

# Análisis Orientado a Objetos en UML

## TEMA 2

## **Tema 2. Análisis Orientado a Objetos en UML**

**2.1. Actividades del análisis de requisitos del software**

**2.2. Principios fundamentales del análisis de requisitos del software**

**2.3. Especificación de requisitos del software**

**2.4. Análisis orientado a objetos en UML**

**2.5 Modelo de casos de uso en UML**

**2.6. Modelo conceptual de datos en UML**

**2.7. Modelo de comportamiento del sistema en UML**

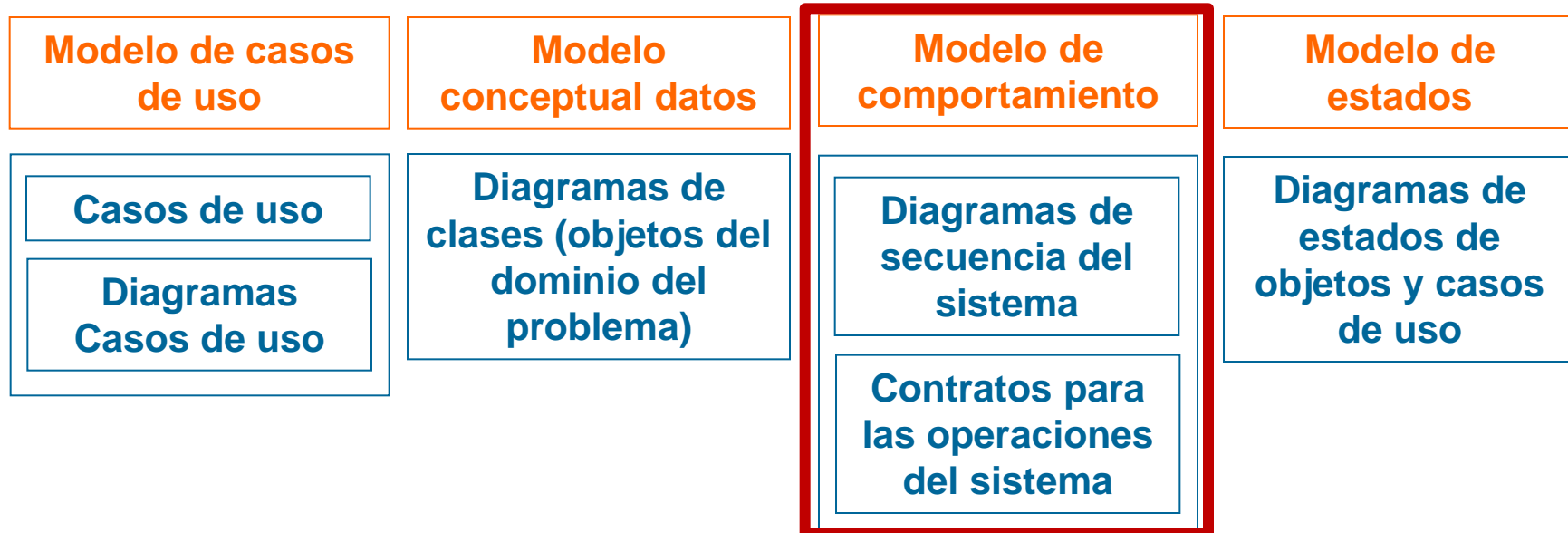
**Contenido de  
este fichero**

**Bibliografía**

## El alumno debe ser capaz de:

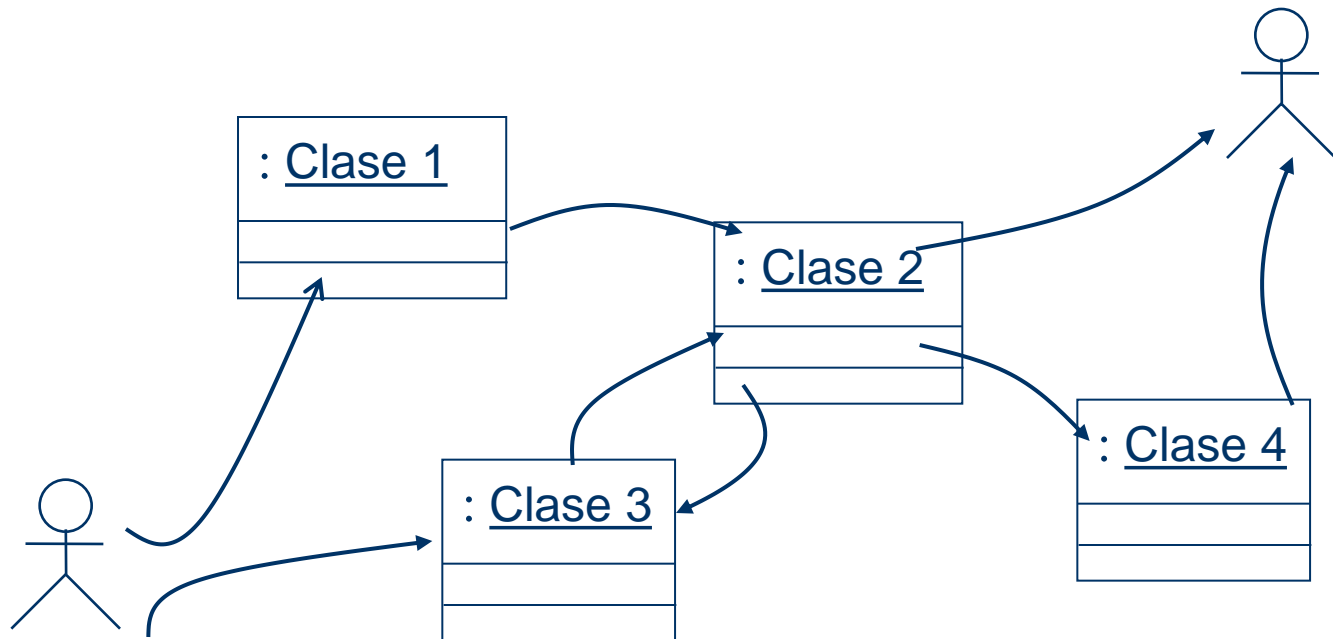
- Describir el objetivo del **modelo de comportamiento**.
- Definir los conceptos **evento** y **operación** del sistema.
- Describir el objetivo del **diagrama de secuencia de sistema**.
- Describir el objetivo del **contrato de una operación** del sistema.
- Describir la **plantilla** del **contrato de una operación** del sistema.
- Realizar el modelo de comportamiento de un sistema propuesto.

# Modelo de comportamiento

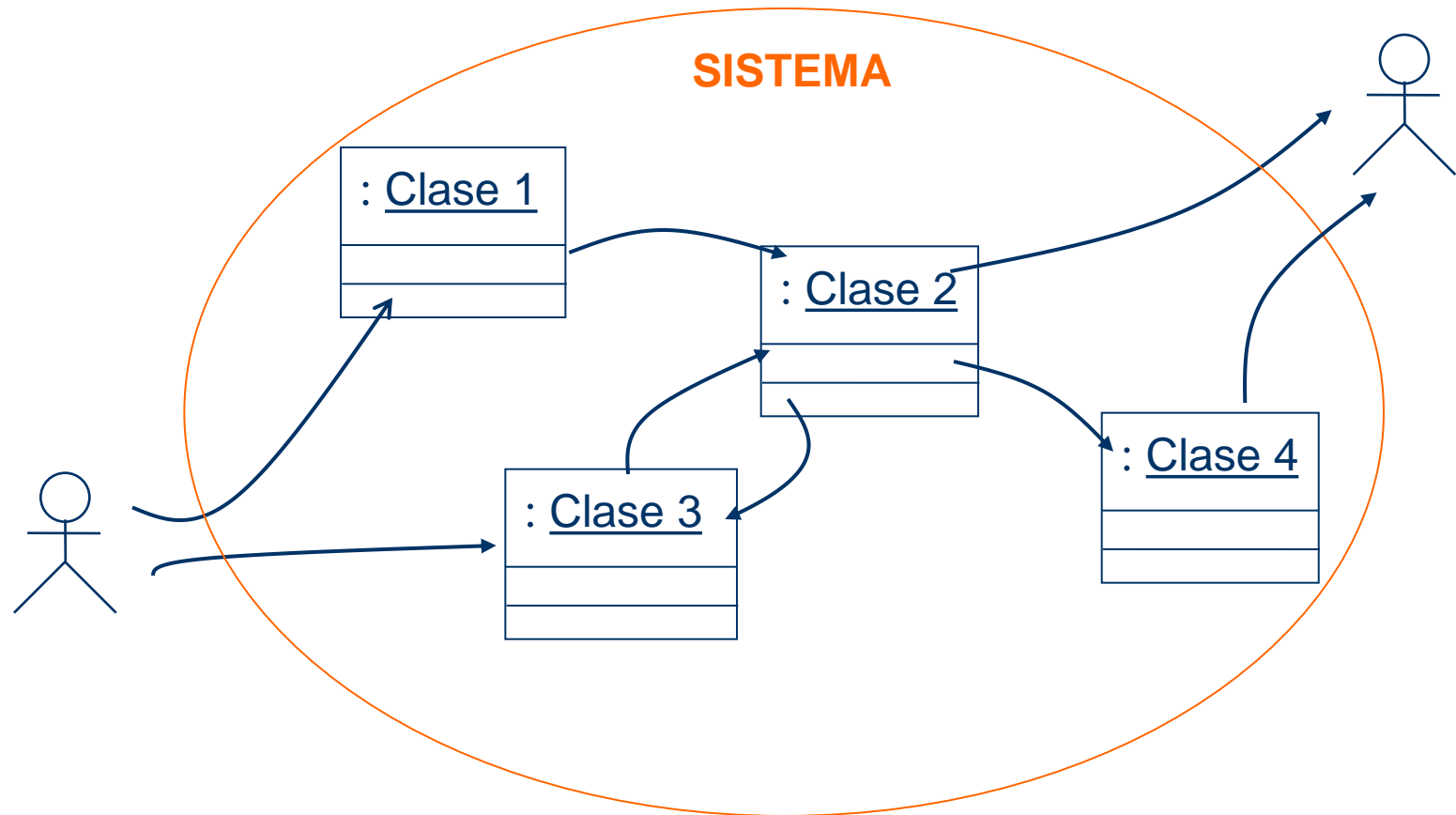


# Modelo de comportamiento

Los **objetos** se comunican mediante la **invocación de operaciones** de otros objetos



# Modelo de comportamiento



- Consideramos un “**sistema**” que engloba a todos los objetos.
- La **especificación del comportamiento** del sistema se realiza mediante el **modelo de comportamiento** del sistema.

# Modelo de comportamiento

## Diagramas de secuencia del sistema

- Muestran la **secuencia de eventos** entre los actores y el sistema
- Permiten identificar las **operaciones del sistema**

## Contratos de las operaciones del sistema

- Describen los **efectos** que las **operaciones del sistema** tienen en el **estado** del sistema

# Modelo de comportamiento

## Diagramas de secuencia del sistema

- Muestran la **secuencia de eventos** entre los actores y el sistema
- Permiten identificar las **operaciones del sistema**

## Contratos de las operaciones del sistema

- Describen los **efectos** que las **operaciones del sistema** tienen en el **estado** del sistema



## Diagrama de secuencia del sistema

### Objetivo

Identificar los eventos y las operaciones del sistema.

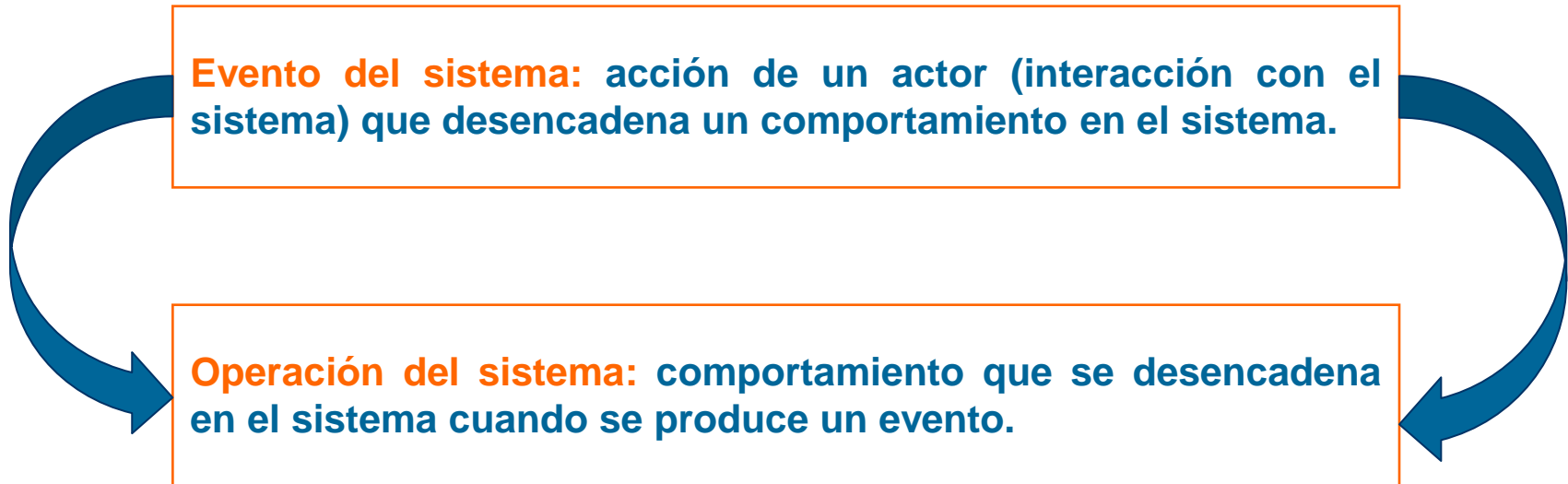
### Punto de partida

- **Descripciones de los casos de uso.**
- **Modelo conceptual de datos.**

### ¿Cuántos se realizan?

Uno por cada escenario relevante de un caso de uso.

## Diagrama de secuencia del sistema



## Diagrama de secuencia del sistema

Construcción

## Diagrama de secuencia del sistema

### Construcción

1. Dibujar una **línea vertical** que representa el **sistema**.

## Diagrama de secuencia del sistema

### Construcción

1. Dibujar una **línea vertical** que representa el **sistema**.
2. Dibujar una **línea vertical** por cada actor que interactúa con el **sistema**.


## Diagrama de secuencia del sistema

### Construcción

1. Dibujar una **línea vertical** que representa el **sistema**.
2. Dibujar una **línea vertical** por cada **actor** que interactúa con el sistema.
3. Identificar los **eventos externos** generados por los actores del sistema (a partir de la descripción de los escenarios de los casos de uso).

## Diagrama de secuencia del sistema

### Construcción

1. Dibujar una **línea vertical** que representa el **sistema**.
2. Dibujar una **línea vertical** por cada **actor** que interactúa con el sistema.
3. Identificar los **eventos externos** generados por los actores del sistema (a partir de la descripción de los escenarios de los casos de uso).
4. Representar los eventos externos (**mensajes actor**  **sistema**).

# Modelo de comportamiento

## Ejemplo

**Caso de uso:** Venta de productos

**Descripción:** Capturar una venta y su pago en efectivo

**Actores:** Cliente, Cajero (principal)



¿Eventos del sistema?

**Resumen:** un cliente llega a la caja con productos para comprar. El cajero registra los productos y gestiona el pago en efectivo. Al acabar, el cliente se va con los productos.

### Escenario principal:

1. El caso de uso comienza cuando un Cliente llega a la caja con los productos para comprar.
  2. El Cajero indica que comienza una nueva venta e introduce el punto de venta (*wnum-pv*).
  3. El Sistema comprueba que existe el punto de venta de la compra y registra el inicio de una nueva venta.
  4. El Cajero introduce el identificador (*wcodigo*) y la cantidad (*wcantidad*) de un producto
  5. El Sistema comprueba que existe el producto y registra la línea de venta.
- Repetir los pasos 4-5 hasta que el Cajero indique que no hay más productos
6. El Sistema calcula el total de la compra y lo muestra.
  7. El Cajero dice el total de la compra al Cliente.
  8. El Cliente entrega una cantidad de dinero posiblemente superior al total de la compra.
  9. El Cajero introduce el dinero que ha recibido (*wimporte*).
  10. El Sistema calcula y muestra el cambio al Cliente y registra el pago.
  11. El Cajero deposita el dinero recibido en la caja y coge el cambio. El Cajero da el cambio al Cliente.
  12. El Cliente se va con los productos comprados.



# Modelo de comportamiento

## Ejemplo

**Caso de uso:** Venta de productos

**Descripción:** Capturar una venta y su pago en efectivo

**Actores:** Cliente, Cajero (principal)

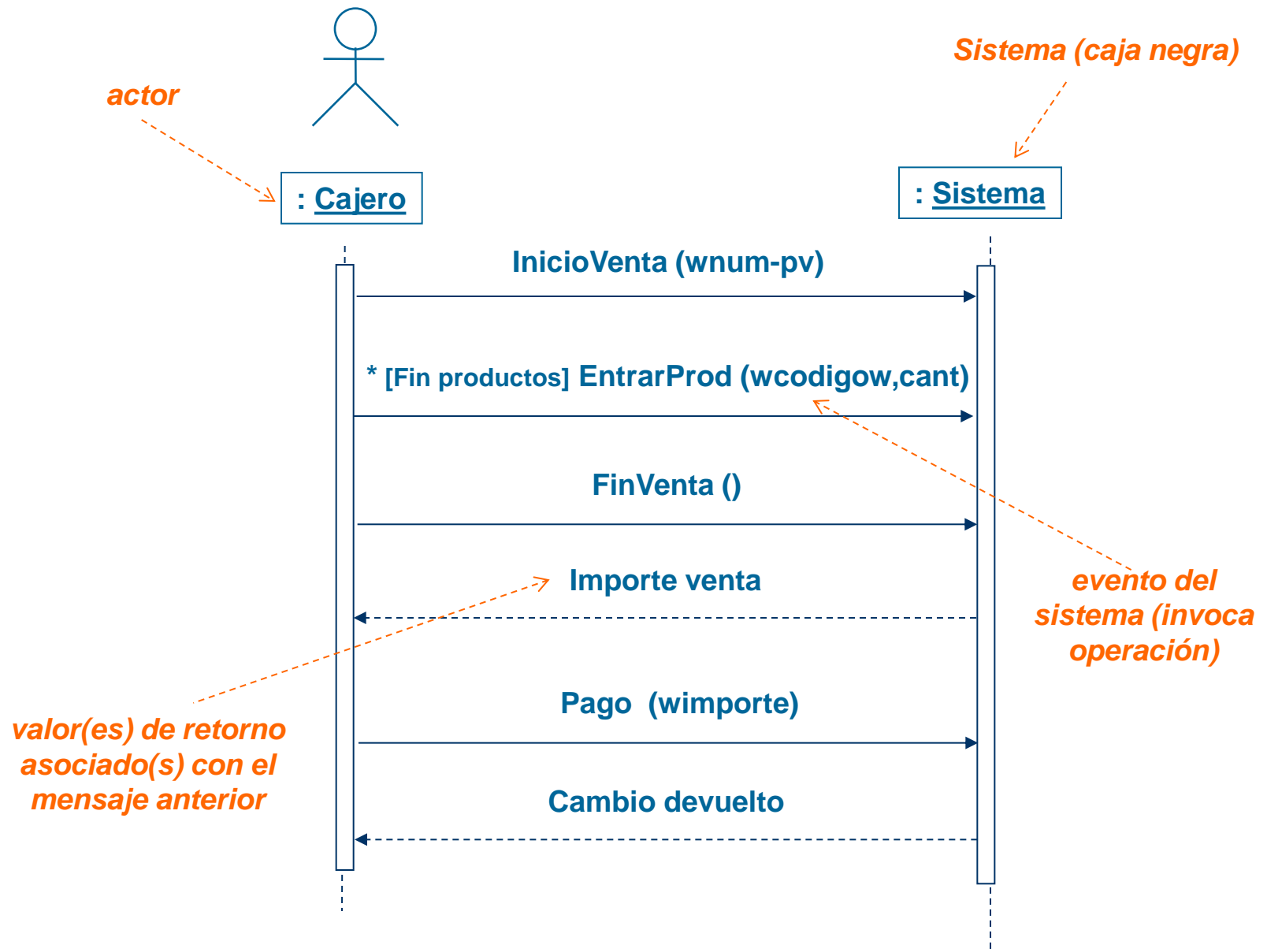


**Resumen:** un cliente llega a la caja con productos para comprar. El cajero registra los productos y gestiona el pago en efectivo. Al acabar, el cliente se va con los productos.

### Escenario principal:

1. El caso de uso comienza cuando un Cliente llega a la caja con los productos para comprar.
  2. El Cajero indica que comienza una nueva venta e introduce el punto de venta (*wnum-pv*).
  3. El Sistema comprueba que existe el punto de venta de la compra y registra el inicio de una nueva venta.
  4. El Cajero introduce el identificador (*wcodigo*) y la cantidad (*wcantidad*) de un producto
  5. El Sistema comprueba que existe el producto y registra la línea de venta.
- Repetir los pasos 4-5 hasta que el Cajero indique que no hay más productos
6. El Sistema calcula el total de la compra y lo muestra.
  7. El Cajero dice el total de la compra al Cliente.
  8. El Cliente entrega una cantidad de dinero posiblemente superior al total de la compra.
  9. El Cajero introduce el dinero que ha recibido (*wimporte*).
  10. El Sistema calcula y muestra el cambio al Cliente y registra el pago.
  11. El Cajero deposita el dinero recibido en la caja y coge el cambio. El Cajero da el cambio al Cliente.
  12. El Cliente se va con los productos comprados.

# Modelo de comportamiento



# Modelo de comportamiento

Un **evento del sistema** provoca la **ejecución de una operación del sistema** (mismo nombre y mismos parámetros).

## Sistema

**InicioVenta (wnum-pv)**

**EntrarProd (wcodigo,wcantidad)**

**FinVenta ( )**

**Pago (wimporte)**

# Modelo de comportamiento

## Diagramas de secuencia del sistema

- Muestran la **secuencia de eventos** entre los actores y el sistema
- Permiten identificar las **operaciones del sistema**

## Contratos de las operaciones del sistema

- Describen los **efectos** que las **operaciones del sistema** tienen en el **estado** del sistema

## Contrato de operación del sistema

### Objetivo

Describir el comportamiento del sistema en base a los **cambios** que se producen en el **estado del sistema** (objetos del modelo conceptual de datos) después de ejecutar la operación.

### Punto de partida

- **Descripciones de los casos de uso.**
- **Modelo conceptual de datos.**

### ¿Cuántos se realizan?

Un contrato por cada **operación** del sistema **relevante** o que **no esté clara** en la descripción de los casos de uso.

## Plantilla de un contrato de operación del sistema

- **Nombre:** Signatura de la operación.
- **Responsabilidades:** descripción informal del propósito de la operación.
- **Referencias cruzadas (opcional):** casos de uso en los que puede tener lugar la operación.
- **Precondiciones y Postcondiciones.**

### Precondiciones

Suposiciones relevantes sobre el **estado del sistema** (objetos del modelo conceptual de datos) **antes** de ejecutar la **operación**.

### Postcondiciones

Cambios relevantes en el **estado del sistema** (objetos del modelo conceptual de datos) **después** de ejecutar la **operación**.

## Precondiciones de un contrato de operación

**Condiciones** que tiene que cumplir el **estado** del sistema para que la operación pueda ejecutarse (**no** son las **acciones** que se realizan antes de que la operación se ejecute).

- Existe un objeto que cumple una determinada condición.
- Existe una relación concreta entre objetos del sistema.
- Un atributo de un objeto tiene un valor determinado.
- Etc.

## Postcondiciones de un contrato de operación

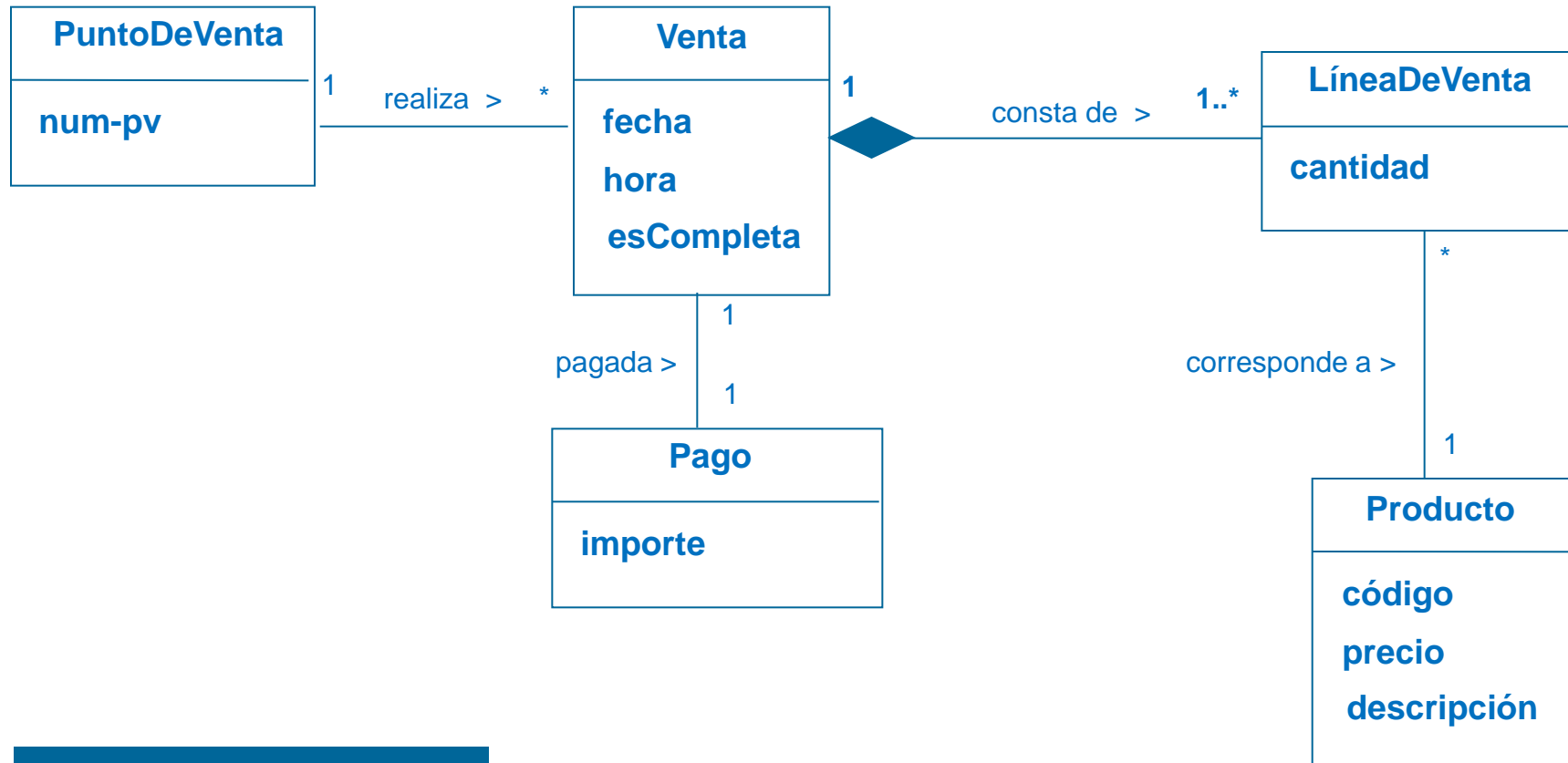
**Cambios** que se producen en el **estado** del sistema cuando se ejecuta la operación del sistema (**no** son las **acciones** que realiza la operación del sistema).

- Creación y eliminación de objetos.
- Creación y eliminación de relaciones entre objetos.
- Modificación de valores de los atributos.
- Etc.



# Modelo de comportamiento

## Ejemplo

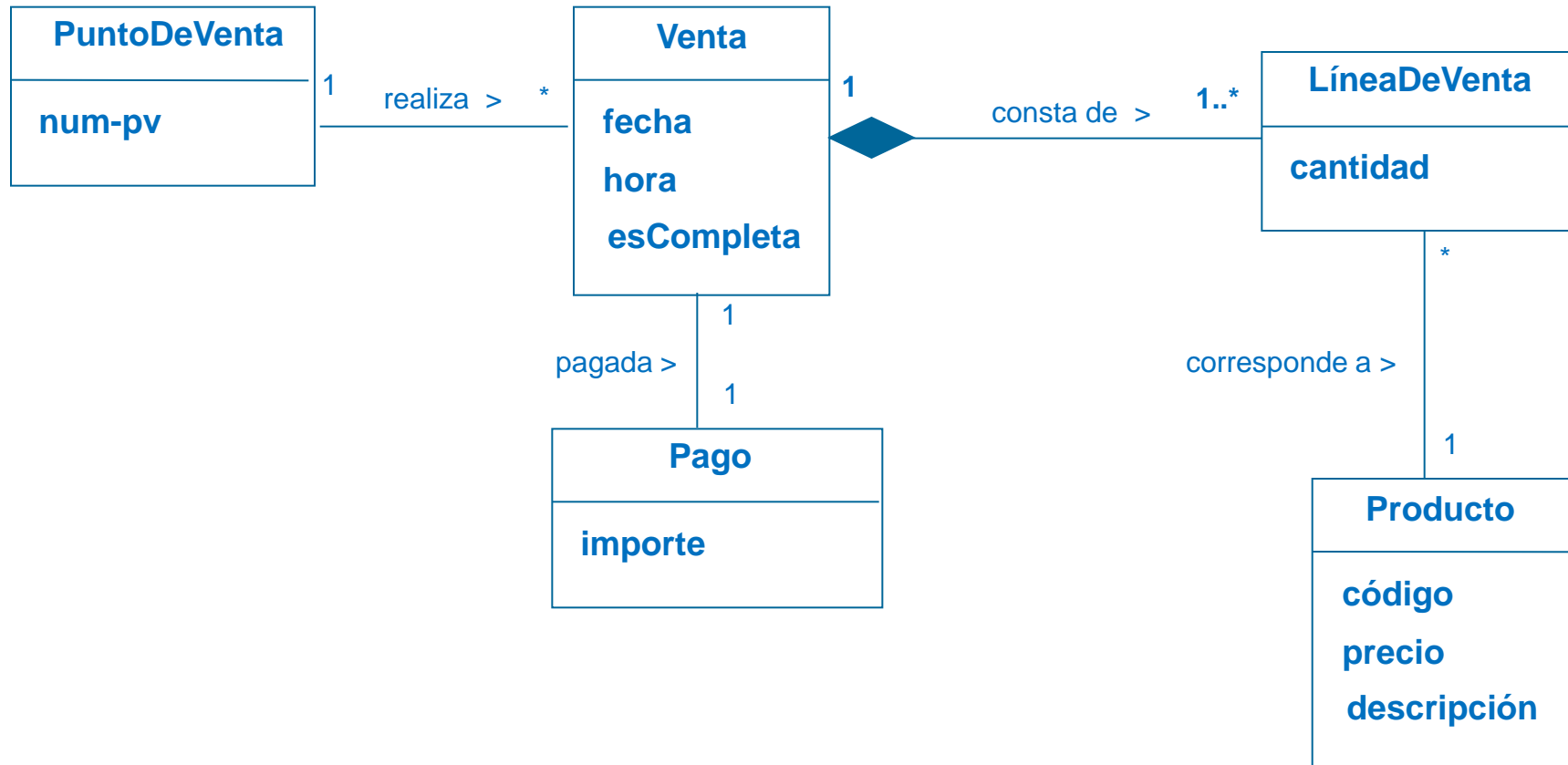


## Restricciones textuales

- La clave externa de *PuntoDeVenta* es *num-pv*.
- La clave externa de *Producto* es *código*.
- Un punto de venta no puede tener más de una venta el mismo día y a la misma hora

# Modelo de comportamiento

## Ejemplo



## Evento

## Operación

1. El Cajero indica que comienza una nueva venta e introduce el punto de venta (*wnum-pv*).
2. El Sistema comprueba que existe el punto de venta de la compra y registra el inicio de una nueva venta.

## Operación: InicioVenta

**Operación:** InicioVenta (wnum-pv)

**Responsabilidades:** Iniciar el registro de una nueva venta

**Referencias Cruzadas:** Casos de Uso: Venta de Productos

**Precondiciones:**

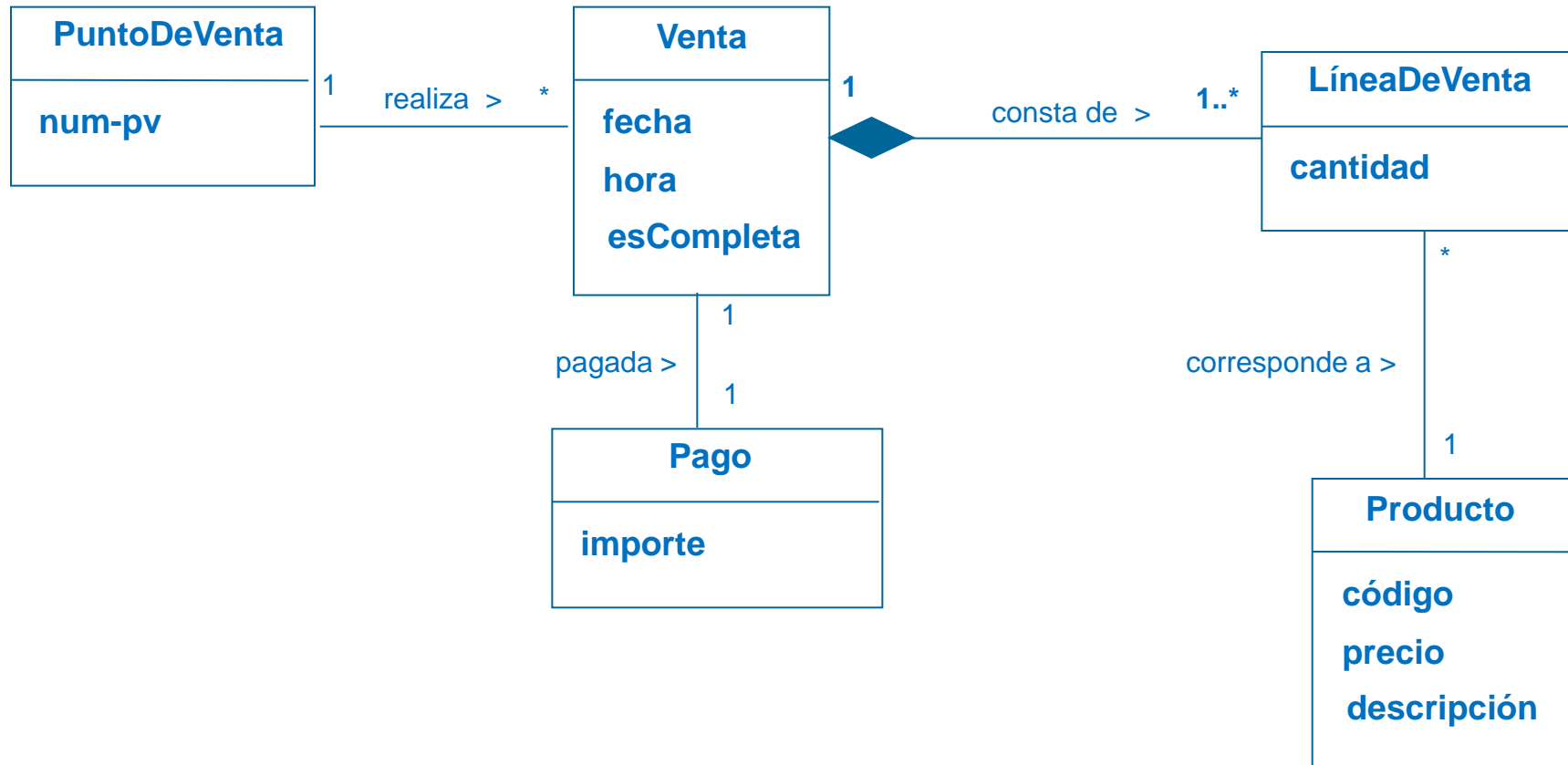
- Existe un objeto punto de venta *PV* con *num-pv* = *wnum-pv*

**Postcondiciones:**

- Se crea una instancia *V* de *Venta* (creación de objeto)
- Se inicializan los atributos de *V* (*V.fecha* = fecha del sistema, *V.hora* = hora del sistema y *V.esCompleta* = “false”) (modificación de atributo)
- Se crea un enlace de la asociación “realiza” entre *V* y *PV* (creación de enlace)

# Modelo de comportamiento

## Ejemplo



## Evento

## Operación

1. El Cajero introduce el identificador (*wcodigo*) y la cantidad (*wcantidad*) de un producto
2. El Sistema comprueba que existe el producto y registra la línea de venta.

## Operación: EntrarProd

**Operación:** EntrarProd (*wcódigo*, *wcantidad*)

**Responsabilidades:** registrar un nuevo producto de la venta en curso (línea de venta)

**Referencias Cruzadas:** Casos de Uso: Venta de Productos

**Precondiciones:**

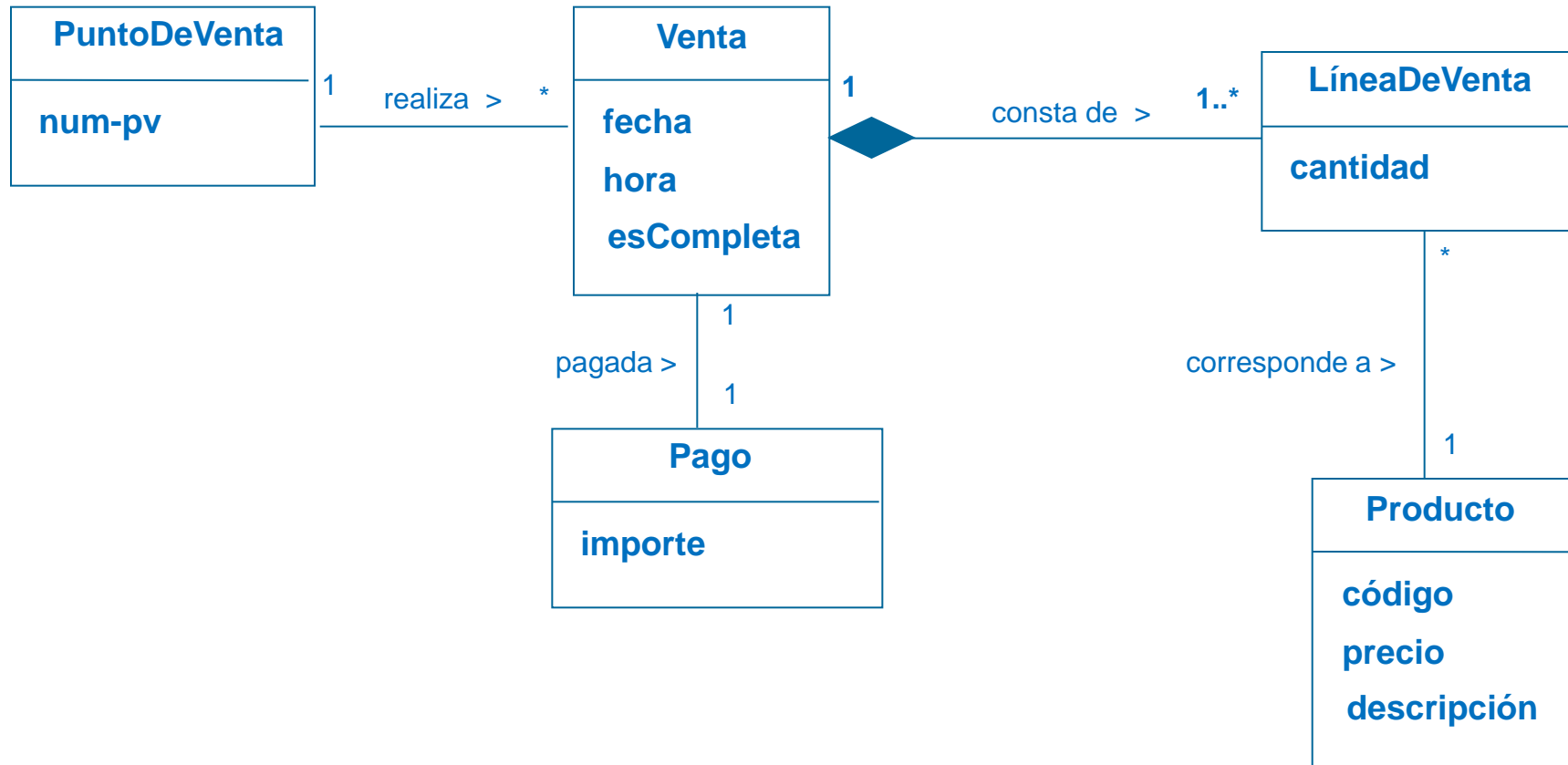
- Existe una Venta en curso *V* y un Producto *P* con *código* = *wcódigo*

**Postcondiciones:**

- Se crea una instancia LíneaDeVenta *LDV* en el objeto venta *V* (creación de un objeto componente de *V*)
- Se asigna *wcantidad* al atributo *LDV.cantidad* (*LDV.cantidad* = *wcantidad*) (modificación de atributo)
- Se crea un enlace de la asociación “corresponde a “ entre *LDV* y *P* (creación de enlace)

# Modelo de comportamiento

## Ejemplo



Evento

Operación

Repetir los pasos 4-5 hasta que el **Cajero** indique que no hay más productos

6. El Sistema calcula el total de la compra y lo muestra.

## Operación: FinVenta

**Operación:** FinVenta ()

**Responsabilidades:** finalizar el registro de una venta y mostrar el importe a pagar

**Referencias Cruzadas:** Casos de Uso: Venta de Productos

**Precondiciones:**

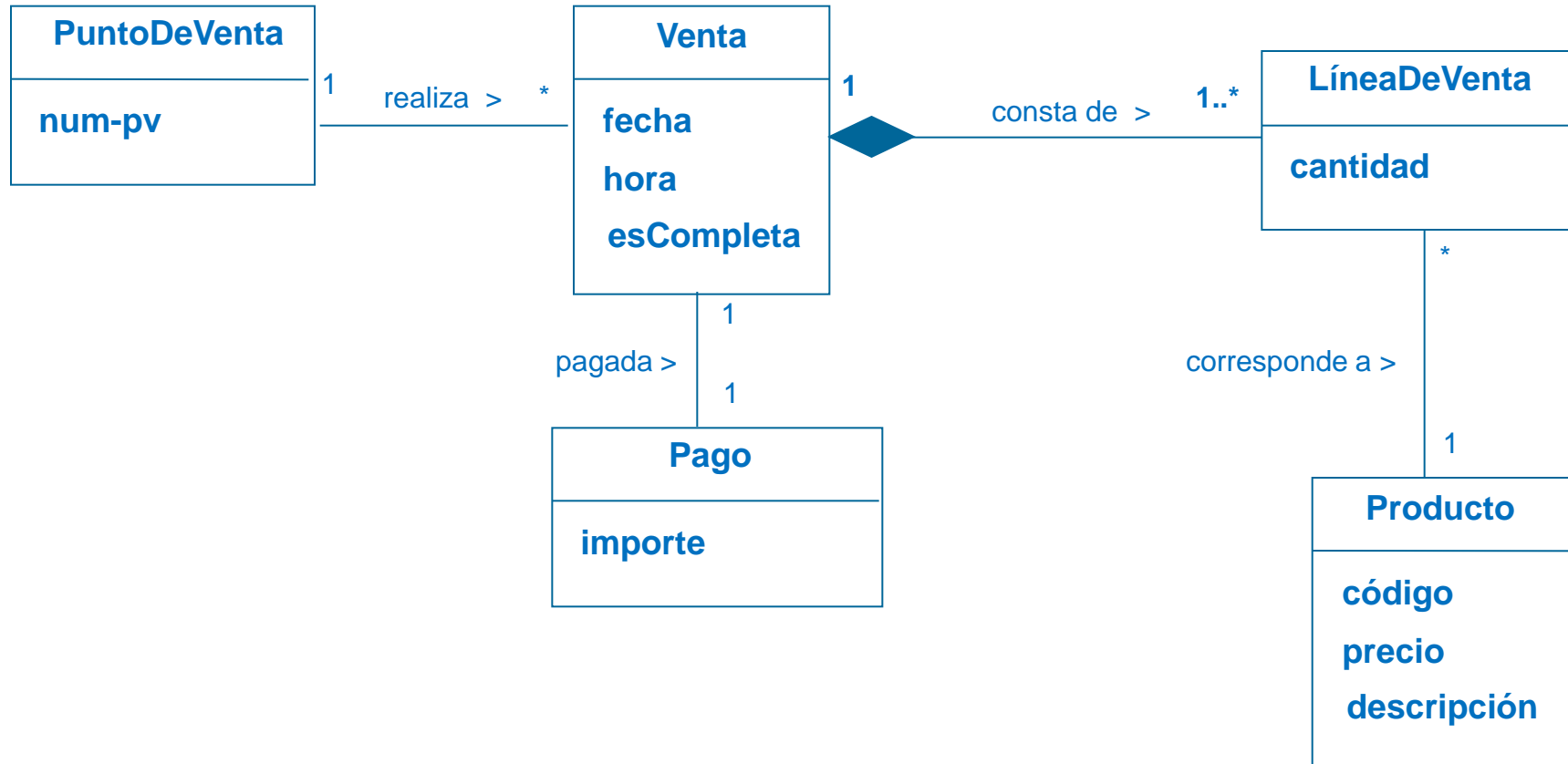
- Existe una Venta en curso *V*

**Postcondiciones:**

- Se asigna “true” al atributo *esCompleta* de *V* (*V.esCompleta* = true” (modificación de atributo)

# Modelo de comportamiento

## Ejemplo



Evento

Operación

6. El Cajero introduce el dinero que ha recibido (*wimporte*).
7. El Sistema calcula y muestra el cambio al Cliente y registra el pago.



## Operación: Pago

**Operación:** Pago (*wimporte*)

**Responsabilidades:** Registrar el pago de la venta y mostrar el cambio a devolver

**Referencias Cruzadas:** Casos de Uso: Venta de Productos

**Precondiciones:**

- Existe una Venta en curso *V* y el atributo *V.esCompleta* = “*true*”

**Postcondiciones:**

- Se crea un objeto *P* de la clase *Pago* (creación de objeto)
- Se asigna *wimporte* al atributo *P.cantidad* (modificación de atributo)
- Se crea un enlace de la asociación “pagada” entre *V* y *P* (creación de enlace)