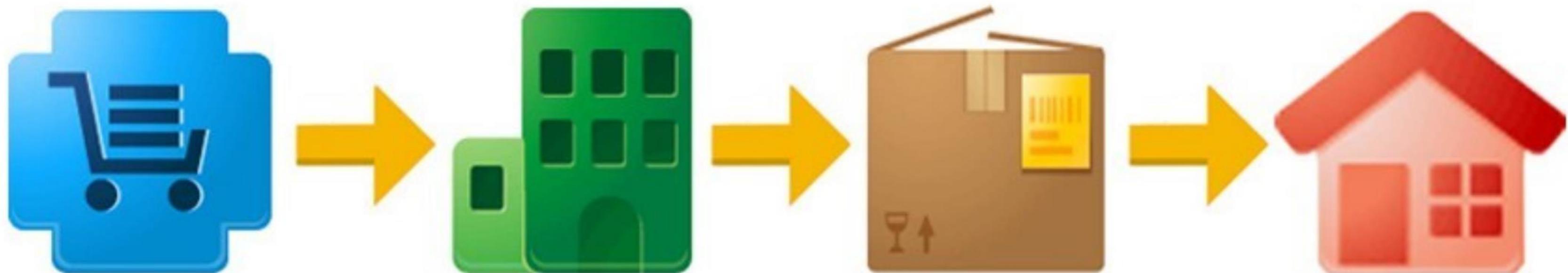
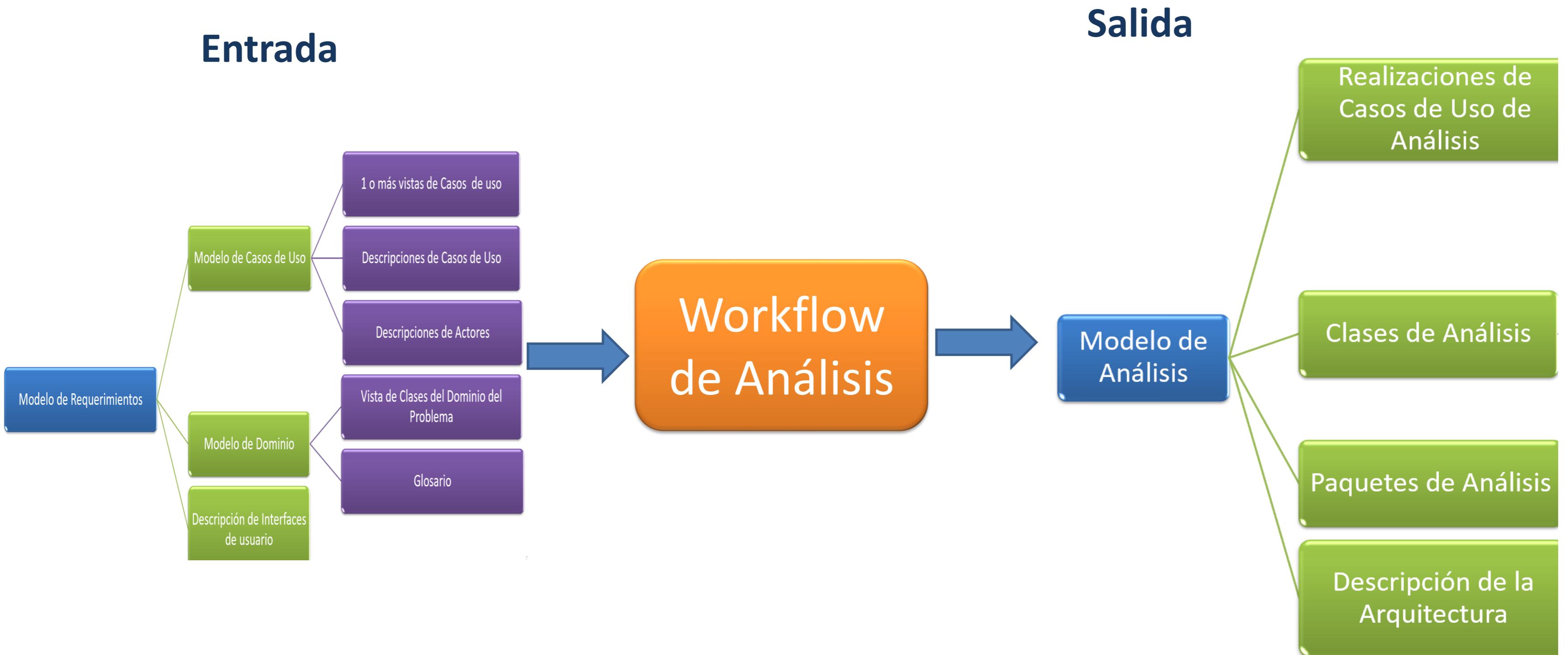


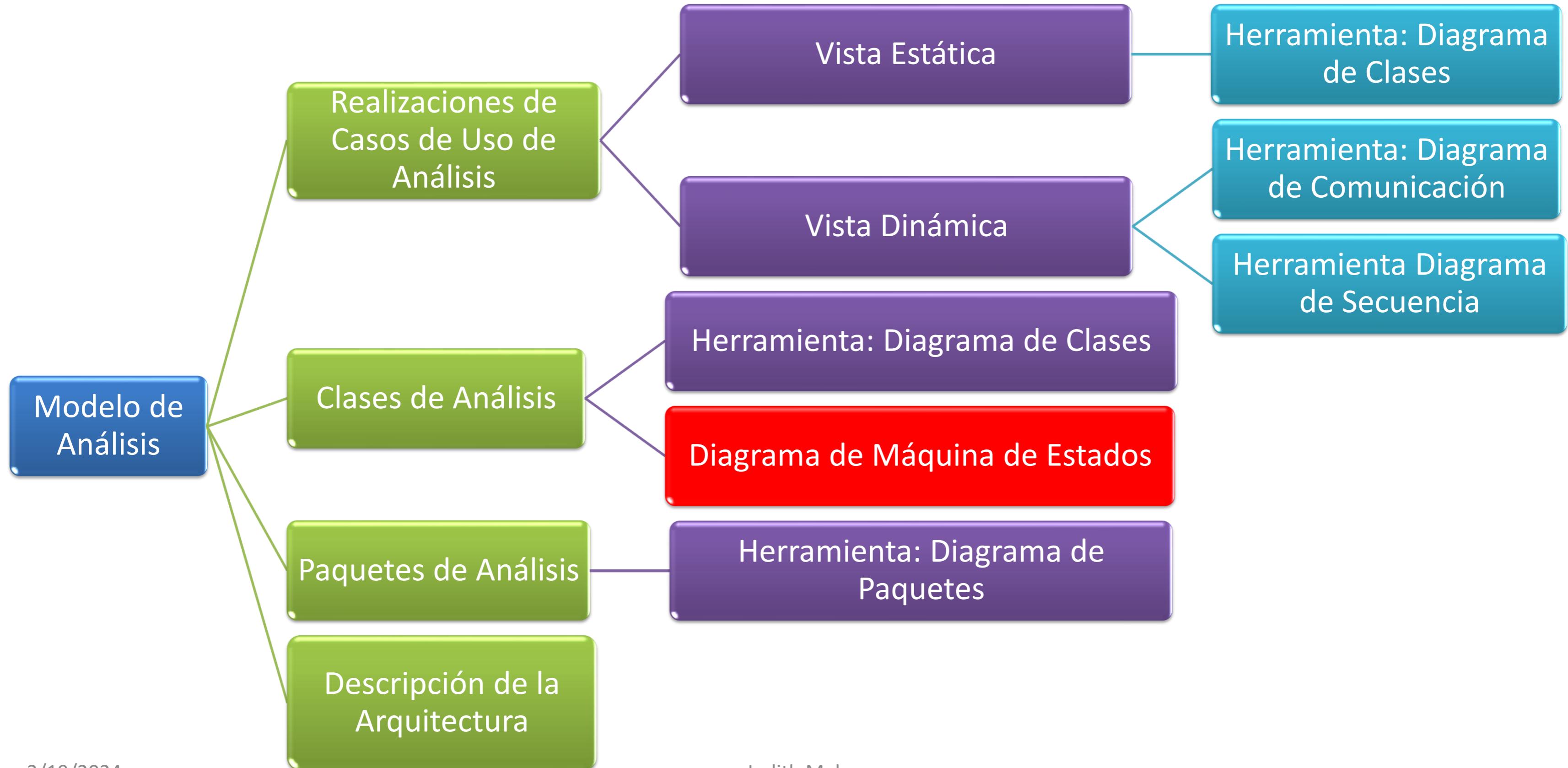
Modelado de Comportamiento con el Diagrama de Máquina de Estados



Workflow de Análisis como proceso...



¿Cómo construimos el Modelo de Análisis?



¿Qué es Diagrama de Máquina de Estados?

Un diagrama de transición de estados muestra una máquina de estados, la cual especifica la secuencia de estados en los que un objeto puede estar, los eventos y condiciones que causan que los objetos alcancen esos estados, y las acciones que ocurren cuando esos estados son alcanzados.



Son usados para modelar el comportamiento dinámico de un elemento de modelado, comúnmente clases cuyos objetos varían el comportamiento en función de su estado.

Diagrama de Máquina de Estados

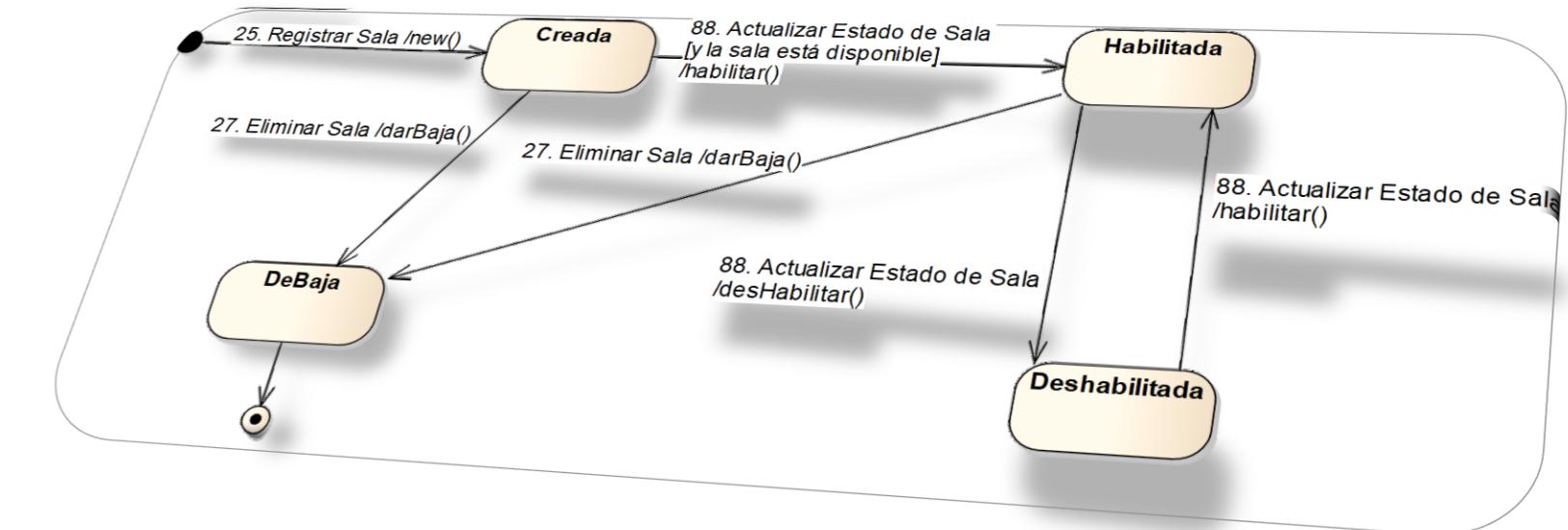
→ **CLASIFICACIÓN:** De comportamiento, dinámico, lógico.

→ **Uso:**

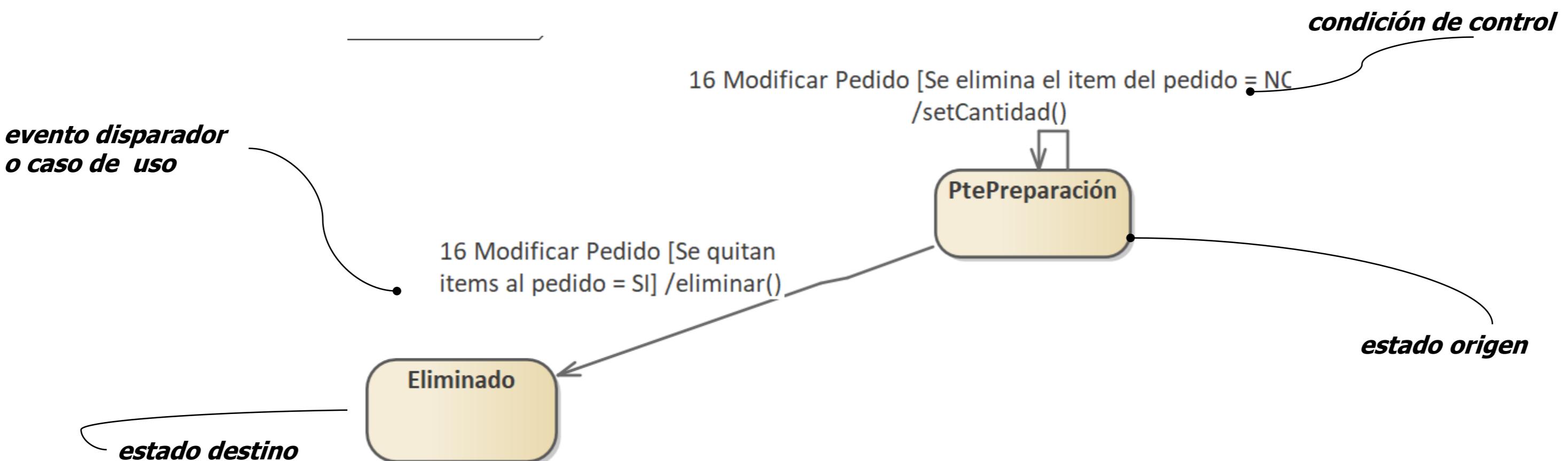
- Explorar el comportamiento de una clase, actor, subsistema o componente.
- Modelar sistemas de tiempo real.

→ **Contiene comúnmente:**

- Estados
- Transiciones
- Eventos



Sobre el Diagrama de Máquina de Estados vimos estos elementos para modelar...



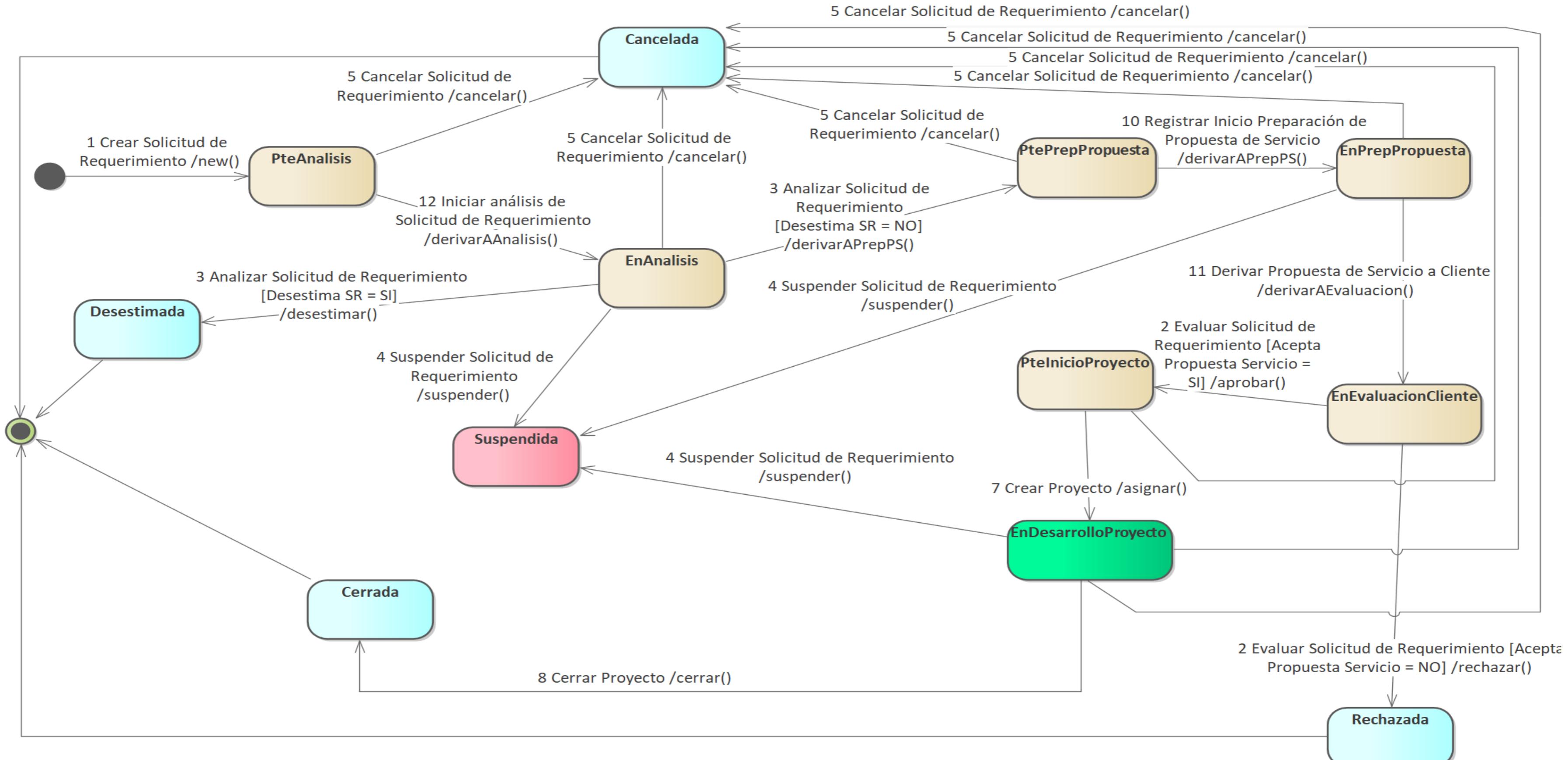
Caso de Estudio: Software para Gestión de Requerimientos



Reglas de Negocio para la Solicitud de Requerimientos

- Las SR las **crea el cliente** ingresando los siguientes datos: Cliente (si no está registrado como cliente, primero deberá registrarse); producto de software asociado (si existe), tipo de SR (selección de uno de los indicados anteriormente) y una descripción de lo que necesita. Al crearse la solicitud se le asigna un número y se registra la fecha y hora de creación.
- **La SR creada por el cliente queda pendiente de análisis por parte de un Analista de Requerimientos** de la empresa que será el responsable de evaluar la pertinencia y factibilidad de lo requerido. De análisis de factibilidad realizado, puede resultar que la **SR sea desestimada, o derivada para que sea preparada una propuesta** que responda a lo pedido por el Cliente.
- Luego, un Consultor asignado **toma la SR y prepara un Propuesta de Servicios y se la envía al cliente** para su consideración
- El **Cliente recibe la propuesta de servicio asociada a la SR y pueda aceptarla o rechazarla.**
- Si el Cliente acepta la propuesta, la SR queda en estado pendiente de inicio. Dado que la empresa necesita un tiempo para armar el equipo de desarrollo realizará el trabajo.
- Para realizar el trabajo, se **inicia un proyecto de desarrollo**, que pasará por diferentes etapas (Requerimientos, Análisis & Diseño, Implementación, Prueba y Despliegue); el proyecto de desarrollo termina con la aceptación del despliegue por parte del Cliente.
- Con la **aceptación del despliegue por parte del cliente se cierra la SR** que dio origen al proyecto de desarrollo de software.
- Independientemente de las razones y los aspectos contractuales, un **Cliente puede cancelar la SR en cualquier momento anterior a la aceptación del despliegue del producto de software.**
- También debe contemplarse la situación de **interrupción del trabajo en cualquiera de las etapas en las que personal de la empresa esté dedicando tiempo a una SR**, para que el tiempo no sea computado como esfuerzo asociado. Cuando se **retoma el trabajo, la SR vuelve al estado en el que estaba al momento de la interrupción**; a partir de ahí seguirá con la evolución correspondiente a ese estado.

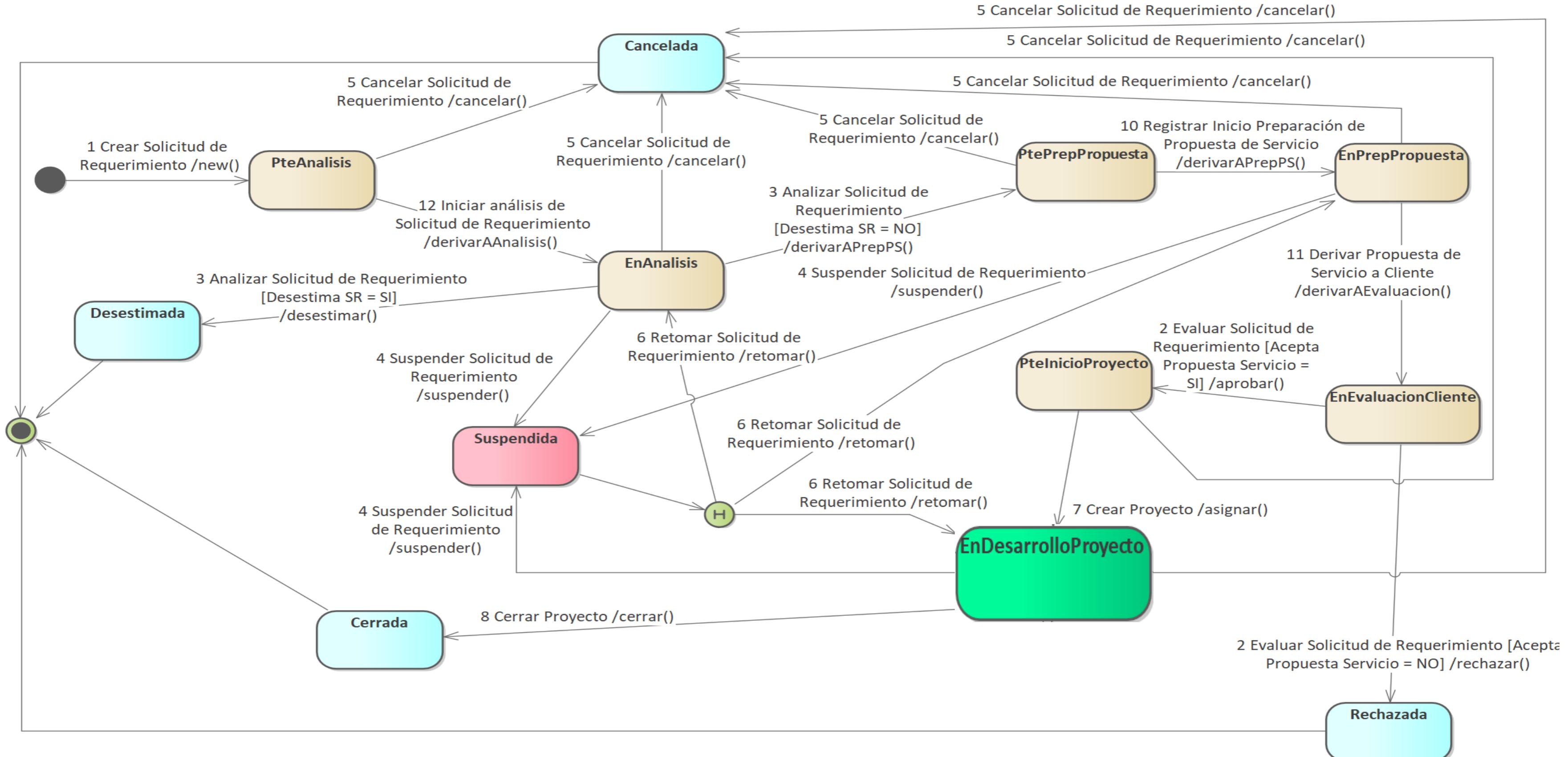
Vamos cómo se vé la máquina de estados de la clase SolicitudDeRequerimientos, con lo que sabemos...



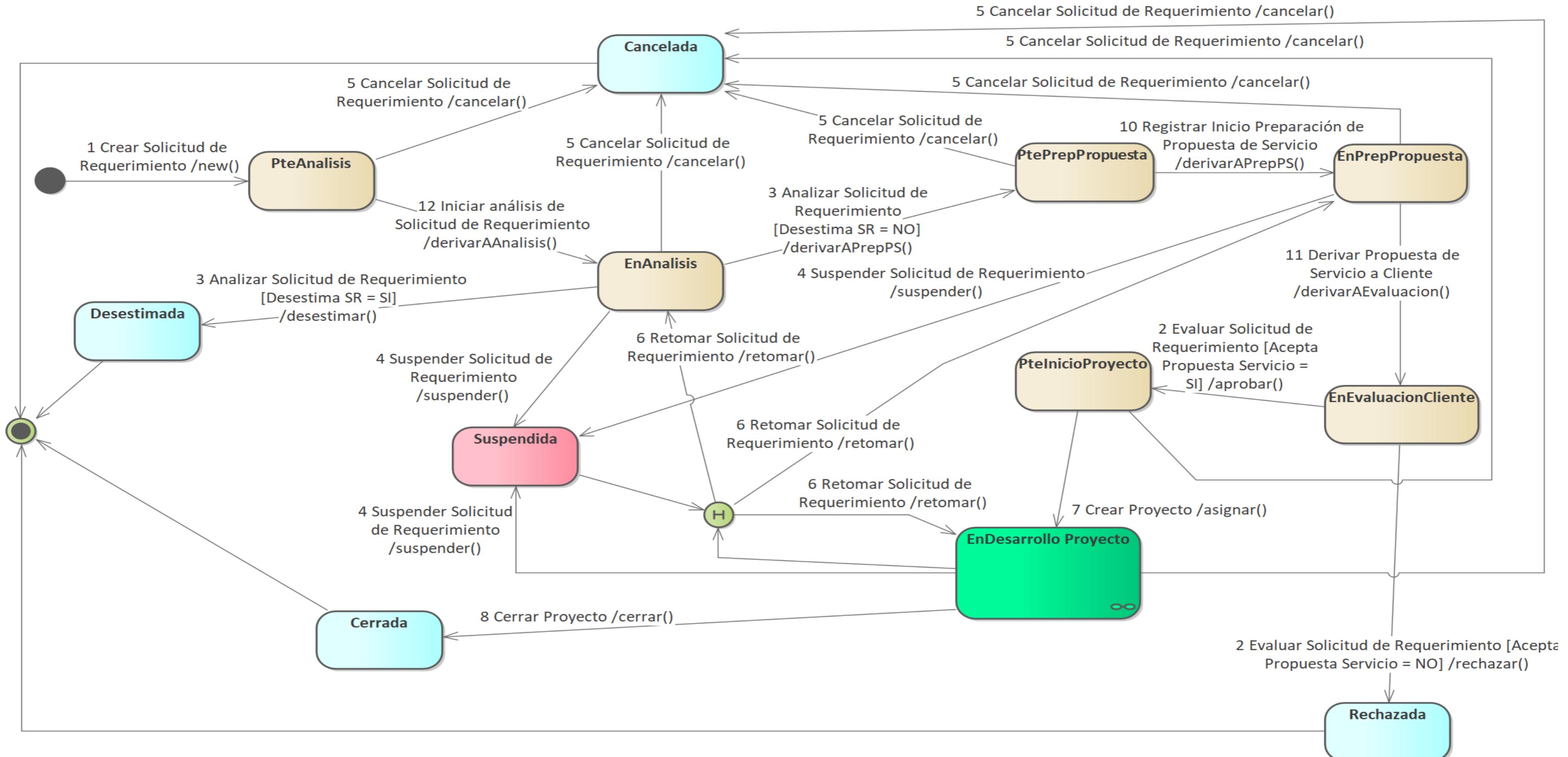
Reglas de Negocio para la Solicitud de Requerimientos

- Para realizar el trabajo, se inicia un **proyecto de desarrollo**, que pasará por diferentes etapas (**Requerimientos, Análisis & Diseño, Implementación, Prueba y Despliegue**); el proyecto de desarrollo termina con la aceptación del despliegue por parte del Cliente.
- Con la aceptación del **despliegue por parte del cliente se cierra la SR que dio origen al proyecto de desarrollo de software**.
- Es necesario poder informar a los clientes del **estado de la solicitud de requerimientos en todo momento y del avance del proyecto de desarrollo asociado a la SR**.
- También es necesario poder informar **cuánto tiempo permanece una solicitud en cada estado y quién es el responsable de cada intervención** (cambio de estado).

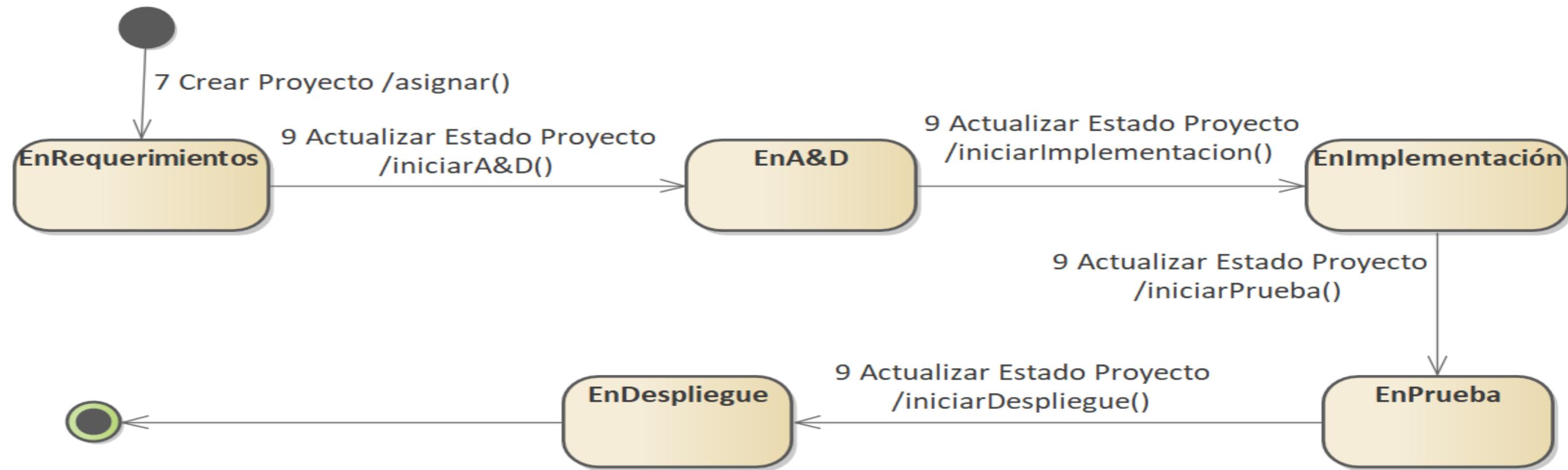
**Veamos cómo se vé la máquina de estados de la clase
SolicitudDeRequerimientos, ahora...**



Vamos cómo se vé la máquina de estados de la clase SolicitudDeRequerimientos, ahora...

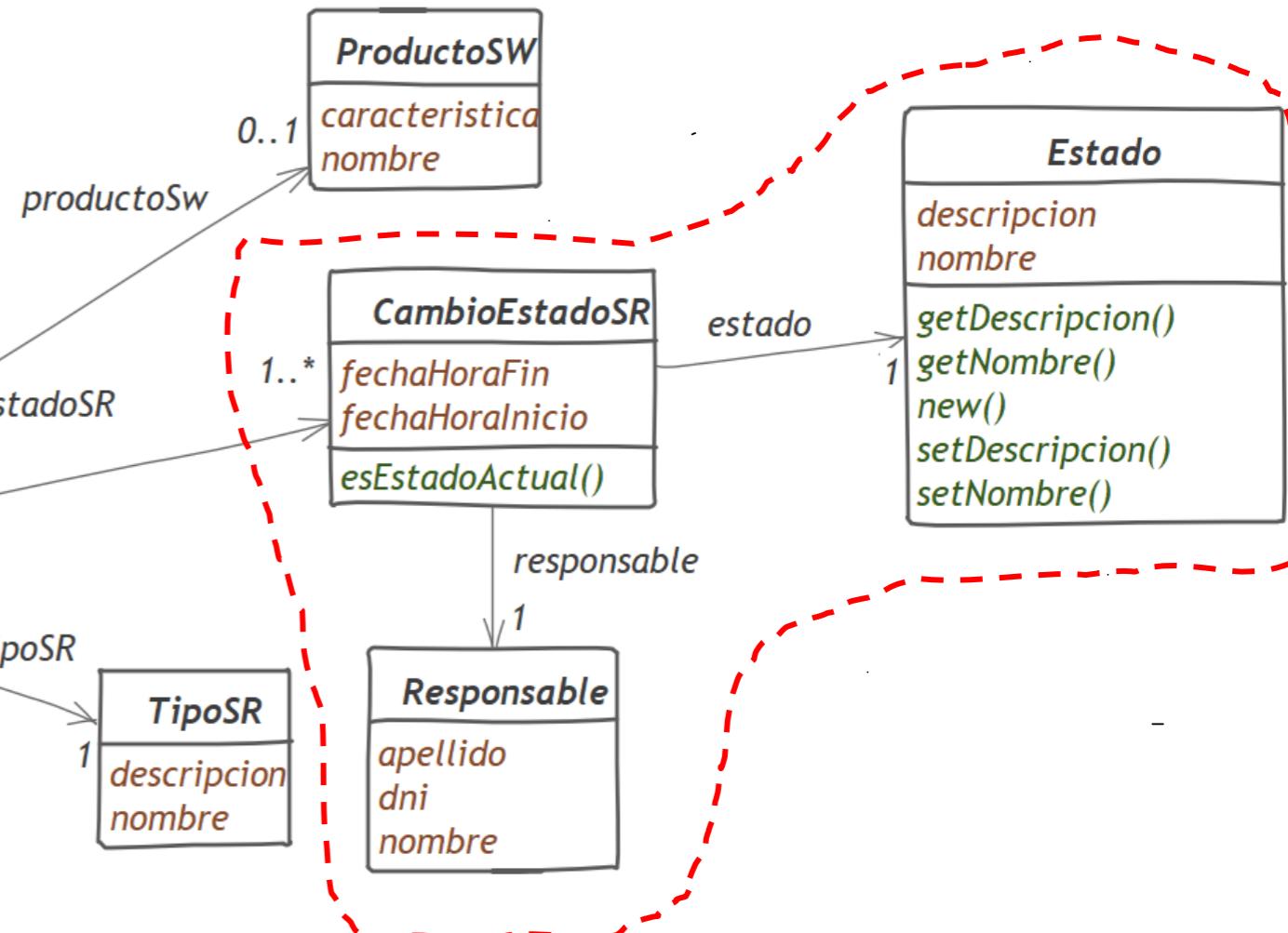
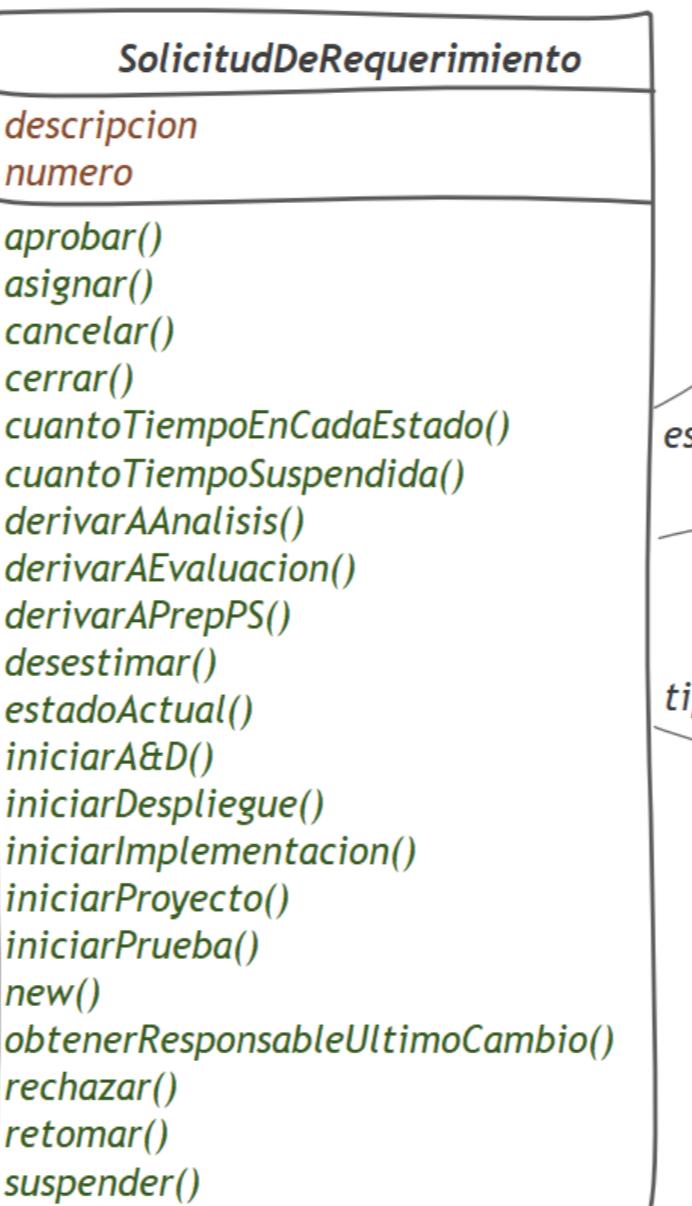
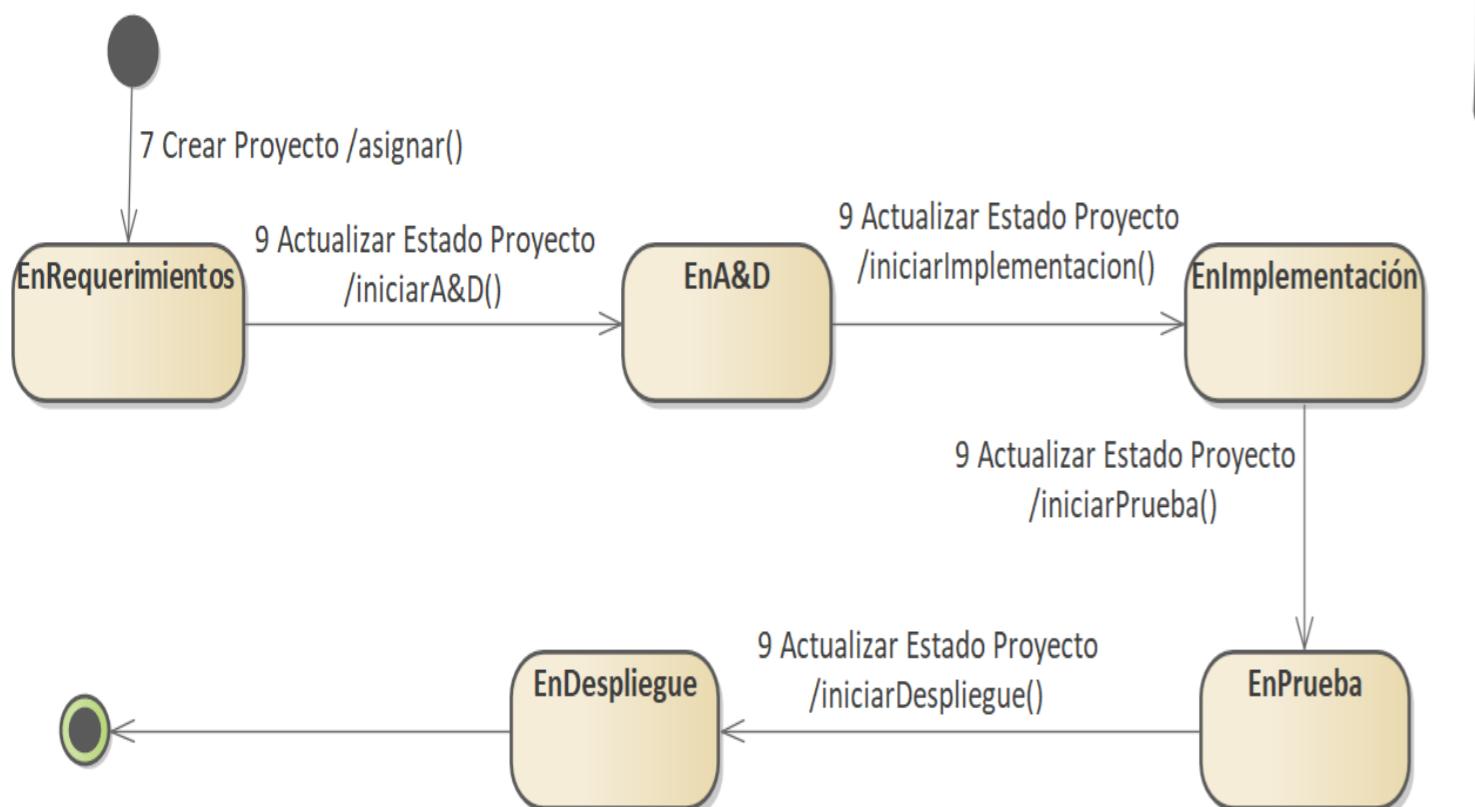
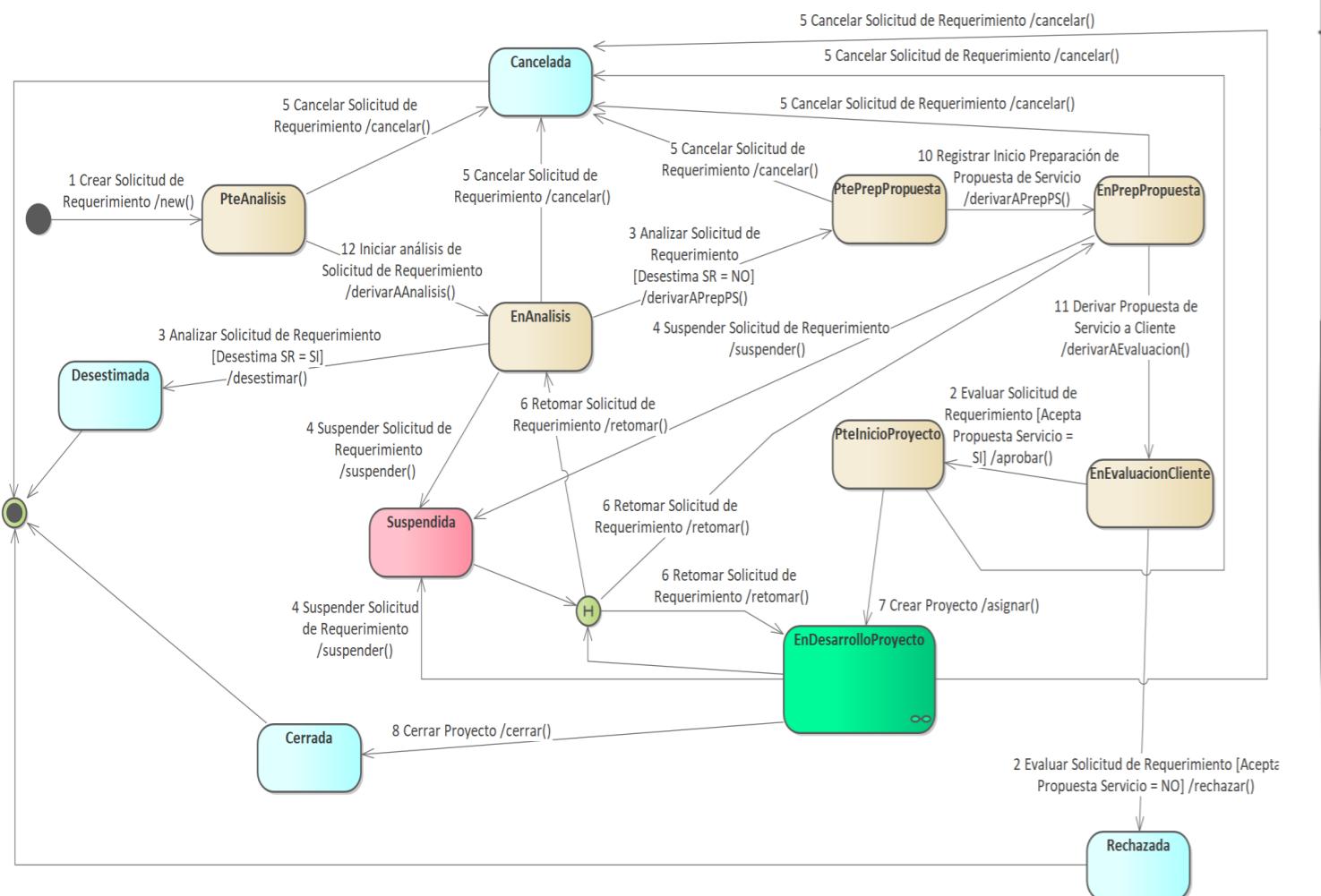


¿Qué vemos dentro del estado compuesto EnDesarrollo Proyecto?

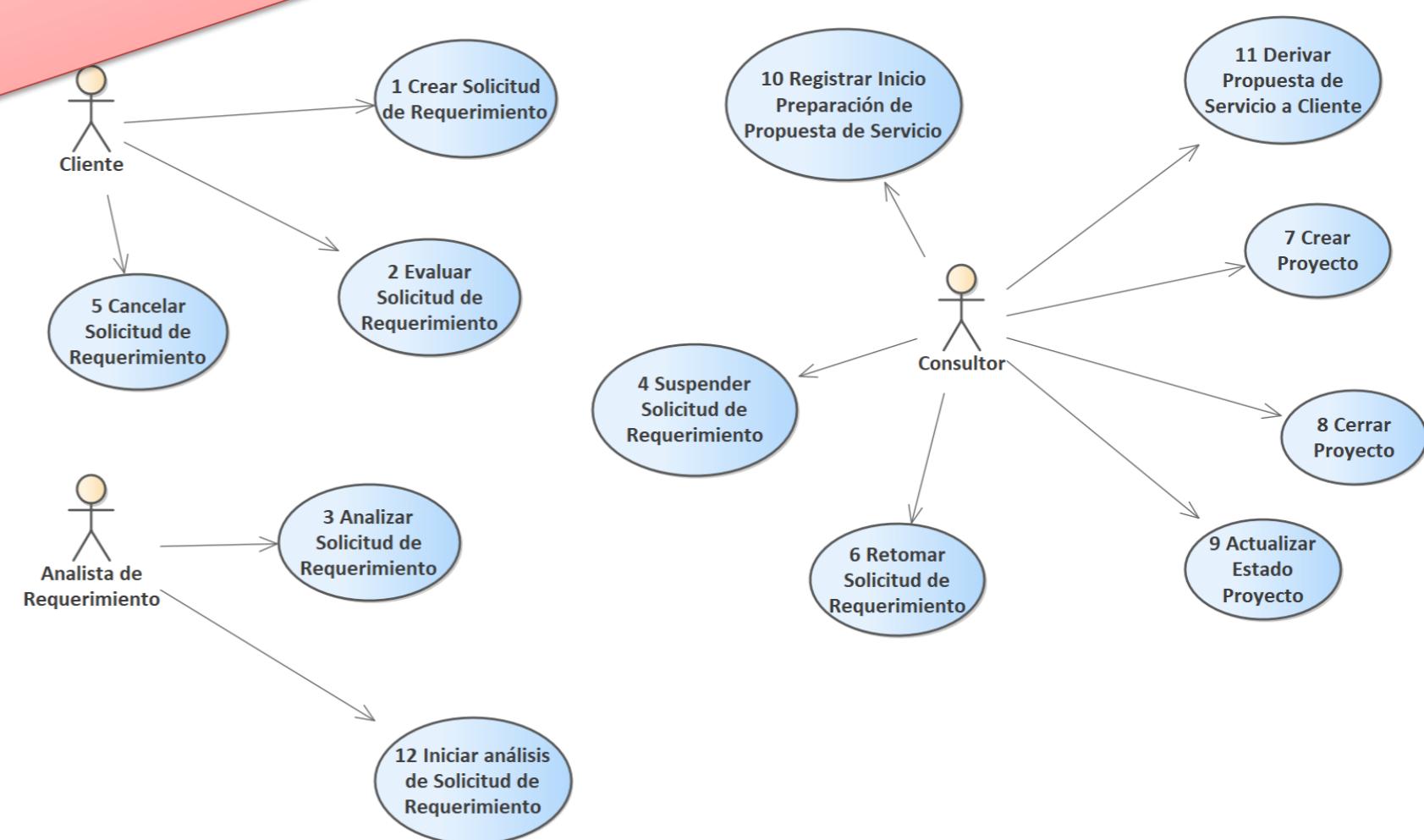
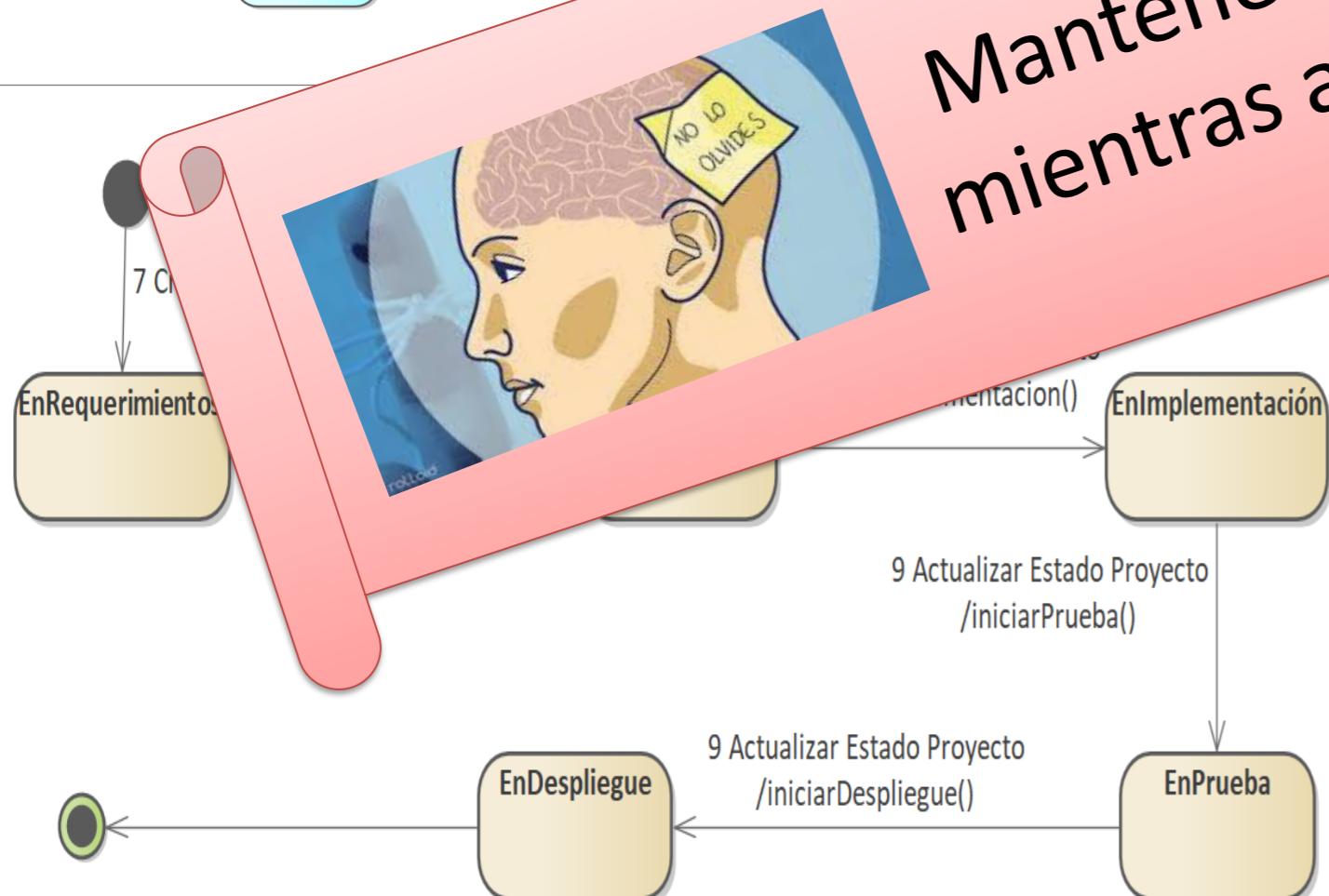
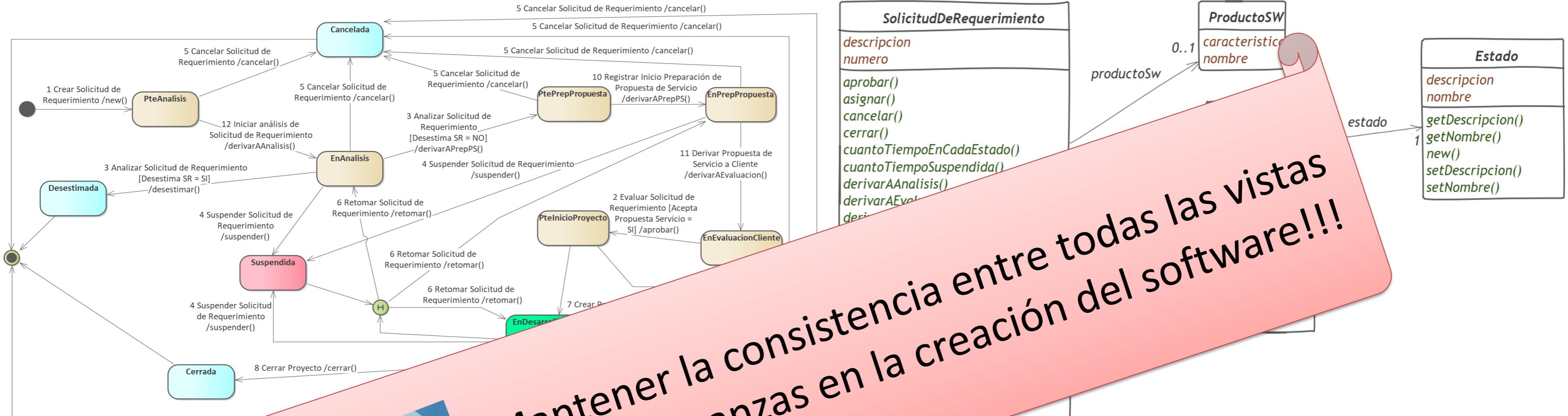


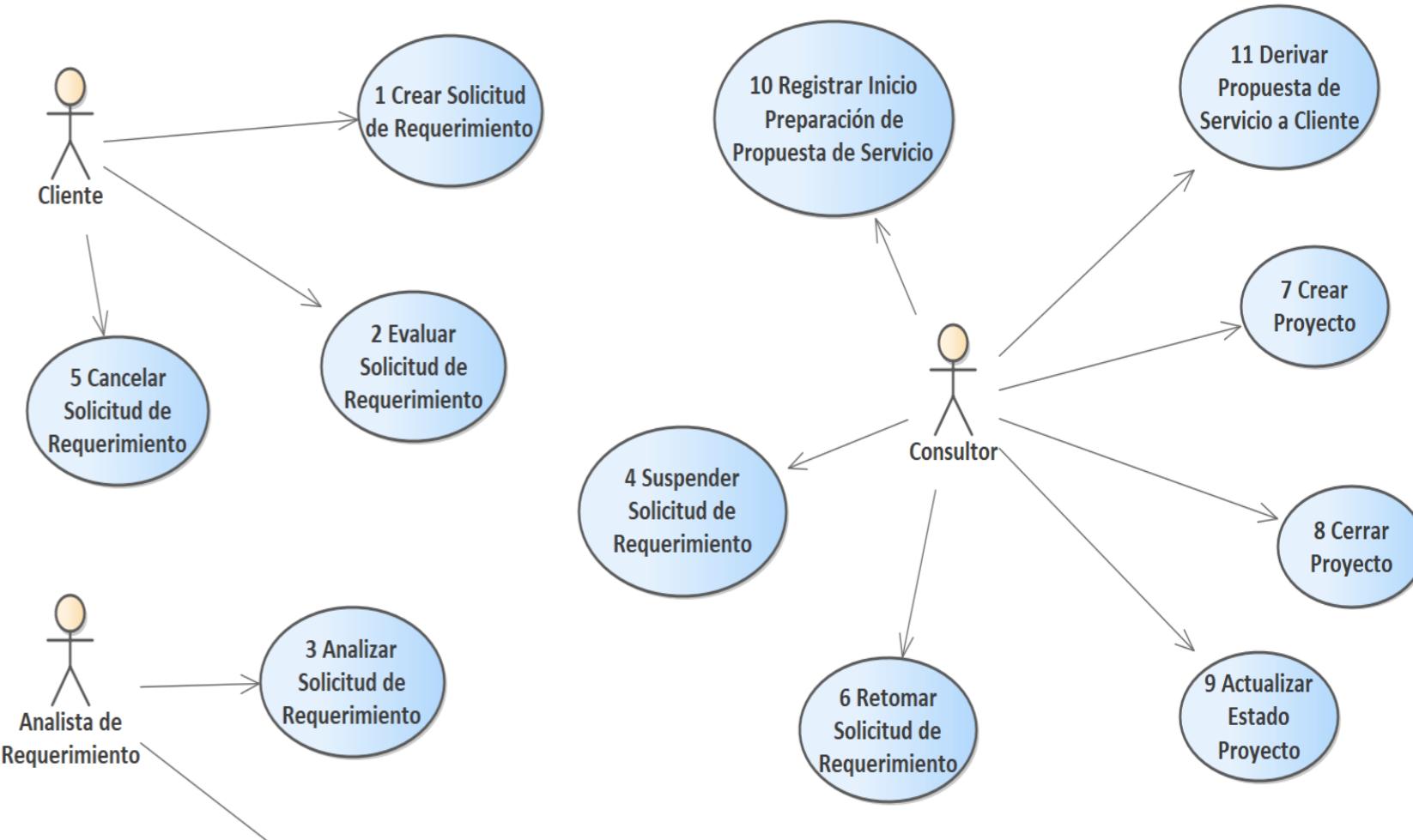
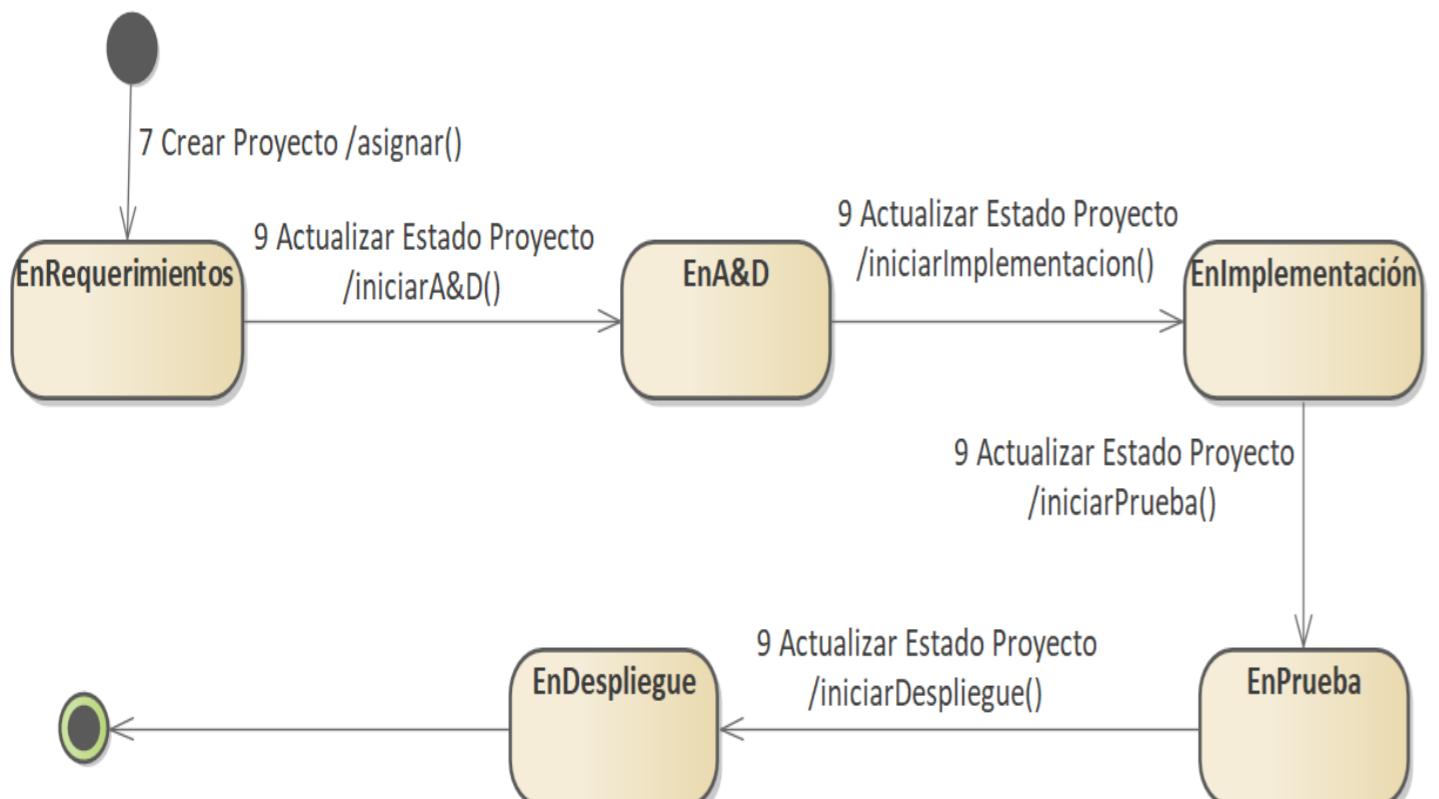
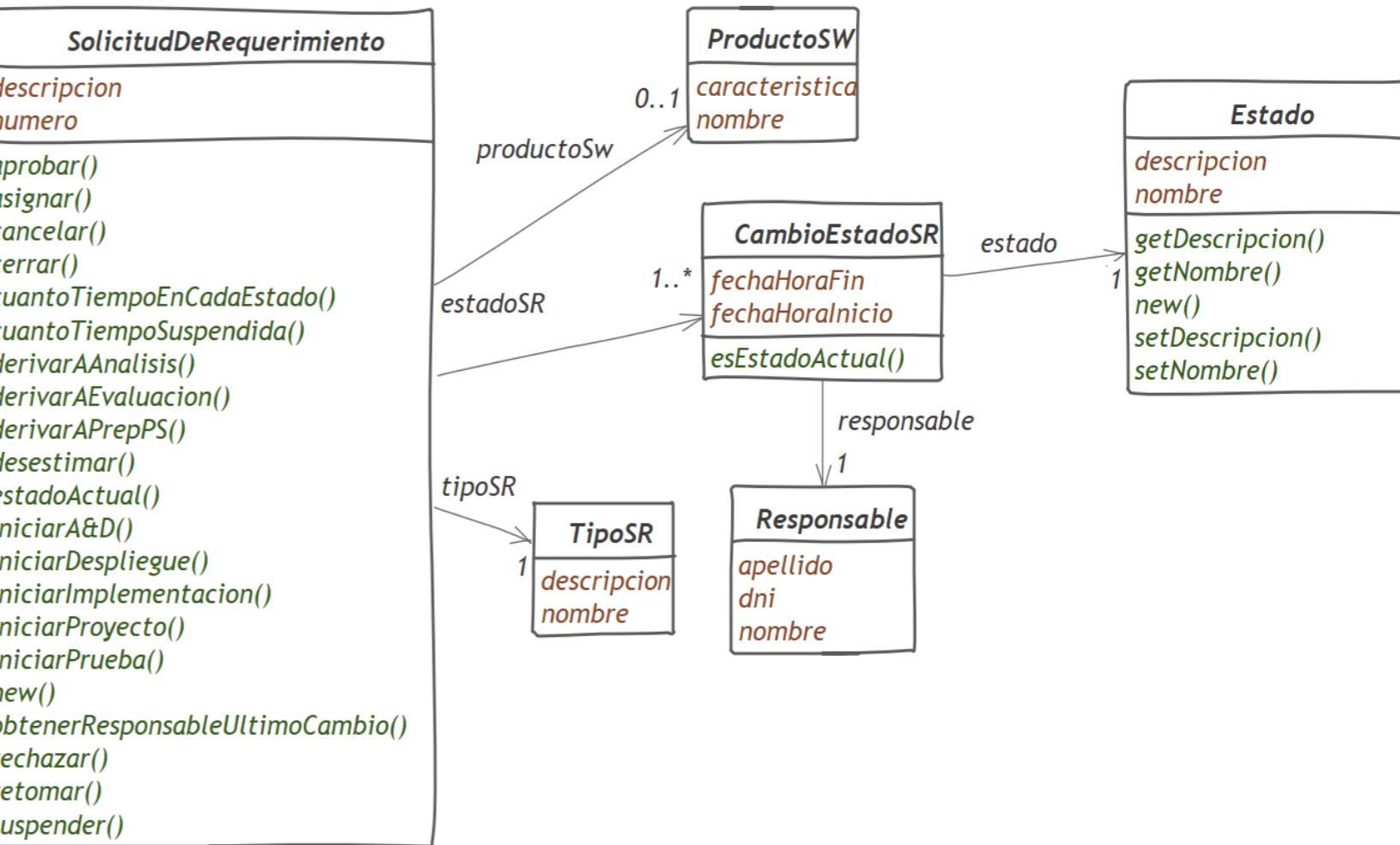
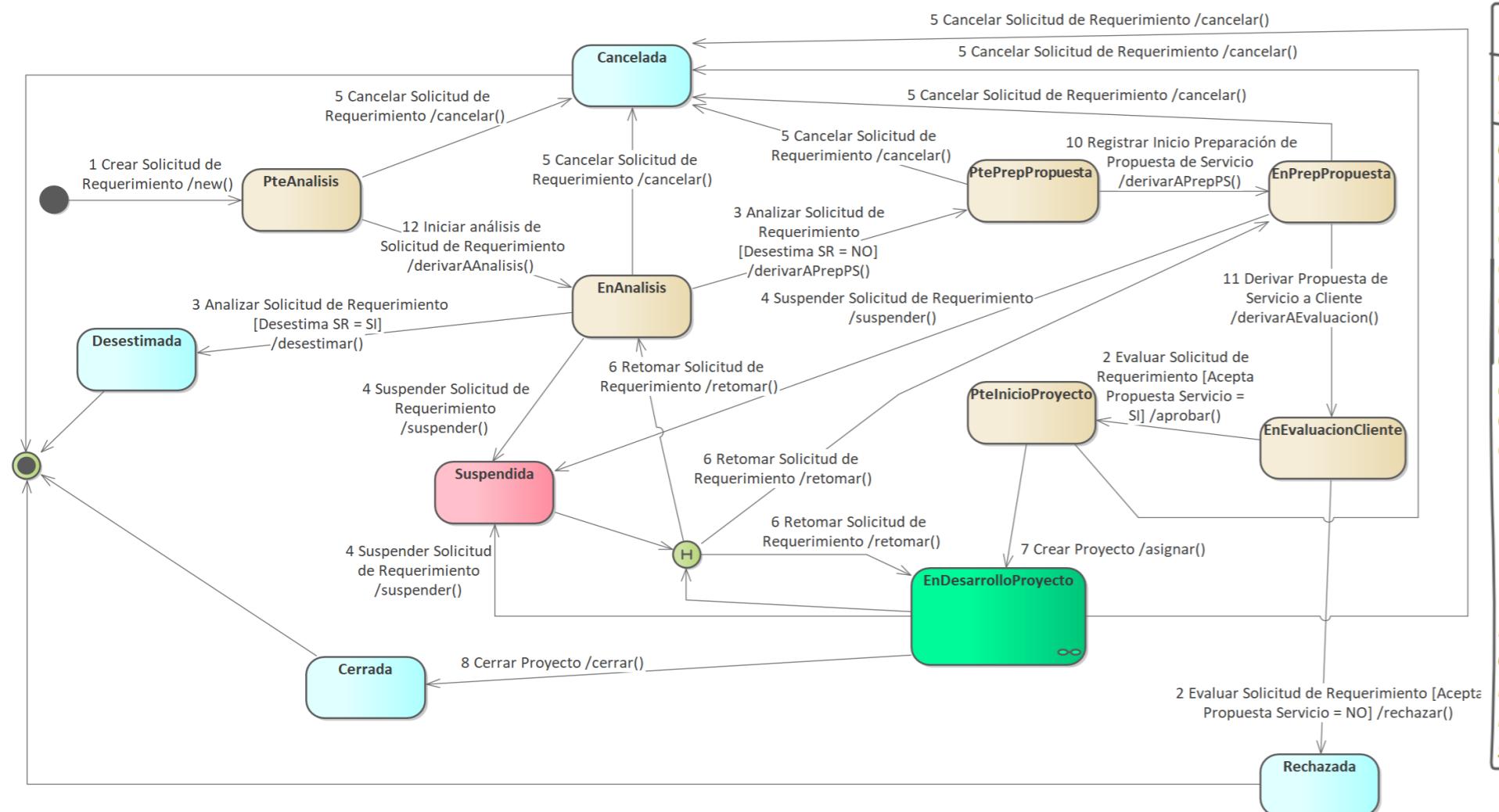
Proyecto de Desarrollo

- ✓ Para realizar el trabajo, se inicia un **proyecto de desarrollo**, que pasará por diferentes etapas (**Requerimientos, Análisis & Diseño, Implementación, Prueba y Despliegue**); el proyecto de desarrollo termina con la aceptación del despliegue por parte del Cliente.
- ✓ Es necesario poder informar a los clientes del estado de la solicitud de requerimientos en todo momento y del avance del proyecto de desarrollo asociado a la SR.
- ✓  **Esto es una simplificación!!! El desarrollo de software no es lineal, ni secuencial, es un proceso de refinamientos sucesivos.**



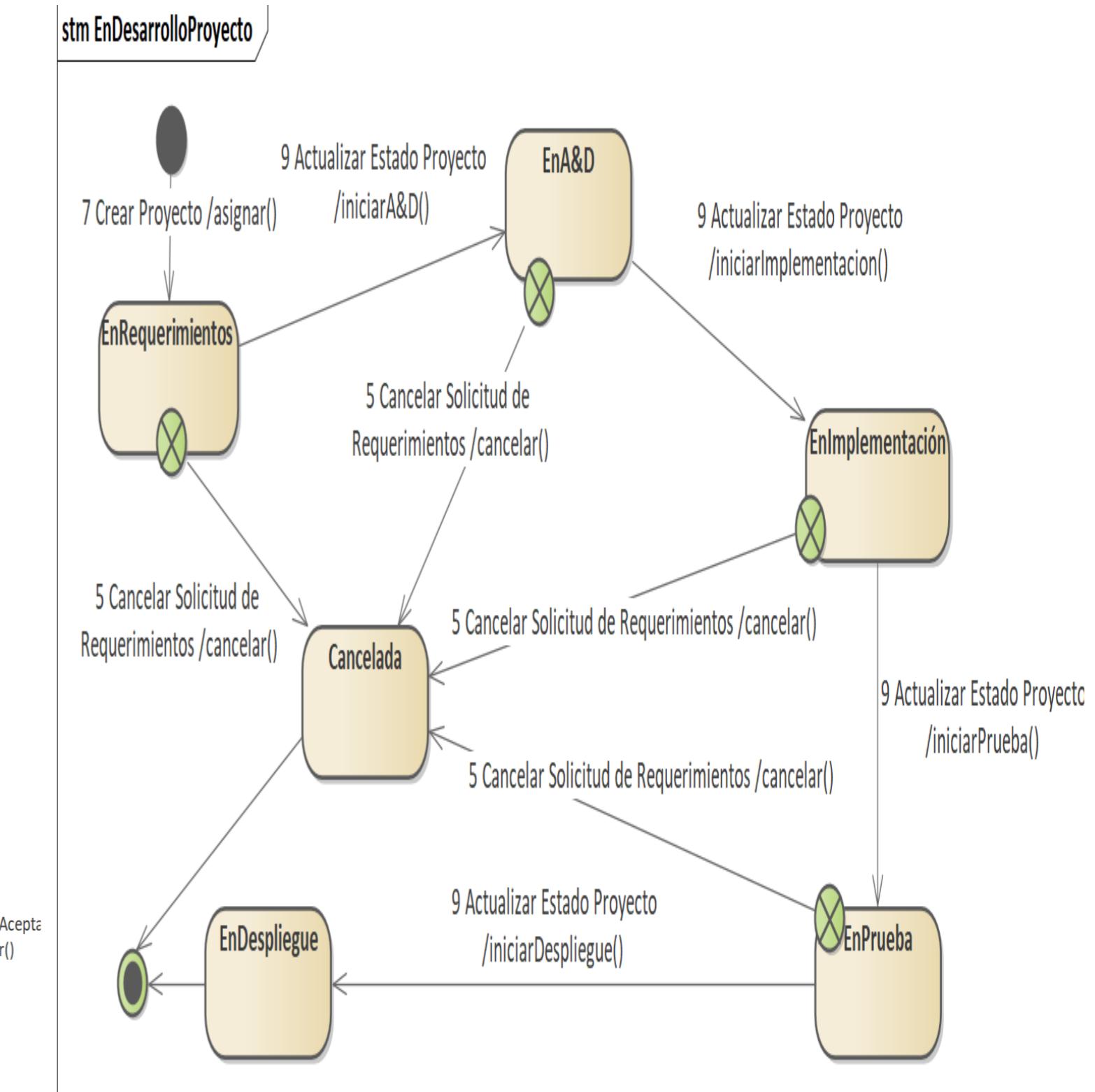
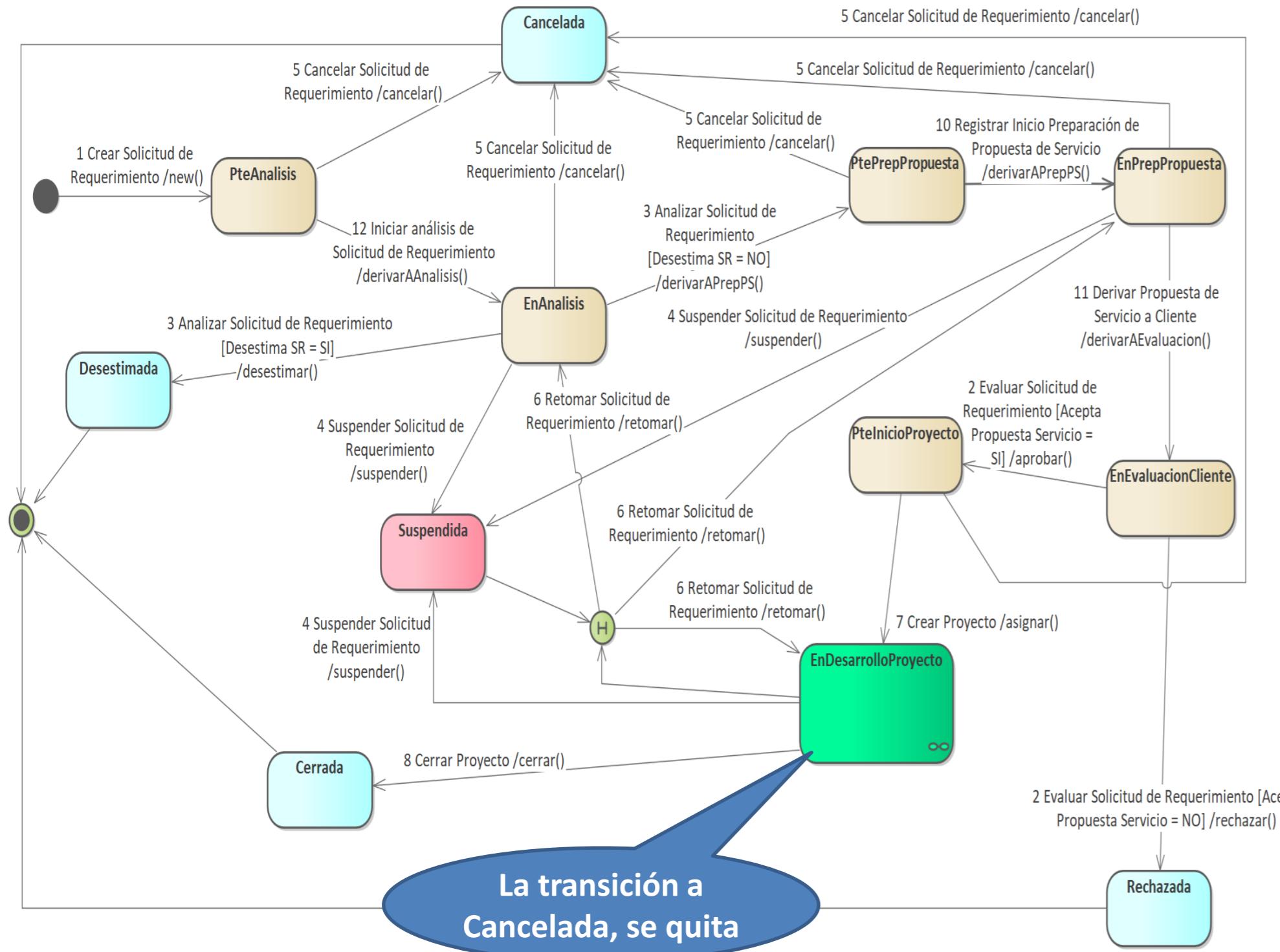
- ✓ También es necesario poder informar **quién es el responsable de cada intervención** (cambio de estado).
- ✓ Es necesario poder informar a los clientes del **estado de la solicitud de requerimientos en todo momento y del avance del proyecto de desarrollo asociado a la SR.**



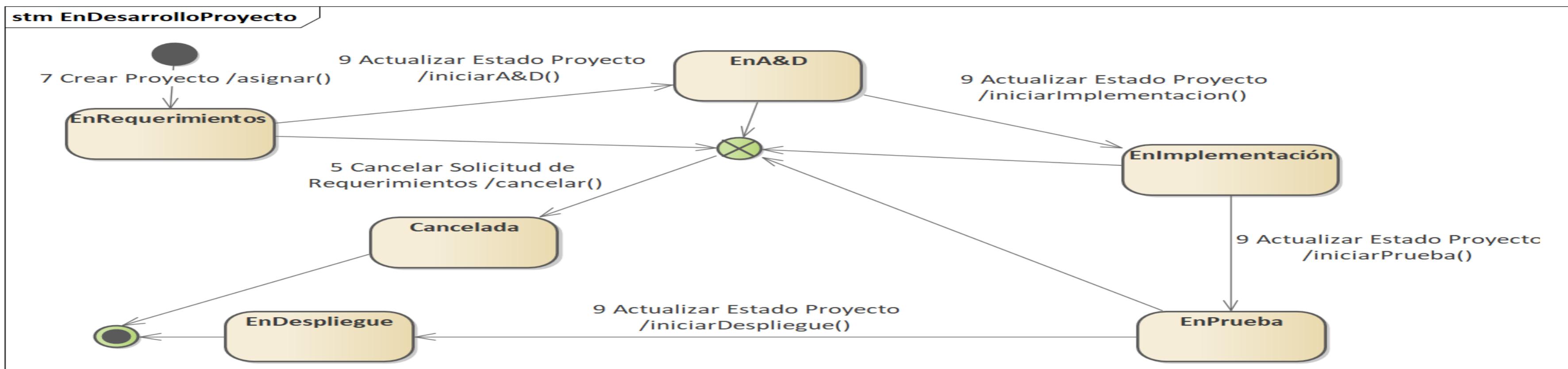
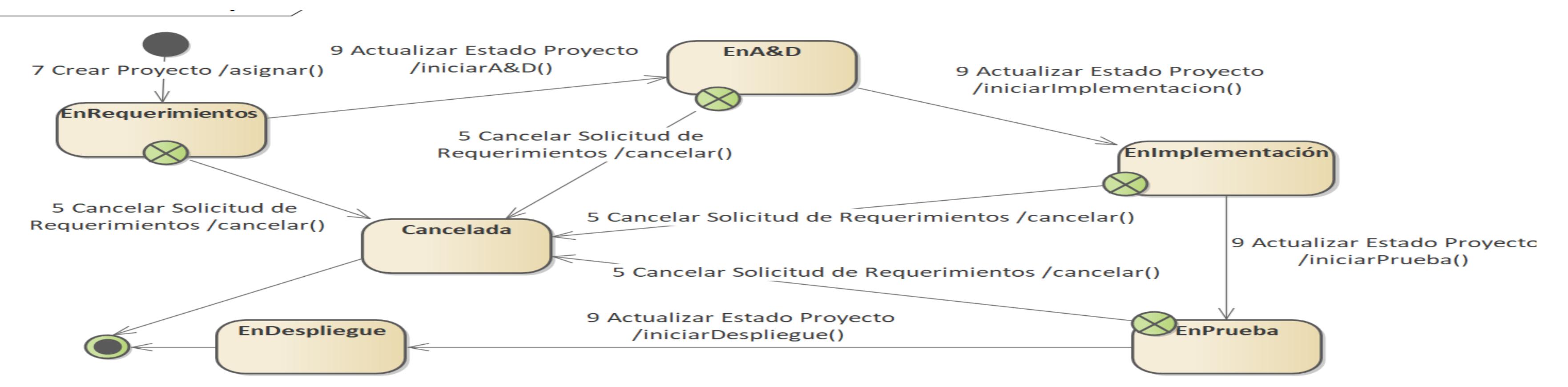


¿Qué pasa si la regla de negocio cambia y dice esto?

“Independientemente de las razones y los aspectos contractuales, un Cliente puede cancelar la SR **en cualquier momento anterior al inicio del despliegue del producto de software.**”

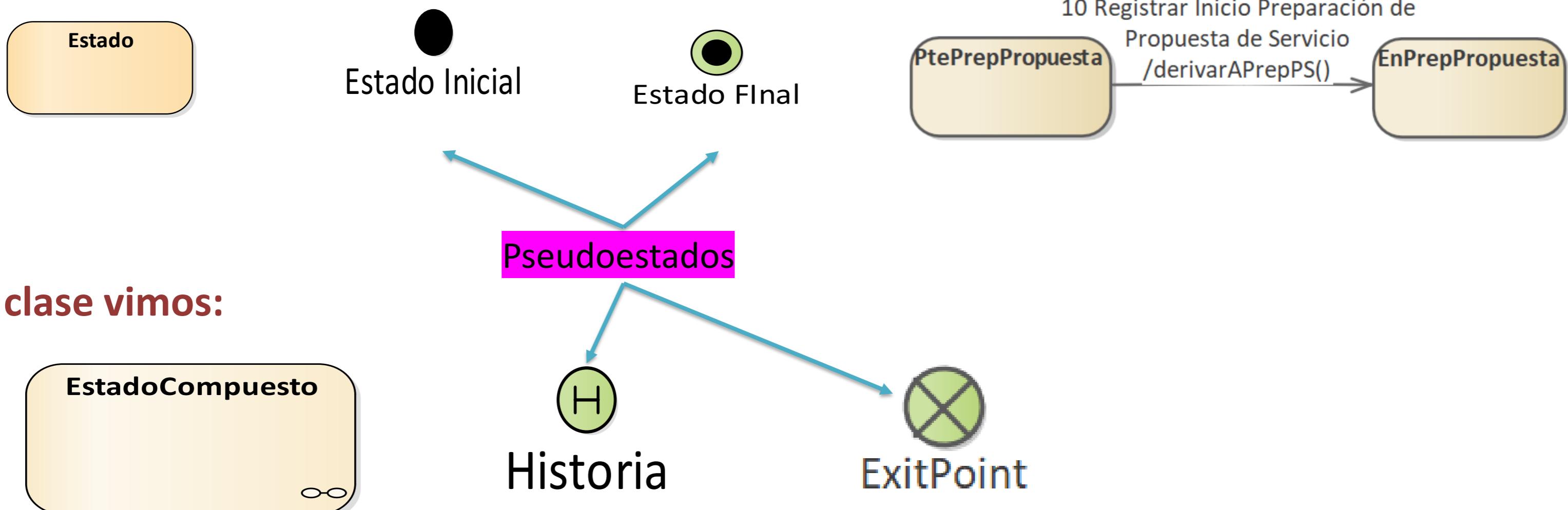


El pseudoestado Exit Point se puede modelar de cualquiera de estas dos formas...



Repasemos los elementos de modelado para el Diagrama de Máquina de Estados que utilizamos en la cátedra:

Esto vimos en la clase anterior

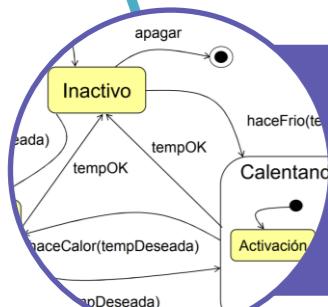


En esta clase vimos:

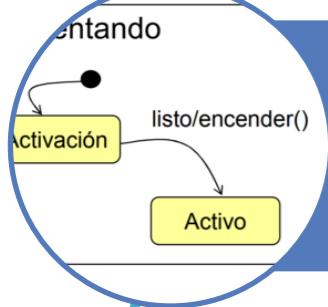
Algunas cosas para tener cuenta sobre el modelado con Diagrama de Máquina de Estados



UML 2.0 tiene más elementos de modelado que los que vemos en esta materia.



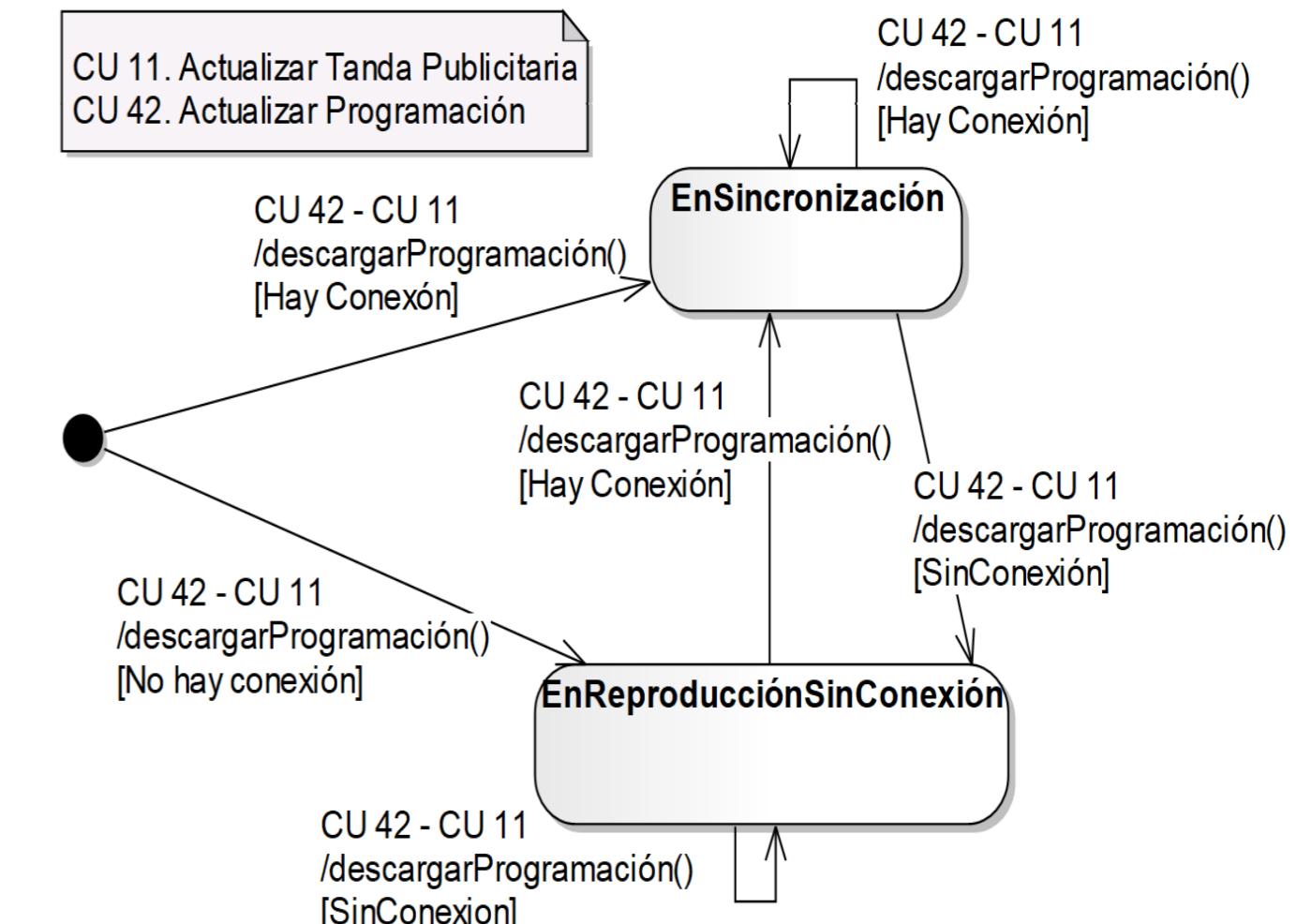
El estado compuesto se indica con el símbolo infinito en la esquina inferior del estado. Para ver la máquina de estado contenida hacemos doble click en el estado compuesto.



Las máquinas de estado dentro de un estado compuesto pueden no tener estado final.



Estado inicial, estado final, historia y Exit son PSEUDOESTADOS.



Para terminar, recordemos...



El estado es el valor de los atributos en un momento de tiempo

No todas las clases necesitan una máquina de estados, cuáles si?

La máquina de estados representa los estados por los que todos los objetos de la clase modelada pueden pasar.

Los objetos NO van a pasar por todos los estados identificados.

La máquina de estado de una clase se construye en función de uno o más atributos de la clase.

Mantener la integración y la consistencia entre las vistas del software.



Finalmente...
Muchas
Gracias!!!