

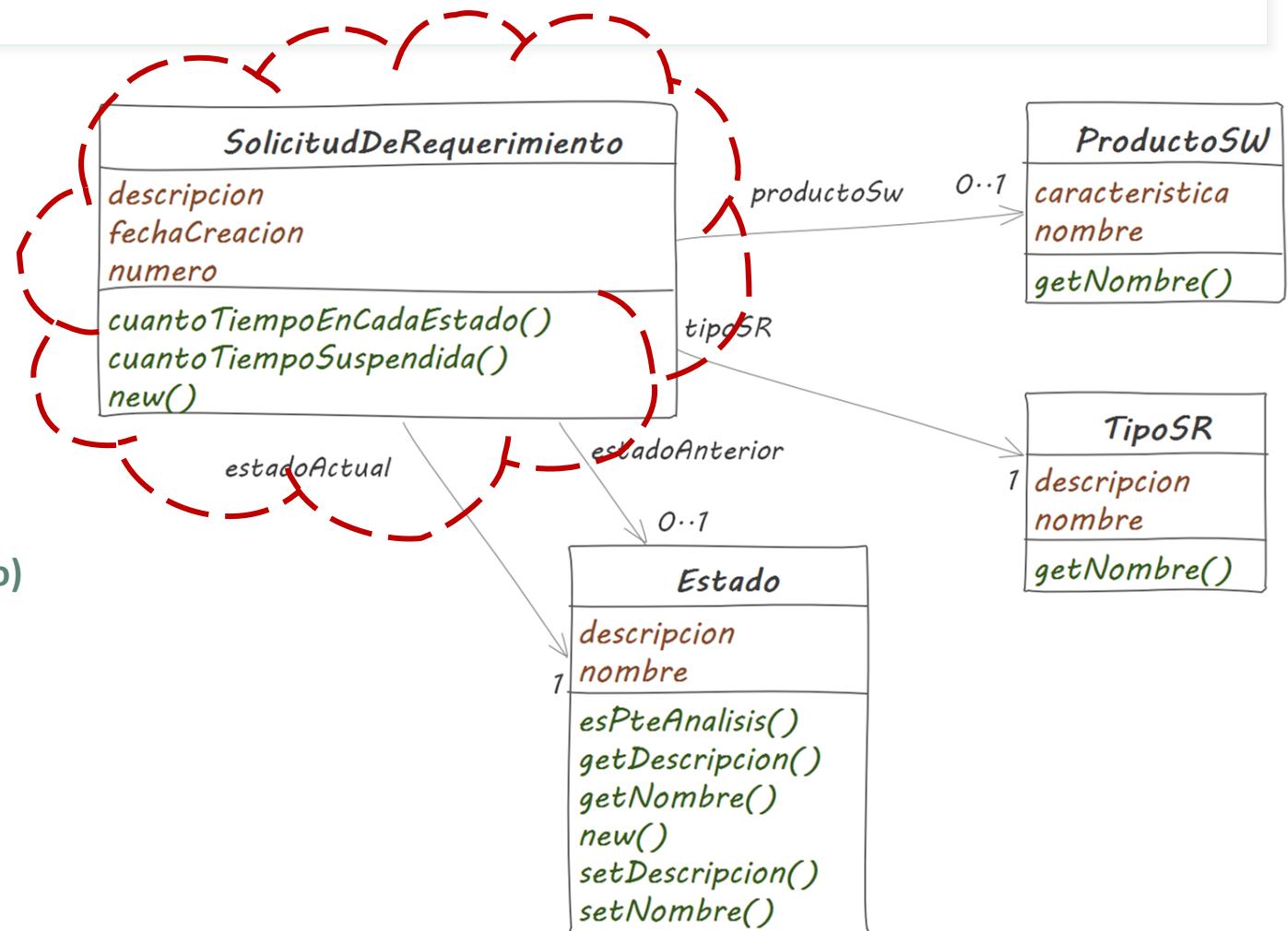
The background features a green-to-black gradient with a dotted line forming a path. Four grey paper planes are positioned along this path: one at the start, two in the middle, and one further ahead. A single magenta paper plane is located on the right side of the slide.

# Tips para el Modelado de Realizaciones de Caso de Uso

Cátedra de Diseño de Sistemas de Información - Judith Meles

# Creación de Objetos: Creación de un objeto

- Vemos la estructura para analizarla e identificar el objeto a crear.
- Objeto de la clase **SolicitudDeRequerimiento**
  - ✓ Tiene atributos por valor (propios)
    - ✓ Que el actor ingresa (descripción)
    - ✓ Que asigna el sistema (fechaCreacion/número)
  - ✓ Tiene atributos por referencia
    - ✓ Que el actor elige (ProductoSW/TipoSR)
    - ✓ Que asigna el sistema (Estado)



# Creación de Objetos: Creación de un objeto

En la dinámica simplificada, el Gestor se encarga primero de:

1. Obtener los atributos que necesita

✓ Atributos por valor:

✓ Que ingresa el actor

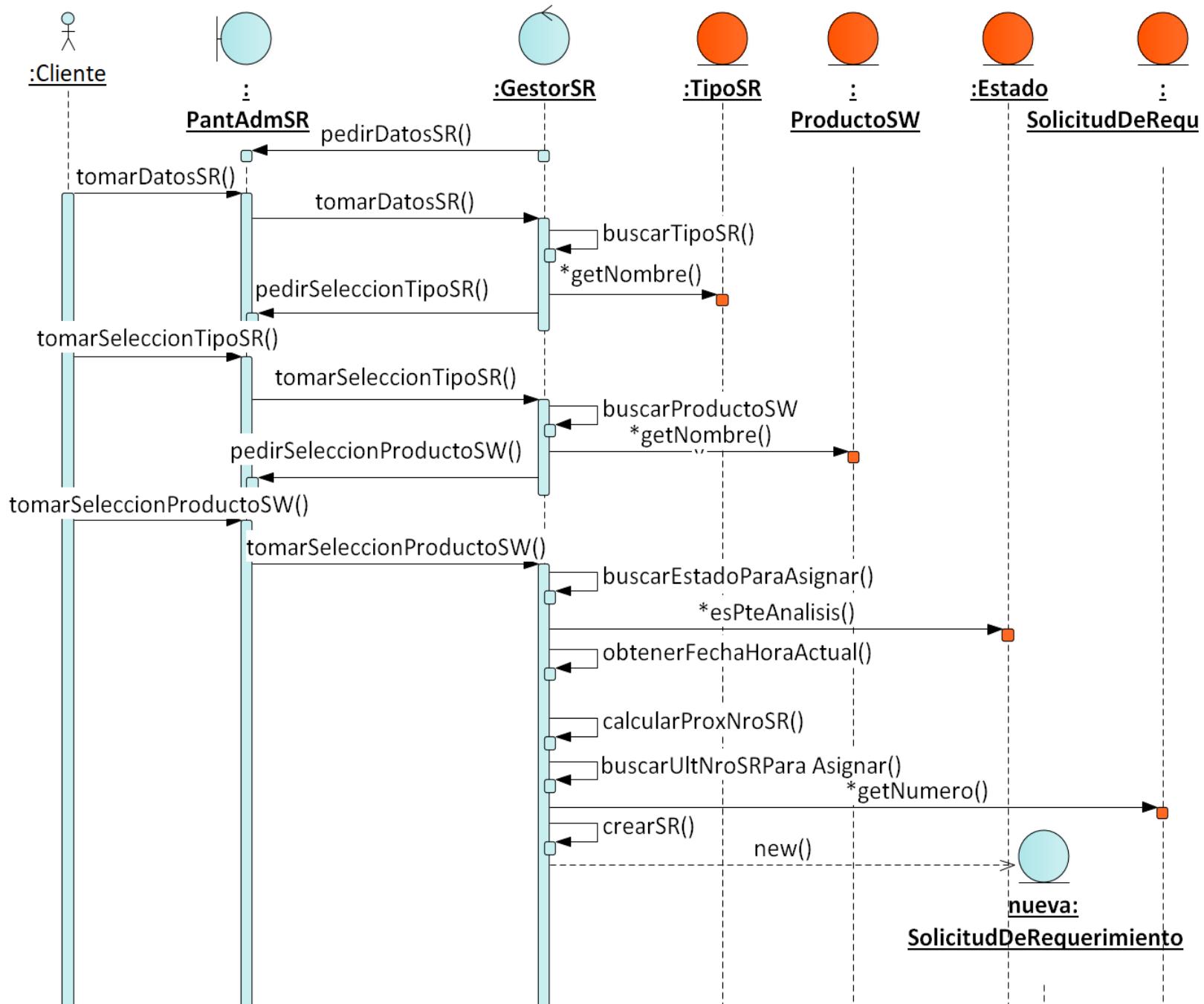
✓ Que obtiene el sistema  
(Gestor)

✓ Atributos por referencia:

✓ Que el actor selecciona  
(ProductoSW/TipoSR)

✓ Predefinidos en el sistema  
(Estado); en el caso de la  
creación no hay estado  
anterior.

2. Crear el objeto



# ¿Qué vimos ...?



Búsqueda de todos los objetos de una clase para pedir que el actor seleccione uno.



Búsqueda de un objeto con un criterio.



Obtención de fecha actual



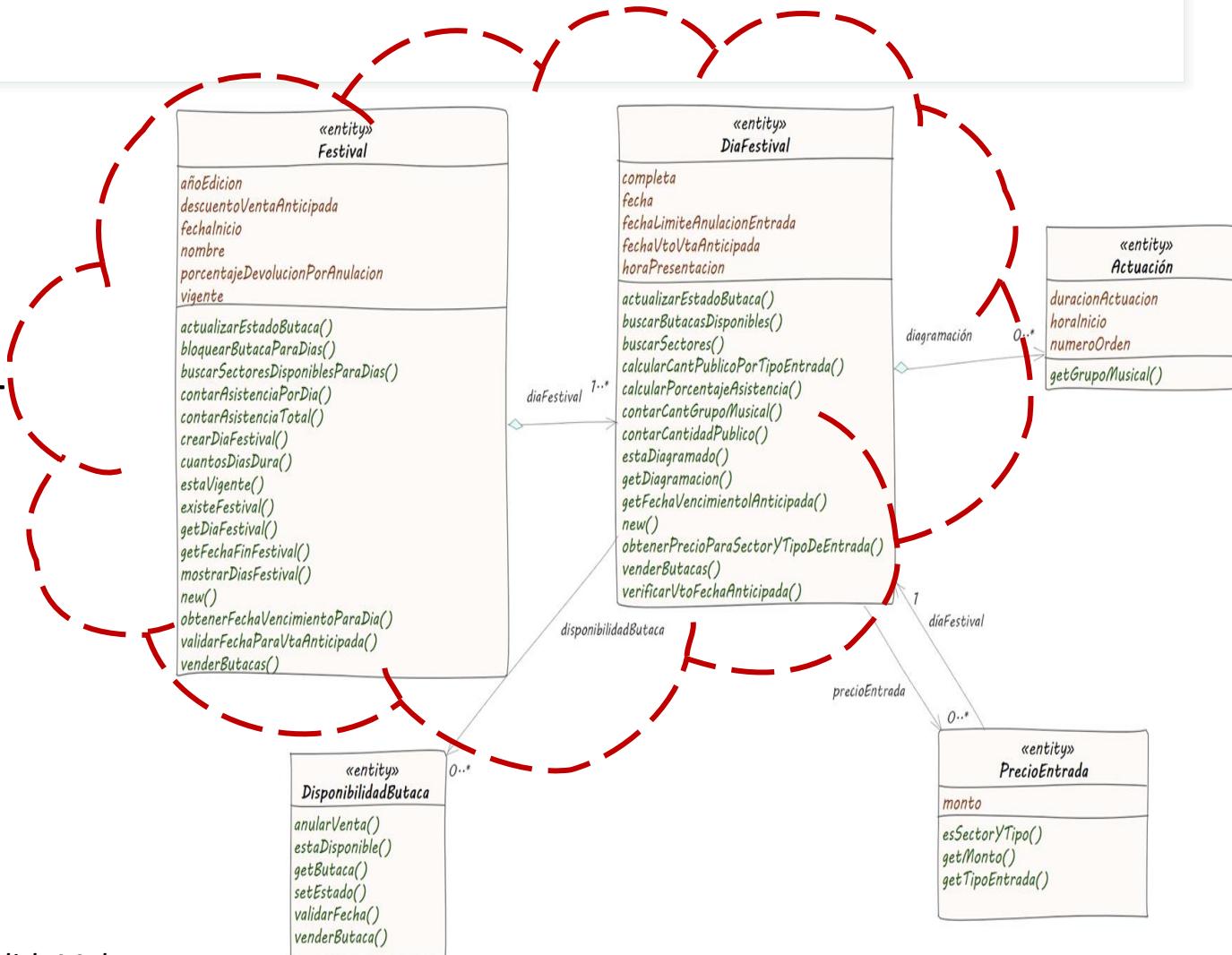
Obtención del último número para asignar el siguiente al momento de creación del nuevo objeto



Creación de un objeto luego de tener todos los atributos (propios y por referencia) que necesitamos para crearlo.

# Creación que implica crear objetos de más de una clase

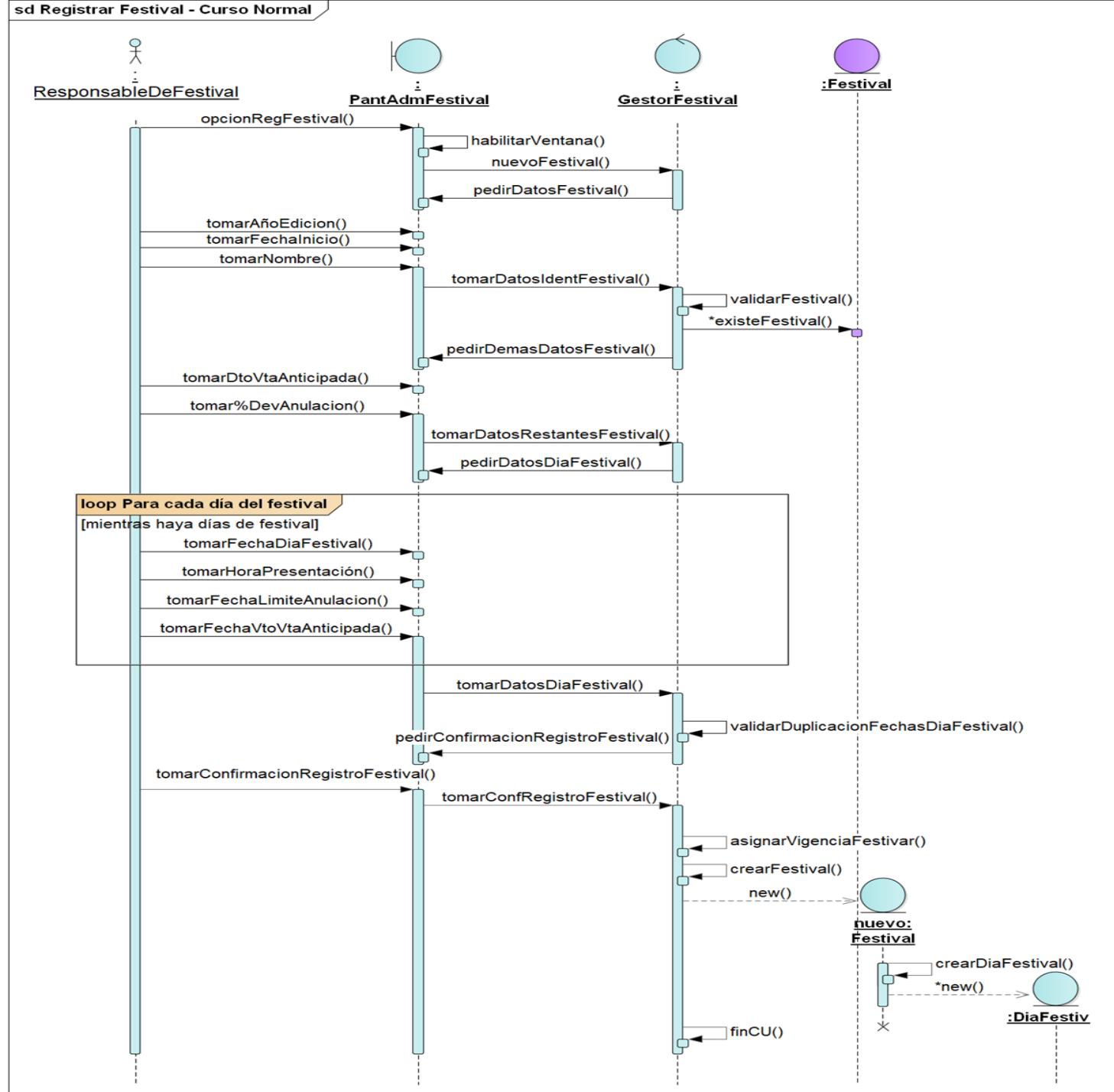
- Vemos la estructura para analizarla e identificar los objetos a crear.
- 1 objeto de la clase **Festival** y al menos 1 objeto de la clase **DiaFestival**.
- Resto de las referencias tienen multiplicidad **0..\***, no se las tiene en cuenta en la creación.



# Creación que implica crear **objetos de más de una clase**

En la dinámica el Gestor se encarga primero de:

1. Obtener los atributos que necesitan para el todo y para las partes.
  - ✓ Atributos por valor: los ingresa el actor
  - ✓ Validar la existencia previa
2. Crear el objeto “Todo”
3. Delegar al objeto “Todo” la responsabilidad de creación de los objetos que agrega/contiene/conoce



# ¿Qué vimos hasta acá...?



Búsqueda de todos los objetos de una clase para pedir que el actor seleccione uno.



Búsqueda de un objeto con un criterio.



Obtención de fecha actual



Obtención del último número para asignar el siguiente al momento de creación del nuevo objeto



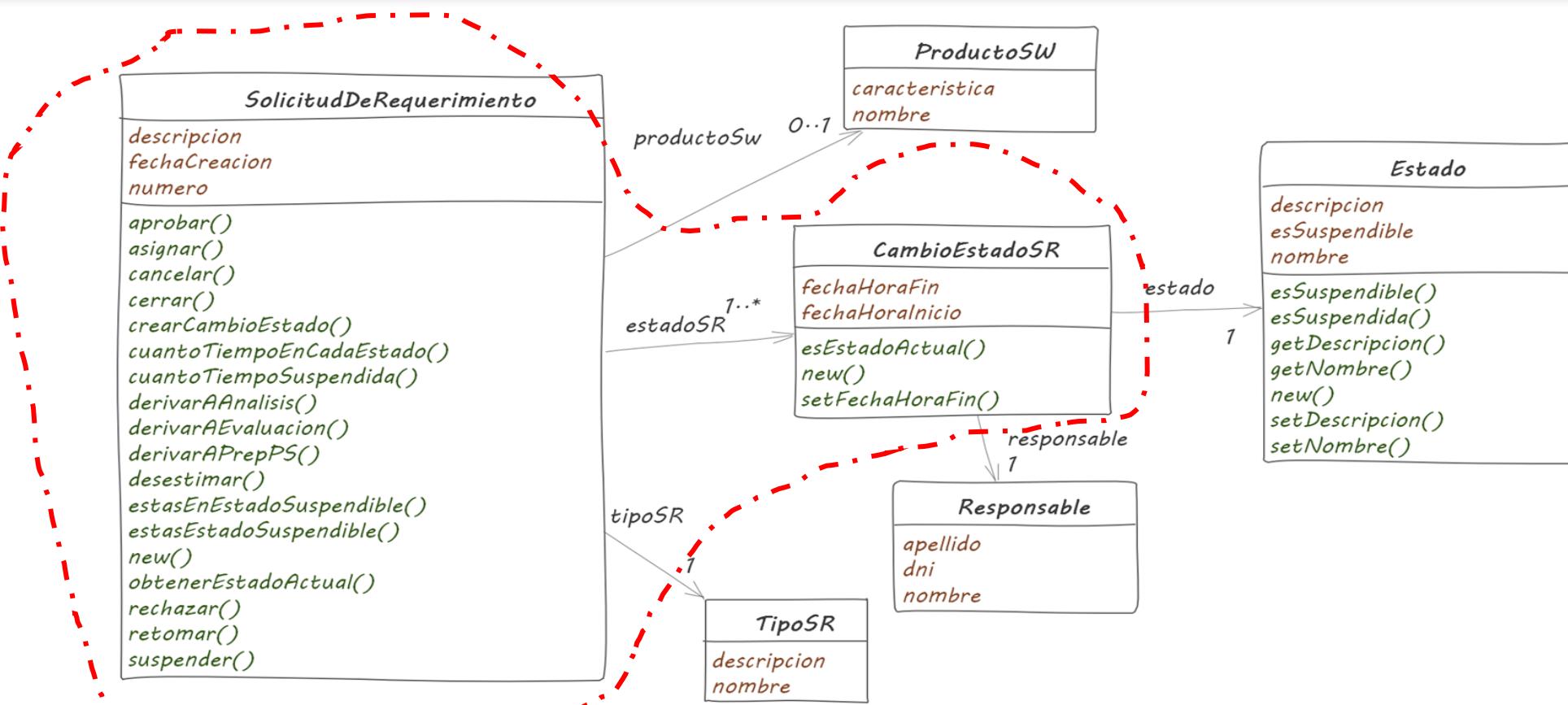
Creación de un objeto luego de tener todos los atributos (propios y por referencia) que necesitamos para crearlo.



Creación de objetos de más de una clase (cuando hay una estructura tipo Todo-Parte)

# Creación de objetos de más de una clase: Objeto con su cambio de estado

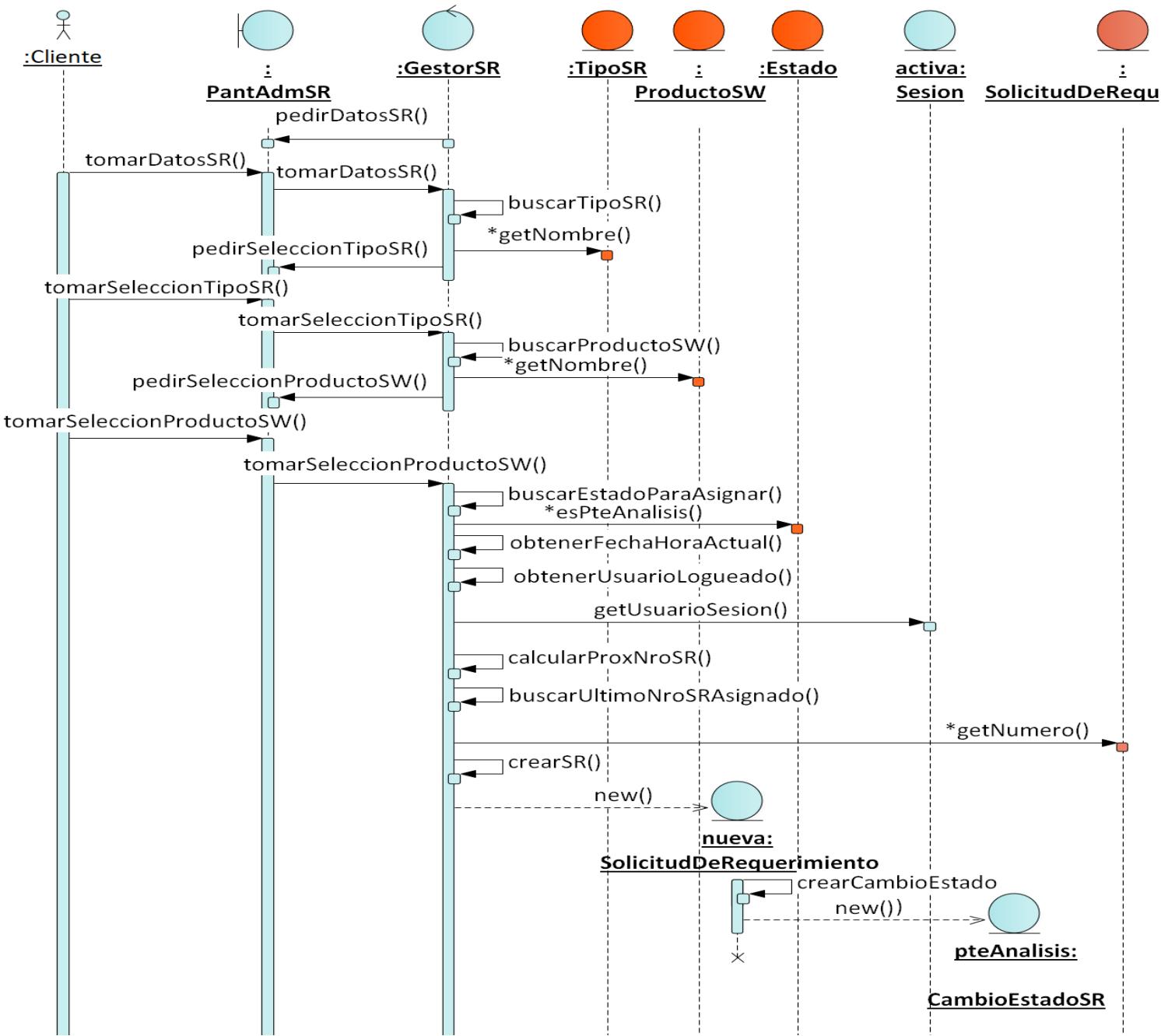
- Vemos la estructura para analizarla e identificar el objeto a crear.
- Objeto de la clase **SolicitudDeRequerimiento** y Objeto de la clase **CambioEstadoSR**
- ✓ Tiene atributos por valor (propios)
- ✓ Que asigna el sistema (FechaHoralInicio)
- ✓ Tiene atributos por referencia
  - ✓ Que el actor elige (ProductoSW/TipoSR)
  - ✓ Que asigna el sistema (Estado/Responsable)



# Creación de objetos de más de una clase: Objeto con su cambio de estado

En la dinámica simplificada, el Gestor se encarga primero de:

1. Obtener los atributos que necesita
  - ✓ Atributos por valor: los ingresa el actor
  - ✓ Atributos que selecciona el actor
2. Gestor obtiene los atributos que necesita Estado/FechaHoraInicio/ Responsable.
3. Gestor crea la SR y le delega a esta la responsabilidad de **crear su cambio de estado**.



# ¿Qué vimos ...?

---



Búsqueda de todos los objetos de una clase para pedir que el actor seleccione



Obtención de fecha actual



Obtención del usuario de la sesión



Creación de un objeto luego de tener todos los atributos (propios y por referencia) que necesitamos para crearlo.



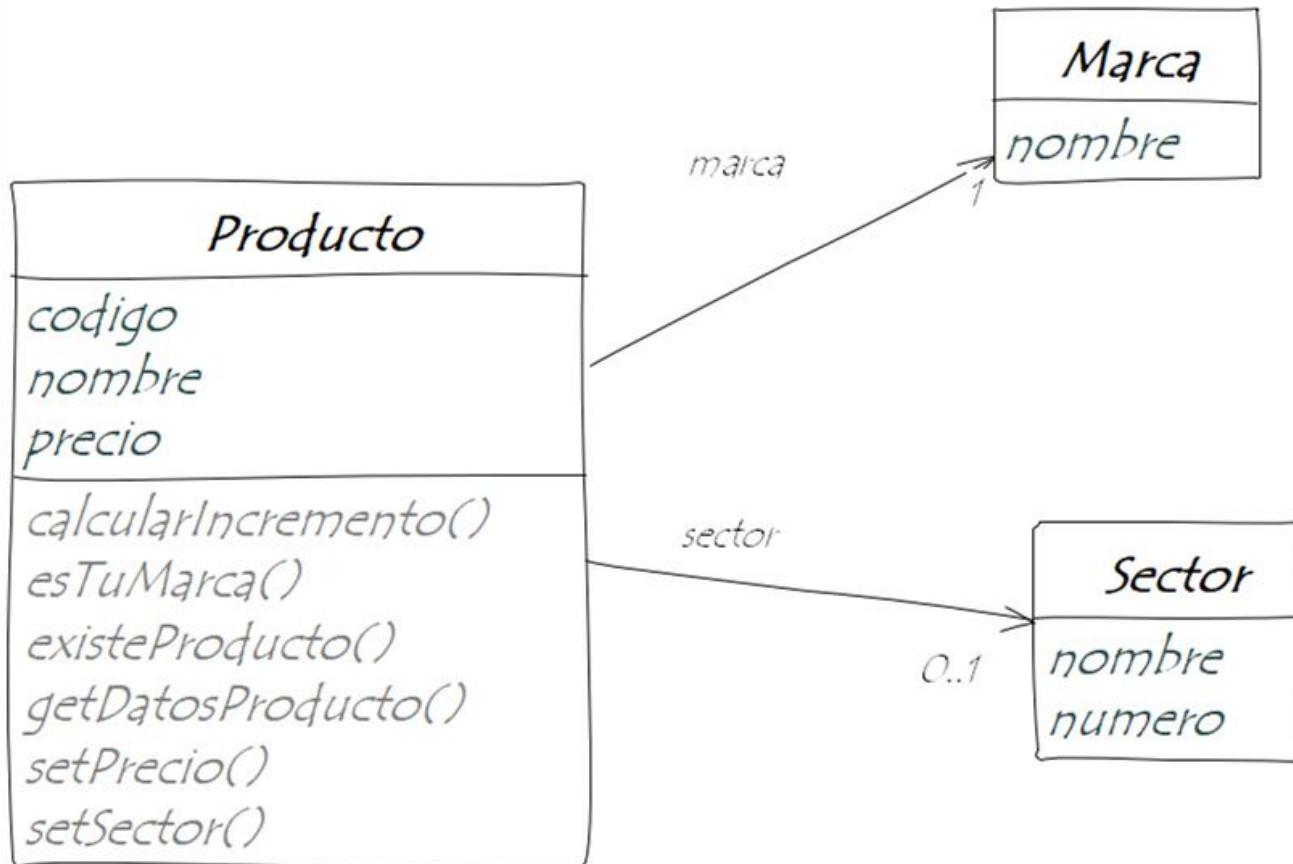
Creación de objetos de más de una clase (cuando hay una estructura tipo Todo-Parte)



Creación de objetos de más de una clase cuando el objeto crea su cambio de estado.

# Modificación de objetos:

## Modificación de atributos propios de un Objeto



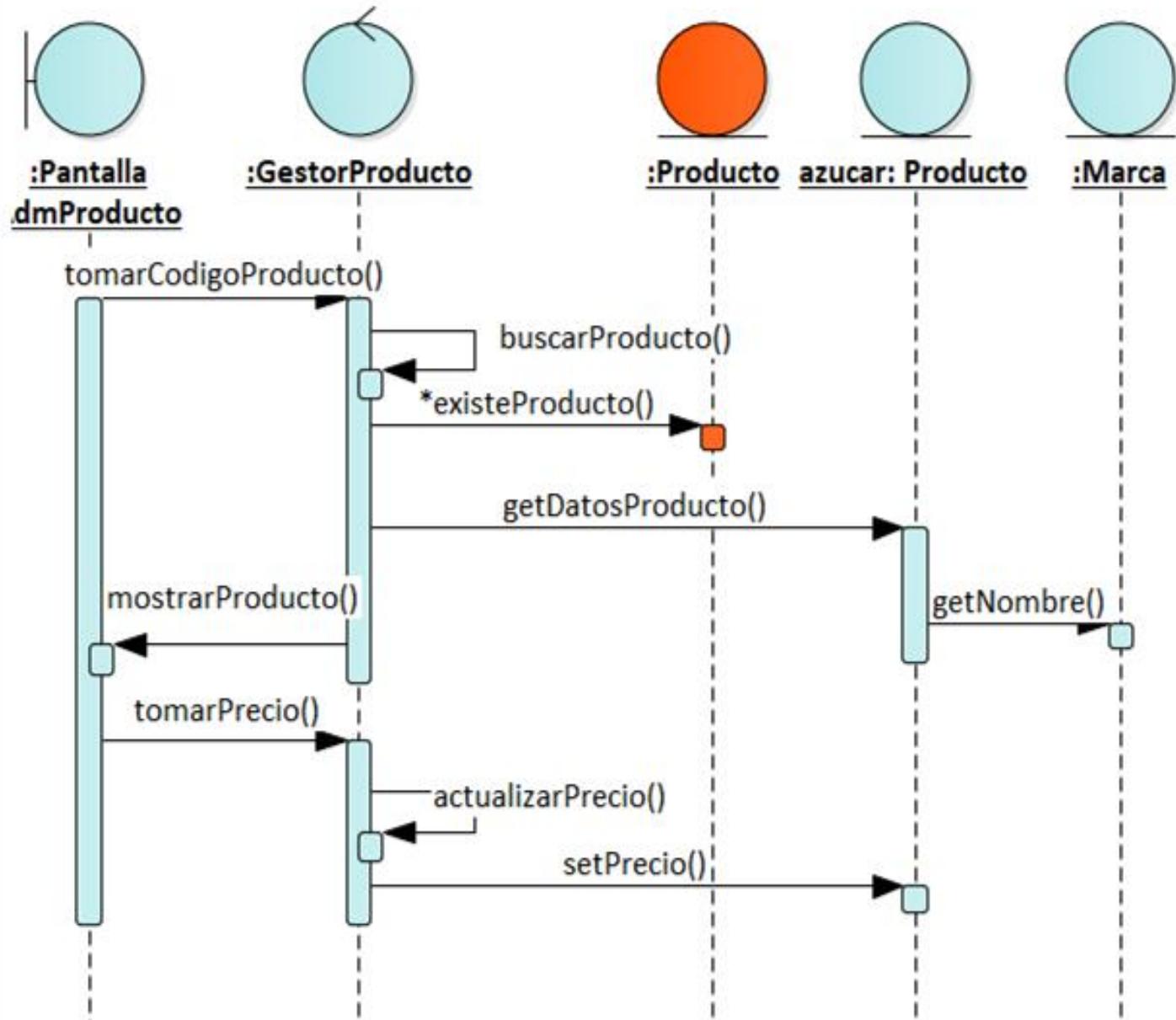
- Vemos la estructura para analizarla e identificar el objeto a modificar.
- Objeto de la clase **Producto**
  - **Vamos a modificar el precio del producto cuyo código es “3d4f6789”**

Modificación de objetos:

Modificación de atributos propios de un Objeto

• En la dinámica simplificada primero:

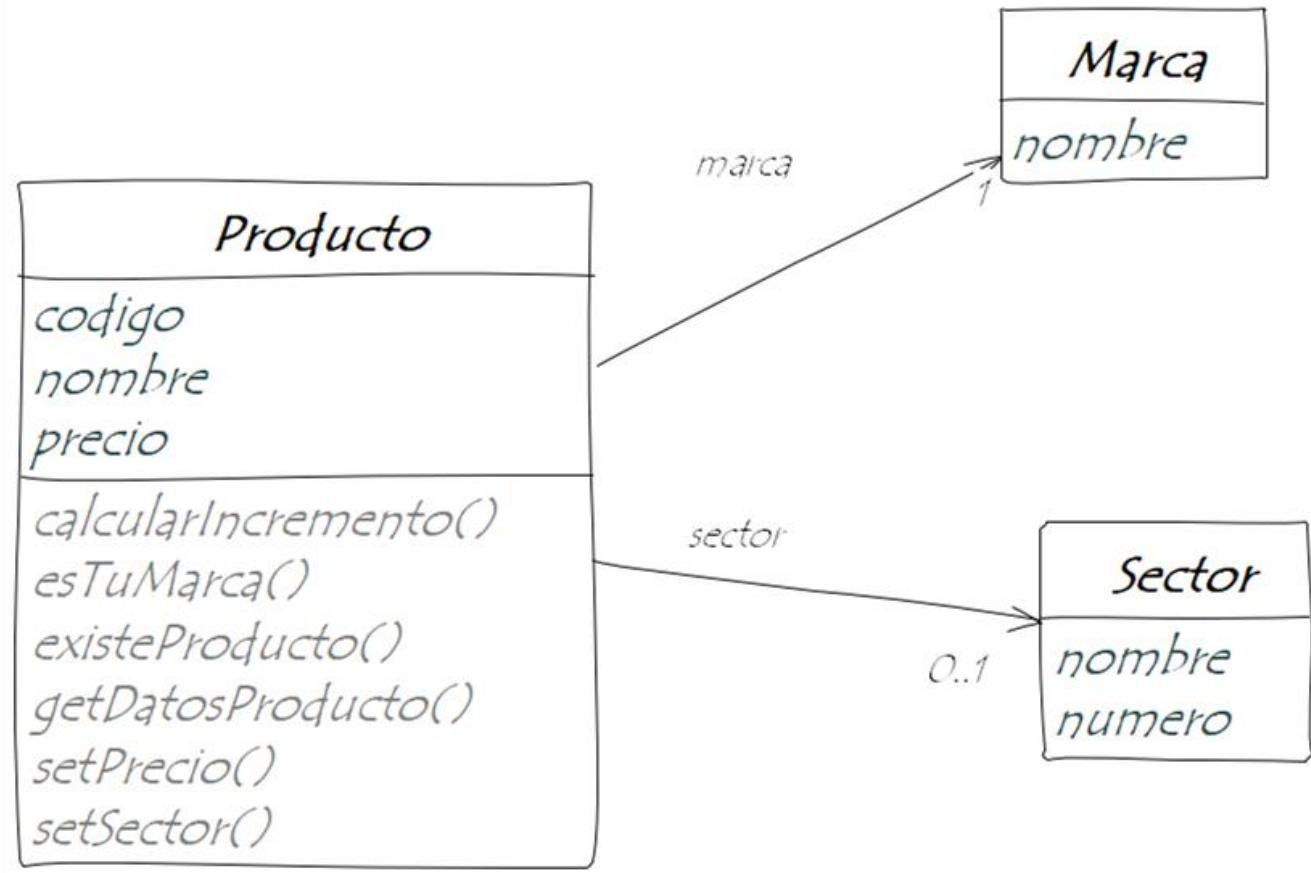
- Se ingresa el código para buscar el producto
- Gestor busca el producto muestra sus datos.
- Se ingresa el nuevo precio
- Gestor invoca el método de seteo del atributo **precio, actualizando el valor.**



## Modificación de objetos:

Modificación de atributos propios **para todos** los objetos de la clase.

- Vemos la estructura para analizar e identificar los objetos a modificar
- Todos los objetos de la clase **Producto tienen un incremento de 10%.**

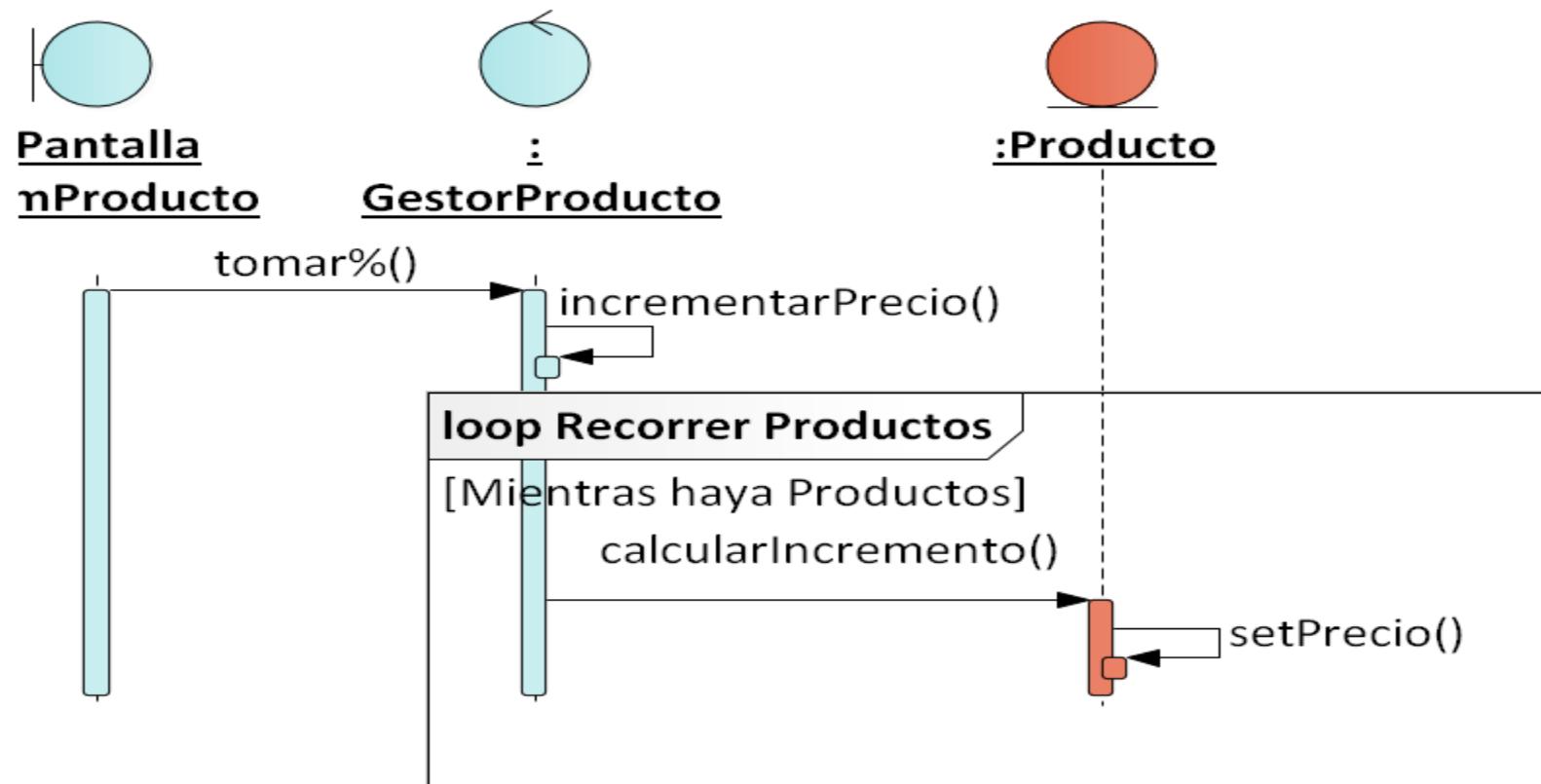


## Modificación de objetos:

Modificación de atributos propios **para todos** los objetos de la clase.

• En la dinámica simplificada primero:

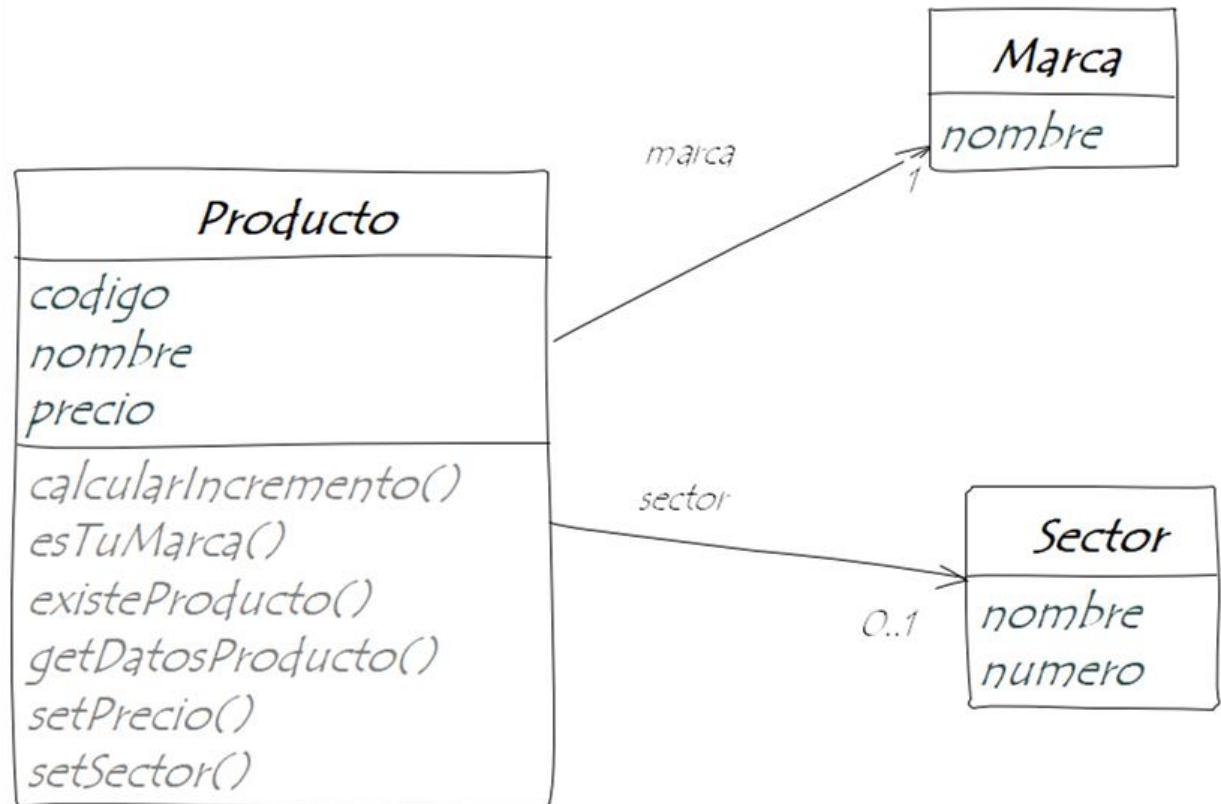
- Se ingresa el % de aumento
- Gestor con su método **incrementarPrecio()** invoca el método **calcularIncremento()** de todos los objetos producto de la clase.
- Luego cada objeto producto se setea su nuevo precio calculado



## Modificación de objetos:

Modificación de atributos propios **para todos** los objetos que cumplen un criterio

- Vemos la estructura para analizarla e identificar los objetos a modificar.
- Objetos de la clase **Producto** que tengan la marca “Sancor”, le vamos a incrementar el precio en un porcentaje.

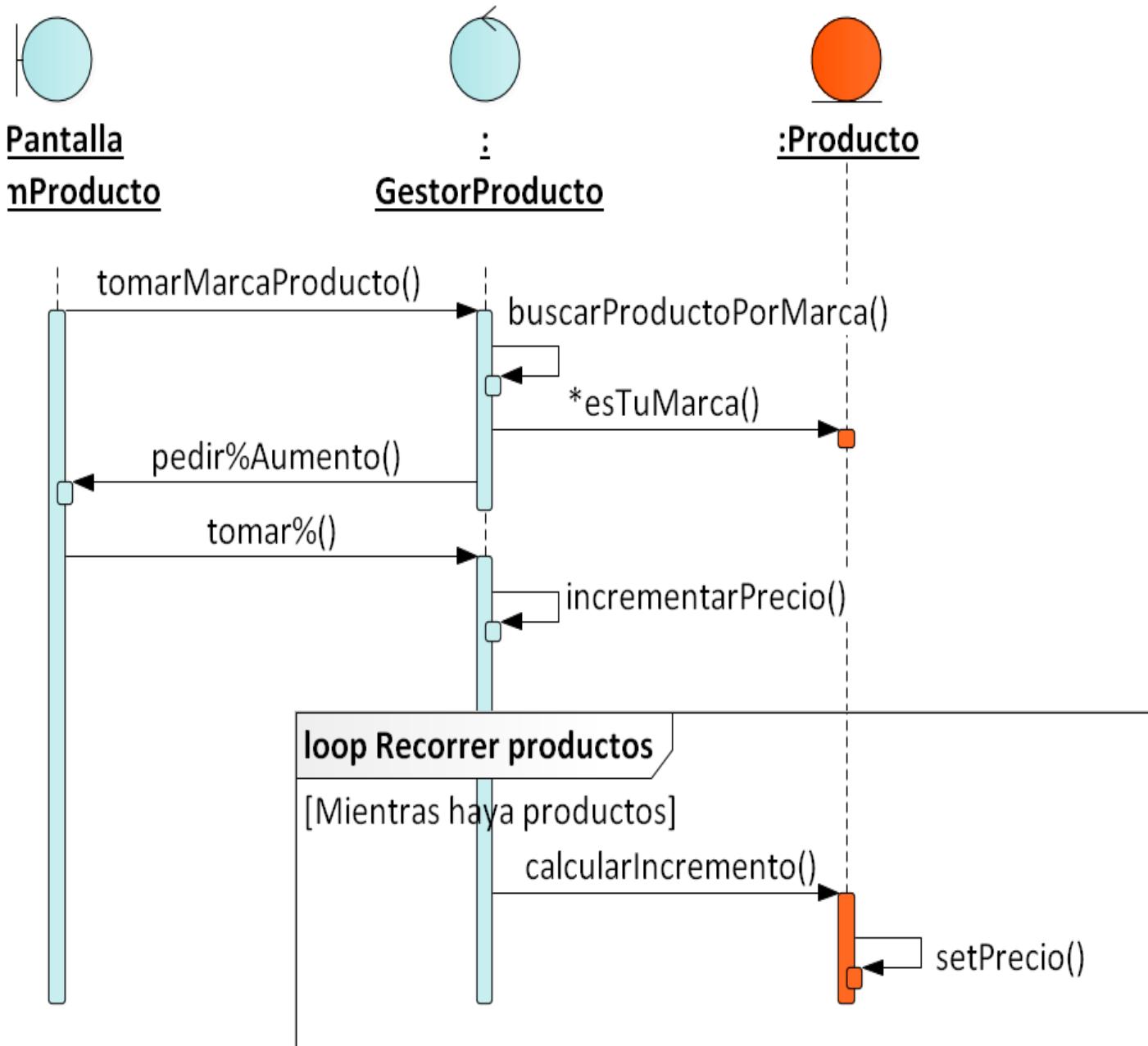


## Modificación de objetos:

Modificación de atributos propios **para todos** los objetos que **cumplen un criterio**

- En la dinámica simplificada primero:

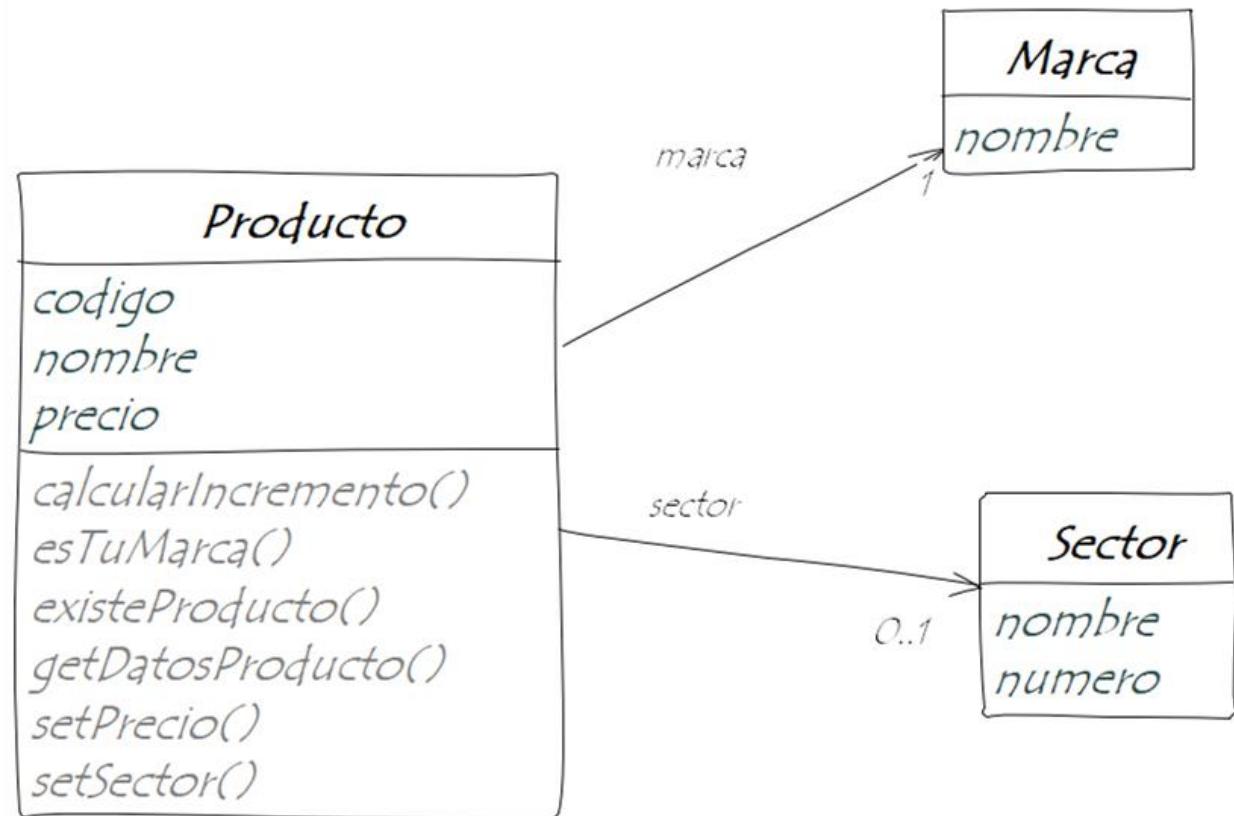
- Se selecciona la marca
  - Gestor busca todos los productos de esa marca, **hacienda una validación con atributos de referencia.**
  - Se ingresa el % de incremento.
  - Gestor invoca el método **calcularIncremento()** de los objetos producto de esa marca
  - Luego cada objeto producto se setea su **nuevo precio calculado**



## Modificación de objetos:

Modificación de un atributo de referencia **para todos** los objetos que **cumplen un criterio**

- Vemos la estructura para analizarla e identificar los objetos a modificar.
- Objetos de la clase **Producto** que **tengan la marca “Sancor”, los vamos a cambiar de sector.**

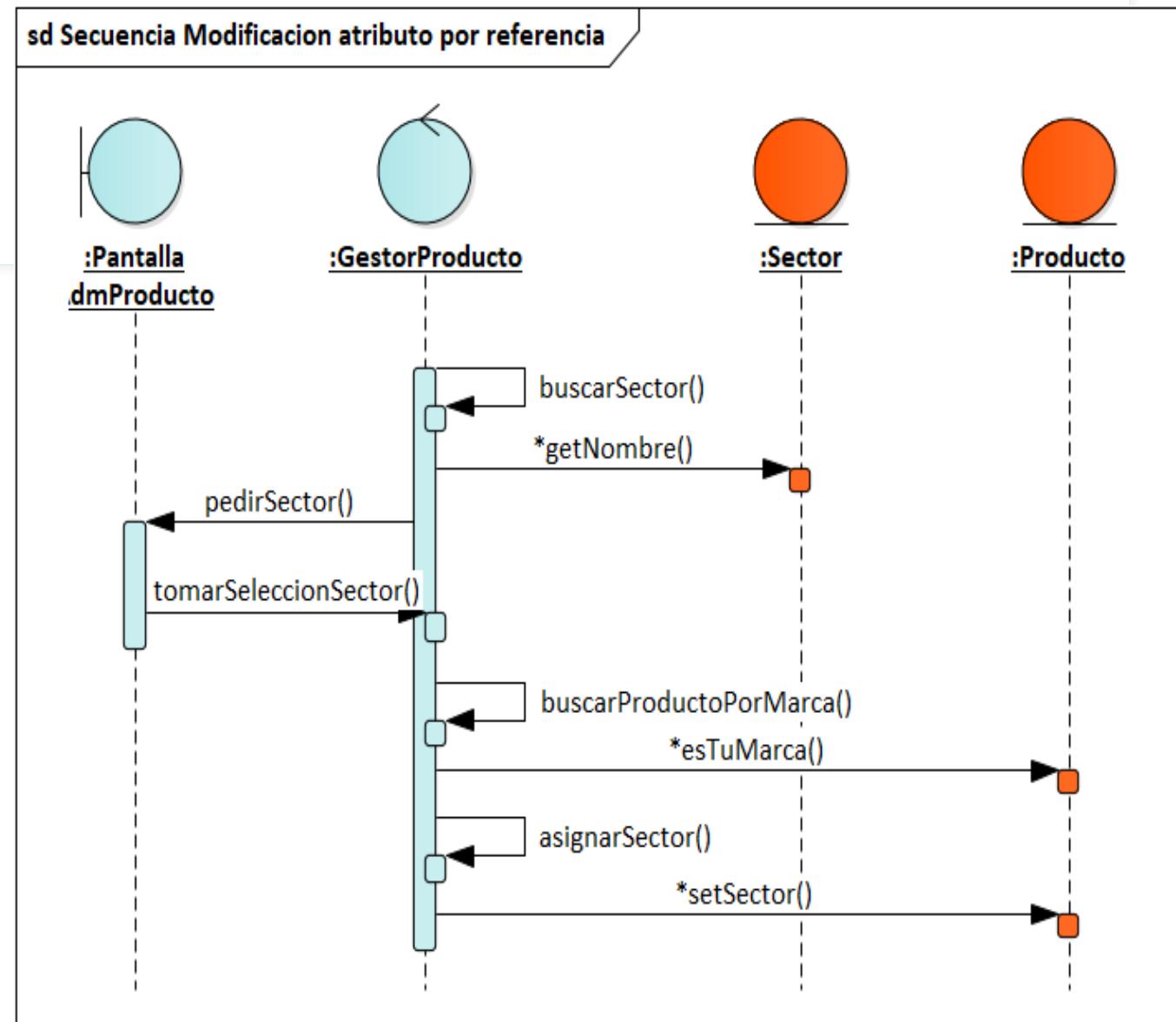


# Modificación de objetos:

Modificación de un atributo de referencia para todos los objetos que cumplen un criterio

- En la dinámica simplificada, primero:

- Gestor busca todos los sectores y los muestra.
- Se selecciona sector destino para los productos.
- Gestor busca los objetos producto de esa marca
- **Luego asigna el sector elegido a los objetos de la marca.**

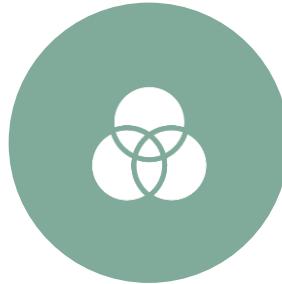


# ¿Qué vimos...?

---



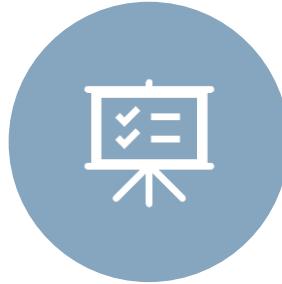
Modificación de atributos propios de **un objeto**



Modificación de atributos propios **para todos** los objetos que **cumplen un criterio**



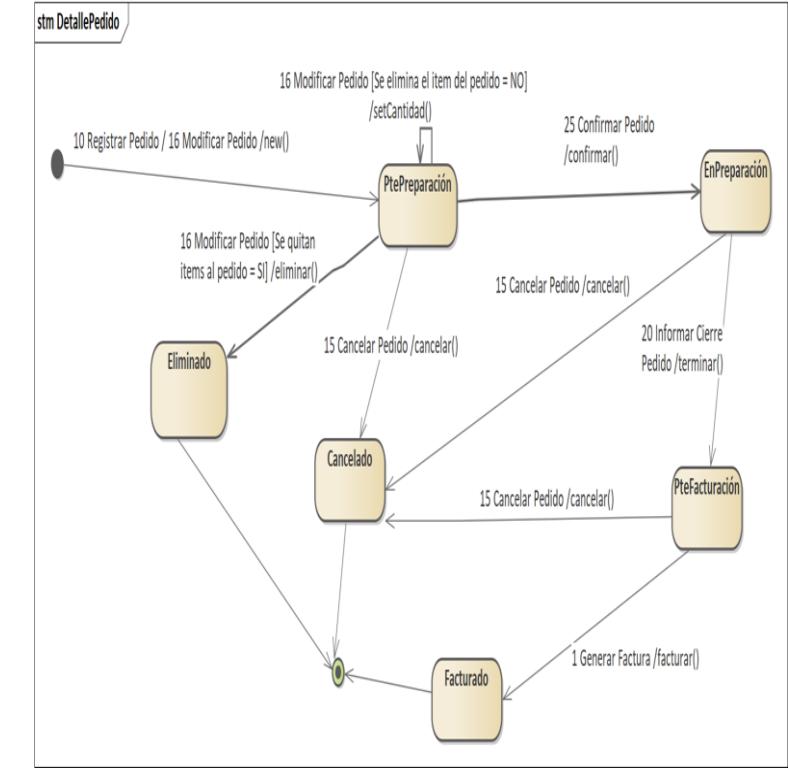
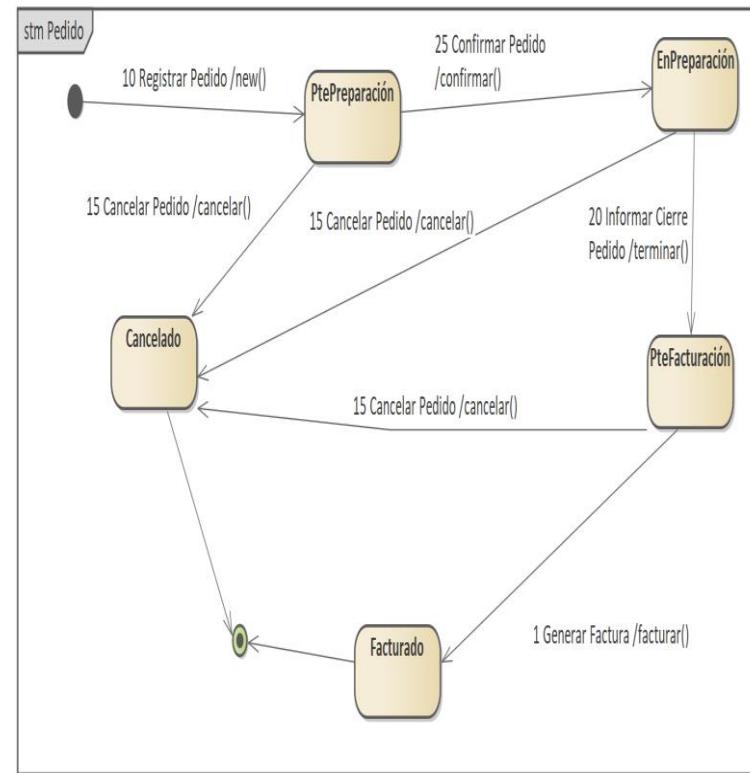
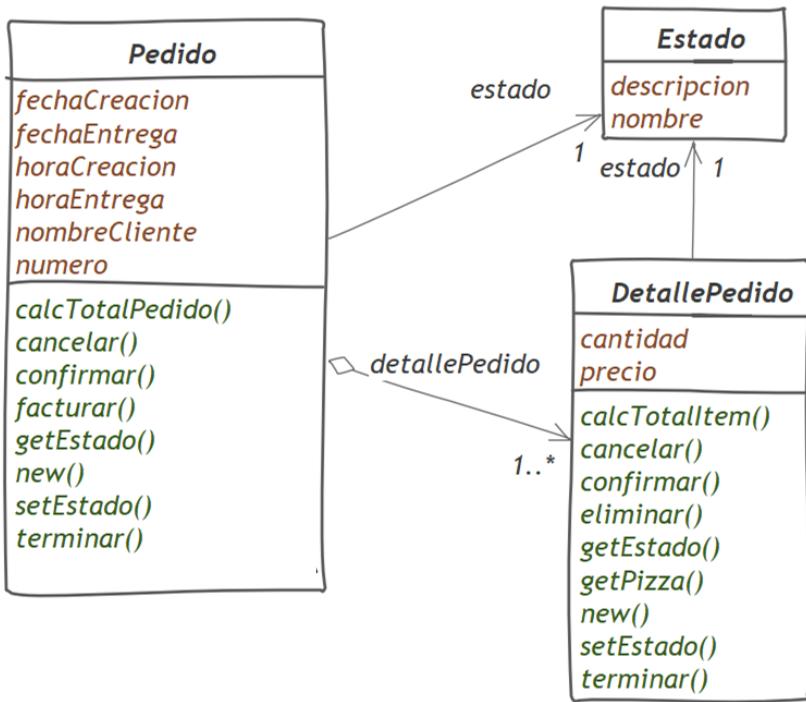
Modificación de atributos propios **para todos** los objetos de la clase.



Modificación de un atributo de referencia **para todos** los objetos que **cumplen un criterio**

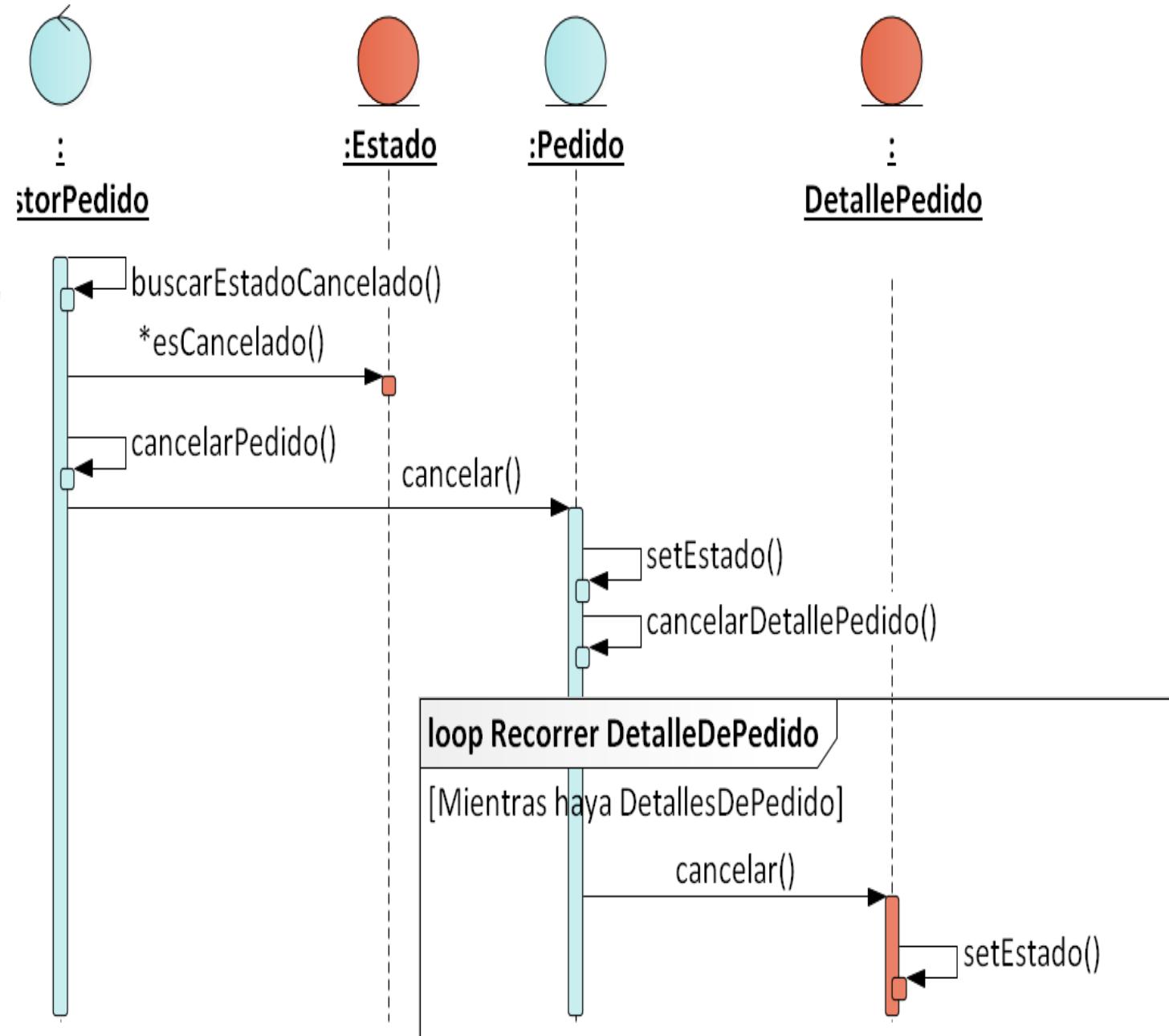
# Manejo de Estados: Actualizar el estado de un objeto

- Por ejemplo: Queremos **cancelar un pedido** y por consiguiente **cancelar sus detalles** de pedido.



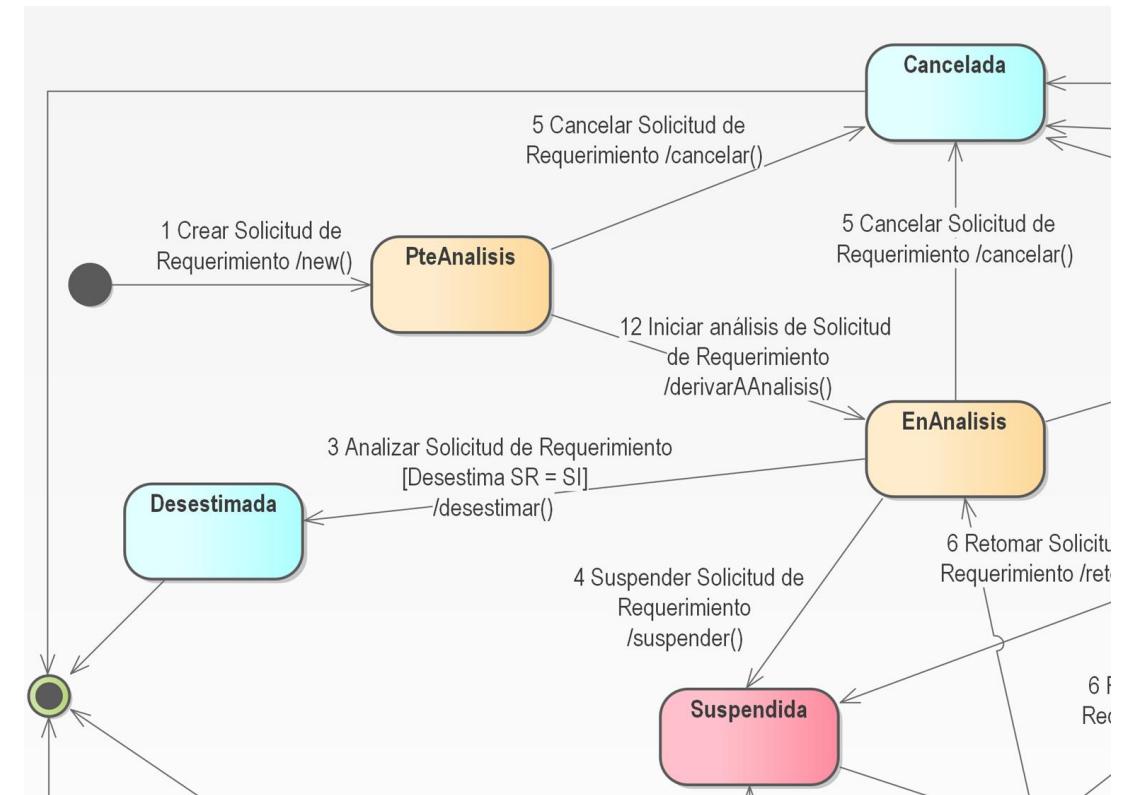
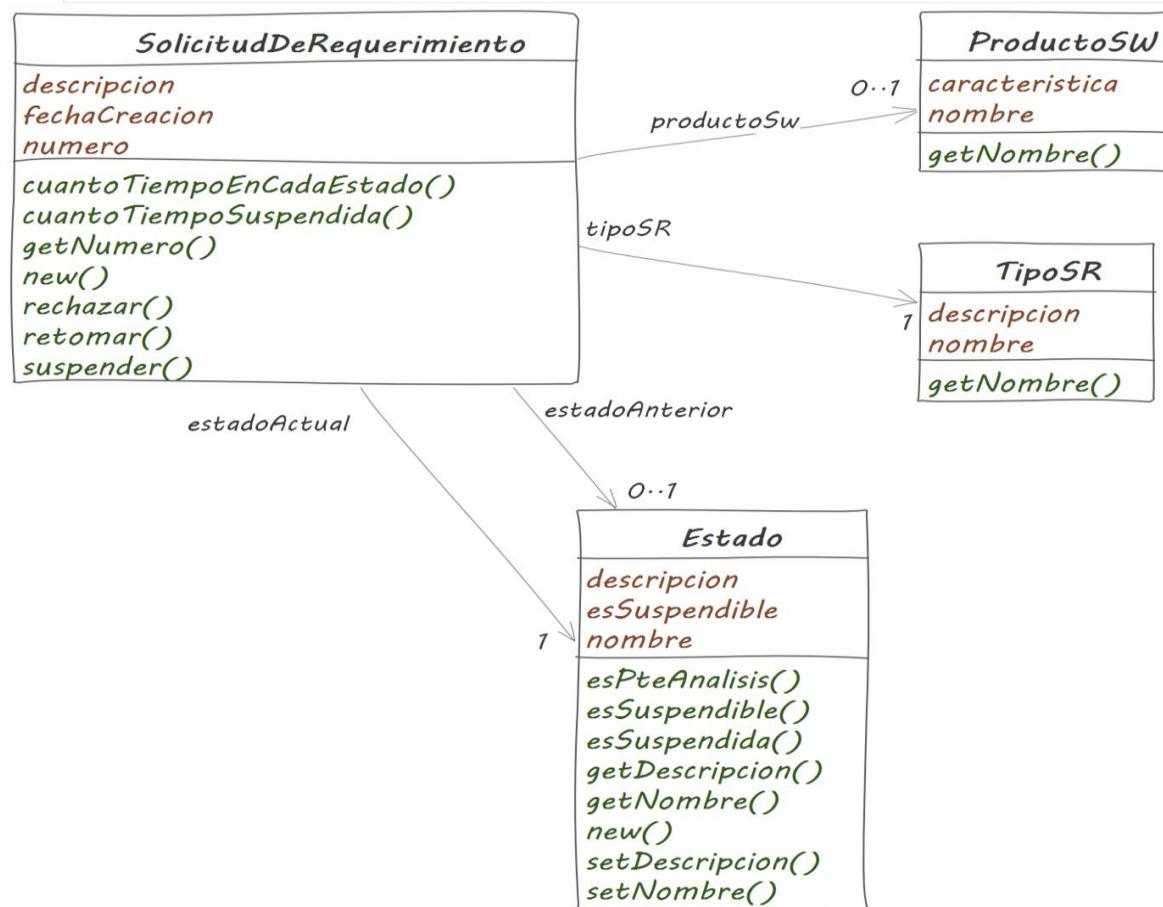
# Manejo de Estados: Actualizar el estado de un objeto

- En la dinámica simplificada, el GestorPedido **busca el estado** que necesita asignar al pedido y sus detalles; en este caso el estado “cancelado”.
- Se usa el método modelado en la transición de correspondiente de la máquina de estados, en este caso **“cancelar()”** con la referencia al objeto cancelado como parámetro del método.
- Luego el pedido cancela del mismo modo sus detalles de pedido.



## Manejo de Estados: Actualizar estado cuando hay estado actual y anterior

- Por ejemplo: Queremos cambiar el estado de la solicitud elegida de **EnAnálisis** a **Suspendida**.



Manejo de Estados: Actualizar  
estado cuando hay estado  
actual y anterior

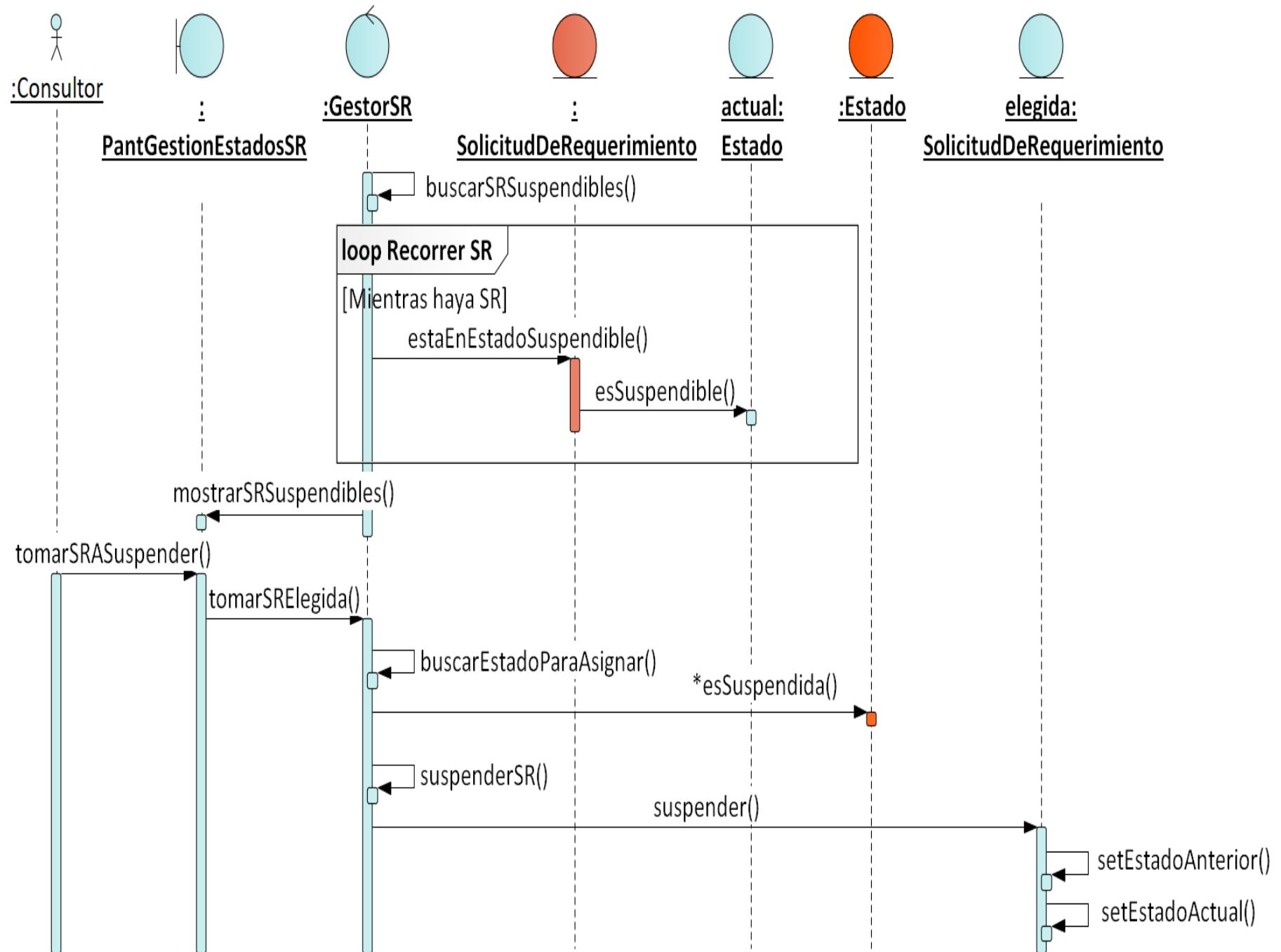
### **ámina simplificada, primero:**

or busca las SR que pueden  
suspendidas y las muestra.

or selecciona la SR que desea  
suspender.

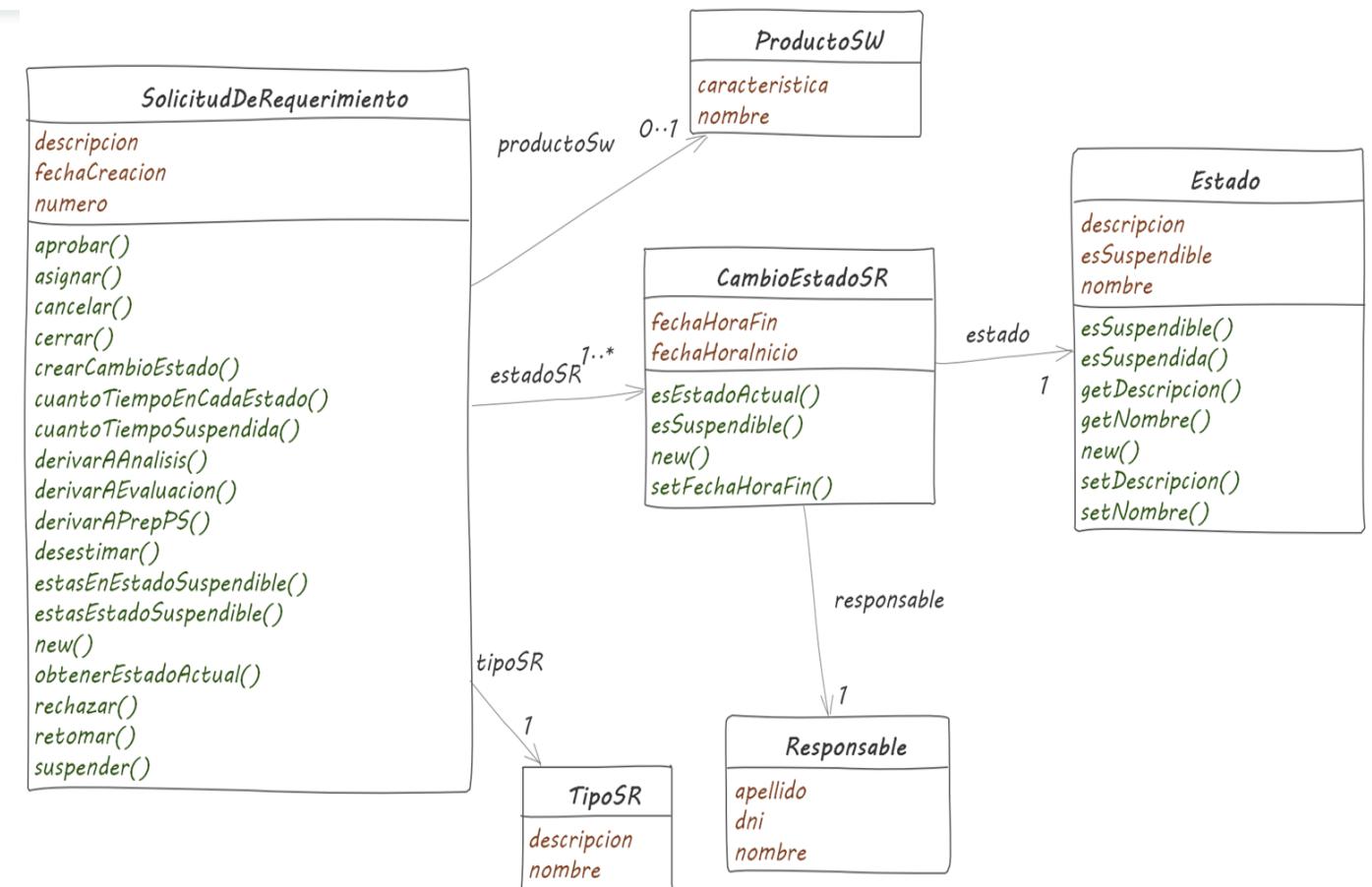
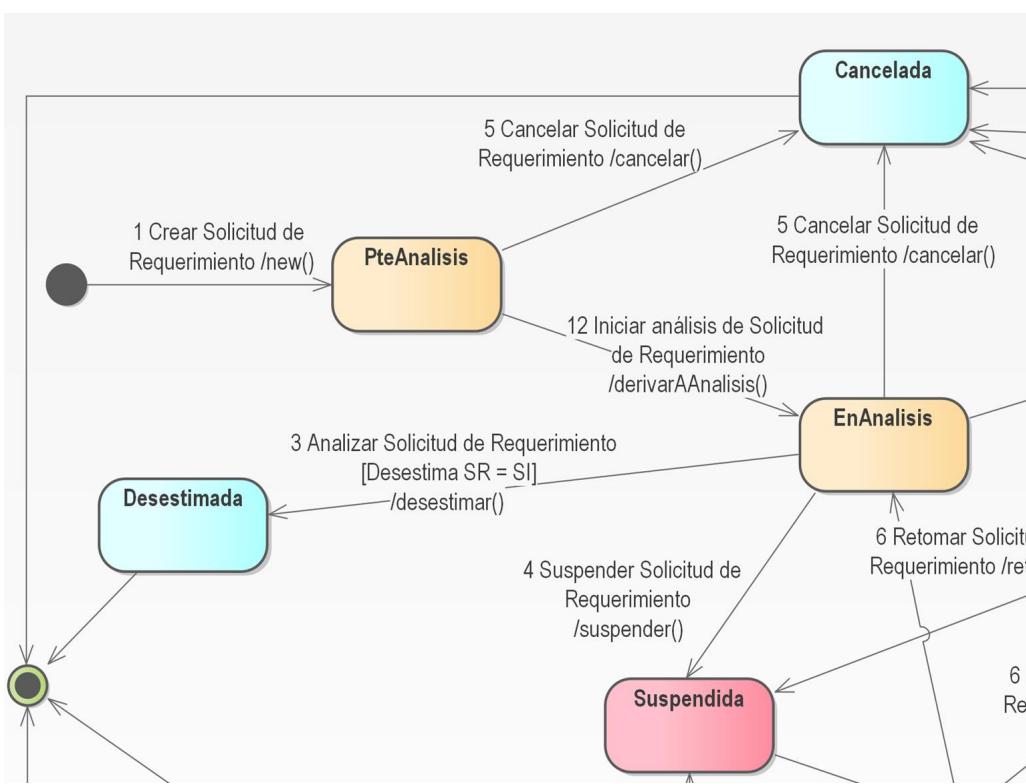
or busca el estado a asignar,  
este caso “Suspendida”.

suspende la SR, lo que implica  
cambiar el estado anterior con el  
estado actual y el estado actual  
“Suspendida”.



# Manejo de Estados: Actualizar estado **cuando** **se mantienen los** **cambios de estado**

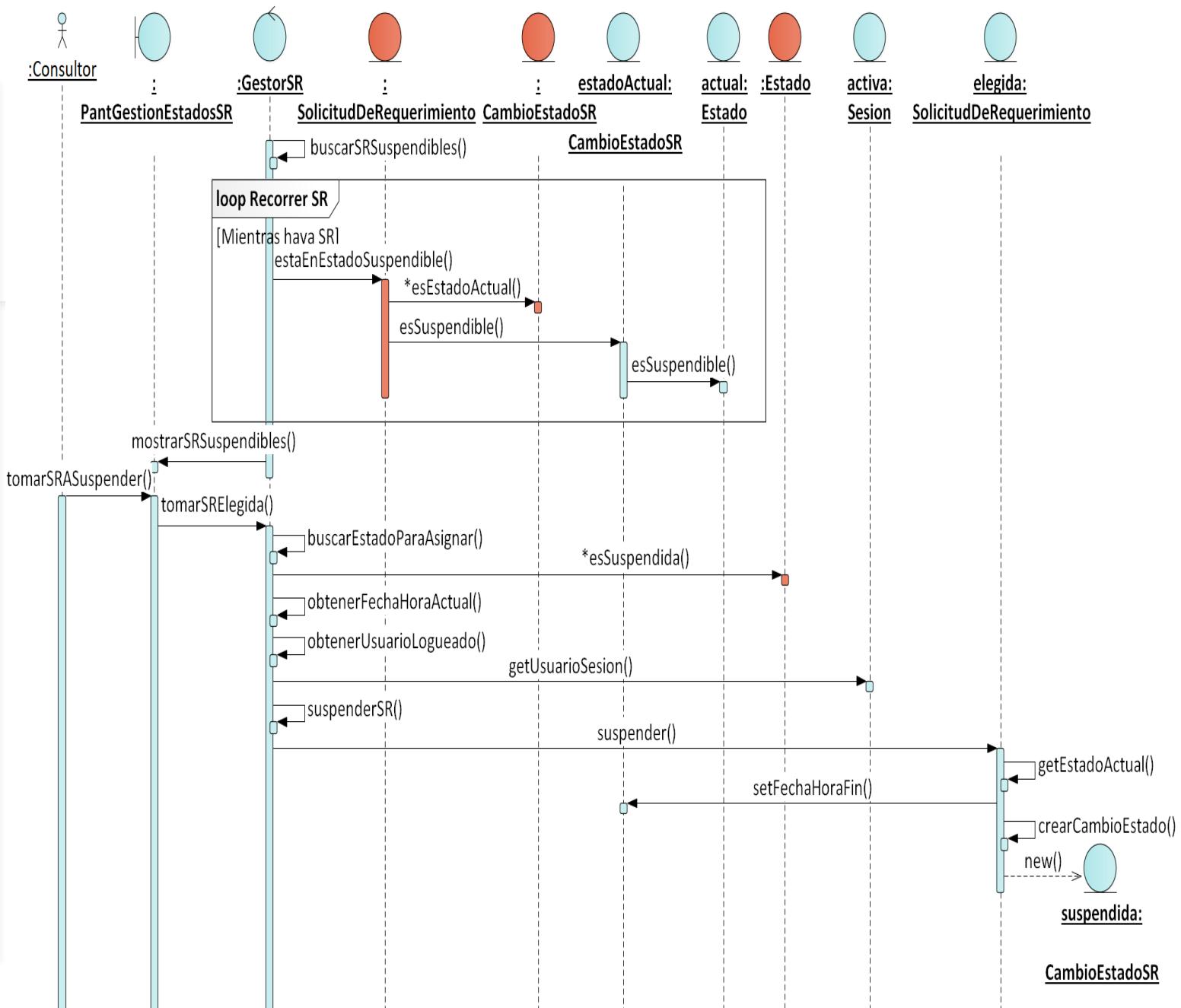
- Por ejemplo: Queremos cambiar el estado de la solicitud elegida de **EnAnálisis** a **Suspendida**.
- Queremos saber quién hizo el cambio de estado y cuándo.



## Manejo de Estados: Actualizar estado cuando se mantienen los cambios de estado

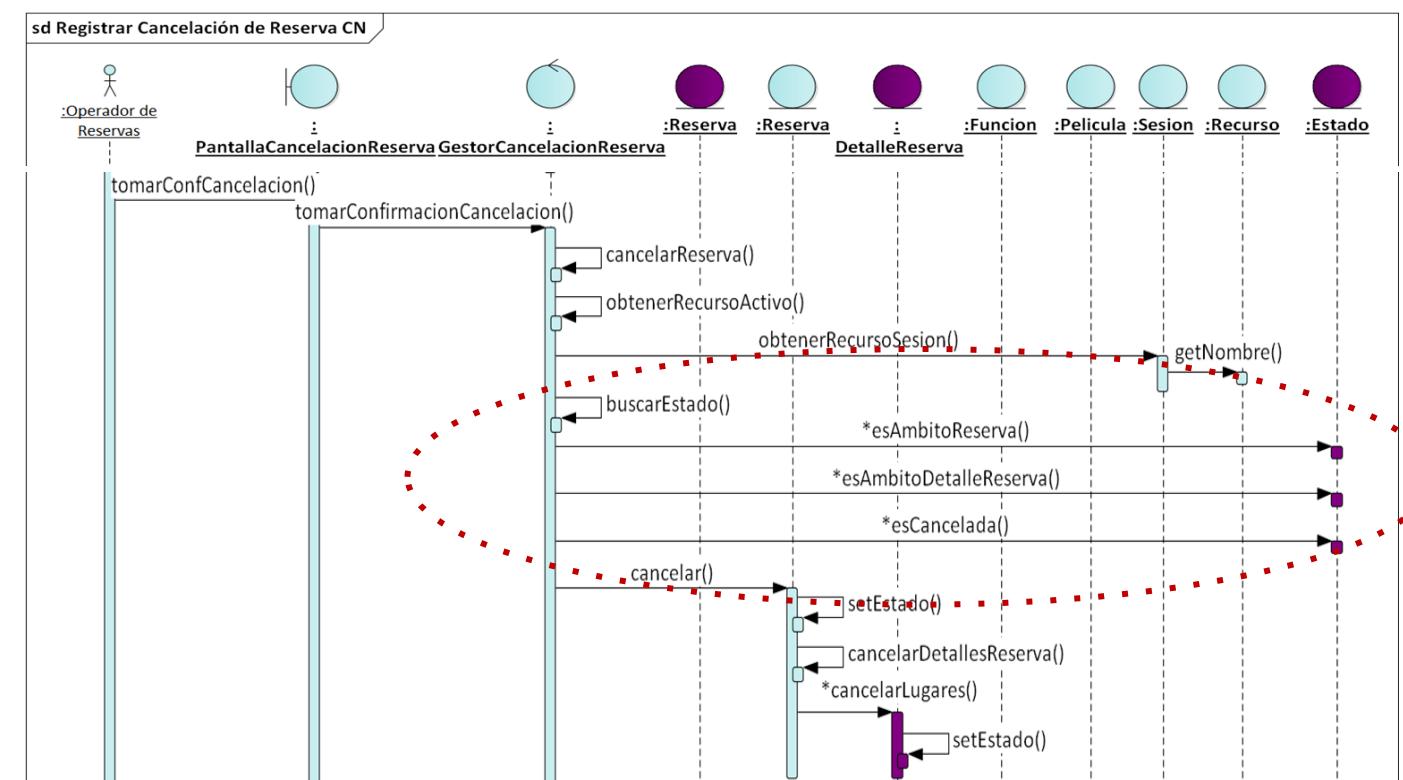
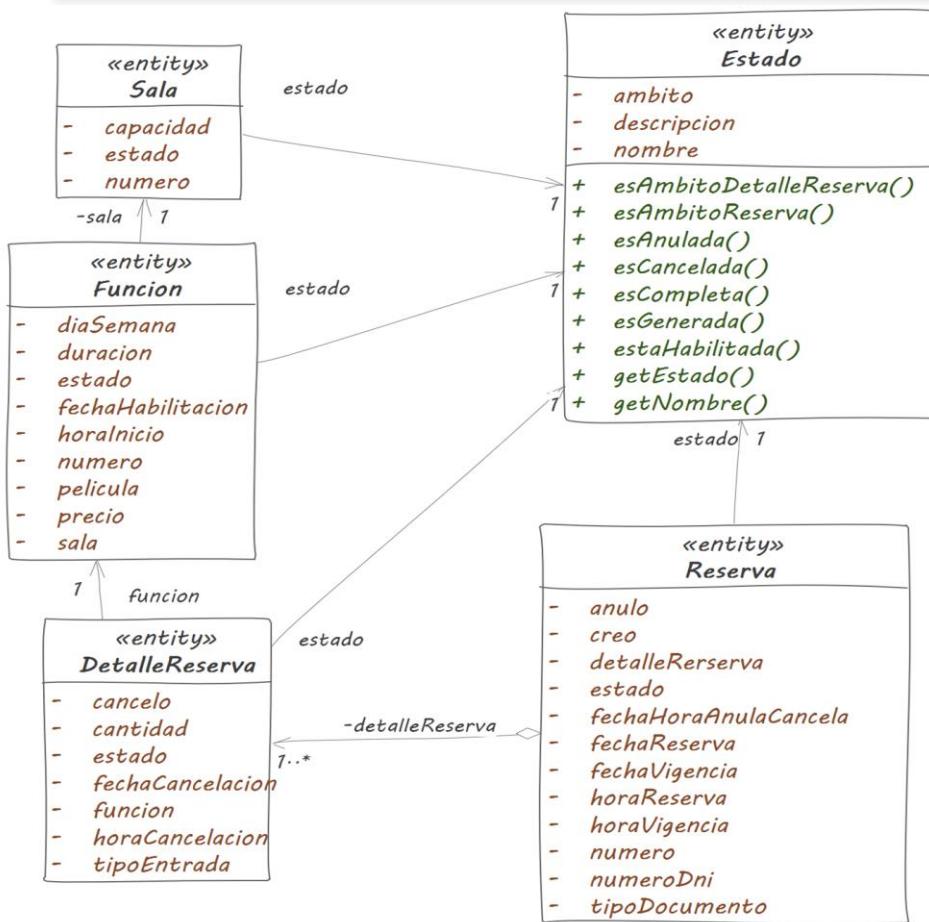
- En la dinámica simplificada, primero:

- Gestor busca las SR que pueden ser suspendidas y las muestra.
- Actor selecciona la SR que desea suspender.
- Gestor busca el estado a asignar, en este caso “Suspendida”.
- Gestor obtiene fecha y hora actual y obtiene el responsable desde la sesión de usuario activa, utilizando el atributo de referencia. En este caso no se requiere pedirle nada al objeto Responsable.
- Gestor suspende la solicitud.
- La SR busca el estado actual para cerrarlo y luego crea el nuevo cambio de estado, en este caso **Suspendida**



# ¿Cómo se modela cuando tenemos un atributo ámbito en la clase Estado?

- Queremos modelar varias clases que tienen comportamiento dependiente del estado utilizando una sola clase Estado
- Una forma es con un atributo **ámbito**



# ¿Qué vimos...?



Manejo de Estados: Actualizar el estado de un objeto



Manejo de estados: Actualizar estado **cuando hay estado actual y anterior** en la estructura.



Manejo de Estados: Actualizar estado **cuando se mantienen los cambios** de estado, (historial de estados).



Manejo de Estados: Cómo se modela con el atributo ámbito dentro de la clase estado.



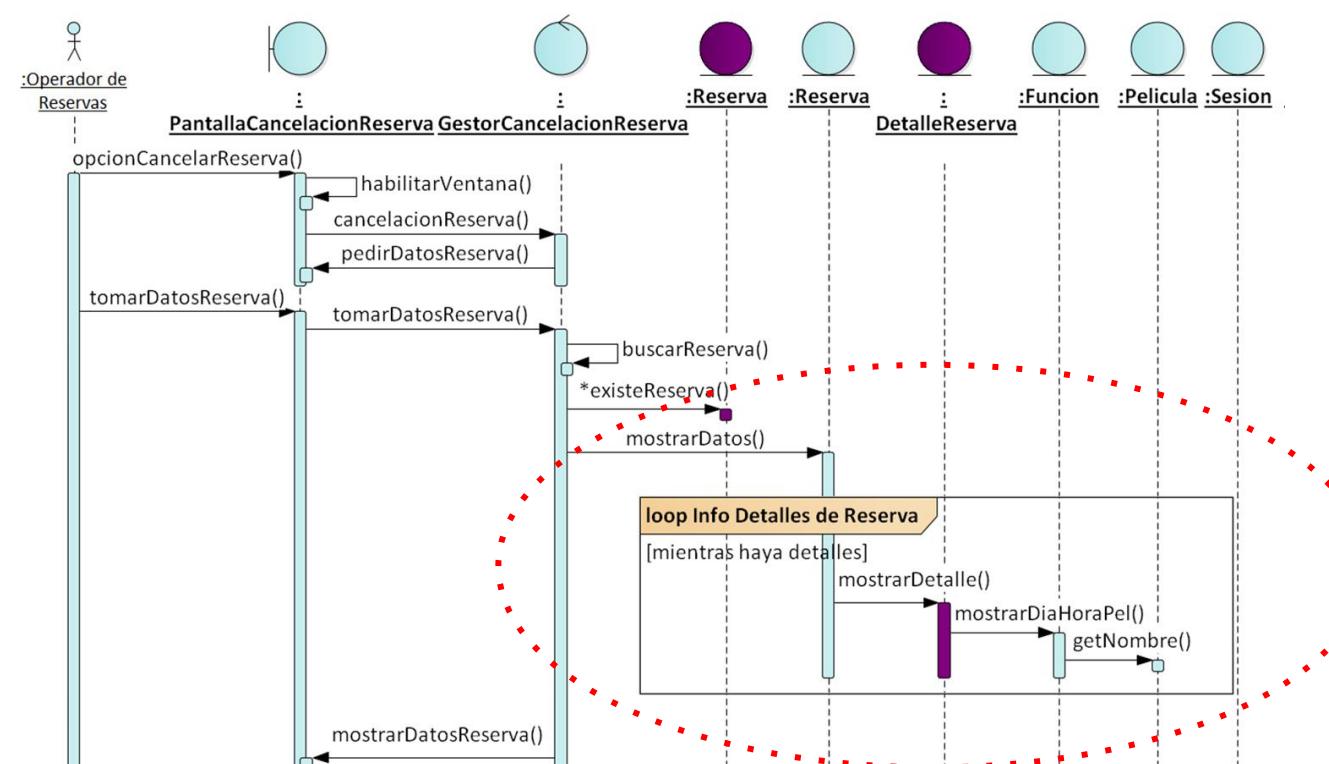
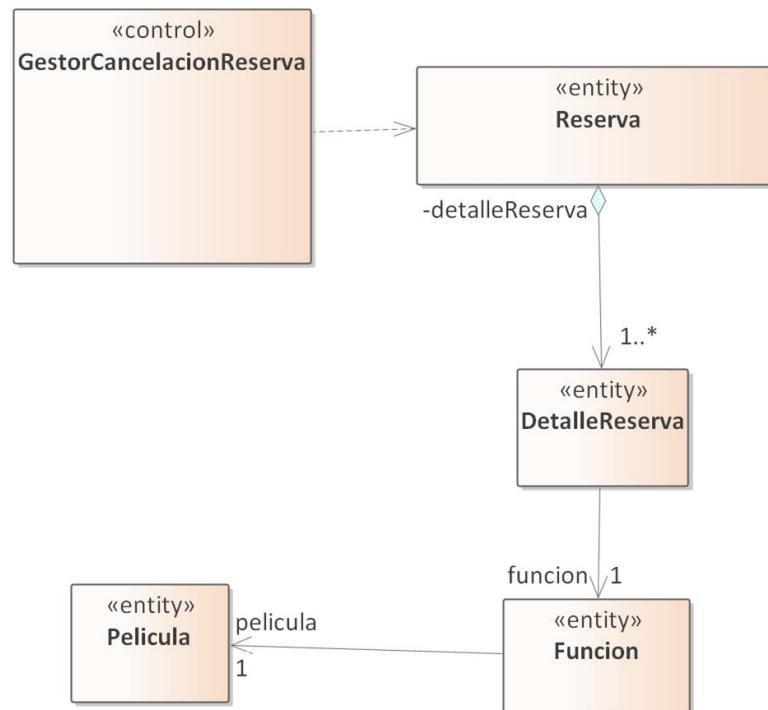
Obtener usuario logueado



Obtener fecha y hora actual del sistema.

# ¿Cómo funciona la colaboración entre los objetos?

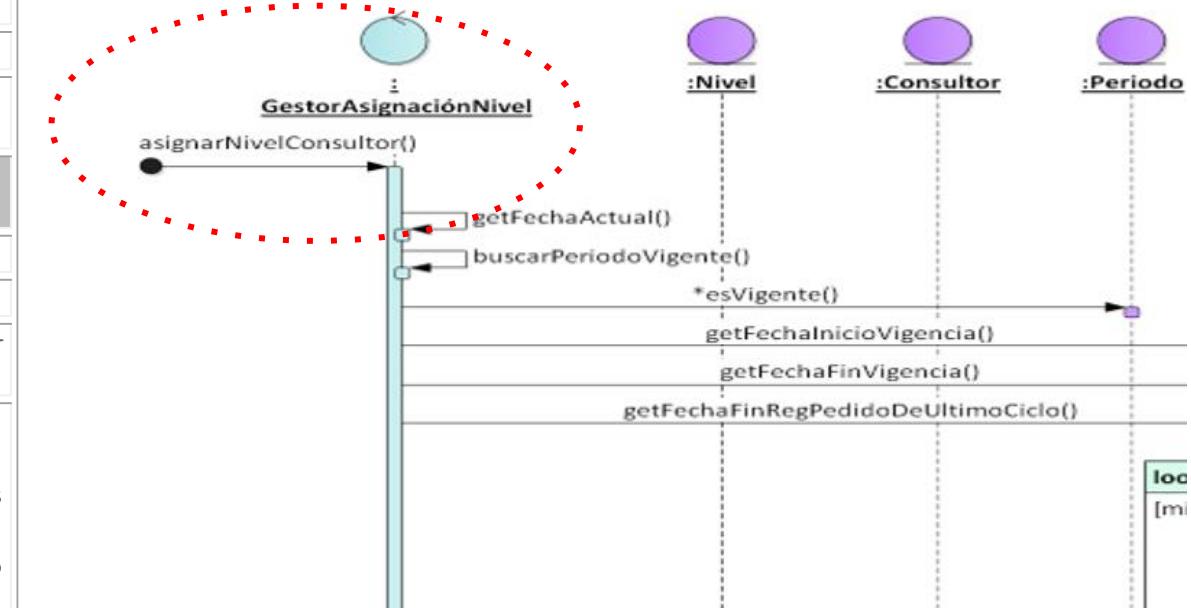
- El Gestor necesita mostrar la información sobre las reservas y pide colaboración, utilizando el patrón experto a quién tiene la información.
- ¿Puedo yo responder a este mensaje solo? Si no, pido colaboración... así hasta llegar al objeto que puede, porque tiene la información.



# ¿Todas las realizaciones de casos de uso tienen un objeto Boundary?

- Queremos modelar un caso de uso abstracto
- No hay actor, no hay objeto boundary

Nombre del Caso de uso: Asignar Nivel a Consultores			Nro. de orden: 14
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Complejidad:	<input type="checkbox"/> Simple	<input checked="" type="checkbox"/> Mediano	<input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo
Actor Principal:	no aplica		Actor Secundario: No aplica
Tipo de Caso de uso:	<input checked="" type="checkbox"/> Concreto		<input type="checkbox"/> Abstracto
Objetivo:	Asignar a cada Consultor el nivel que le corresponde en función de sus pedidos entregados dentro de un período.		
Flujo Descripto:	Procesamiento exitoso con al menos un Consultor que sube de nivel, al menos un consultor que baja de nivel y al menos un consultor que queda en el nivel que estaba inicialmente.		
1.	Sistema: El caso de uso inicia a las 24:00 hs del último día del período vigente.		
2.	Sistema: Busca para todos los niveles el rango de puntajes mínimo y máximo.		
3.	Sistema: Busca para cada Consultor activo todos los pedidos no procesados para el período vigente ( <i>ver observación 2</i> ), y suma el total de los puntos obtenidos en cada pedido pendiente de procesar del período.		
4.	Sistema: Compara para los consultores el puntaje total obtenido con los del nivel del consultor y: <ul style="list-style-type: none"> <li>• Hay al menos un consultor que debe subir el nivel dado que el puntaje es mayor que el máximo de su nivel actual. Para este caso busca el nivel correspondiente al puntaje total obtenido y se lo asigna a esos consultores.</li> <li>• Hay al menos un consultor que permanece en su nivel dado que el puntaje obtenido está dentro del rango de su nivel actual.</li> <li>• Hay al menos un consultor que deben bajar el nivel dado que el puntaje es menor que el mínimo de su nivel actual entonces busca el nivel correspondiente al puntaje total obtenido y se lo asigna a esos consultores.</li> </ul>		
5.	Sistema: Marca los pedidos como procesados. ( <i>Observación 1</i> )		
6.	Sistema: Envía al mail registrado de cada Consultor el aviso del nivel actual asignado. Fin del caso de uso.		



Tipo de Caso de uso:

Concreto

Abstracto

**Objetivo:** Asignar a cada Consultor el nivel que le corresponde en función de sus pedidos entregados dentro de un período.

#### Flujo Descripto: Procesamiento exitoso con al menos un Consultor que sube de nivel.

1. **Sistema:** El caso de uso inicia a las 24:00 hs del último día del período vigente.
2. **Sistema:** Busca y muestra para el período vigente fechas de inicio y de fin de vigencia y el tipo de procesamiento de los puntos y es “**Por puntos Totales con Piso Mínimo**”. (*Leer Observación 2*)
3. **Sistema:** Busca para todos los niveles el rango de puntajes mínimo y máximo.
4. **Sistema:** Busca **para cada Consultor activo** el tipo de notificación por defecto asociada. (*Leer Observación 5*)
5. **Sistema:** Busca **para cada Consultor activo** todos los pedidos entregados no procesados.
6. **Sistema:** Suma el total de puntos obtenidos según el tipo de procesamiento.
7. **Sistema:** Compara para los consultores el puntaje total obtenido con los del nivel del consultor y:
  - Hay al menos un consultor que debe **subir el nivel** dado que el puntaje es mayor que el máximo de su nivel actual. Para este caso busca el nivel correspondiente (inmediato superior) y se lo asigna a esos consultores.
  - Hay al menos un consultor que **permanece en su nivel** dado que el puntaje obtenido está dentro del rango de su nivel actual.
8. **Sistema:** Marca los pedidos como procesados. (*Observación 1*)
9. **Sistema:** Envía a la opción registrada por defecto (mail, sms, WhatsApp) de cada Consultor el aviso del nivel actual asignado. Fin del caso de uso. (*Observación 3 y 4*)

#### Flujos Alternativos

A1: No existen pedidos pendientes de procesar.

A2: El procesamiento de pedidos quedó sin terminar.

A3: Se canceló el proceso.

A4: Forma de procesamiento por “puntos totales”.

A5: Forma de procesamiento por “puntos totales en todos los ciclos”.

**Observación 1:** Se debe indicar que los pedidos fueron procesados para no procesarlos nuevamente.

**Observación 2:** Los tipos de procesamiento de puntos para el camino de crecimiento de los

Consultores actualmente son tres:

- “Por puntos Totales” =  $\Sigma (\sum \text{total de puntos pedido} \times \text{ciclo})$
- “Por puntos Totales con piso mínimo” =  $\Sigma (\text{SI } (\sum \text{total de puntos pedido} \times \text{ciclo}) > \text{MIN (puntos del Nivel Consultor)} \text{ entonces } (\sum (\text{puntos pedido} \times \text{ciclo})) / \text{cantidad de ciclos})$
- “Por puntos en todos los ciclos” =  $\text{SI } (\sum \text{total de puntos pedido} \times \text{ciclo}) > 0 \text{ entonces } \sum (\text{total de puntos pedido} \times \text{ciclo})$

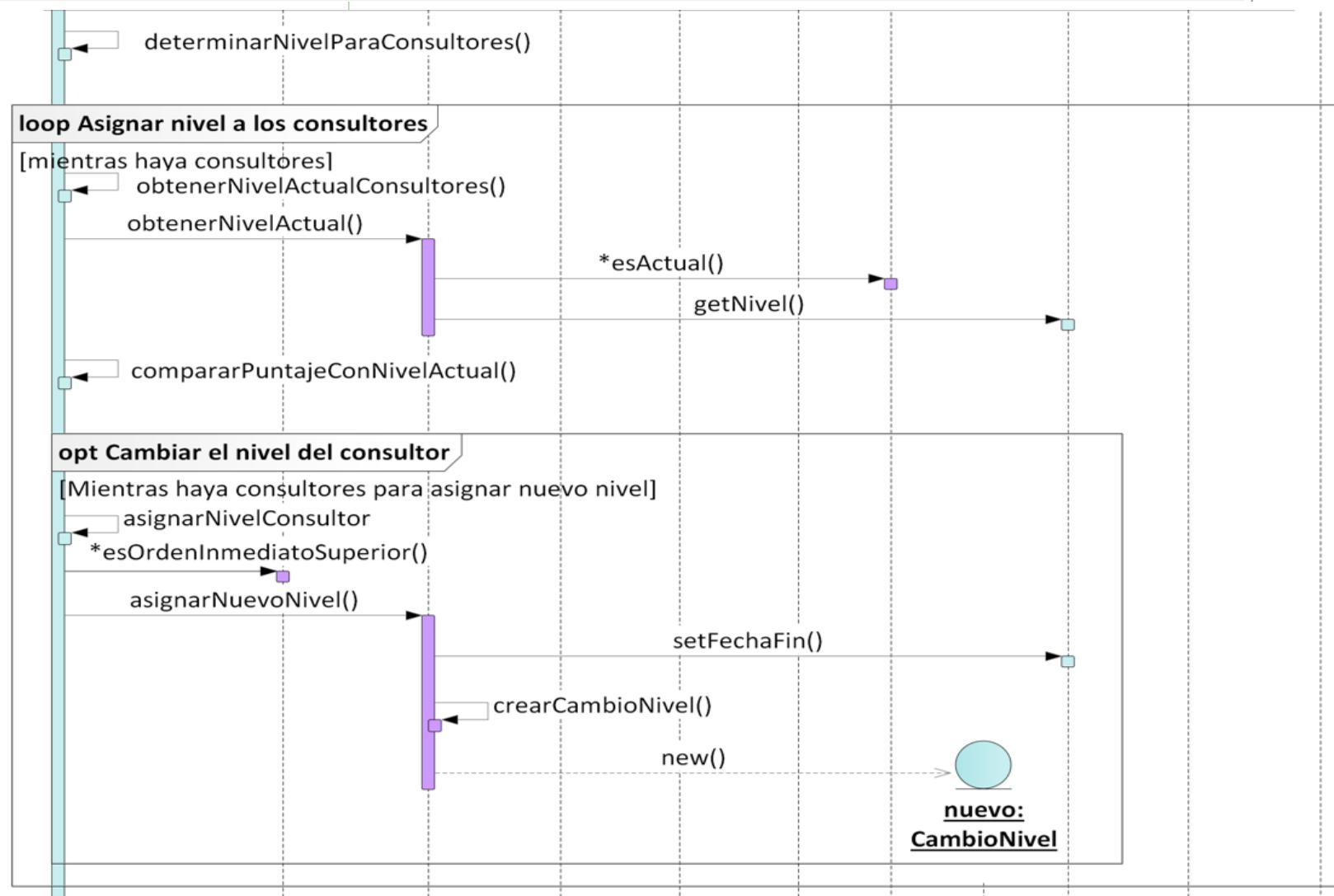
**Observación 3:** Los Consultores al registrarse eligen una opción de notificación por defecto para recibir información.

**Observación 4:** En el aviso se notifica la cantidad de puntos obtenidos como resultado del procesamiento y el nivel actual.

**Observación 5:** Si el tipo de notificación asociada es email, se obtiene el email del Consultor. Si el tipo de notificación es SMS o WhatsApp se obtiene el número de celular del Consultor.

- 7. Sistema: Compara para los consultores el puntaje total obtenido con los del nivel del consultor y:**
- Hay al menos un consultor que debe **subir el nivel** dado que el puntaje es mayor que el máximo de su nivel actual. Para este caso busca el nivel correspondiente (inmediato superior) y se lo asigna a esos consultores.
  - Hay al menos un consultor que **permanece en su nivel** dado que el puntaje obtenido está dentro del rango de su nivel actual.

¿Cómo reflejamos acciones opcionales?

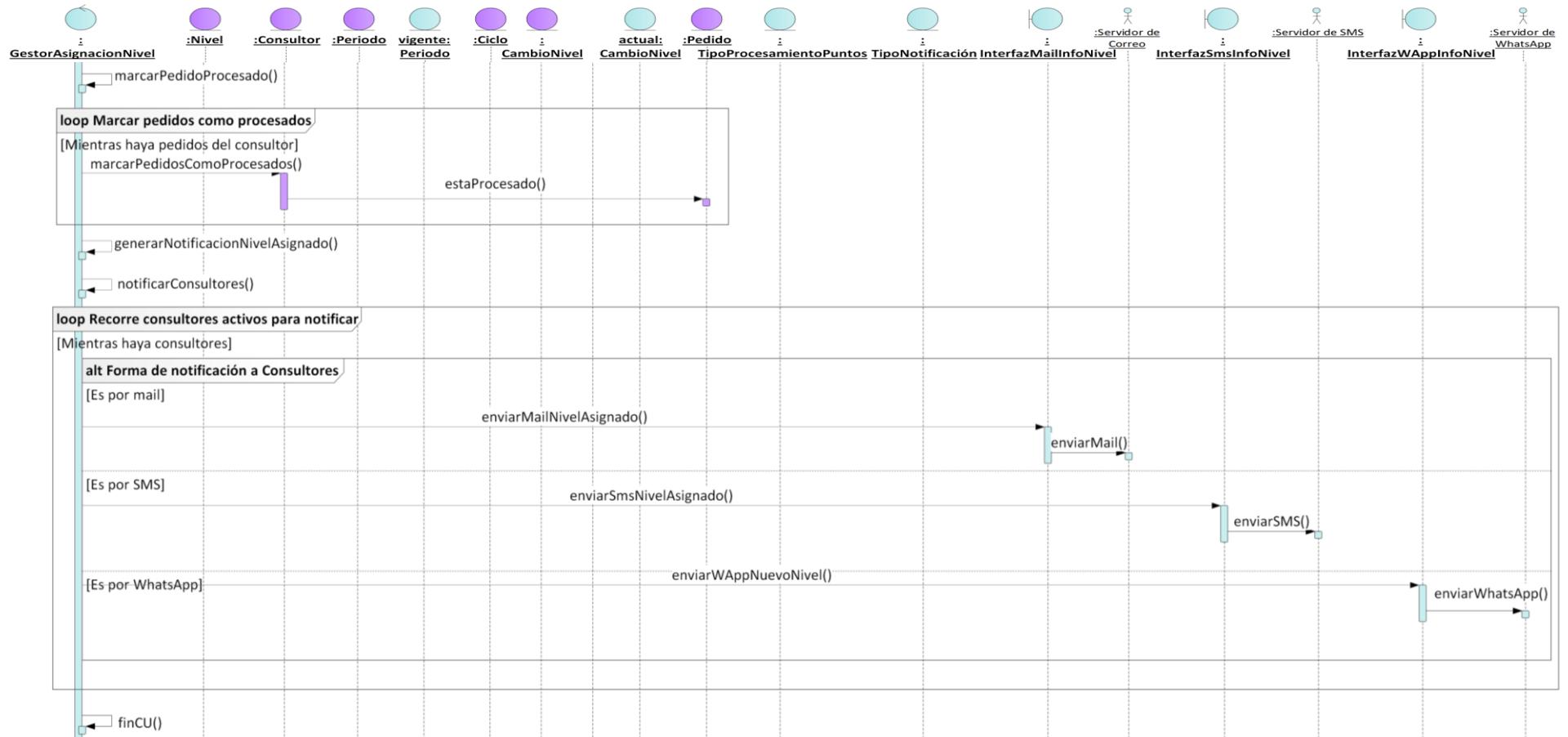


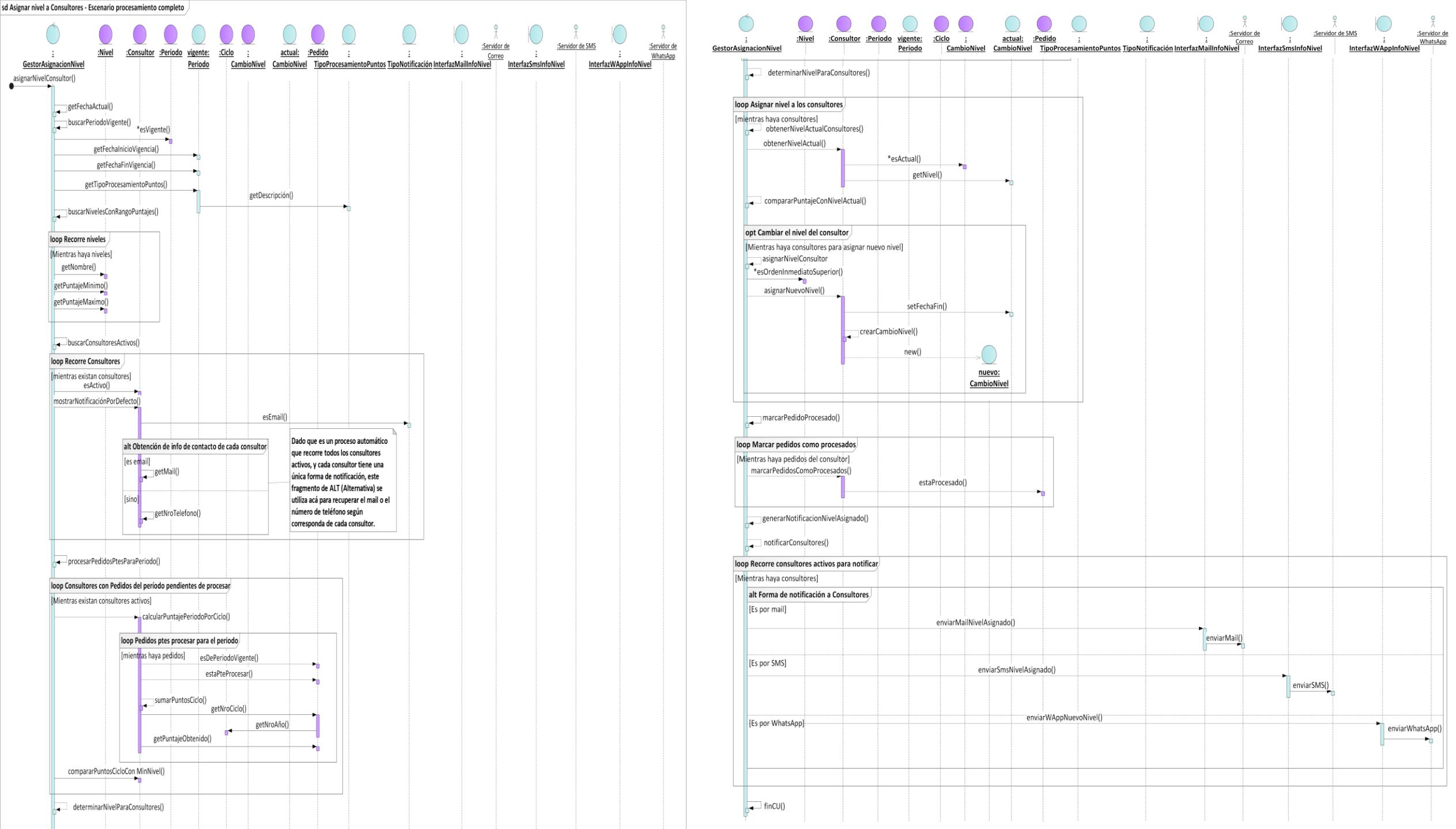
# ¿Cómo reflejamos acciones alternativas?

- **9. Sistema:** Envía a la opción registrada por defecto (mail, sms, WhatsApp) de cada Consultor el aviso del nivel actual asignado. Fin del caso de uso.  
*(Observación 3 y 4)*

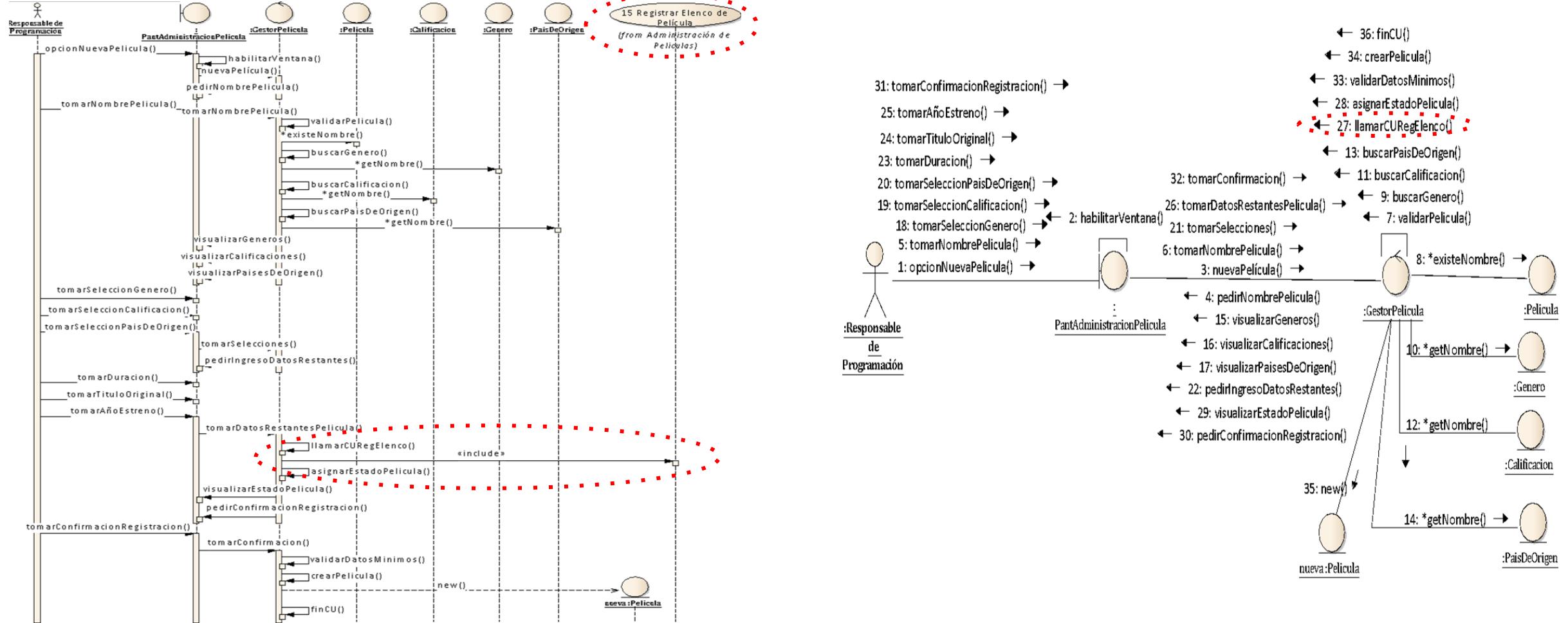
**Observación 3:** Los Consultores al registrarse eligen una opción de notificación por defecto para recibir información.

**Observación 4:** En el aviso se notifica la cantidad de puntos obtenidos como resultado del procesamiento y el nivel actual.





# ¿Cómo se modelan las llamadas a otros casos de uso?



# ¿Qué vimos...?



Cómo funciona la colaboración entre objetos



Cómo se modelan los casos de uso abstractos



Cómo se modelan las llamadas a casos de uso de inclusión y/o extensión con el diagrama de comunicación y de secuencia



Cómo se modelan situaciones alternativas y opcionales con fragmentos en el diagrama de secuencia.



El constructor es un método estático (método de clase)



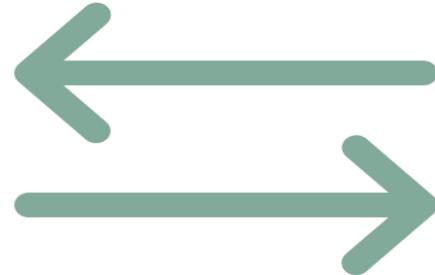
# Las acciones sobre los objetos se resumen a:

## Crear objetos



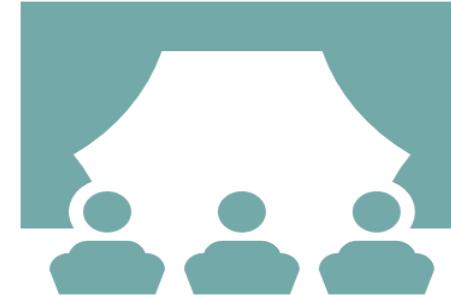
- Tener todos los atributos (por valor o por referencia), necesarios
- Decidir quién crea

## Modificar valores de atributos permitidos



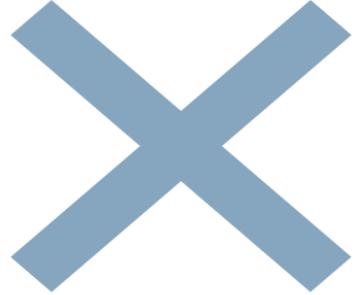
- Localizar el o los objetos a modificar.
- Tener en cuenta las reglas de Negocio.
- Obtener los nuevos valores (puede ser calculándolos o recibiéndolos como parámetros).
- Asignarlos a los atributos correspondientes

## Mostrar valores de atributos



- Saber que queremos mostrar.
- Pedir filtros de consulta si hace falta.
- Iterar para buscar
- Mostrar.
- Decidir quien tiene la responsabilidad

## Eliminar/Anular/Cancelar



- Localizar el o los objetos a eliminar/anular/cancelar.
- Validar las reglas de Negocio.
- Pedir confirmación
- Decidir quién debe hacerlo.
- Eliminación física/lógica ≠ Anular/Cancelar

# Dudas, consultas....



Finalmente...  
Muchas Gracias!!!