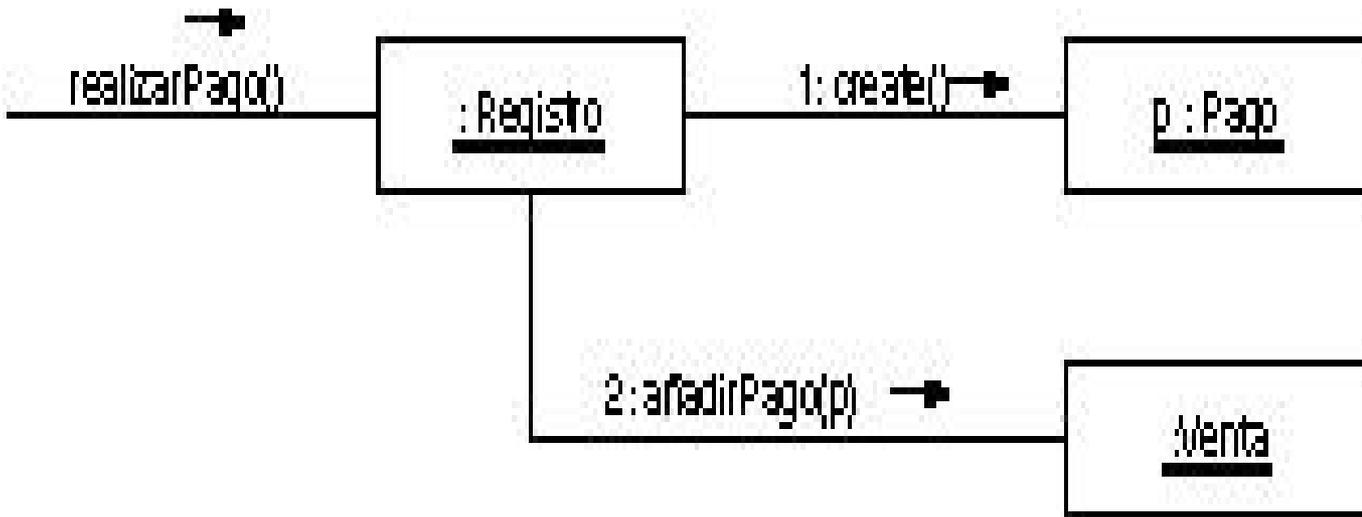
Registro

Venta

- Debemos crear una instancia de Pago y asociarla con la Venta correspondiente:
- ¿A qué clase asignar esta responsabilidad?
- El patrón Creador sugiere que sea Registro la clase responsable



(a)



- Pero no es ésta la única alternativa posible

(b)



Bajo acoplamiento

- Tanto en (a) como en (b) Venta y Pago están acoplados
- Pero en (a) también lo están Registro y Pago,
- Mientras que en (b) no lo están
- Así, desde el punto de vista del Acoplamiento bajo, es preferible la opción (b).

Bajo acoplamiento

- Ejemplos de posibles acoplamientos entre las clases A y B:
 - A tiene un dato miembro del tipo B
 - Un objeto de la clase A llama a una función miembro de B
 - Una función miembro de A tiene un parámetro del tipo B
 - Una función miembro de A devuelve un valor del tipo B
 - A es subclase directa o indirecta de B

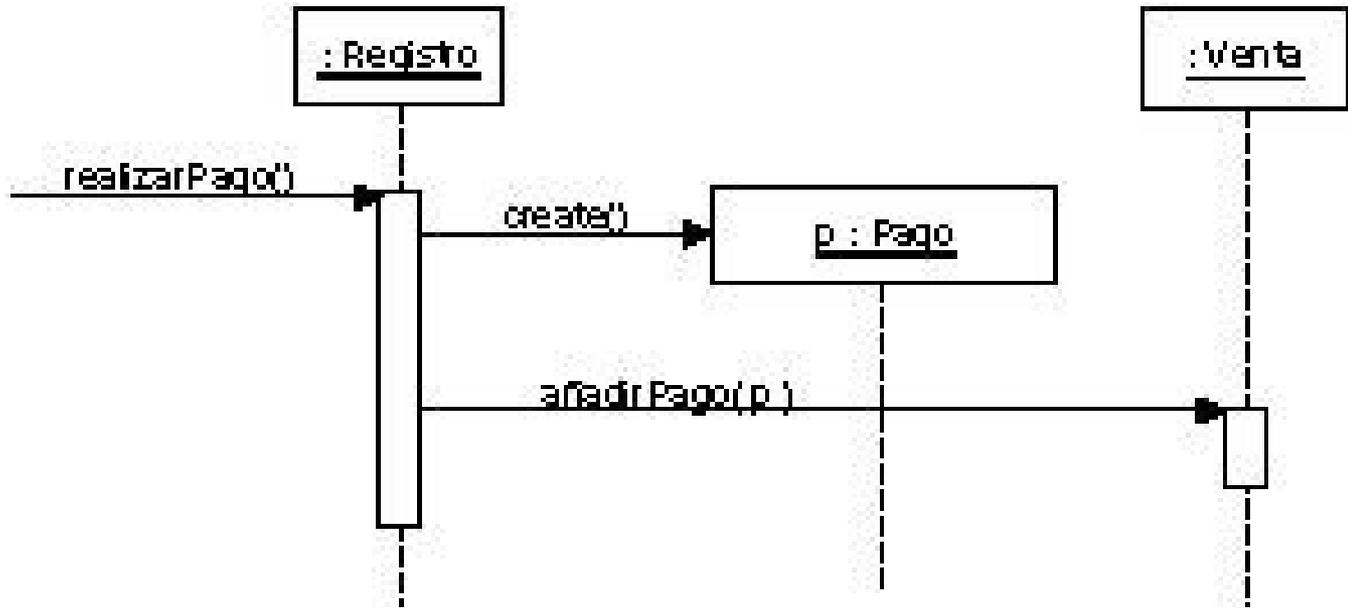
Alta cohesión

- Asignar una responsabilidad de manera que la cohesión permanezca alta.
- ¿Qué significa alta cohesión?
- Una clase con cohesión baja hace demasiadas cosas, inconexas, y es difícil
 - Entenderla
 - Reutilizarla
 - mantenerla

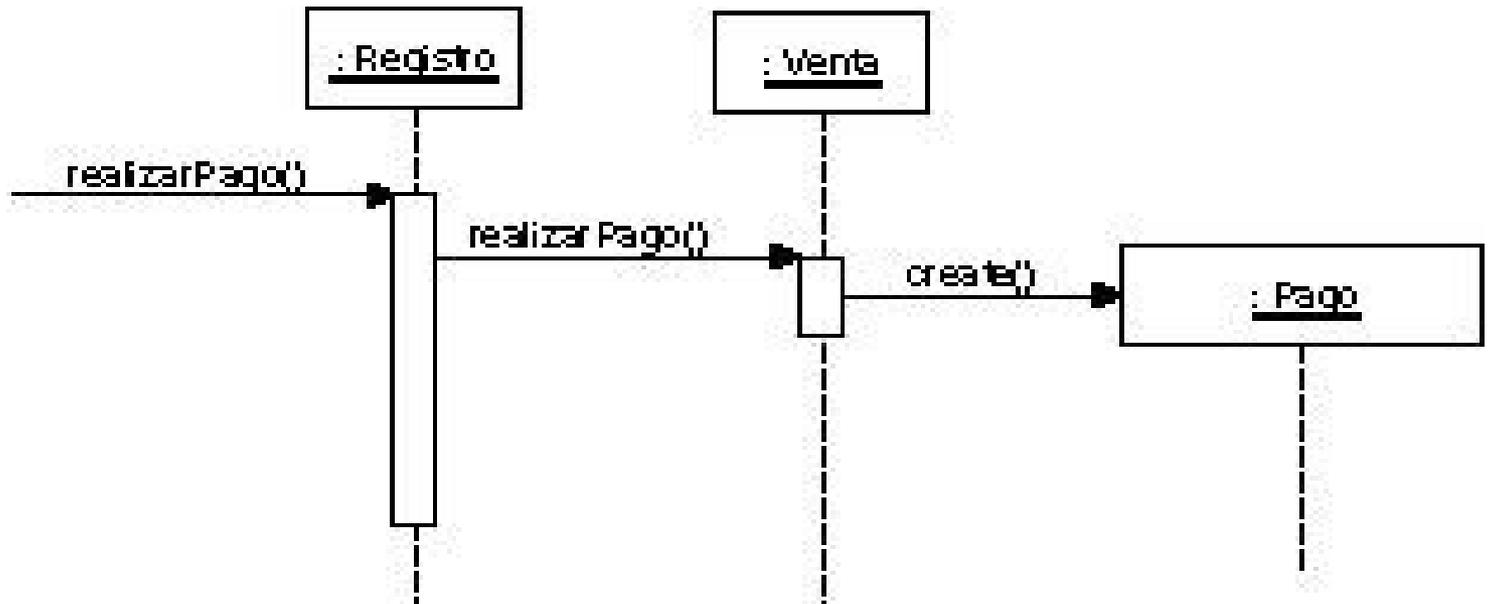
Alta cohesión

- El ejemplo anterior volverá a ser más útil
- Debemos crear una instancia de Pago y asociarla con la Venta correspondiente:
 - ¿A qué clase asignar esta responsabilidad?
 - El patrón Creador sugiere que sea Registro la clase responsable
 - Nuevamente no es esta la única alternativa posible

(a)



(b)

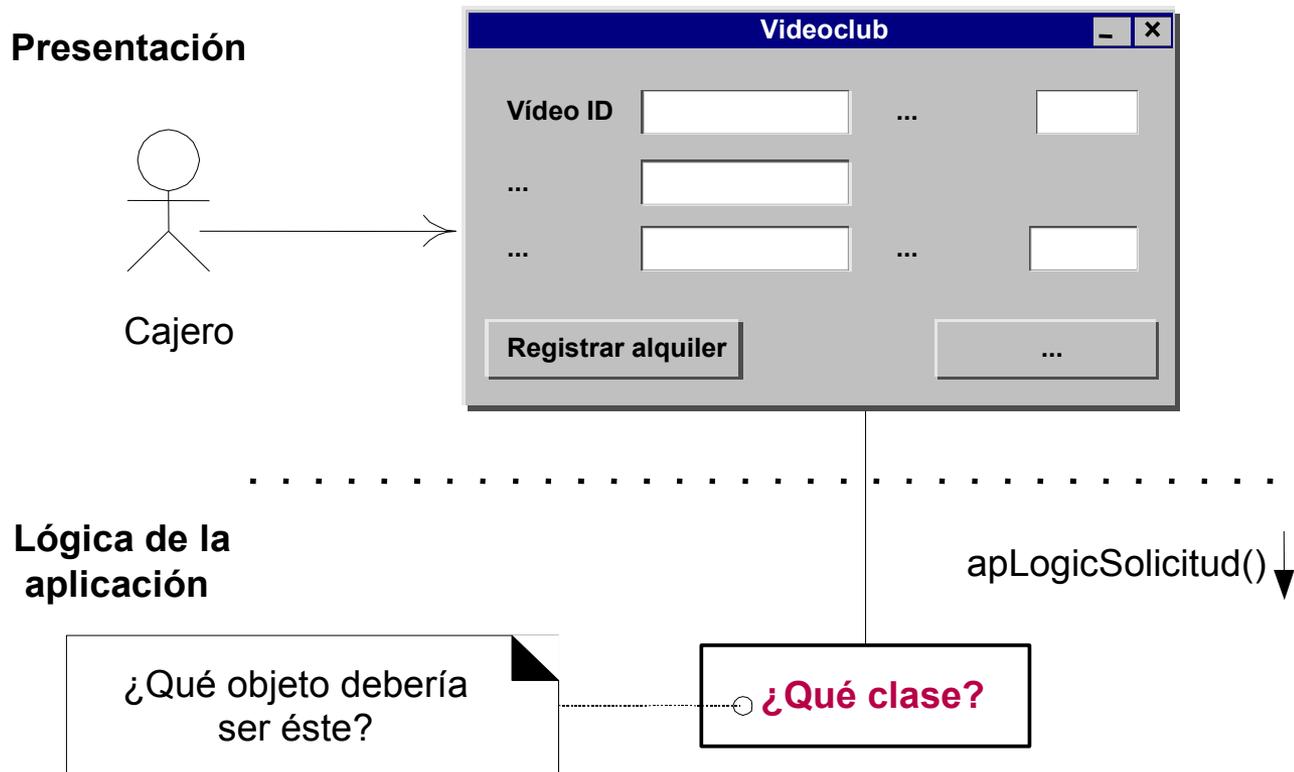


Alta cohesión

- En la opción (a) Registro se hace responsable de dar soporte a la implementación de realizarPago()
- En la opción (b), en cambio, esta responsabilidad se delega en Venta
 - Esta responsabilidad parece apropiada para la clase Venta
 - Si quisiéramos descargar de responsabilidades a Registro, esta podría ser una buena solución
 - Podríamos querer descargar de responsabilidades a registro en el caso en que ésta hubiera llegado a ser una clase con una cohesión baja

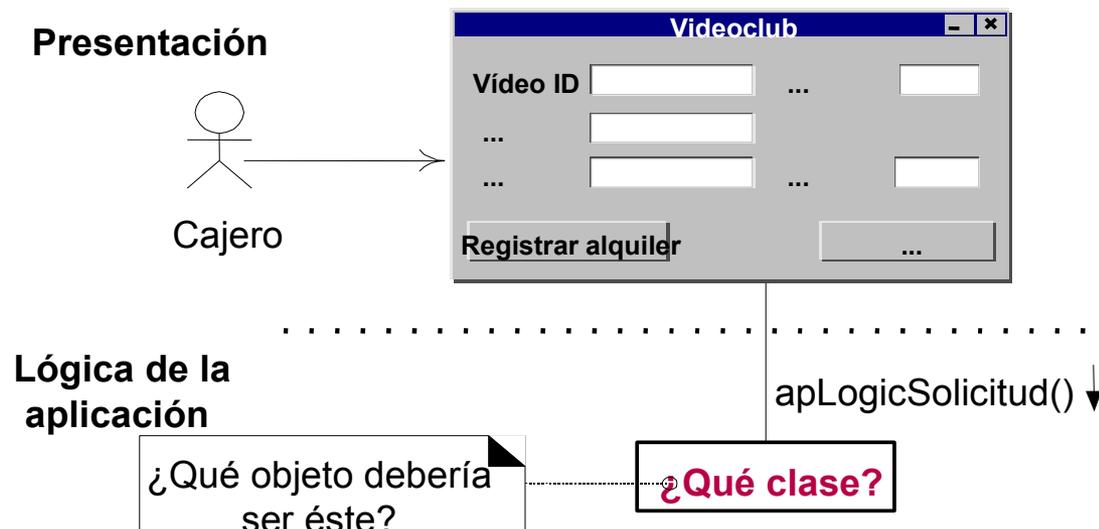
Controlador

- ¿Qué objeto (de la capa de coordinación de la aplicación) debería recibir solicitudes de trabajo procedentes de la capa de UI?



Controlador

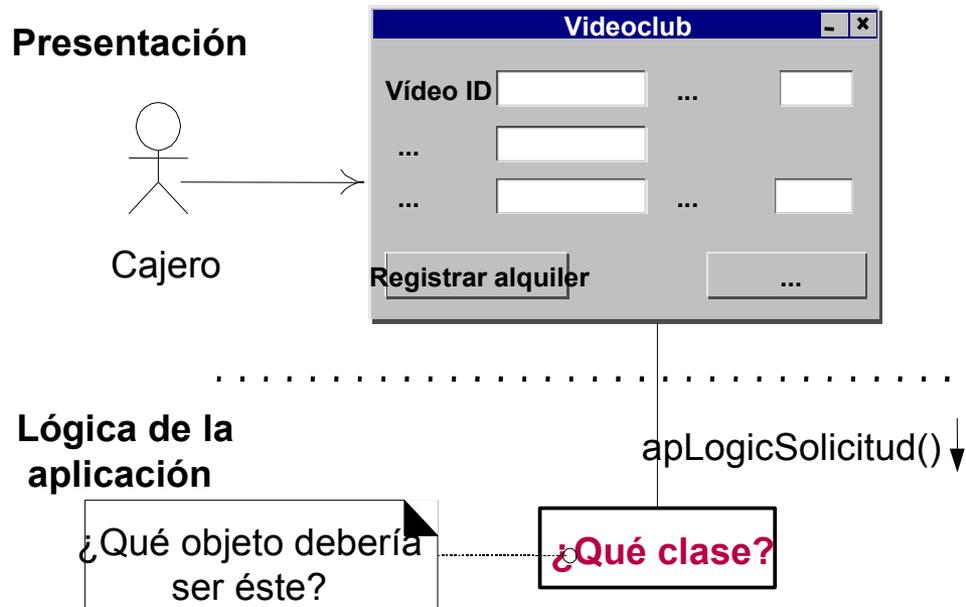
- ¿A qué objeto asignar la responsabilidad de manejar un evento del sistema?
- Un objeto representativo del sistema (**Fachada**)
- Un objeto representativo del escenario del caso de uso correspondiente (*Controlador de sesión, o de caso de uso*)



Controlador

Candidatos:

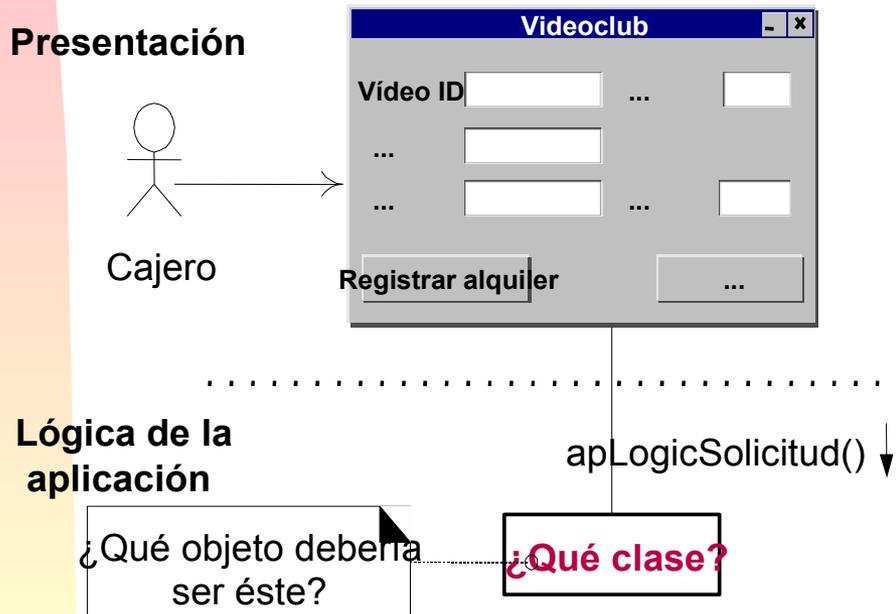
- Un objeto cuyo nombre refleje el servidor global, los negocios o la entidad a gran escala.
- Un tipo de objeto de “fachada”.
- Por ejemplo: Establecimiento, ServidorDeAlquiler.



Controlador

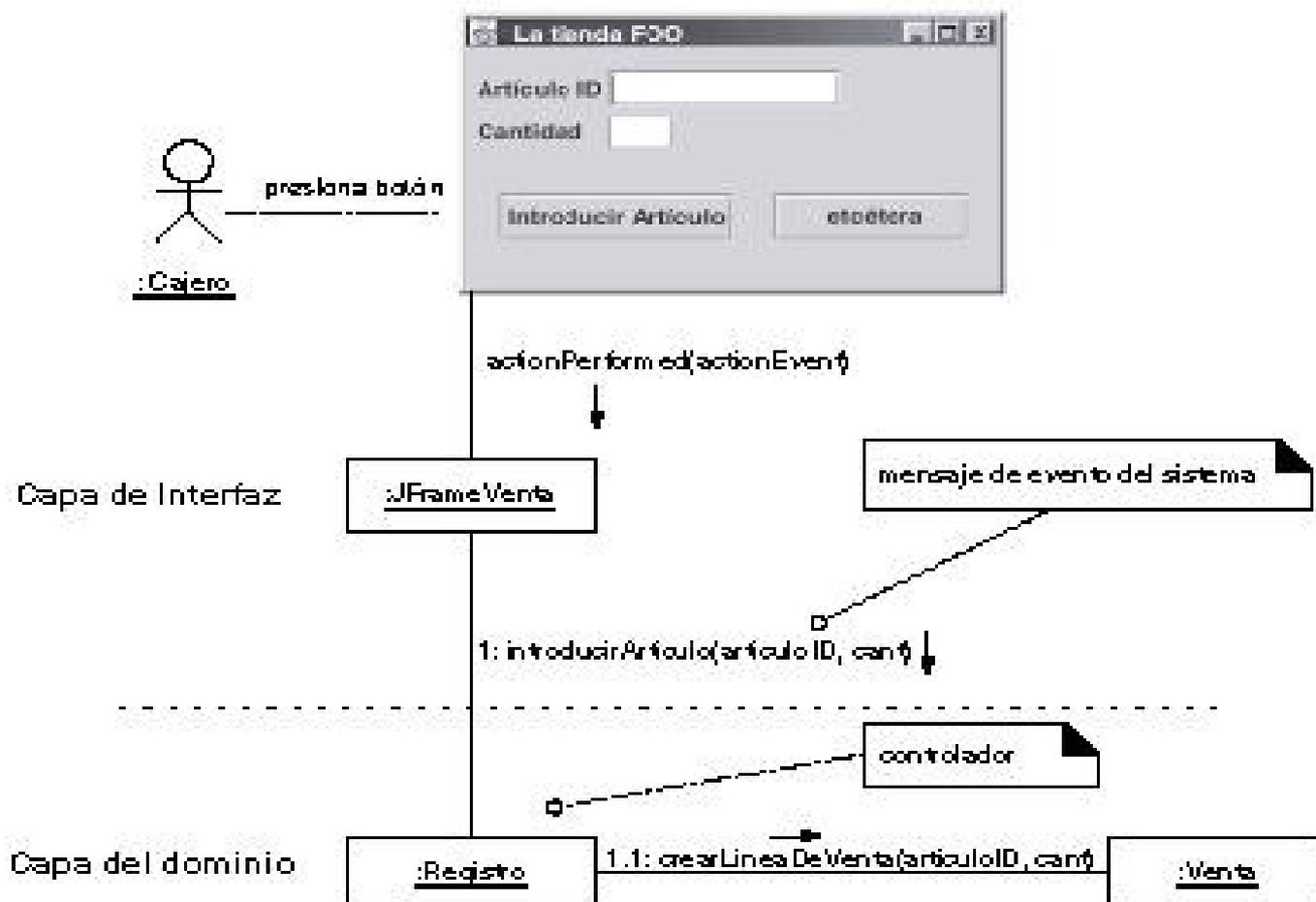
Candidatos:

- Un objeto cuyo nombre refleje el caso de uso,
 - Ejemplo: ManejadorDelCasoDeUsoAlquilerDeVideos
- formando parte de la capa de coordinación de la aplicación.
- Es conveniente usar un solo Controlador para cada escenario
- Una sesión es una instancia de interacción con el actor



Controlador

Figura 16.17. Acoplamiento deseable entre la capa de interfaz y la del dominio.

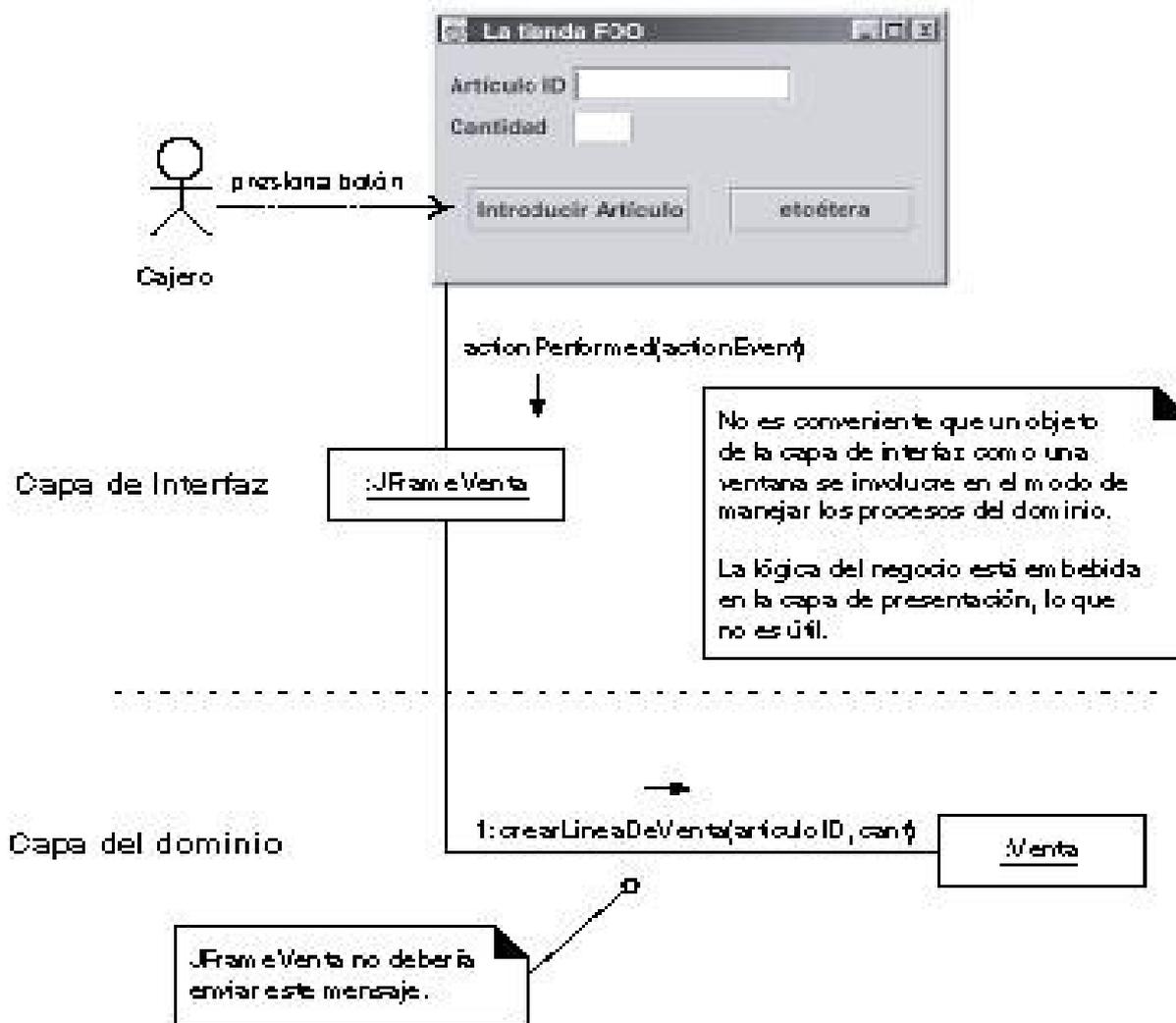


Controlador

- **No son buenos candidatos, en cambio, los objetos de interface de usuario:**
 - **Ventana**
 - **Applet**
 - **Widget**
 - **Vista**
 - **Documento**

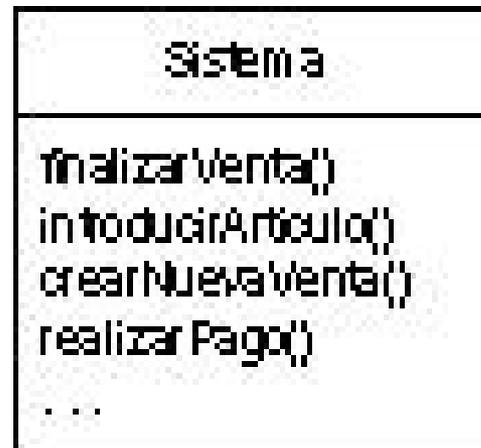
Controlador

Figura 16.18. Acoplamiento menos conveniente entre la capa de interfaz y la del dominio.



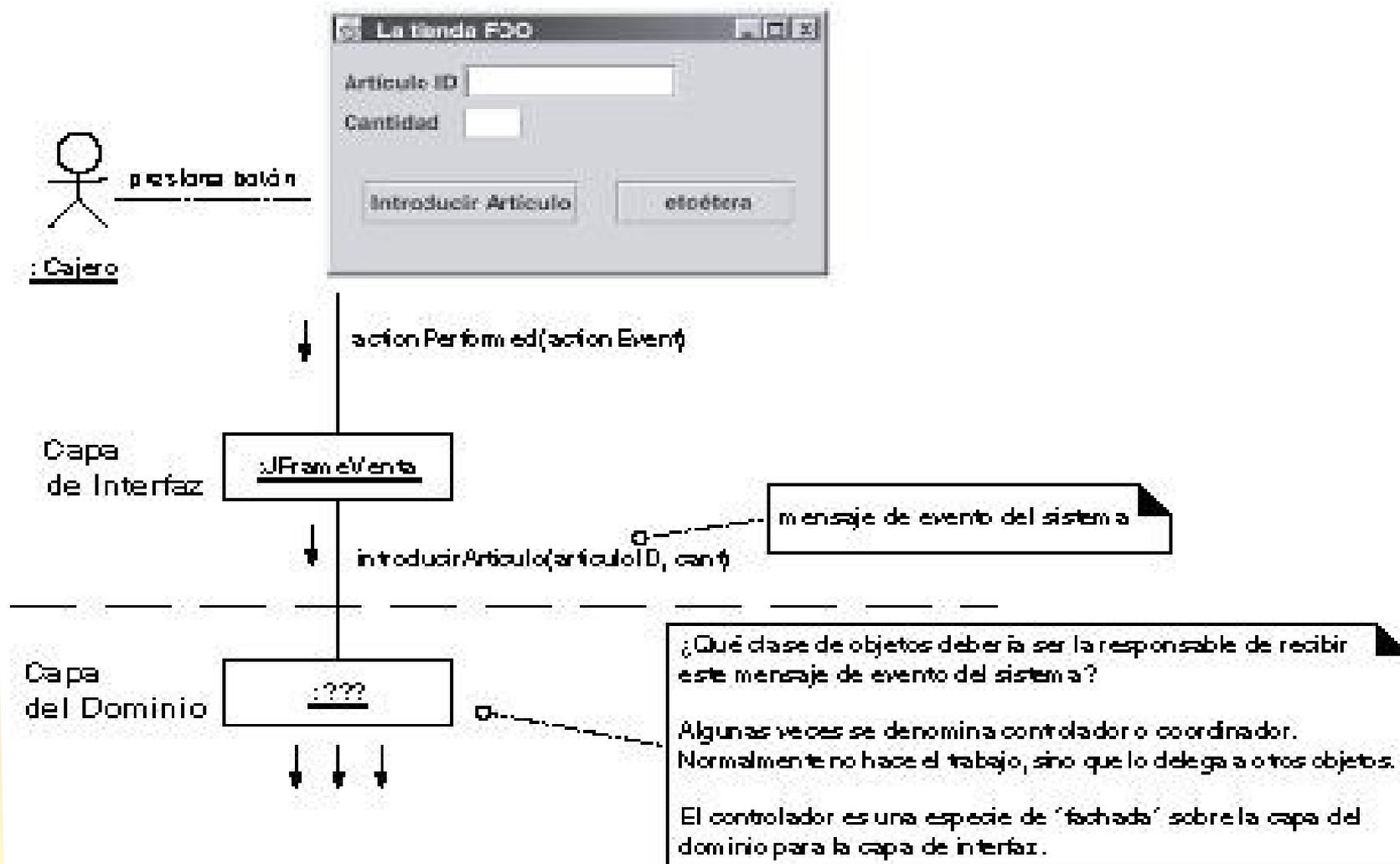
Controlador

- En la aplicación TPDV hay varios eventos del sistema, a los que corresponden las operaciones:
 - IntroducirArticulo
 - FinalizarVenta
 - ...



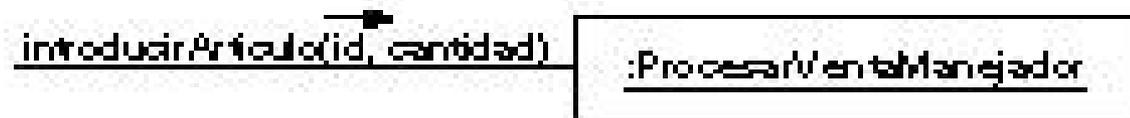
Controlador

Figura 16.14. ¿Controlador para introducirArticulo?



Controlador

- Siguiendo el patrón Controlador podrían ser candidatos:
 - Registro, TPDV, ..., ó
 - ProcesarVentaManejador, ProcesarVentaSesion, ...



Controlador

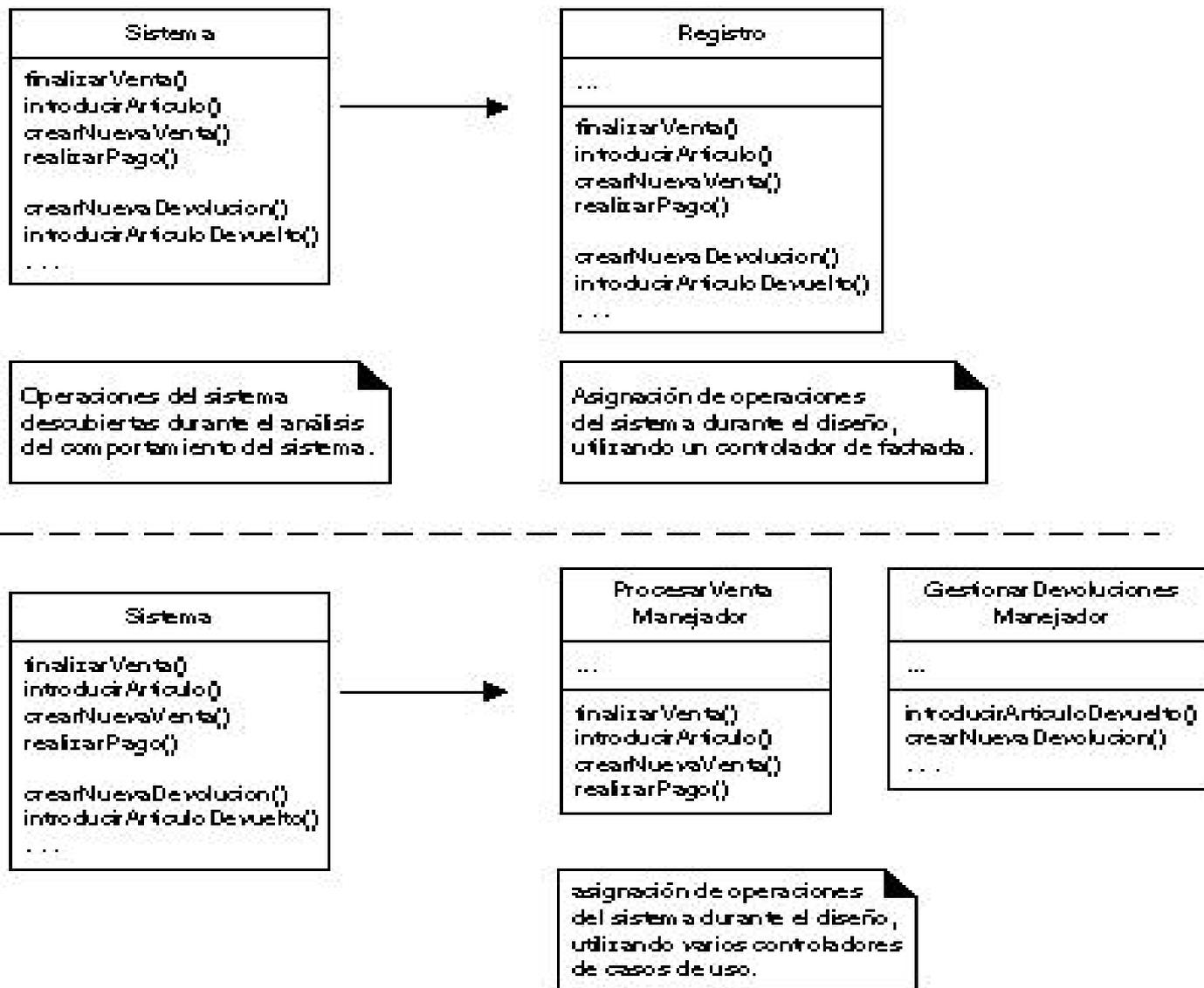


Figura 16.16. Asignación de las operaciones del sistema.

Controlador

- Utilizar el mismo controlador para todos los eventos de un caso de uso,
- Así podremos garantizar que sólo se permite la respuesta al evento que corresponde en cada estado
- Debe procurarse que los Controladores sólo tengan responsabilidades de coordinación,
- No deben sobrecargarse de responsabilidad los Controladores

Controlador

<i>Patrón</i>	
Nombre	Controlador
Problema	¿A qué objetos asignar la responsabilidad de gestionar un evento de entrada del sistema?
Solución	Elegir un controlador de fachada, o de caso de uso, o de sesión.
Ejemplo	Gestión de los eventos del sistema en la aplicación TPDV

Fabricación pura

- **¿Dónde asignar una responsabilidad cuando las opciones normales basadas en el Experto conllevan problemas en el acoplamiento y la cohesión o son, por otra parte, indeseables?**
- **Inventar una clase “artificial”, cuyo nombre no tiene que estar necesariamente inspirado por el vocabulario del dominio.**
- **Ejemplo: persistencia de la base de datos en:**
 - **¿Vídeo?**
 - **¿BaseDeDatosFachada? (una fabricación pura).**

Polimorfismo

- **¿Cómo diseñar para casos similares que varían?**
- **Asignar una operación polimórfica a la familia de tipos para los que varían los casos:**
 - **No utilice la lógica de casos.**
- **Ejemplo: dibujar()**
 - **Cuadrado, Círculo, Triángulo.**

Indirección

- **¿Un mecanismo común para reducir el acoplamiento?**
- **Asignar una responsabilidad a un objeto intermediario para desacoplar la colaboración de otros dos objetos.**

No Hable con Extraños

- **¿Cómo diseñar para restringir el acoplamiento al conocimiento de las conexiones estructurales de objetos?**
- **No recorra una red de conexiones de objetos a la hora de invocar una operación.**
- **En su lugar, promueva la operación a un “conocido” del cliente.**