

Programación 4

Análisis

Especificación del
Comportamiento del Sistema

Contenido

- Introducción
- Modelo de Casos de Uso
 - La Clase Sistema
- Interacciones con el Sistema
- Contratos de Software

Introducción

- Durante esta actividad de análisis se busca describir en forma precisa cuál debe ser el comportamiento esperado del sistema
- Se trabaja sobre el Modelo de Casos de Uso
 - Viendo al sistema como una unidad
 - Se definen protocolos que caractericen el uso del sistema por parte de los actores en cada escenario de los casos de uso
 - El comportamiento completo del sistema es especificado al especificar cada mensaje de los protocolos

Introducción (2)

- Cada escenario de los casos de uso a analizar es entendido en términos de una interacción entre los actores involucrados y el sistema
- Al describir el significado de cada uno de los mensajes identificados en cada interacción se está especificando el comportamiento del sistema

Introducción (3)

- Nos enfocamos en qué es lo que debe hacer el sistema ante cada mensaje
- La forma en cómo el sistema resuelve internamente un mensaje será definida durante la etapa de diseño

Modelo de Casos de Uso

Modelo de Casos de Uso

- Contenido:
 - Introducción
 - Breve descripción textual que sirve como introducción al modelo
 - Relevamiento de funcionalidades
 - Descripción textual de información no reflejada en el resto del modelo, por ejemplo:
 - Secuencias típicas en que los casos de uso son utilizados por los usuarios
 - Otras funcionalidades no capturadas en los casos de uso

Modelo de Casos de Uso (2)

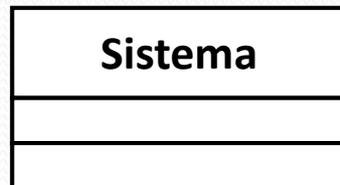
- Contenido (cont.)
 - Actores
 - Todos los actores detectados para el sistema
 - Casos de uso
 - Todos los casos de uso definidos
 - Relaciones
 - Todas las asociaciones entre actores y CU
 - Comportamiento
 - Especificación del comportamiento de cada caso de uso en el modelo, el cual está definido por: Eventos del Sistema y Contratos de Software

La Clase Sistema

- Durante esta actividad el sistema será considerado como un objeto:
 - Que es instancia de una clase Sistema
 - Que tiene operaciones (puede recibir mensajes)
 - Que tiene un estado
- En todo Modelo de Casos de Uso se asume que existe una clase Sistema

La Clase Sistema (2)

- Existe una **única instancia** de esta clase la cual representa al “sistema entero”

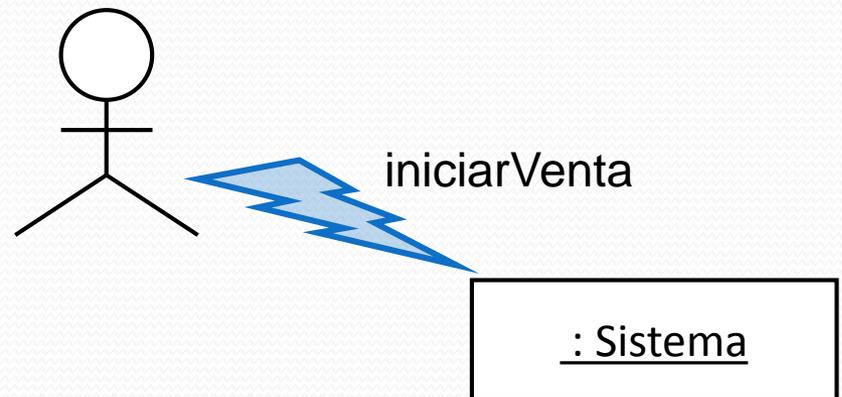
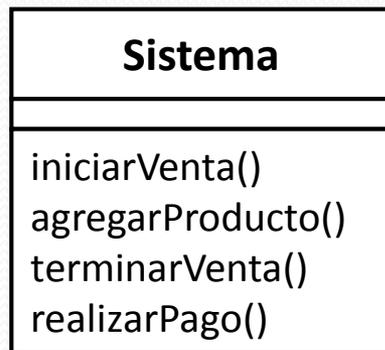


La Clase Sistema (3)

- Las **operaciones** de esta clase permiten que el sistema reciba mensajes de los actores:
 - Se identifican al definir los protocolos que representan los escenarios de los diferentes casos de uso
 - Durante el análisis no se busca diseñarlas
 - Su semántica es definida en términos del efecto que deben tener sobre el estado del sistema

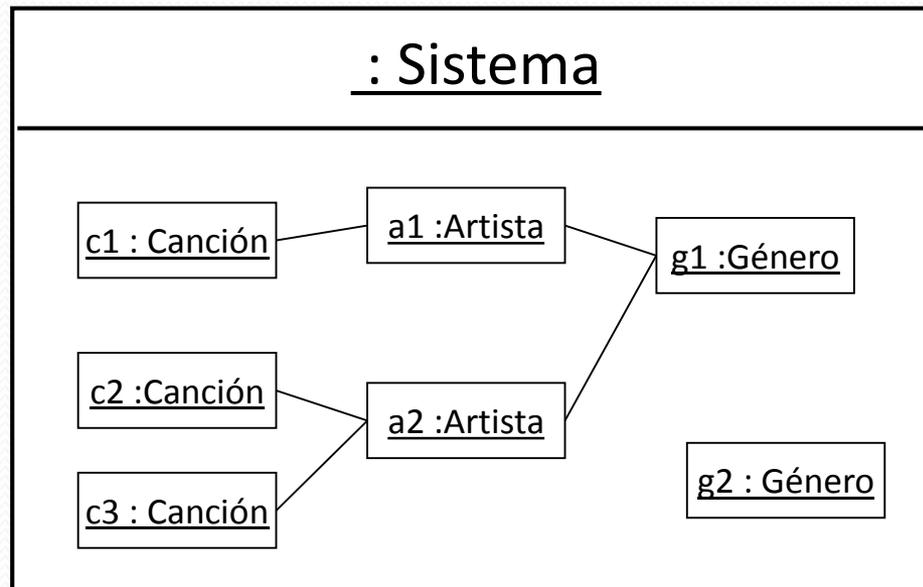
La Clase Sistema (4)

- Un actor puede enviar mensajes al sistema “invocando” sus operaciones



La Clase Sistema (5)

- En esta actividad el **estado** del sistema se asume como una configuración de objetos válida respecto al Modelo de Dominio



Interacciones con el Sistema

Interacciones con el Sistema

- Los casos de uso describen la forma en que actores utilizan al sistema para cumplir con sus objetivos
- Es necesario expresar estas ideas desde un punto de vista técnico
- Para ello se definen protocolos que determinan la interacción entre los actores y el sistema, ya sea para uno o varios escenarios de un caso de uso
 - Cada protocolo es expresado mediante un Diagrama de Secuencia del Sistema (DSS)

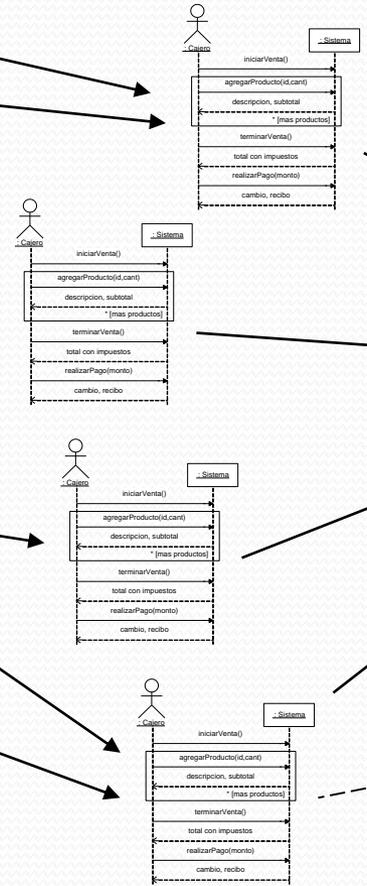
Interacciones con el Sistema (2)

Caso de Uso 1

- Esc. Típico
- Esc. Alternativo 1
- ⋮
- Esc. Alternativo n

Caso de Uso 2

- Esc. Típico
- Esc. Alternativo 1
- ⋮
- Esc. Alternativo m



Los DSSs se incluyen en la secc. "Comportamiento" del modelo

Modelo de Casos de Uso

Un DSS que define la interacción entre los actores y el sistema en el escenario dado

Interacciones con el Sistema

Eventos del Sistema

- Un evento del sistema
 - Es un estímulo externo,
 - Es generado por un actor
 - Ante el cual el sistema debe reaccionar
- Las acciones de los actores (sobre el sistema) descritas en los casos de uso sugieren los eventos del sistema
- Es necesario considerar la definición de evento del sistema para identificarlos

Interacciones con el Sistema

Eventos del Sistema (2)

- Ejemplo:

- *“El Usuario ingresa el nombre de la canción y el año”*
Es un estímulo externo generado por un actor ante el cual el sistema debe reaccionar
⇒ **Es un evento del sistema**
- *“El Usuario inicia una sesión en el sistema operativo”*
Es un evento externo pero no afecta al sistema
⇒ **No es un evento del sistema**

Interacciones con el Sistema

Operaciones del Sistema

- Los eventos del sistema disparan una operación del sistema
- Estas operaciones son ejecutadas por la “instancia sistema” en respuesta a la ocurrencia de un evento del sistema
- Las operaciones del sistema relativas a uno o varios escenarios de un caso de uso permiten definir la interacción entre los actores y el sistema

Interacciones con el Sistema

Operaciones del Sistema (2)

- Las operaciones del sistema pueden tener asociados parámetros

- Ejemplo:

- *“El usuario ingresa con el teclado el cuerpo y el asunto del email”* representa un evento que dispara la operación

escribirEmail(cuerpo:String, asunto:String)

- *“El usuario selecciona varios contactos de una lista desplegable”* representa un evento que dispara la operación

elegirContactos(contactos: Set(String))

Interacciones con el Sistema

Diag. de Secuencia del Sistema

- Es un artefacto incluido en el Modelo de Casos de Uso que define e ilustra la interacción entre los actores y el sistema en uno o varios escenarios de un CU
- Incluye:
 - Una instancia representando a cada participante (sistema y actores)
 - Los mensajes enviados entre ellos en el/los escenario/s correspondiente/s (con sus respuestas)

Interacciones con el Sistema

Diag. de Secuencia del Sistema (2)

- Un Diagrama de Secuencia del Sistema puede ser construido para:
 - Un escenario de un Caso de Uso
 - Varios escenarios de un Caso de Uso
- Un criterio para decidir entre estas alternativas será la complejidad de estos escenarios y la simplicidad (o no) del DSS resultante

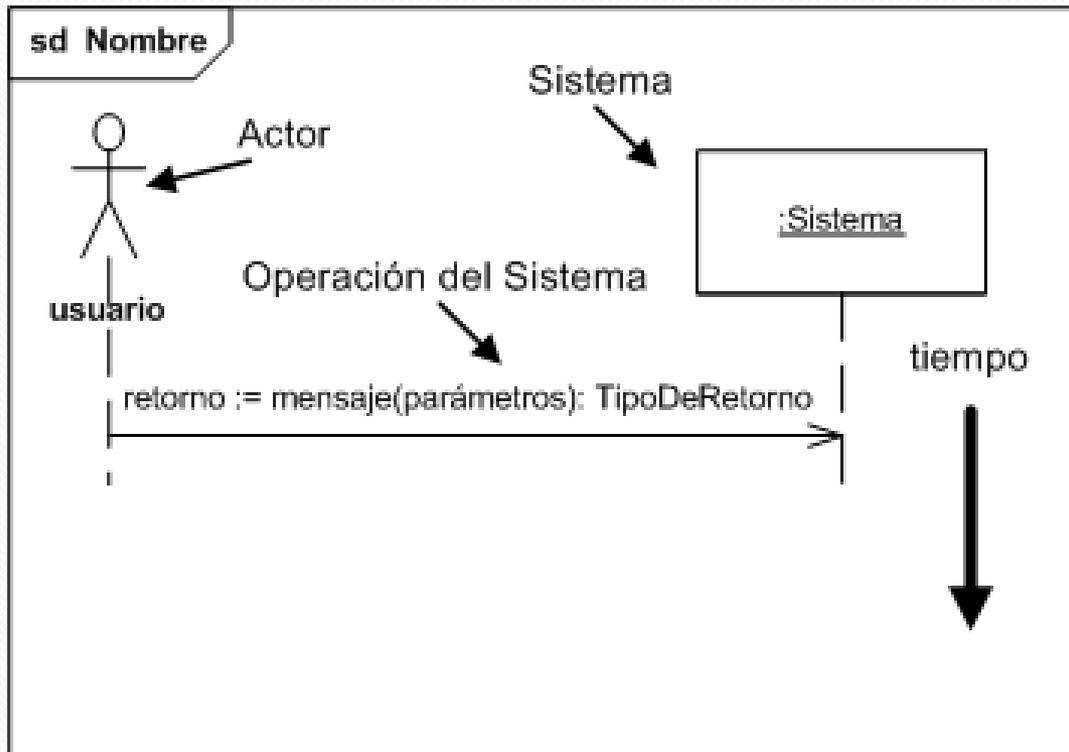
Diag. de Secuencia del Sistema (3)

- Los diagramas de secuencia del sistema definen la conversación entre los actores y el sistema, enfocándose en los mensajes que el sistema recibe
- Sería posible incluir además mensajes enviados desde el sistema hacia los actores:
 - Sin embargo esto no forma parte del conjunto de servicios que el sistema brinda (y cuya especificación es el objetivo de la presente actividad)

Interacciones con el Sistema

Diag. de Secuencia del Sistema (4)

- Notación:



Diag. de Secuencia del Sistema (5)

- Tipo de un atributo (o tipo de retorno) cuando no es simple, sino que es una colección de cierto datatype
Tipo

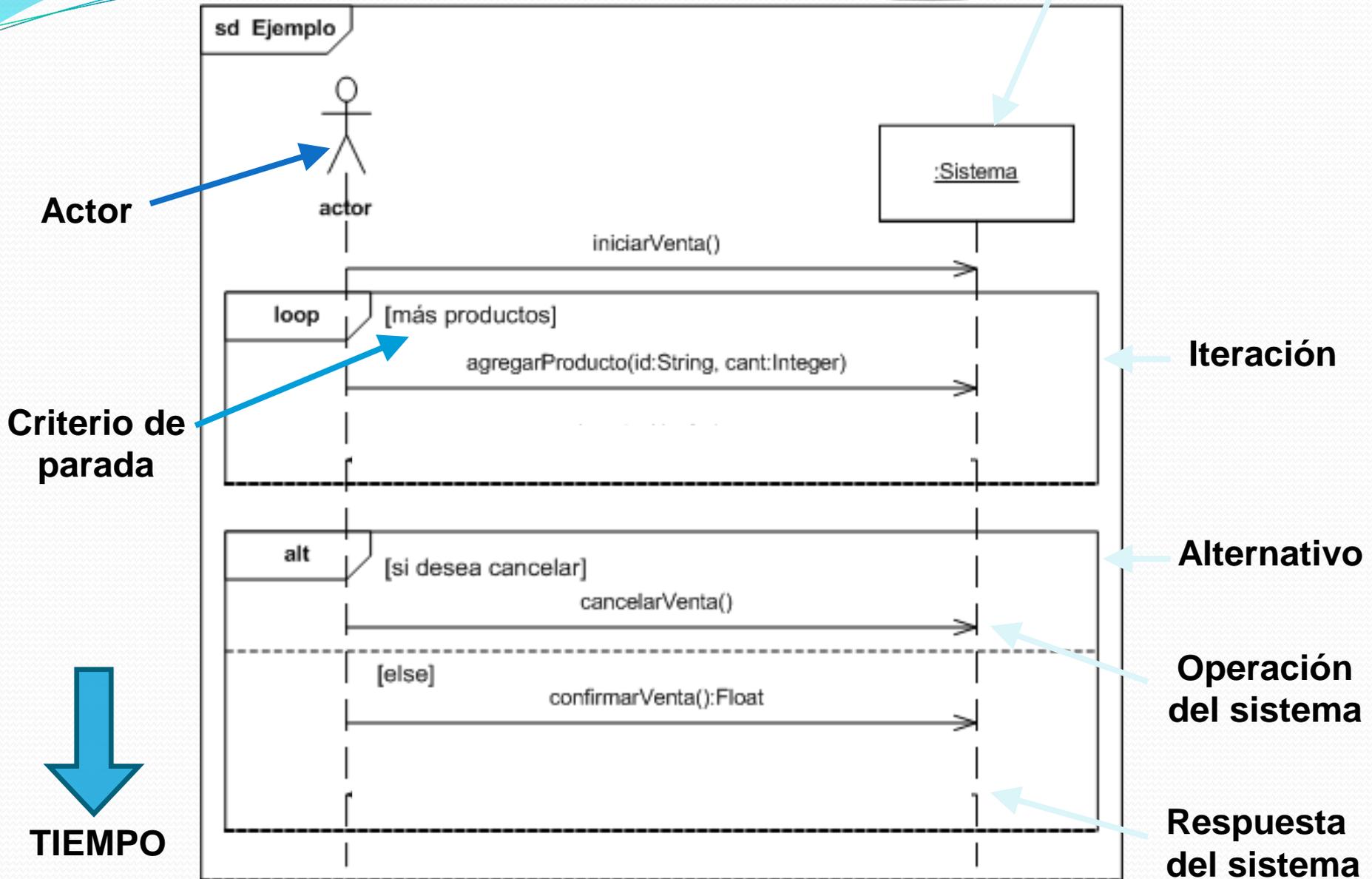
ingresarNombre (nombre : String) : DataCancion

- Puede utilizarse **Set (Tipo)**, si se quiere pasar como parámetro (o retorno) varios elementos de un solo tipo (por ejemplo: una lista de elementos)

listarCanciones () : Set (DataCancion)

Nombre del Diagrama

Sistema



Sugerencias

- Definición de un DSS:
 1. Incluir una instancia que represente al sistema como una unidad
 2. Identificar cada actor que participe en el/los escenario/s considerado/s e incluir una instancia para cada uno
 3. De la descripción del caso de uso identificar aquellos eventos que los actores generen y sean de interés para el sistema e incluir cada uno de ellos como un mensaje

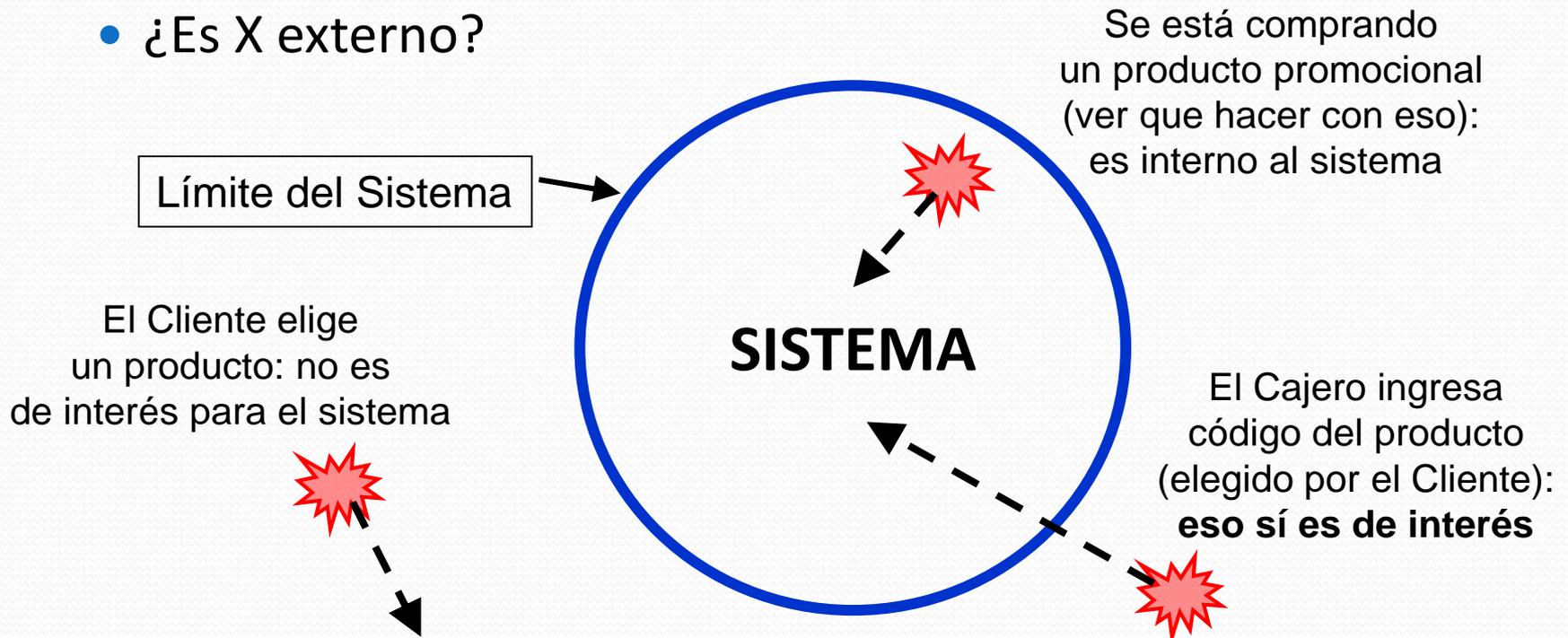
Sugerencias (2)

- Límite del sistema:
 - Para identificar eventos del sistema es útil pensar en el límite del sistema
 - El límite suele determinarse para que coincida con el sistema de software (y el de hardware también)
 - Buscar aquello que ocurra fuera de ese límite y que además lo atraviese

Interacciones con el Sistema

Sugerencias (3)

- Límite del sistema (cont.):
 - ¿Es responsabilidad del sistema reaccionar ante el evento X?
 - ¿Es X externo?



Sugerencias (4)

- Límite del sistema (cont.):
 - Las operaciones del sistema deberían retornar exclusivamente datatypes o tipos primitivos

Sugerencias (5)

- Memoria del Sistema:
 - El sistema puede (o no) tener memoria:
 - Sin memoria, los mensajes son independientes
 - Con memoria, cada mensaje puede “recordar” la información utilizada en un estado previo del sistema
 - Debe indicarse claramente si el sistema tiene o no memoria, y en caso de tenerla, qué información recuerda

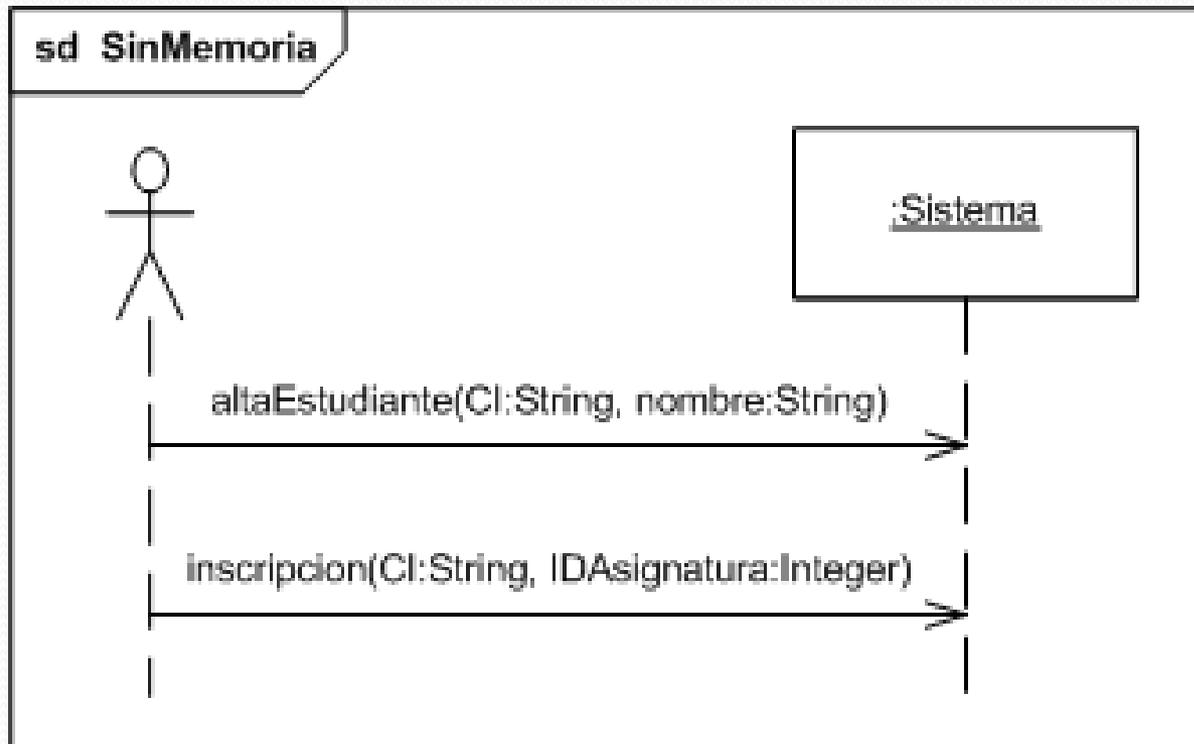
Sugerencias (6)

- Memoria del Sistema (cont.):
 - Para indicar la memoria de un sistema, generalmente basta con indicarlo en el nombre del diagrama y mediante la utilización de notas en el diagrama
 - Alternativamente, puede utilizarse un diagrama de estructura estática en aquellos casos en que interese indicar una estructura compleja de dicha memoria

Interacciones con el Sistema

Sugerencias (7)

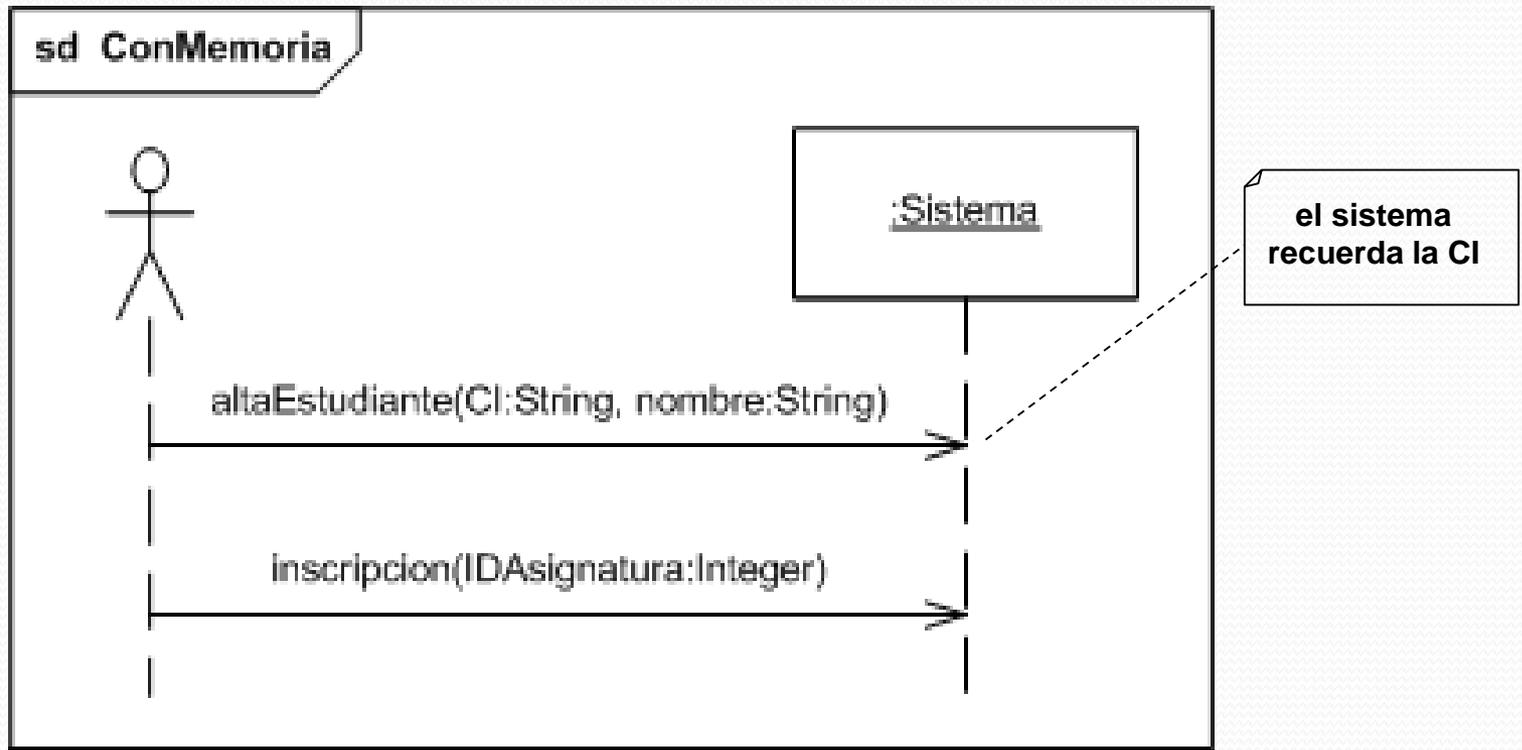
- Ejemplo: DSS sin memoria



Interacciones con el Sistema

Sugerencias (8)

- Ejemplo: DSS con memoria



Interacciones con el Sistema

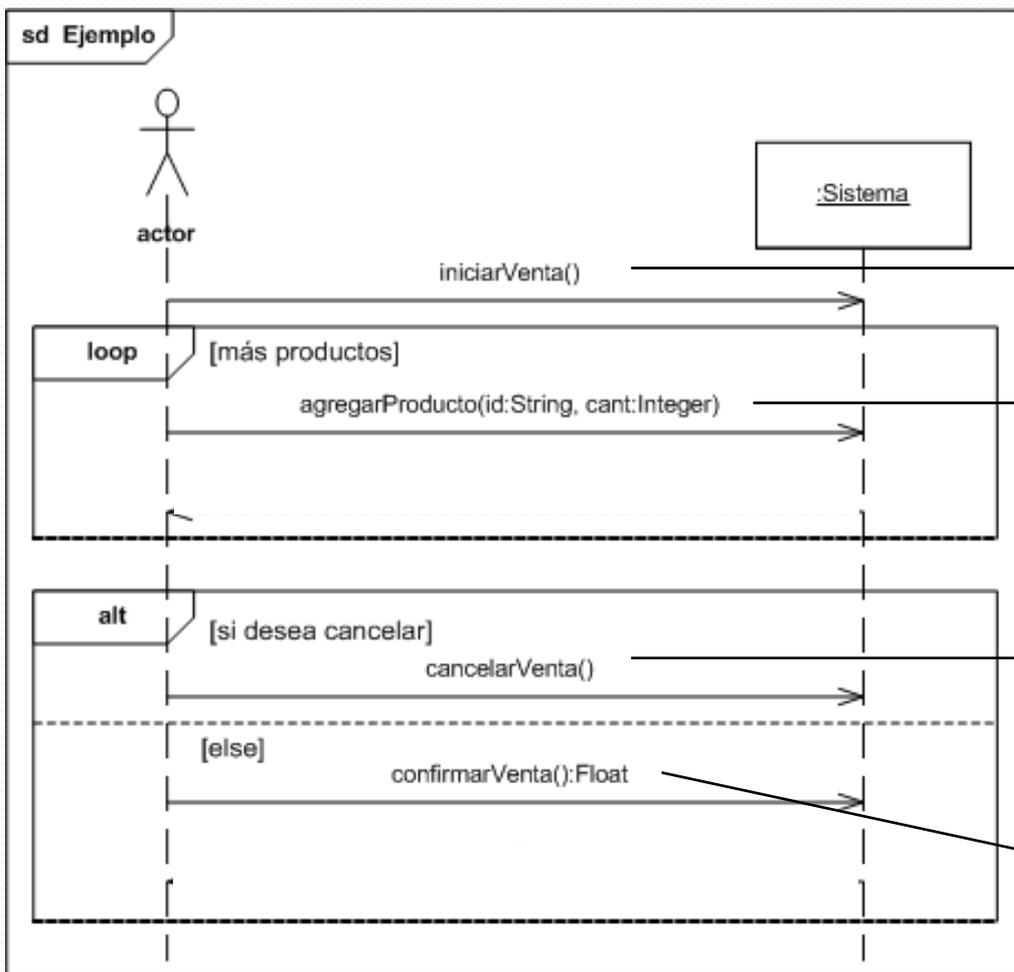
Errores Comunes

- Envío de mensajes hacia el usuario
- Desconocer la memoria del sistema
- No especificar *data types* utilizados
- Sobrecargar de información un diagrama de secuencia pudiendo realizar varios de ellos
- No indicar tipo de parámetros ni valor de retorno de los mensajes

¿Qué Sigue?

- Una vez identificadas las operaciones del sistema es posible especificar su comportamiento
- Esta especificación expresa el efecto que una operación tendrá sobre el sistema
- Para ello se realizará un Contrato de Software para cada operación del sistema

¿Qué Sigue? (2)



Los contratos se incluyen en la secc. "Comportamiento" del modelo

Contrato:
iniciarVenta()

Contrato:
agregarProducto()

Contrato:
cancelarVenta()

Contrato:
confirmarVenta()

Modelo de Casos de Uso

Contratos de Software

Contratos de Software

- Un contrato de software especifica el comportamiento o efecto de una operación
- La especificación es declarativa y no imperativa
- Esta técnica está basada en las ternas de Hoare en las que:
 - Se describen propiedades del resultado, en lugar de dar un conjunto de pasos o instrucciones que indiquen cómo calcularlo

Enfoque de Contratos (1)

	Obligaciones	Derechos
Consumidor	Satisfacer precondiciones	Obtener la postcondición satisfecha
Proveedor	Satisfacer postcondiciones	Procesamiento asumiendo como satisfecha la precondición

Enfoque de Contratos (2)

- El Consumidor se compromete a satisfacer la precondición al invocar la operación:
 - Si la satisface: tiene derecho a exigir que la postcondición se satisfaga
 - Si no la satisface: no se le garantiza la correctitud del resultado de la invocación (se lanzan excepciones)
- Por esta razón es responsabilidad del Consumidor saber cuándo invocar a la operación (y manejar en forma adecuada el resultado)

Enfoque de Contratos (3)

- El Proveedor se compromete a satisfacer la postcondición al finalizar la operación solamente cuando la precondition fue satisfecha al momento de la invocación
- El compromiso no comprende el caso en que la precondition no fue satisfecha:
 - En ese caso el Proveedor puede devolver un valor arbitrario y el Consumidor tiene que aceptarlo y saber qué hacer con él

Enfoque de Contratos (4)

- Consumidor:
 - Prefiere precondiciones débiles: implica menos trabajo
 - Prefiere postcondiciones fuertes: implica más resultados
- Proveedor:
 - Prefiere precondiciones fuertes: implica menos preocupaciones
 - Prefiere postcondiciones débiles: implica menos trabajo

Enfoque de Contratos (5)

- Precondición:
 - Es a lo que debe acceder el Consumidor para obtener el resultado deseado
 - Es lo que debe exigir el Proveedor para llegar al resultado
- Postcondición:
 - Es a lo que accederá el Consumidor
 - Es a lo que se compromete el Proveedor

Enfoque de Contratos (6)

- Tanto las Pre- como las Post- las determina el Proveedor
- El Consumidor:
 - Viendo la Post- sabe qué va a obtener (sin saber cómo)
 - Viendo la Pre- sabe a cambio de qué obtiene el resultado

Contratos de Operaciones

- Los contratos se pueden realizar para operaciones de cualquier tipo de clase
- En esta actividad las realizaremos para operaciones del sistema
- Para una operación X tendremos $\{P\}S\{Q\}$
 - P es la precondition de X (especificada)
 - S es el programa que implementa X (a ser diseñado más adelante en la etapa de Diseño)
 - Q es la postcondición de X (especificada)

Contratos de Operaciones (2)

- ¿Quién utiliza el contrato (partes P y Q) de una operación?
 - Un diseñador de nuestro equipo que deba diseñar S
 - Para saber qué es lo que tiene que lograr su diseño de la operación
 - En función de lo anterior para decidir cómo será el diseño de la operación (parte S)
 - Un desarrollador de otro equipo que deba invocar la operación (el diseño o implementación de S no es su responsabilidad)
 - Para saber qué es lo que la operación hace sin tener que ver el diseño o la implementación de S

Condiciones

- ¿En qué términos se expresan las pre- y postcondiciones?
¿Y para el caso particular de operaciones del sistema?
- En términos generales estas condiciones refieren al estado del sistema antes y después de la invocación a la operación
 - Las precondiciones refieren además a los argumentos de la operación
 - Las postcondiciones refieren además al valor retornado por la operación (si existe)

Condiciones (2)

- Las Precondiciones refieren al momento previo a la invocación y expresan condiciones sobre
 - Los valores de los parámetros de la operación
 - El estado del sistema:
 - La creación de objetos
 - La destrucción de objetos
 - La conexión de objetos
 - La desconexión de objetos
 - La modificación del valor de atributos de objetos

Condiciones (3)

- Las Postcondiciones refieren al momento posterior a la invocación expresan condiciones sobre
 - El valor de retorno (si corresponde)
 - El estado del sistema:
 - La creación de objetos
 - La destrucción de objetos
 - La conexión de objetos
 - La desconexión de objetos
 - La modificación del valor de atributos de objetos

Condiciones (4)

- **Creación de objetos:**
 - Pre: Declarar que el objeto no existe
 - Post: Declarar que el objeto existe
- **Dstrucción de objetos:**
 - Pre: Declarar que el objeto existe
 - Post: Declarar que el objeto no existe y que todos los objetos que estaban conectados a él ya no lo están

Condiciones (5)

- **Conexión de objetos:**

- Pre: Declarar que los objetos no están conectados
- Post: Declarar que los objetos están conectados

- **Desconexión de objetos:**

- Pre: Declarar que los objetos están conectados
- Post: Declarar que los objetos no están conectados

- **Modificación del valor de atributos de objetos:**

- Pre: Declarar que el objeto exista
- Post: Declarar que el atributo del objeto tiene el valor dado

Condiciones (6)

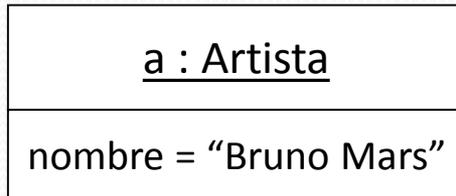
- Ejemplo para operación

`crearCancion(nombreCancion:string, nombreArtista:string)`

- **Precondición:** No existe un objeto Canción de nombre `nombreCancion` en el sistema
- **Precondición:** Existe un objeto Artista `a` de nombre `nombreArtista`
- **Postcondición:** Se crea una nueva instancia de Cancion `c` de nombre `nombreCancion` y con atributo `id` autogenerado tal que el `id` sea único en el sistema.
- **Postcondición:** Hay un link entre `c` y `a`
- Se puede derivar que la operación crea al objeto de tipo Canción y lo conecta con el objeto Artista

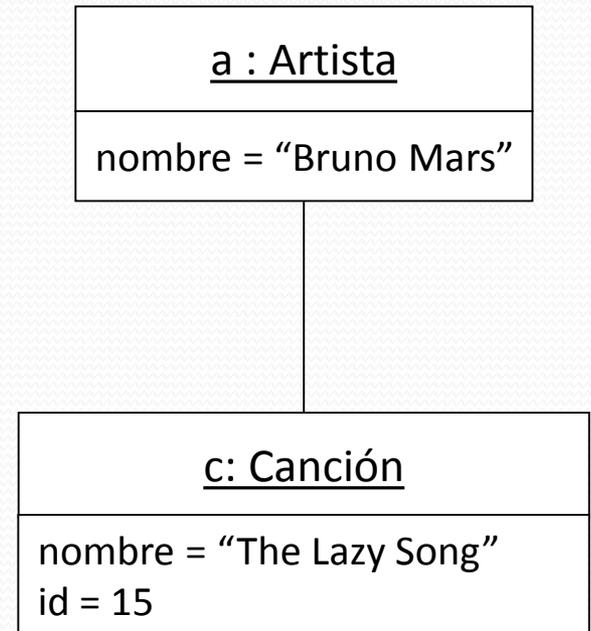
Condiciones (7)

σ_1



σ

2



Se pasa del estado σ_1 al estado σ_2 mediante la ejecución de `crearCancion("The Lazy Song", "Bruno Mars");`

Condiciones (8)

- Expresa condiciones sobre el estado inicial y sobre el estado final que indican qué es lo que la operación hace, pero no cómo lo hace
- NO dice cómo debe implementarse la operación del sistema **crearCancion()**

Estructura de Contratos

- Un contrato es un artefacto textual que se incluye en la sección 'Comportamiento' del Modelo de Casos de Uso
- Está estructurado de la siguiente forma:
 - **Firma:** Cabezal sintáctico de la operación
 - **Parámetros:** Descripción de los parámetros de la operación
 - **Responsabilidades:** Descripción de las responsabilidades, una idea de lo que debe realizar la operación

Estructura de Contratos (2)

- Estructura (cont.)
 - **Referencias cruzadas:** Caso(s) de Uso a los que pertenece la operación
 - **Salida:** Resultado de la operación (sólo si es una función)
 - **Precondición:** Descripción del estado de la instancia del sistema a la que se le aplicará la operación, y otras condiciones que sea necesario asumir previo a la aplicación (por ejemplo, con respecto a los parámetros)

Estructura de Contratos (3)

- Estructura (cont.)
 - **Postcondición:** Descripción del estado de la instancia del sistema a la que se le aplicó la operación
 - **Snapshots:** (Opcional)
 - Pares de *snapshots* que ejemplifiquen el estado de la instancia a la que se le aplicó la invocación, previo y posterior a la invocación
 - La invocación concreta que produce el cambio ejemplificado (mostrando los parámetros efectivos)

Errores Comunes

- Incluir invariantes como postcondiciones
- Omitir el resultado de una operación como postcondición

¿Qué Sigue?

- Hasta el momento se tienen identificadas y especificadas las operaciones del sistema para todos los casos de uso definidos
- Es posible ahora realizar un diseño en el que
 - Se identifiquen los objetos que realmente participarán en la solución
 - Se definan interacciones entre dichos objetos tal que cada una cumpla un contrato correspondiente a una operación del sistema