

Unidad 6

16 - Modelado Orientado a Objetos



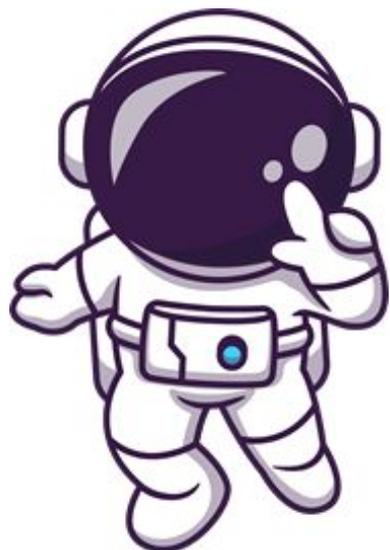
Modelado Orientado a Objetos

- El desarrollo de cualquier sistema consiste en realizar tres etapas:
 - Análisis
 - Diseño
 - Programación
- Durante el proceso de análisis, en el modelo orientado a objetos, se realiza el modelado y la declaración de objetos.
- En el modelo orientado a objetos el manejo de los objetos está determinado por las implementaciones de los métodos detectados en el análisis.





Modelado Orientado a Objetos





Análisis de la estructura

1. Se identifican los tipos de objetos y sus asociaciones representados a través de un esquema de objetos.
2. Se organizan los tipos de objetos en subtipos y supertipos, indicando la jerarquía por medio de herencia.
3. Estudiar la composición de objetos complejos, definiendo el mecanismo que controla la naturaleza de los objetos que se encuentran dentro de otros objetos.





Análisis del comportamiento

1. Se identifican qué estados pueden tener los objetos.
2. Transiciones de estados: se determinan a través de un diagrama de cambios de estado del objeto.
3. Identificación de los eventos que ocurren.
4. Operaciones que se llevan a cabo.
5. Interacción entre objetos.
6. Reglas de activación que se utilizan para reaccionar ante un evento.
7. Representación de todas las operaciones mediante métodos. Este es el paso justo antes de generar código.



Tarjetas CRC

Clases
Responsabilidades
Colaboradores





Tarjetas CRC

- Son una metodología para el diseño de software orientado a objetos creada por Kent Beck y Ward Cunningham.
- Es una técnica para la representación de sistemas OO, para pensar en objetos.
- Son un puente de comunicación entre diferentes participantes.
- Resume las responsabilidades de una clase antes de pasar a realizar el diagrama.
- Sirven para simular escenarios
- No puede ser usado como documentación definitiva y permanente de un proyecto





Tarjetas CRC

- Tarjeta de 6x4 pulgadas
- Responsabilidades
 - Características
 - Comportamientos
- Colaboradores
 - Otras clases que debe colaborar
- Si la tarjeta tiene vacíos, hay que pensar mejor la clase o faltan más clases.
- Si se queda pequeña o necesita más de 1 tarjeta por clase, separar en varias clases porque tiene muchas responsabilidades

Nombre Clase	
Responsabilidades	Colaboradores

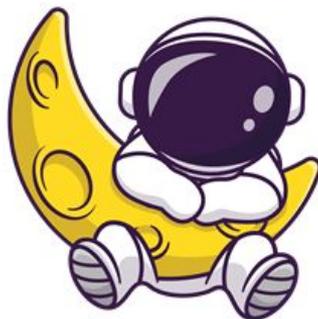




Tarjetas CRC

● Clases

- Colección de objetos similares.
- Se escribe el nombre en la parte superior.
- Al reverso se recomienda escribir una descripción del propósito de la clase.
- Si se tiene una Superclase o Subclase, se puede modificar la tarjeta para recibir esos datos adicionales



Nombre Clase	
Superclase(s):	
Subclase:	
Responsabilidades	Colaboradores



Tarjetas CRC

● Responsabilidades

- Algo que una clase conoce o hace.
- Son todos los servicios que un objeto puede realizar y que mantiene en contacto con otros objetos.
- Se escriben en la parte inferior izquierda.
- Puede ocupar 2/3 partes del espacio de la tarjeta

Nombre Clase	
Responsabilidades	Colaboradores



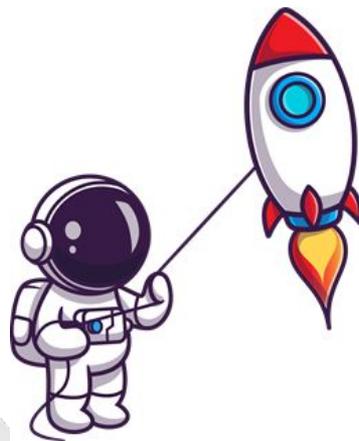


Tarjetas CRC

● Colaboradores

- Otras clases con las que se interactúan en la ejecución de sus responsabilidades.
- Representan las relaciones que hay entre las clases.
- Se escriben en la parte inferior derecha.
- Suele ocupar 1/3 partes del espacio de la tarjeta

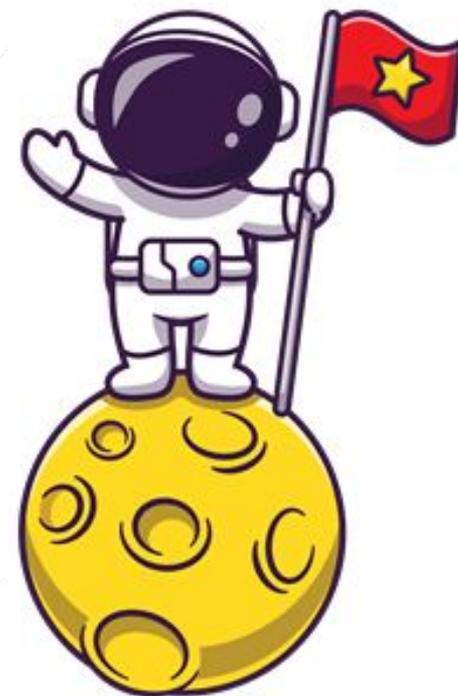
Nombre Clase	
Responsabilidades	Colaboradores





Construcción Tarjetas CRC

- Se debe tener en la mano el documento de especificación de requerimientos.
- Conocer a fondo las clases involucradas
- Podemos apoyarnos de:
 - **Diagramas de jerarquías:** Ayuda a distinguir las clases abstractas de las clases concretas.
 - **Diagramas de Venn:** Mejor conocimiento de las relaciones de herencia que existen en el sistema.
 - **Contratos y formatos:** Ayudan a entender el diseño, debido a que las responsabilidades están agrupadas dentro de ellas





Ejemplo 1

Transacción Bancaria

Datos de transacción

Usuario actual

Giro de dinero

Abono de dinero

Consulta de saldo

Repositorio Cuentas

Repositorio Usuarios





Ejemplo 2

Pacman

Come Frutas

Camina sobre laberinto

Pierde una vida cuando lo atrapan

Frutas

Laberinto

Fantasmas





Sesión de CRC

- Para la elaboración de las tarjetas CRC es recomendable realizarlas un grupo de 3 a 6 personas:
 - Facilitador
 - Arquitecto de software
 - Experto de dominio
- La clave de la sesión CRC es la identificación de las responsabilidades de todas las clases.
 - Lluvia de ideas
- No se debe pensar en cómo será su implementación.





Vuelo	
Origen Destino Número Realizar depósito	Avión

Avión	
Marca Modelo Capacidad Registrar avión	Vuelo

Pasajero	
Nombre Dirección Ciudad País Teléfono Registrar pasajero Modificar pasajero	Reserva

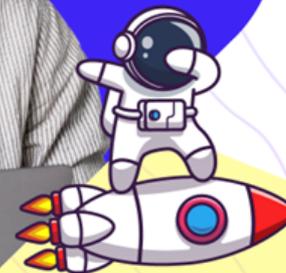
Pago	
Tipo Fecha Cantidad Realizar pago	Pasajero Reserva

Reserva	
Costo Fecha reserva Fecha expiración Nro asiento Registrar reserva Modificar reserva	Pasajero Vuelo



Ejemplo: Agencia de Viajes

Diagrama de Casos de Uso





Diagramas en UML

● Estructurales

- *Diagrama de clases*
- Diagrama de componentes
- Diagrama de despliegue
- *Diagrama de objetos*
- *Diagrama de paquetes*
- Diagrama de perfiles
- Diagrama de estructura compuesta

● De comportamiento

- Diagrama de actividades
- ***Diagrama de casos de uso***
- Diagrama de máquina de estados
- Diagrama de interacción
 - Diagrama de secuencia
 - Diagrama de comunicación
 - Diagrama de tiempos
 - Diagrama global de interacciones

Diagrama de Casos de Uso

- Captura la funcionalidad de un sistema, de un subsistema, o de una clase, tal como se muestra a un usuario exterior.
- Reparte la funcionalidad del sistema en transacciones significativas para los usuarios ideales de un sistema.
- Los usuarios del sistema se denominan **actores** y las particiones funcionales se conocen con el nombre de **casos de uso**.
- La técnica que se utiliza para modelar esta vista es el diagrama de casos de uso

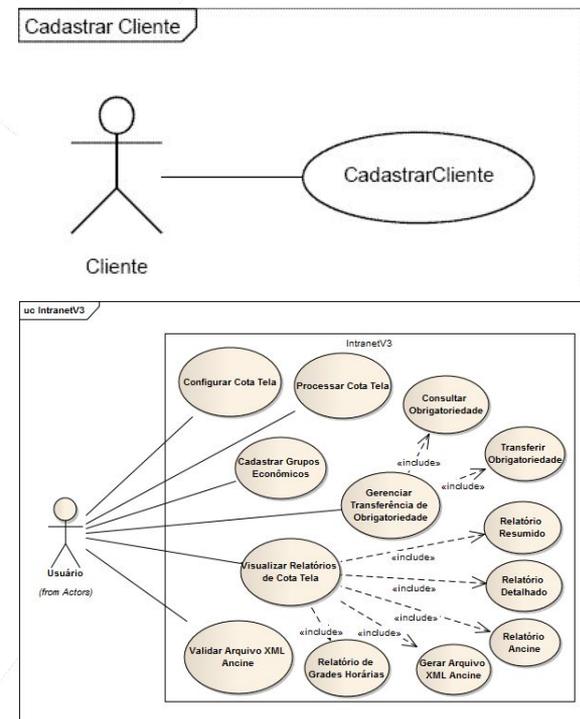
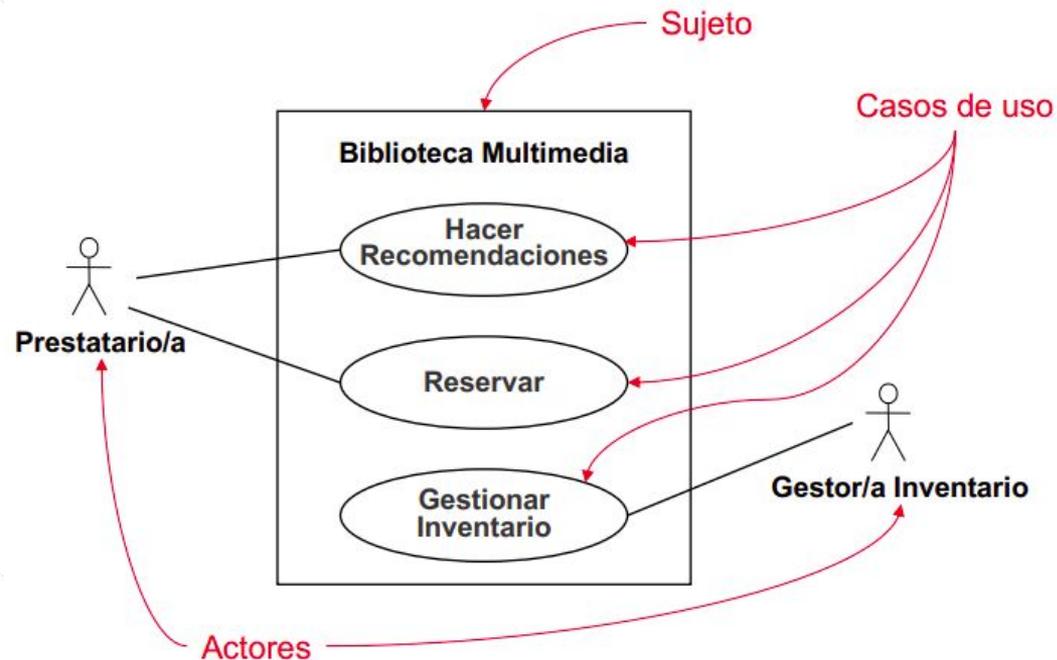


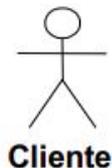
Diagrama de Casos de Uso

- Sus componentes principales son:
 - **Sujeto:** sistema que se modela
 - **Casos de uso:** unidades funcionales completas
 - **Actores:** entidades externas que interactúan con el sistema





Actores



Cliente



Un actor es un clasificador que modela un tipo de rol que juega una entidad que interacciona con el sujeto pero que es externa a él

- Un actor puede tener múltiples instancias físicas
- Una instancia física de un actor puede jugar diferentes papeles

Los actores se comunican con el sujeto intercambiando mensajes (señales, llamadas o datos)

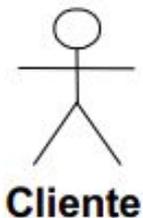
Notación:

- Se representan con el icono estándar de “*stick man*” o “*monigote*” con el nombre del actor (obligatorio) cerca del símbolo, normalmente se pone encima o debajo
- También se puede representar mediante un símbolo de clasificador con el estereotipo «*actor*»
- Los nombres de los actores suelen empezar por mayúscula
- Se pueden usar otros símbolos para representar tipos de actores, por ejemplo para representar actores no humanos



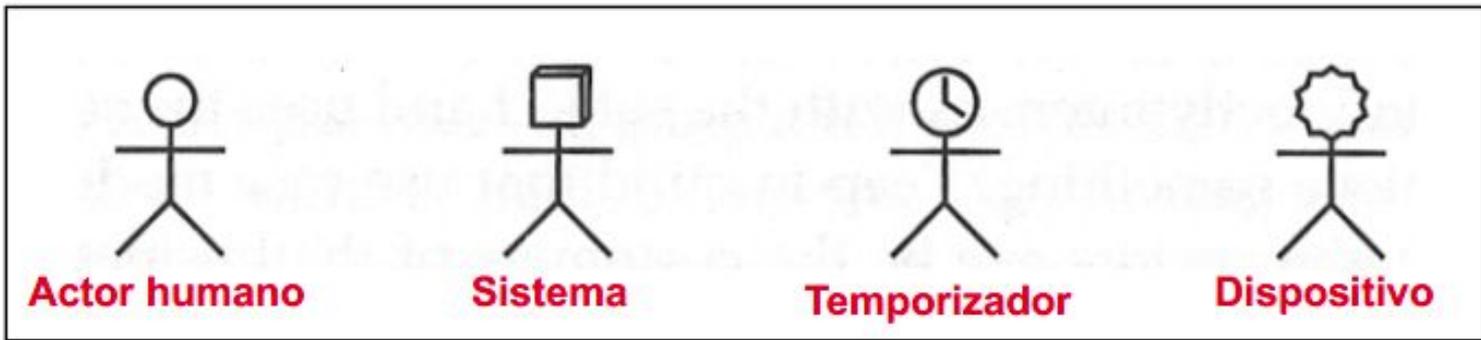
Actores

Stick man



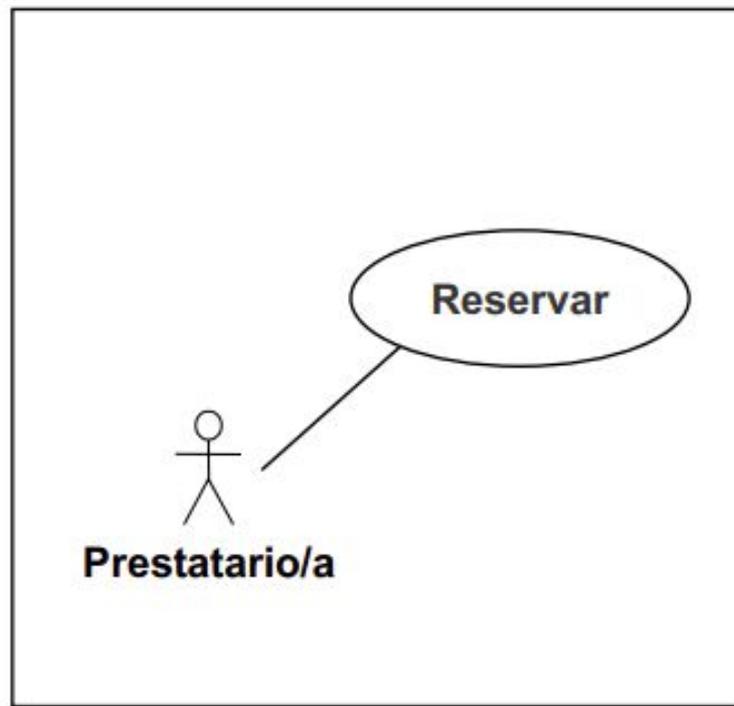
«actor»
Cliente

Clasificador

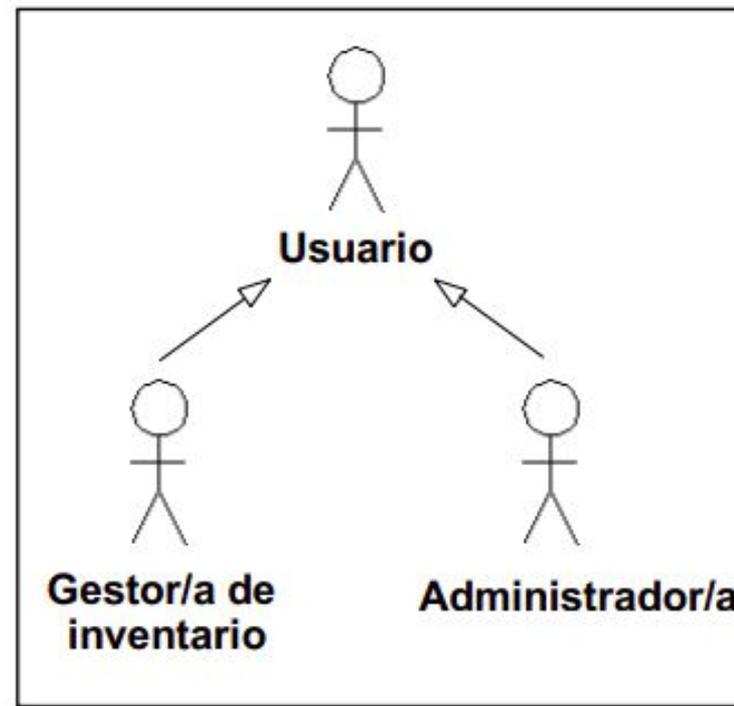




Actores



Asociación entre un actor y un caso de uso



Relaciones de generalización entre actores

Casos de Uso

Un caso de uso se define como un conjunto de acciones realizadas por el sistema que dan lugar a un resultado observable. Especifica un comportamiento que el sujeto puede realizar en colaboración con uno o más actores, pero sin hacer referencia a su estructura interna.

El caso de uso puede contener posibles variaciones de su comportamiento básico incluyendo manejo de errores y excepciones.

Características de los casos de uso:

- Un caso de uso se inicia por un actor
- Los casos de uso proporcionan valores a los actores
- La funcionalidad de un caso de uso debe ser completa



El comportamiento de un caso de uso se puede describir mediante interacciones, actividades, máquinas de estado ...

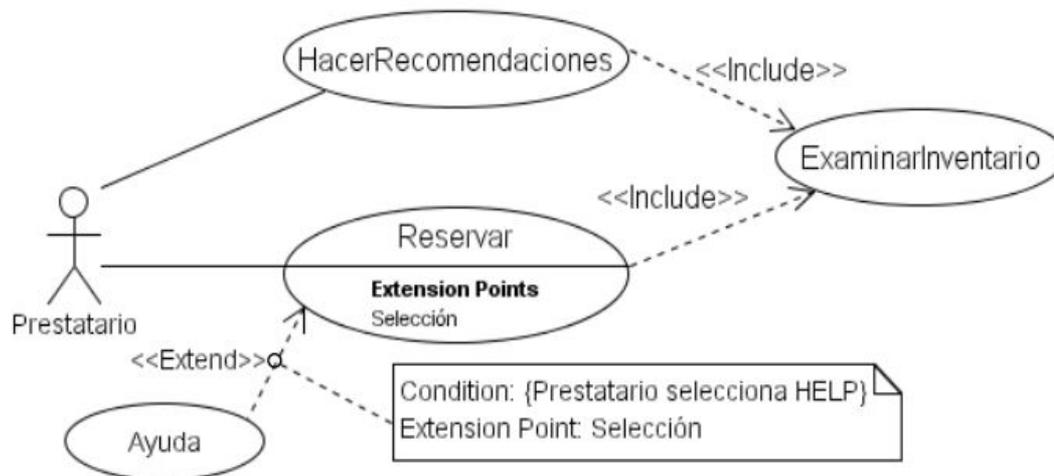
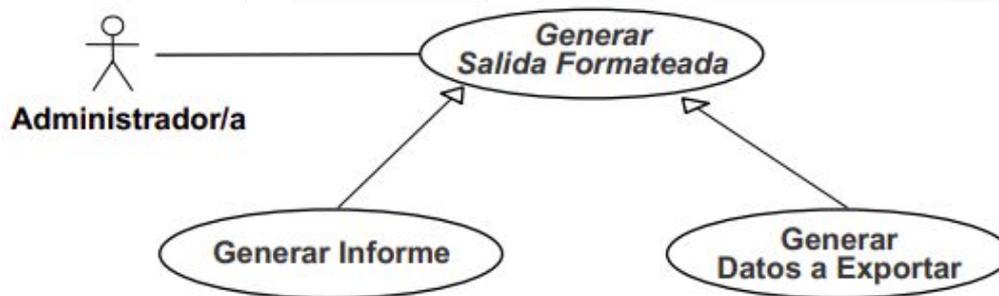


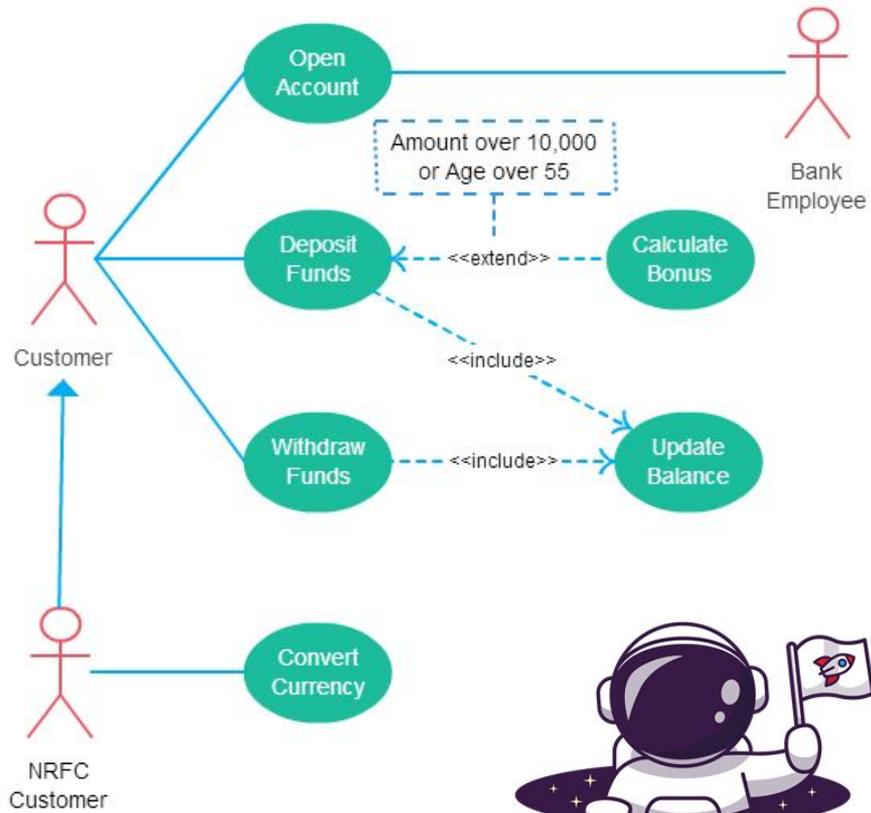


Relaciones entre Casos de Uso

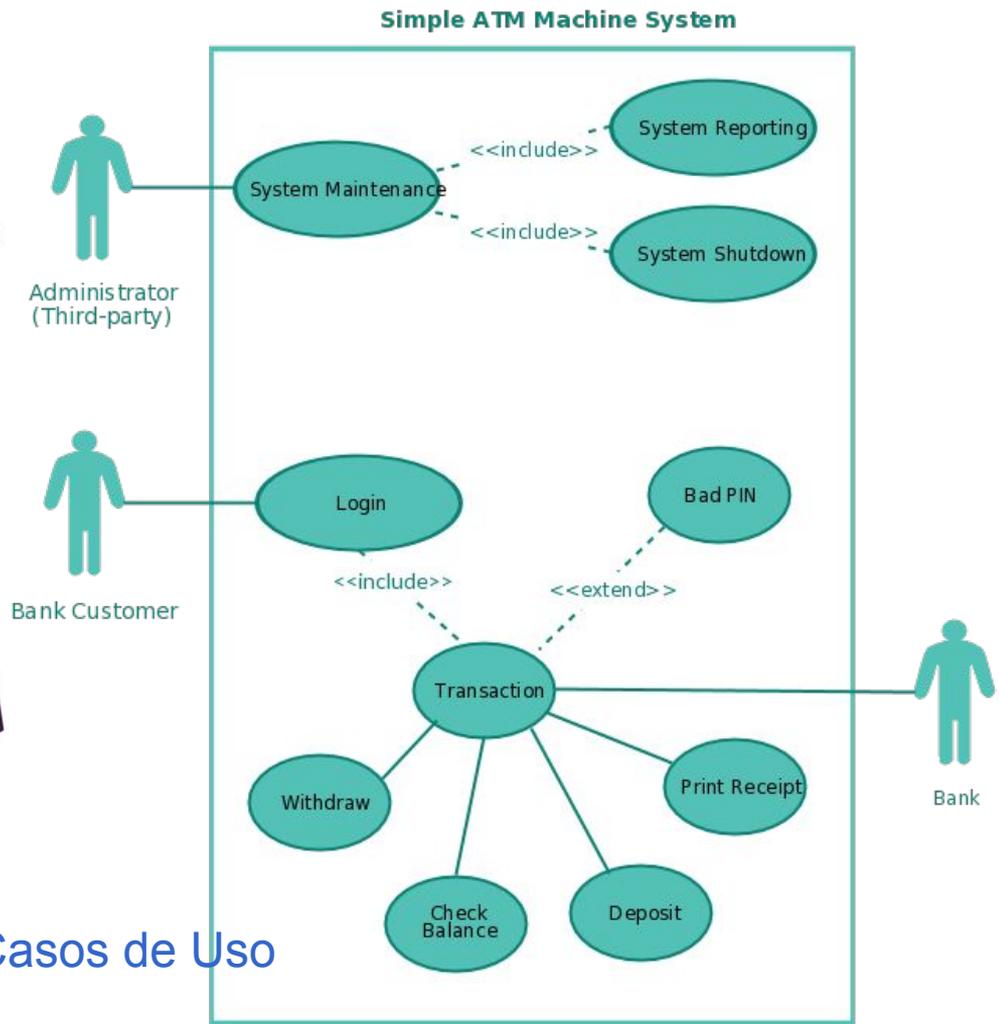
Relación	Descripción	Notación
Asociación	Línea de comunicación entre un actor y un caso de uso en el que participa	—————
Generalización	Una relación entre un caso de uso general y un caso de uso más específico, que hereda y añade propiedades al caso de uso base	—————>
Inclusión	El comportamiento de un caso de uso (incluido) se inserta en el comportamiento de otro caso de uso (includor)	- - - - -> «include»
Extensión	El comportamiento de un caso de uso base (extendido) puede ser extendido con comportamiento adicional definido por otro caso de uso (extensor).	- - - - -> «extend»
Realización	Establece una relación entre el caso de uso y los diagramas que describen la funcionalidad del caso de uso	- - - - ->

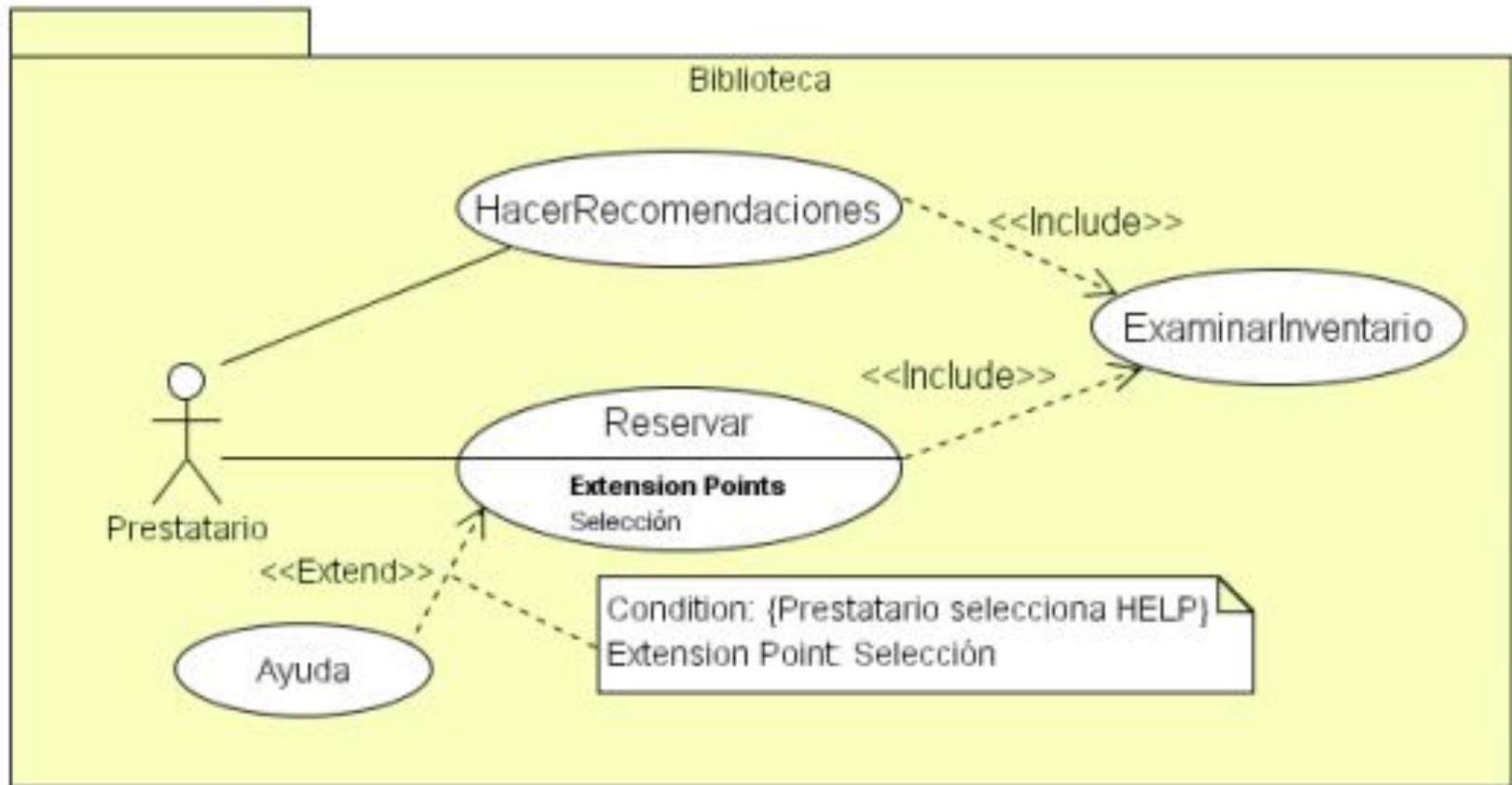






Ejemplos Diagramas Casos de Uso





Diagramas agrupados en un paquete



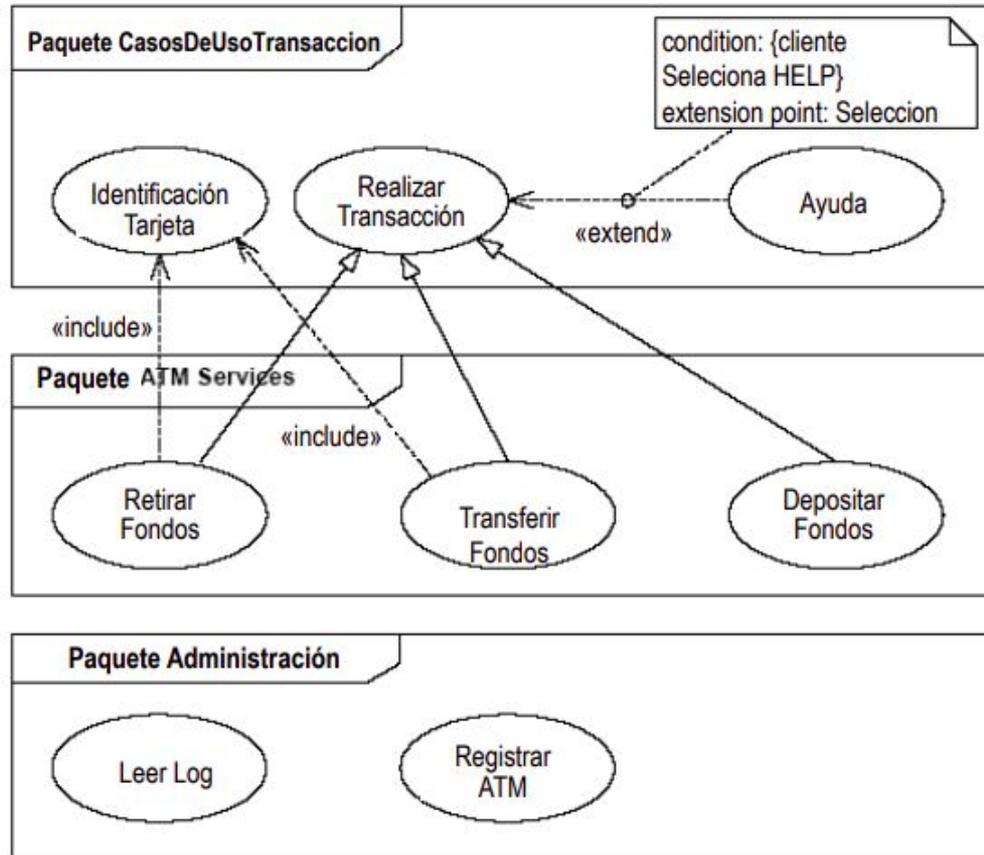
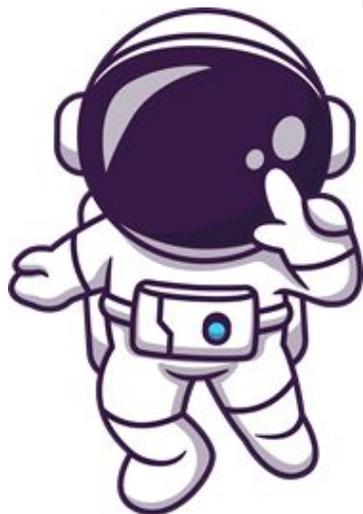


Diagrama con casos de uso en diferentes paquetes

Descripción de los actores



Nombre:	<i><nombre del actor></i>
----------------	---------------------------------

Descripción:	<i><descripción del actor></i>
---------------------	--------------------------------------

Nombre:	Usuario no Autenticado
----------------	------------------------

Descripción:	
---------------------	--

Representa a un usuario que no se a identificado frente al sistema. Generalmente estos usuarios deberían poder registrarse (crear un nuevo usuario) o ingresar al sistema para transformarse en usuarios autenticados, en moderadores o en administradores del sistema





Descripción del Caso de Uso



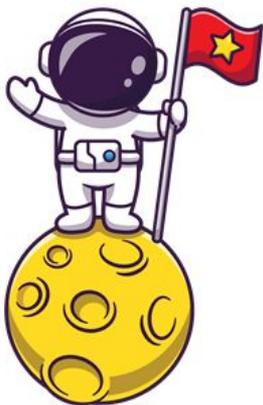
Nombre:	<i><nombre del caso de uso></i>
Autor:	<i><nombre del autor (o autores) del caso de uso></i>
Fecha:	<i><fecha de creación del caso de uso></i>
Descripción:	<i><breve descripción del caso de uso></i>
Actores:	<i><actores participantes en el caso de uso></i>
Precondiciones:	<i><condiciones que deben cumplirse para poder ejecutar el caso de uso></i>
Flujo Normal:	<i><flujo normal (feliz) de ejecución del caso de uso></i>
Flujo Alternativo:	<i><flujos alternativos de ejecución del caso de uso></i>
Poscondiciones:	<i><condiciones que deben cumplirse al finalizar la ejecución del caso de uso></i>





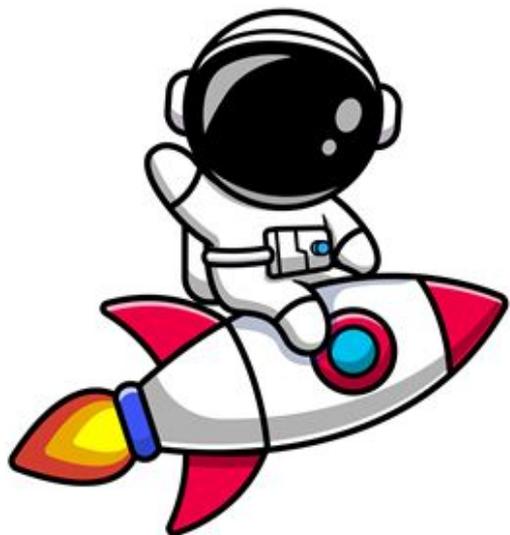
Descripción del Caso de Uso

Nombre:	Crear mensaje foro
Autor:	Pedro Pérez
Fecha:	21/04/09
Descripción:	Permite crear un nuevo mensaje (hilo) en el foro de discusión.
Actores:	Usuario / Moderador
Precondiciones:	El usuario debe de estar autenticado en el sistema.





Descripción del Caso de Uso



Flujo Normal:

- 1.- El actor pulsa sobre el botón para crear un nuevo mensaje.
- 2.- El sistema muestra una caja de texto para introducir el título del mensaje y una zona de mayor tamaño para introducir el cuerpo del mensaje.
- 3.- El actor introduce el título del mensaje y el cuerpo del mismo.
- 4.- El sistema comprueba la validez de los datos y los almacena.
- 5.- El moderador recibe una notificación de que hay un nuevo mensaje.
- 6.- El moderador acepta y el sistema publica el mensaje si éste fue aceptado por el moderador.

Flujo Alternativo:

- 4.A.- El sistema comprueba la validez de los datos, si los datos no son correctos, se avisa al actor de ello permitiéndole que los corrija.
- 7.B.- El moderador rechaza el mensaje, de modo que no es publicado sino devuelto al usuario.

Poscondiciones:

El mensaje ha sido almacenado en el sistema y fue publicado.

Otros diagramas de comportamiento





Diagramas en UML

● Estructurales

- *Diagrama de clases*
- Diagrama de componentes
- Diagrama de despliegue
- *Diagrama de objetos*
- *Diagrama de paquetes*
- Diagrama de perfiles
- Diagrama de estructura compuesta

● De comportamiento

- ***Diagrama de actividades***
- *Diagrama de casos de uso*
- ***Diagrama de máquina de estados***
- Diagrama de interacción
 - ***Diagrama de secuencia***
 - Diagrama de comunicación
 - Diagrama de tiempos
 - Diagrama global de interacciones



Diagrama de actividades

- Un diagrama de actividad representa el comportamiento mediante un modelo de flujo de datos y flujo de control
 - **Actividad:** especificación de un comportamiento parametrizado que se expresa como un flujo de ejecución por medio de una secuencia de unidades subordinadas
 - **Acción:** especificación de una unidad fundamental de comportamiento que representa una transformación o procesamiento
 - Las acciones están contenidas en actividades que le proporcionan su contexto
 - Los diagramas de actividad capturan las acciones y sus resultados
- Los pasos de ejecución dentro de una actividad pueden ser concurrentes o secuenciales
- Una actividad involucra constructores de sincronización y de bifurcación



Diagrama de actividades

Usos de los diagramas de actividad

- Capturar las acciones que se realizan cuando se ejecuta una operación
- Capturar el trabajo interno de un objeto
- Mostrar cómo se pueden realizar un conjunto de acciones relacionadas y cómo afectan a los objetos
- Mostrar como se puede realizar una instancia de un caso de uso en términos de acciones y cambios de estado de los objetos
- Mostrar como trabaja un negocio en términos de trabajadores (actores), flujos de trabajo, organización y objetos (factores intelectuales y físicos usados en un negocio)

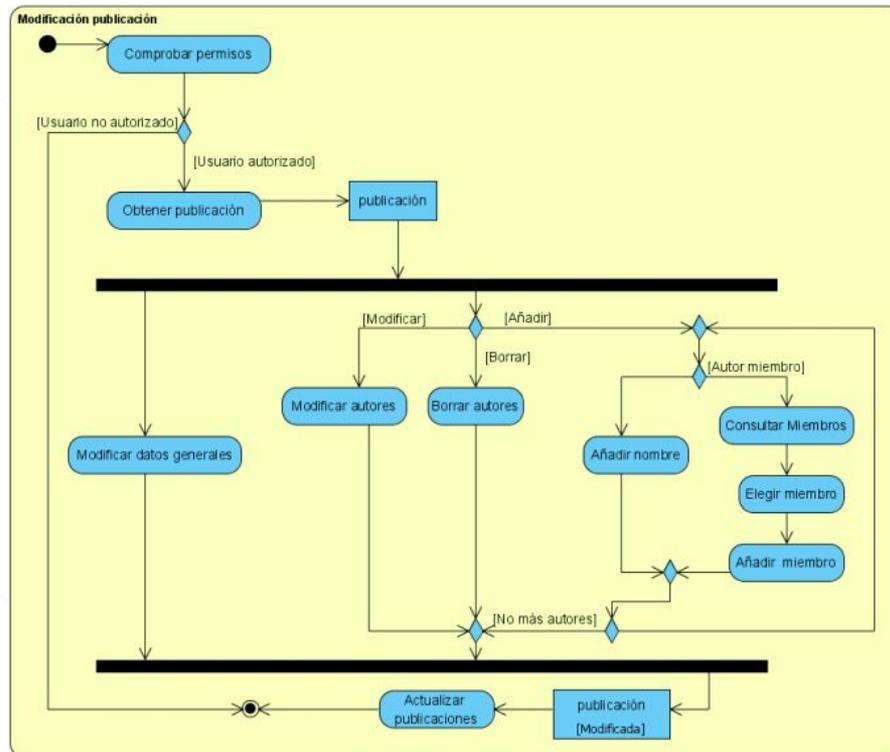


Diagrama de actividades

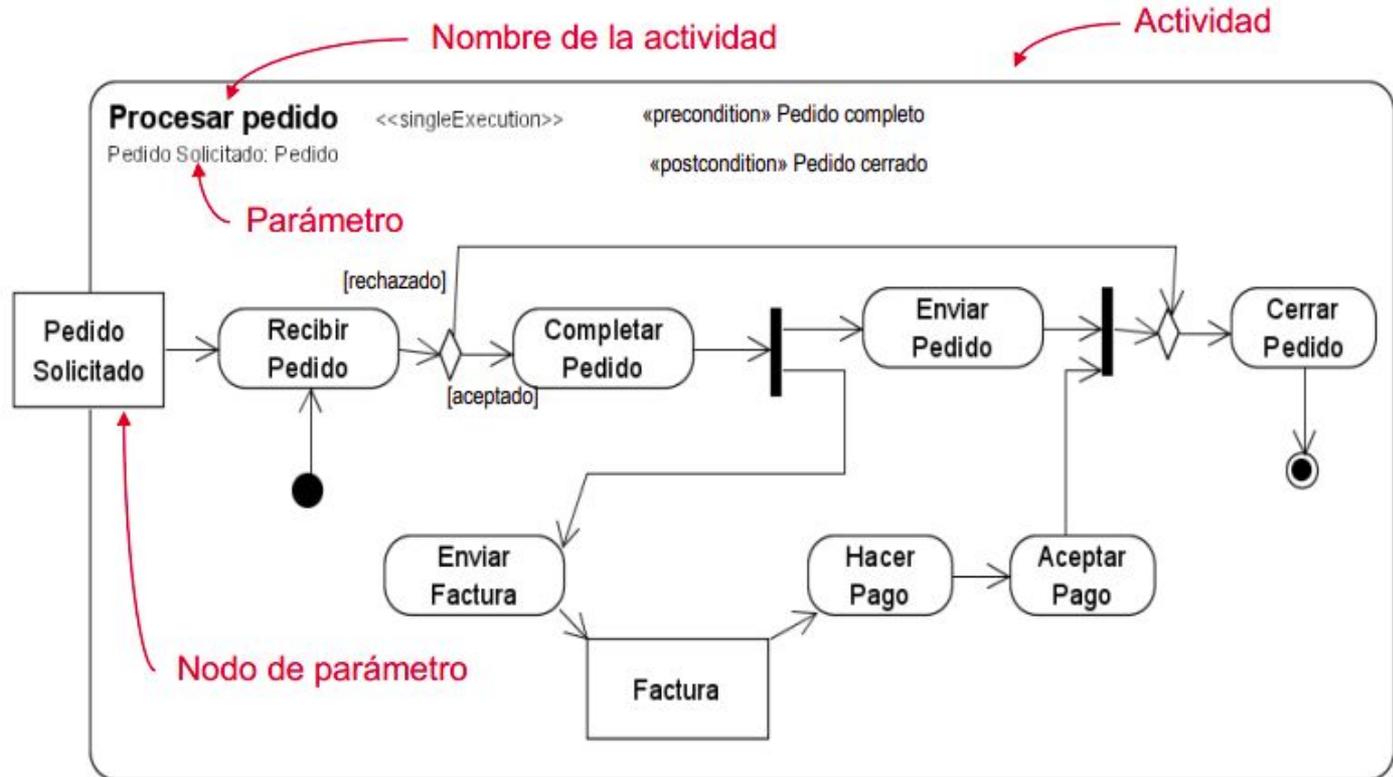
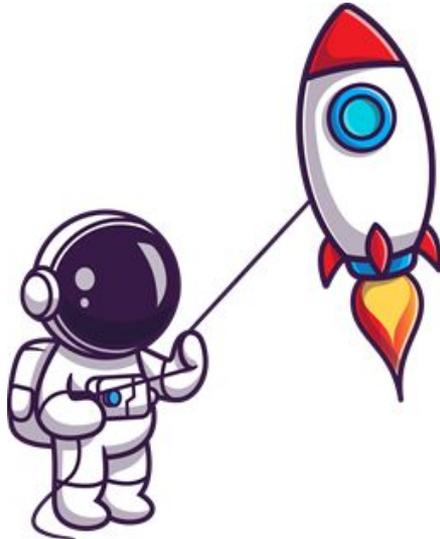




Diagrama de secuencia

- Los diagramas de secuencia muestran la interacción entre los **objetos** centrándose en la **secuencia de mensajes que envían y reciben**
- Tiene dos usos diferentes
 - **Forma de instancia:** describe un escenario específico, una posible interacción
 - **Forma genérica:** describe todas las posibles alternativas en un escenario. Puede incluir ramas, condiciones y bucles
- Un diagrama de secuencia representa una interacción como un diagrama bidimensional
 - La dimensión vertical es el eje de tiempos
 - La dimensión horizontal muestra la línea de vida (lifeline) de los objetos implicados en la interacción.



Diagrama de secuencia

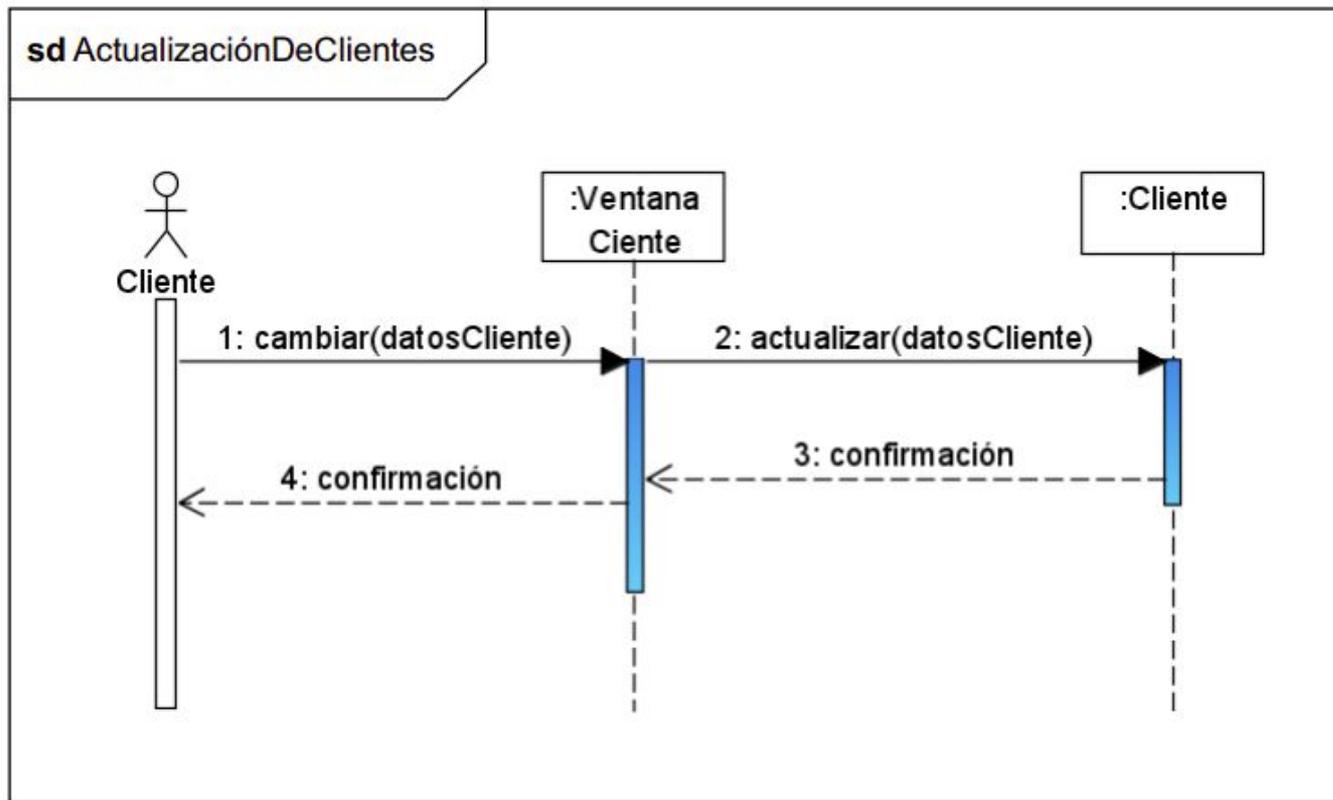




Diagrama de secuencia

- Un **mensaje** representa una comunicación entre objetos
- Transporta información para la realización de una acción. Cuando un objeto recibe un mensaje realiza una actividad.
- Los mensajes pueden ser señales, invocaciones a operaciones, llamadas a procedimientos remotos...

Notación:

- Se muestran como flechas entre las líneas de vida de los objetos
- Existen símbolos específicos para representar diferentes tipos de mensajes
- Pueden tener una signatura: nombre, parámetros y valor de retorno

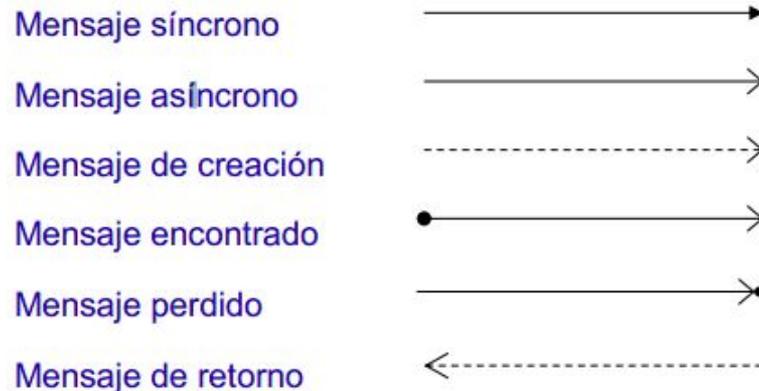


Diagrama de secuencia

Fragmentos

- **alt**: estructura alternativa
- **opt**: comportamiento opcional
- **loop**: bucle, comportamiento repetitivo.
- **par**: comportamientos paralelos
- **critical**: región crítica

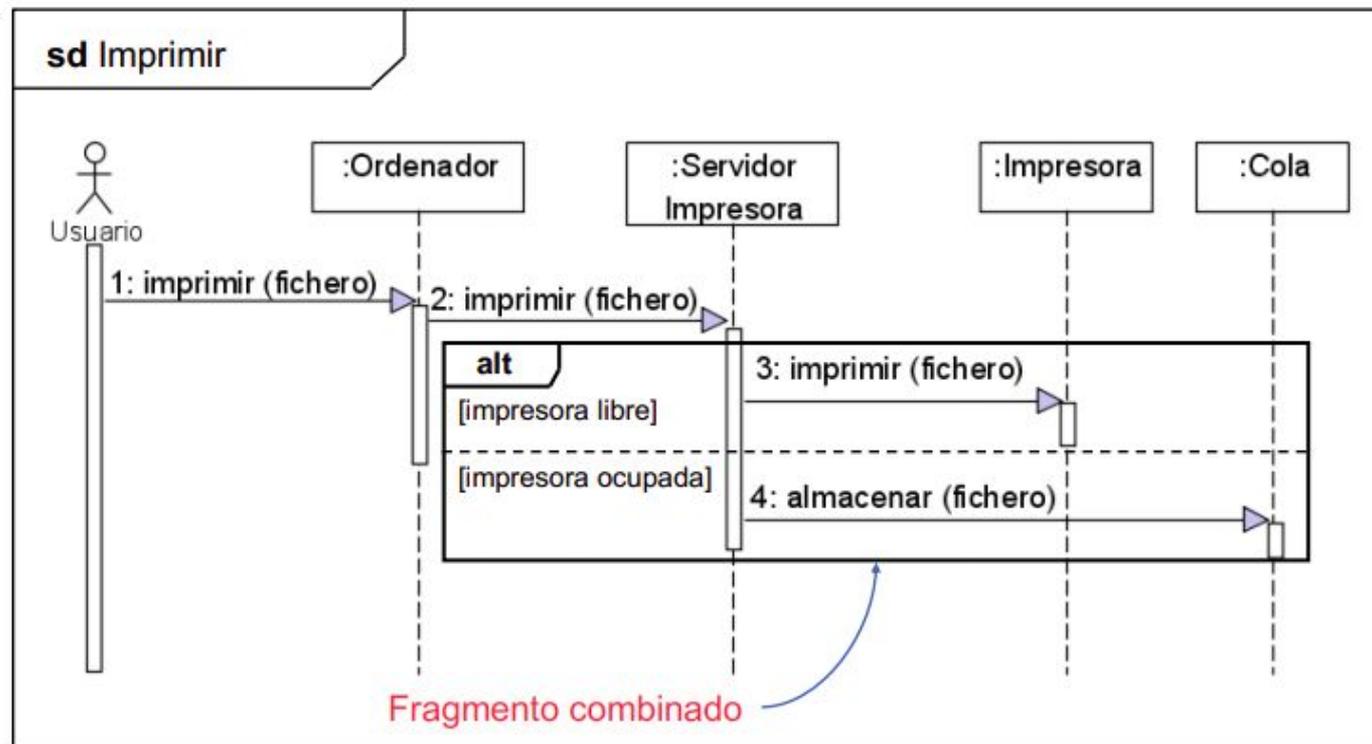




Diagrama de estados

- Las máquinas de estados describen los estados que un objeto puede tener durante su ciclo de vida, el comportamiento en esos estados y los eventos que causan los cambios de estado
- UML define dos tipos de máquinas de estados:
 - **De comportamiento:** capturan los ciclos de vida de los objetos, subsistemas y sistemas
 - **De protocolo:** se usan para especificar las transformaciones legales que pueden ocurrir en un clasificador abstracto como una interfaz o un puerto (protocolos de uso)

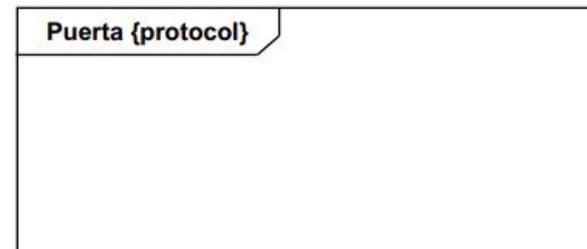
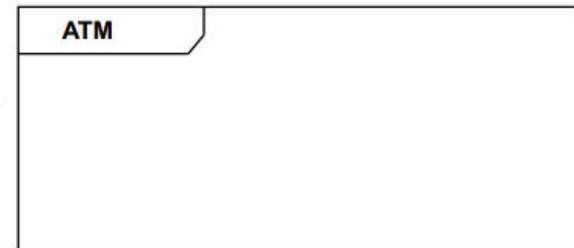


Diagrama de estados

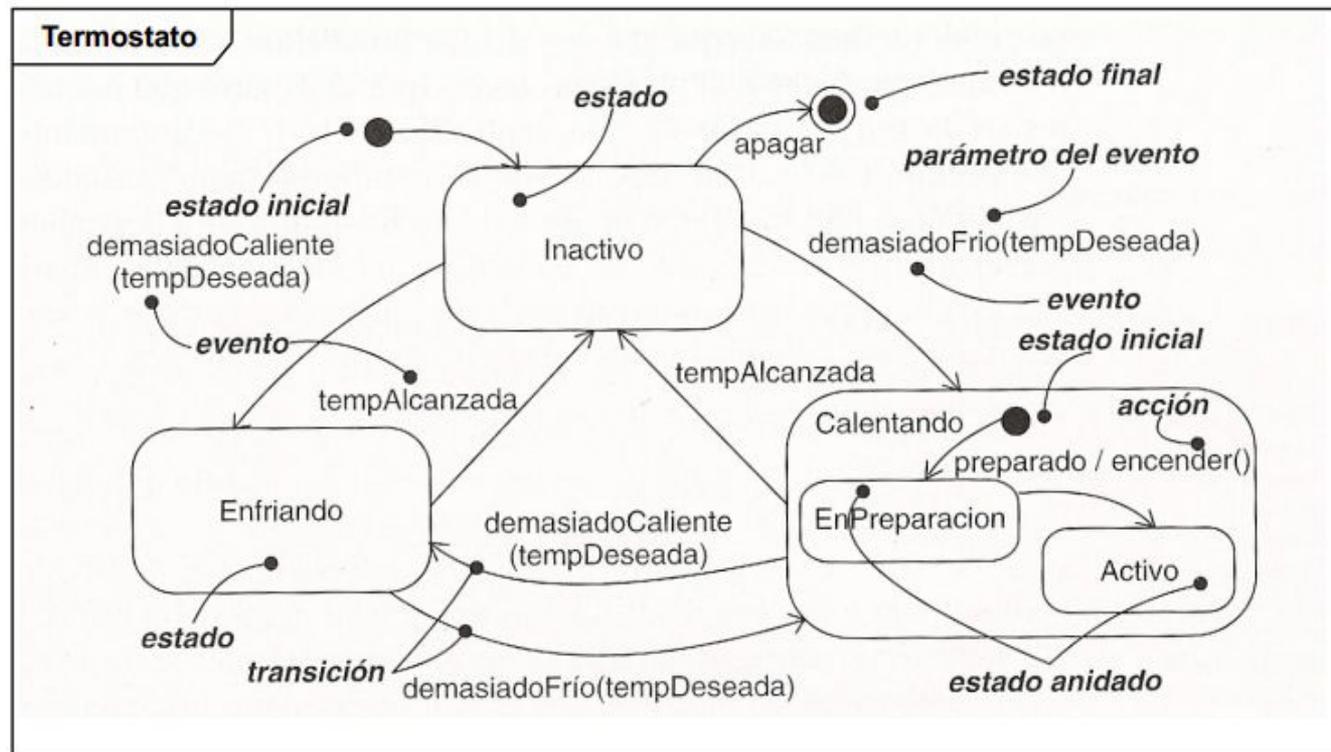


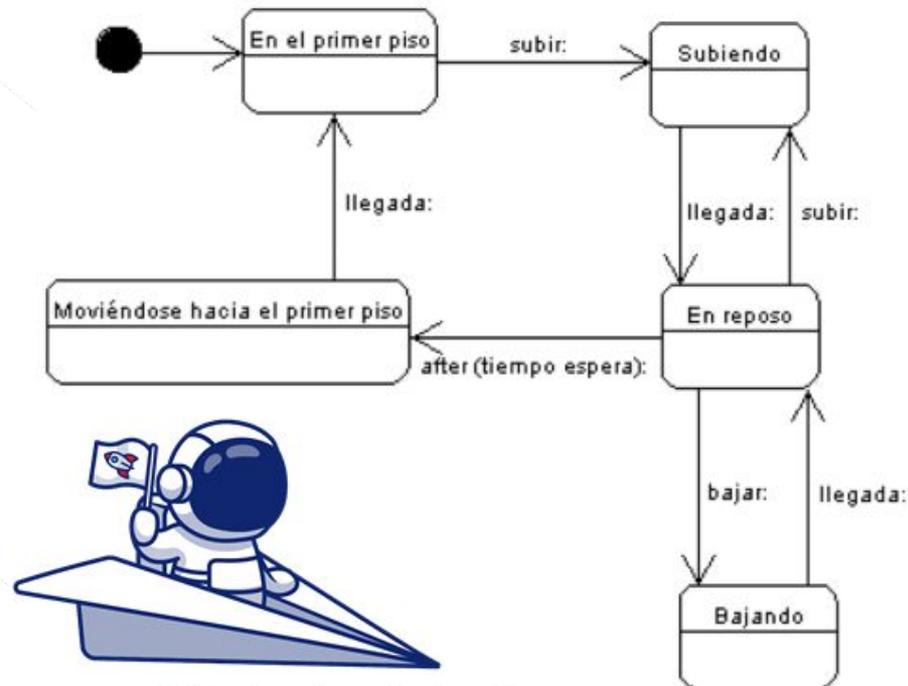
Diagrama de estados

- El estado de un objeto es consecuencia de las actividades que ha realizado previamente y está determinado por los valores de sus atributos y los enlaces con otros objetos

Las clases pueden tener un atributo específico que indique su estado

- Las **transiciones** de un estado a otro se producen cuando ocurre un evento
- Un **evento** es algo que sucede y que puede causar alguna acción. Son disparadores que activan las transiciones de estado

Una clase puede recibir o enviar eventos



Máquina de estados de un ascensor

Diagrama de estados

