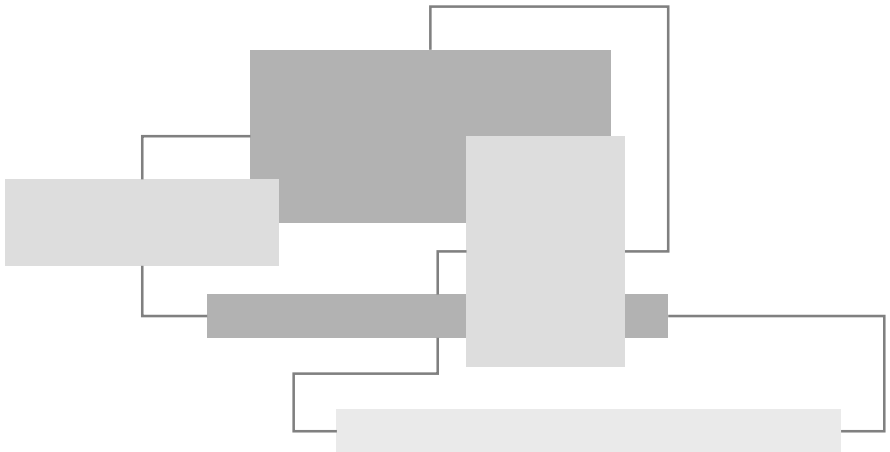
tema 2 - ingeniería de requerimientos

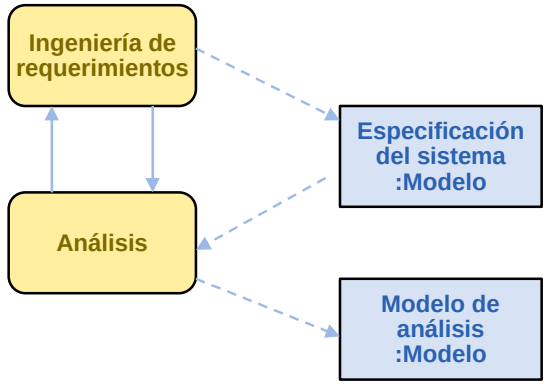


enrique barreiro
departamento de informática
universidade de vigo

escuela superior de ingeniería informática
ingeniería del software de gestión

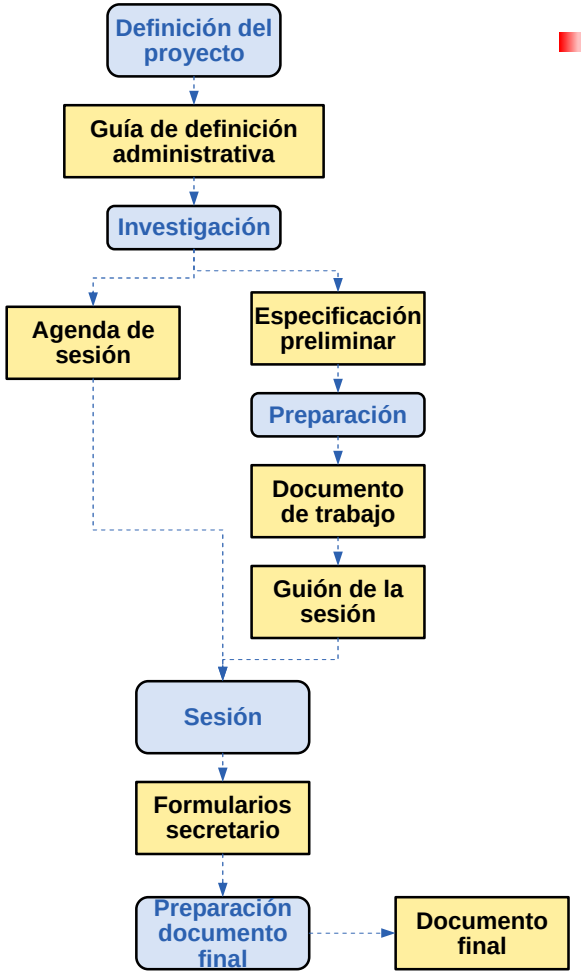
- errores en la especificación de requerimientos
- los requerimientos precisan comunicación entre desarrolladores, clientes y usuarios:
- errores: se descubren tarde y son caros de corregir a posteriori
  - falta de funcionalidad
  - funcionalidad mal especificada
  - interfaces confusas o inútiles
  - funcionalidad obsoleta
- los analistas
  - construyen un modelo del dominio de la aplicación observando a los usuarios en su entorno
  - seleccionan una representación comprensible para clientes y usuarios (por ejemplo, casos de uso)
  - validan el modelo del dominio construyendo prototipos de la interfaz y buscando retroalimentación con los usuarios y clientes.

- la obtención de requerimientos
  - identificación de un área del problema
  - definición de un sistema que soluciona el problema y sirve como contrato con el cliente: *especificación del sistema*
  - en el análisis se estructura y formaliza la especificación para producir el *modelo de análisis*.
  - *especificación* vs *modelo de análisis*:
    - representan la misma información
    - difieren en el lenguaje y la notación
      - especificación: lenguaje natural
      - modelo de análisis: notación formal o semiformal
    - sirven de elemento de comunicación
      - especificación: comunicación con cliente y usuarios
      - modelo de análisis: comunicación entre desarrolladores



- actividades de la obtención de requerimientos
  - identificación de actores
  - identificación de escenarios
  - identificación de casos de uso
  - refinamiento de casos de uso
  - identificación de relaciones entre casos de uso
  - identificación de requerimientos no funcionales

- sistema de entrevistas (preguntas-respuestas)
  - adecuado para las primeras tomas de contacto
  - conveniente comenzar por preguntas de *contexto libre*, para entender el problema, personas interesadas en la solución, naturaleza de ésta, y efectividad de la reunión
    - preguntas centradas en el cliente, objetivos globales y beneficios
      - ¿quién solicita el trabajo?
      - ¿quién utilizará la solución?
      - ¿cuál será el beneficio económico de una buena solución?
      - ¿existen otras alternativas a esta solución?
    - preguntas sobre el problema y la solución
      - ¿qué entiende el cliente por una solución "correcta"?
      - ¿qué problemas afrontará esta solución?
      - ¿en qué entorno se va a implantar la solución?
      - ¿existen restricciones o aspectos de rendimiento importantes?
    - preguntas sobre la efectividad de la reunión
      - ¿es usted la persona adecuada para responder a estas preguntas? ¿Sus respuestas son "oficiales"?
      - ¿son relevantes mis preguntas para su problema?
      - ¿hay alguien más que pueda proporcionar información adicional?
      - ¿hay algo más que debería preguntar?
  - problemas de las entrevistas
    - malentendidos
    - omisión de información
    - mala relación de trabajo ("nosotros-ellos")
    - ...



- diseño conjunto de aplicaciones (JAD, "joint application design")
  - desarrollado por IBM a finales de los setenta
  - una sesión de trabajo con participación de todos los involucrados
  - resultado de la sesión: documento de especificación que incluye definiciones de elementos de datos, flujos de trabajo y pantallas de interfaz
  - representa un acuerdo entre usuarios, clientes y desarrolladores y minimiza los cambios posteriores de requerimientos
  - actividades
    - definición del proyecto: el coordinador se entrevista con gerentes y clientes para determinar objetivos y alcance del proyecto, creando la "guía de definición administrativa".
    - investigación: entrevista con usuarios, recopilación de información del dominio, descripción de flujos de trabajo y asuntos a tratar en la reunión. Se crea la "agenda de sesión" y la "especificación preliminar".
    - preparación: el coordinador crea un "documento de trabajo" o primer borrador del documento final.
    - sesión: el coordinador guía al equipo para crear la especificación del sistema en una reunión que puede durar varios días. Se definen los flujos de trabajo, elementos de datos, pantallas, informes,... Las decisiones se documentan en unos formularios.
    - documento final: el coordinador prepara el "documento final" usando los "formularios" y se distribuye a los asistentes para su revisión. Reunión para discutir revisiones y finalizar el documento.

- modelo FURPS+ de requisitos:
  - Funcionalidad (*Functional)*: características, capacidades y seguridad.
  - Facilidad de uso (*Usability*): factores humanos, ayuda, documentación.
  - Fiabilidad (*Reliability*): frecuencia de fallos, capacidad de recuperación de un fallo y grado de previsión.
  - Rendimiento (*Performance*): tiempos de respuesta, productividad, precisión, disponibilidad, uso de los recursos.
  - Soporte (*Supportability*): adaptabilidad, facilidad de mantenimiento, internacionalización, configurabilidad.
  - El "+" indica requisitos adicionales como:
    - Implementación: limitación de recursos, lenguajes y herramientas, hardware,...
    - Interfaz: restricciones referentes a la interacción con sistemas externos
    - Operaciones: gestión del sistema en su puesta en marcha
    - Empaquetamiento
    - Legales: licencias,...
- Otra clasificación:
  - Requisitos funcionales
  - Requisitos no funcionales
  - Requisitos de implementación

- requerimientos funcionales
  - describen las interacciones entre el sistema y su entorno (usuarios u otros sistemas), sin tener en cuenta cuestiones de implementación.
  - se estudian y representan en el Modelo de Casos de Uso

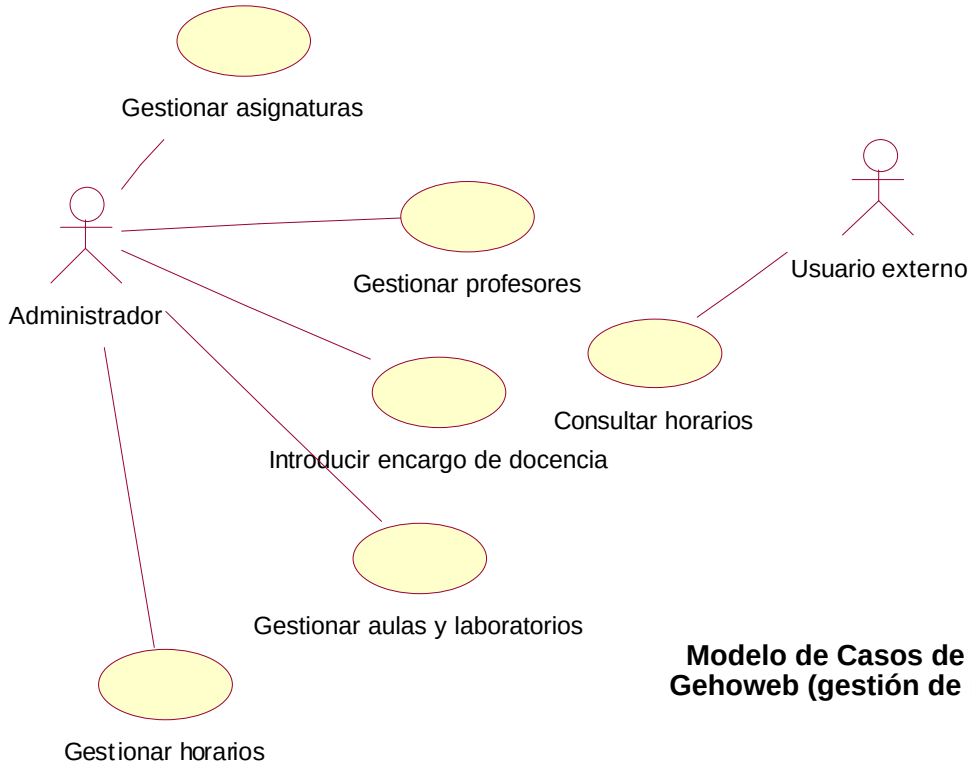
Requerimientos funcionales de GeHoWeb.

GeHoWeb es un sistema para la gestión de horarios de la Escuela Superior de Ingeniería Informática (ESEI). El administrador del sistema, que se tendrá que identificar al acceder al mismo, es el encargado de introducir las asignaturas que se imparten en cada curso, así como los datos del encargo de docencia anual (grupos de teoría y práctica de cada asignatura).

Además, el sistema permite introducir los datos de las aulas de teoría (ubicación y aforo) y de prácticas (ubicación, sistemas operativos, software,...).

La configuración del horario se lleva a cabo directamente sobre una plantilla horaria semanal, en la que cada casilla representará una hora en un determinado día de la semana. Cuando el administrador pulsa esa casilla se mostrarán las asignaturas del curso que se esté configurando en ese momento. Una vez escogida las asignaturas se mostrarán los grupos de teoría y práctica a los que todavía no se les ha asignado un horario. Al escoger un grupo se muestran las aulas disponibles (si es un grupo de teoría) o los laboratorios que cumplen las restricciones de sistemas operativos establecidas para esa materia y que no están ocupados a esa hora.

El sistema podrá ser consultado por cualquier usuario, que podrá consultar el horario de una asignatura, un curso, o de un aula o laboratorio concretos.



Modelo de Casos de Uso de Gehoweb (gestión de horarios)

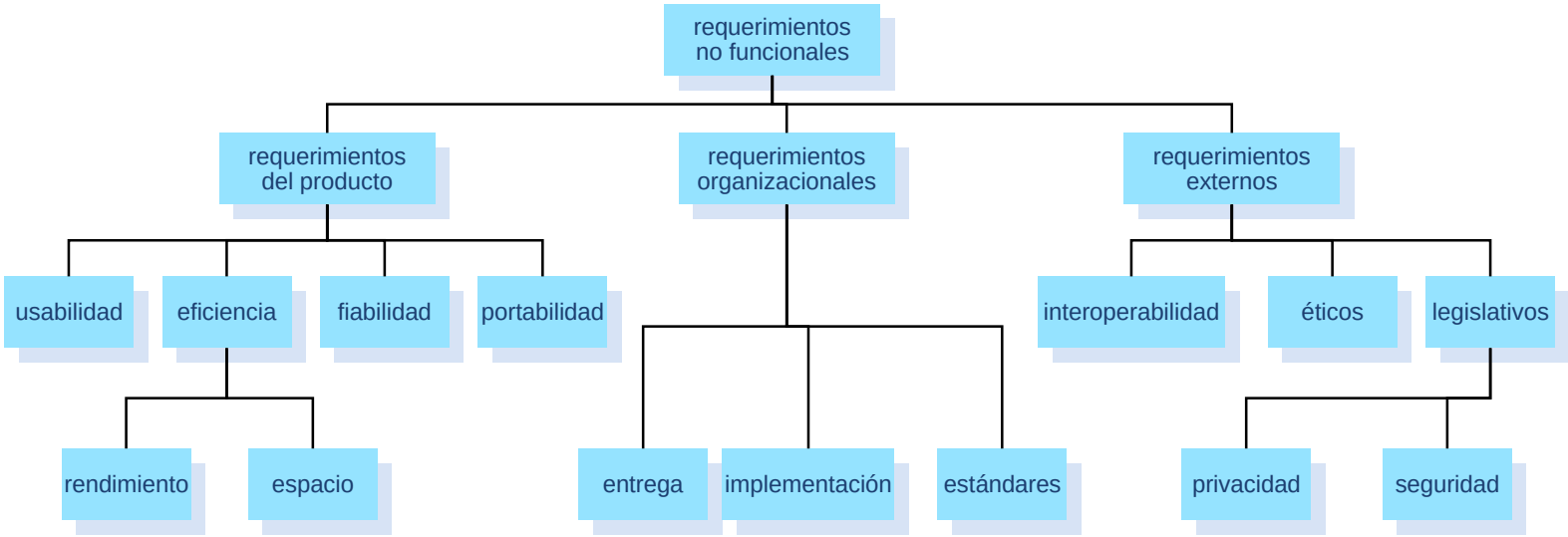
- requerimientos no funcionales
  - describen aspectos del sistema visibles por el usuario que no se relacionan en forma directa con el comportamiento funcional del sistema.
  - se recogen en los casos de uso con los que están relacionados, o en la Especificación Complementaria.
  - en el Glosario se agrupan y clarifican los términos que se utilizan en los requisitos
  - ejemplos: restricciones en el tiempo de respuesta, precisión de los resultados,...

Requerimientos no funcionales de GeHoWeb.

La tasa de disponiblidad de Gehoweb debe ser de un 99%.

El sistema debe tener una interfaz de uso intuitiva y sencilla, complementada con un buen sistema de ayuda (la administración puede recaer en personal con poca experiencia en el uso de aplicaciones informáticas).

El sistema debe disponer de una documentación fácilmente actualizable que permita realizar operaciones de mantenimiento con el menor esfuerzo posible.



fuente: *Ingeniería de Software*, I. Sommerville, p. 102

- requerimientos de implementación
  - son necesidades del cliente que restringen la implementación (por ejemplo, lenguaje de programación, plataforma hardware, servidor de páginas web, libro de estilo,...)

Requerimientos de implementación de GeHoWeb.

Con el fin de ajustarse a la arquitectura de la intranet actual de la ESEI, GeHoWeb debe desarrollarse como un servicio web, accesible desde cualquier navegador Explorer 5.0, Netscape 5.0 o superior, y estará instalado en un servidor Windows 2000, actuando como servidor de páginas web Internet Information Server. La base de datos a utilizar será SQL Server 7.0.

La interfaz de usuario debe de ajustarse a las características de la web de la ESEI, dentro de la cual estará integrado Gehoweb.

Además, en el desarrollo de GeHoWeb deberán tenerse en cuenta las directrices de política de seguridad recomendadas por el Grupo de Seguridad de la ESEI.

características de la especificación de requerimientos

- la validación de requerimientos es continua y muy importante, para asegurarse de que la especificación es:
  - **correcta**: la especificación debe representar la visión que el cliente tiene del sistema
  - **completa**: describe todos los escenarios posibles, incluyendo el comportamiento excepcional
  - **consistente**: no se contradice a sí misma
  - **no ambigua**: no es posible interpretar aspectos de la especificación de dos o más formas diferentes
  - **realista**: el sistema se puede implementar con las restricciones documentadas
  - **verificable**: una vez que se construye el sistema, se puede diseñar una prueba repetible que demuestre que se satisfacen los requerimientos
    - ejemplos de requerimientos no verificables:
      - *"el producto debe tener **una buena** interfaz de usuario"*
      - *"el producto debe responder en un tiempo **razonable**"*
      - *"el sistema debe ser **seguro**"*
  - **rastreable**: la especificación se debe organizar de tal forma que cada función del sistema se pueda rastrear hasta su conjunto de requerimientos correspondiente. Facilita las pruebas y la validación del diseño

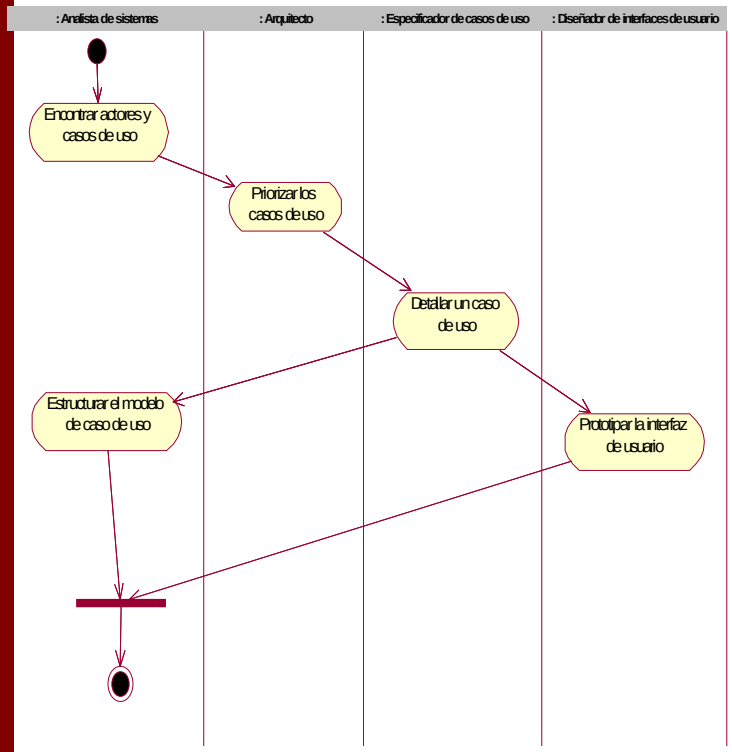
estructura de un documento de requerimientos

- Capítulo 1: propósito y ámbito
  - ¿Cuáles son los objetivos y el ámbito general?
  - Participantes (¿a quién le interesa el sistema?)
  - ¿Qué hay dentro del ámbito? ¿Qué hay fuera del ámbito?
- Capítulo 2: Términos usados (Glosario)
- Capítulo 3: Casos de uso
  - Actores primarios y sus objetivos generales
  - Casos de uso de negocio
  - Casos de uso de sistema
- Capítulo 4: Tecnología utilizada
  - Requerimientos tecnológicos para este sistema
  - Sistemas con los que interactúa este sistema. Requerimientos de esta interacción.

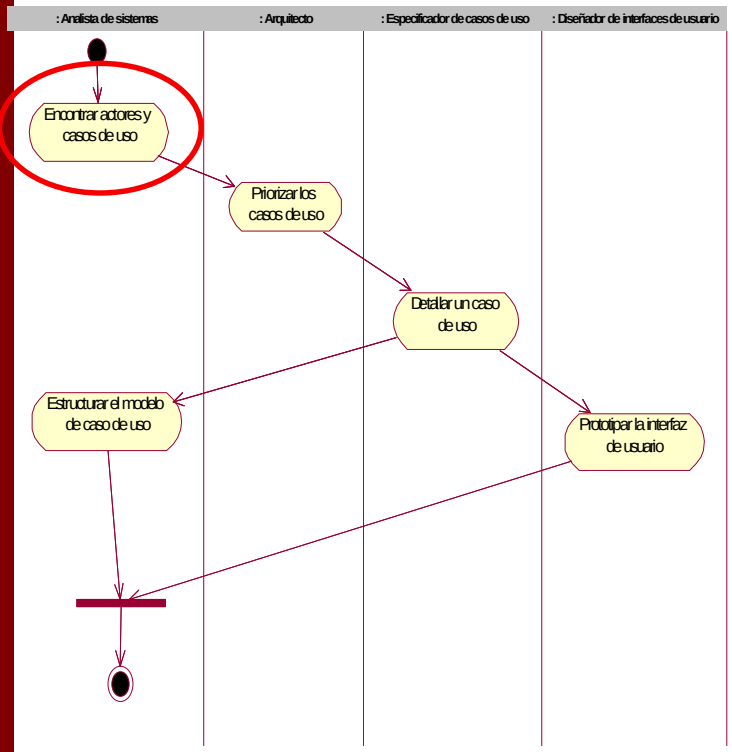
- Capítulo 5: Otros requerimientos
  - Proceso de desarrollo
    - Participantes en el proyecto
    - Feedback o visibilidad del proyecto que quieren los usuarios y clientes
    - ¿Qué podemos comprar, qué debemos construir y quién es nuestra competencia?
    - Otros requerimientos del proceso (prueba, instalación,...)
  - Reglas de negocio
  - Utilización y usabilidad
  - Fiabilidad
  - Rendimiento
  - Soporte, mantenimiento y portabilidad
  - Seguridad, documentación
  - Requisitos sin resolver o diferidos
- Capítulo 6: factores organizativos, legales y humanos
  - Requerimientos legales y políticos
  - Consecuencias humanas de la finalización del sistema
  - Requerimientos de formación
  - Dependencias y restricciones del entorno humano

- artefacto de UML: Modelo de Casos de Uso
- casos de uso
  - propuestos inicialmente por Jacobson
  - mecanismos para ayudar a representar y comprender los objetivos y requisitos de un sistema, de forma simple y comprensible para todo el personal involucrado.
  - de forma informal, *son historias del uso de un sistema para alcanzar sus objetivos*.
  - ejemplo (Larman, 2002, pág. 44):
    - **Procesar Venta:** un cliente llega a una caja con artículos para comprar. El cajero utiliza el sistema PDV (punto de venta) para registrar cada artículo comprado. El sistema presenta una suma parcial y detalles de cada línea de venta. El cliente introduce los datos del pago, que el sistema valida y registra. El sistema actualiza el inventario. El cliente recibe un recibo del sistema y luego se va con los artículos.

- mediante el Modelo de Casos de Uso se muestran cuatro niveles de descripción:
  - **división del trabajo**: casos de uso que describen los procesos de trabajo de los usuarios, relevantes para el sistema. Definición de las fronteras entre usuarios y sistema.
  - **funciones del sistema específicas de la aplicación**
  - **funciones de apoyo del sistema**, como administración de archivos, deshacer, políticas de gestión de excepciones, seguridad,...
  - **diálogo**: casos de uso que describen las interacciones entre usuarios e interfaz de usuario del sistema.



- Según el Proceso Unificado de Desarrollo, los principales pasos para capturar los requerimientos son:
  - identificación de actores y casos de uso
  - priorizar casos de uso
  - detallar casos de uso
  - prototipar la interfaz de usuario
  - estructurar el Modelo de Casos de Uso



- objetivos
  - delimitar el sistema y su entorno
  - esbozar quién y qué (actores) interactuarán con el sistema, y qué funcionalidad (casos de uso) se espera del sistema
  - capturar y definir un glosario de términos comunes esenciales para poder describir detalladamente los casos de uso del sistema.
- es la actividad más decisiva para obtener adecuadamente los requisitos
- responsabilidad del analista de sistemas
- actividades (no tienen por qué seguir este orden)
  - establecer el límite del sistema: solo software, hardware y software como un todo, lo utiliza una persona, una organización,...
  - encontrar actores principales: los que tienen objetivos de usuario que se satisfacen mediante el uso de los servicios del sistema
  - para cada actor, identificar sus objetivos de usuario y escenarios asociados
  - definir los casos de uso que satisfagan los objetivos de usuario. Nombrarlos de acuerdo con sus objetivos. Normalmente los casos de uso del nivel de objetivo de usuario se corresponderán uno a uno con los objetivos de usuario.
  - describir brevemente (descripción *informal*) cada caso de uso

Actores del Sistema de Pagos y Facturación
Comprador Representa a una persona responsable de adquirir bienes o servicios. Puede ser un comprador individual o alguien perteneciente a una empresa. El Comprador de bienes y servicios necesita el Sistema de Facturación y Pagos para enviar pedidos y pagar las facturas.
Vendedor Representa a una persona que vende y distribuye bienes o servicios. El Vendedor utiliza el sistema para conseguir nuevos pedidos y entregar las confirmaciones de pedido, facturas y avisos de pago.
Sistema de cuentas bancarias El Sistema de Facturación y Pagos envía verificaciones de transacciones al Sistema de Cuentas Bancarias

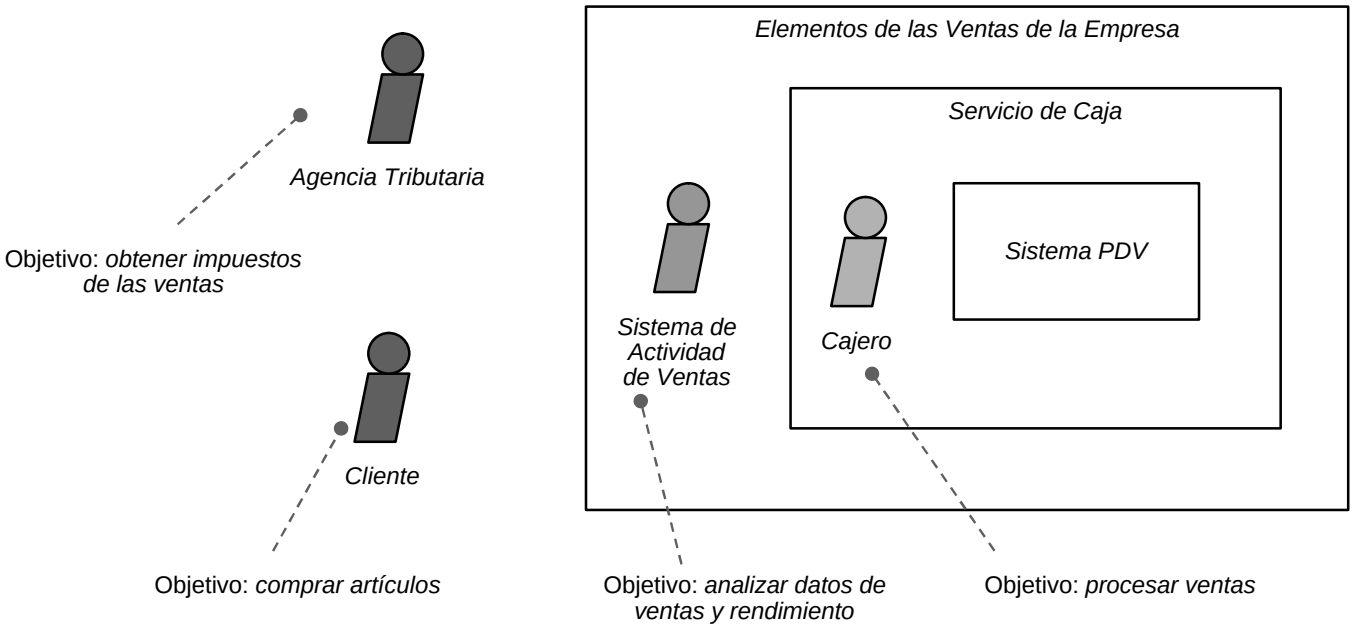
- actores
  - representan entidades externas que interactúan (mantenimiento y/o operación) con el sistema
  - puede ser un usuario o un sistema externo
  - un actor representa un **rol**:
    - no se corresponde directamente con personas concretas
    - toda persona que interactúa con el sistema tiene que estar representado al menos por un actor en el modelo de casos de uso
  - identificación de actores
    - ¿qué grupos de usuarios necesitan el sistema para su trabajo?
    - ¿qué usuarios realizan las funciones principales del sistema?
    - ¿qué usuarios realizan funciones secundarias, como mantenimiento o administración?
    - ¿existe algún sistema externo de hardware o software?
  - se da nombre a los actores y se describen brevemente sus papeles y para qué utilizan el sistema

- identificar actores principales y objetivos
  - además de actores principales y objetivos obvios, se pueden utilizar diferentes preguntas para identificar otros menos evidentes:
    - ¿quién arranca y detiene el sistema?
    - ¿quién administra el sistema?
    - ¿quién gestiona los usuarios y la seguridad?
    - ¿es un actor el "tiempo" porque el sistema hace algo como respuesta a un evento de tiempo?
    - ¿quién evalúa la actividad o el rendimiento del sistema?
    - ...
- tipos de actores
  - **actor principal**: tiene objetivos de usuario que se satisfacen mediante el uso de los servicios del sistema (por ejemplo, el cajero)
    - se identifica para encontrar los objetivos de usuario, que dirigen los casos de uso
  - **actor de apoyo**: proporciona un servicio (por ejemplo, información) al sistema en desarrollo. Por ejemplo, el servicio de autorización de pago). Normalmente es un sistema informático, pero puede ser una organización o una persona
    - se identifica para clarificar las interfaces externas y los protocolos
  - **actor pasivo**: está interesado en el comportamiento del caso de uso, pero no es principal ni de apoyo. Por ejemplo, la Agencia Tributaria.
    - se identifica para asegurar que *todos* los intereses necesarios se han identificado y satisfecho

- actor principal
  - tienen objetivos de usuario que se satisfacen mediante el uso de los servicios del sistema (acuden al sistema para que les ayude)
  - la lista actor-objetivo
    - recoge los actores principales y sus objetivos de usuario

Actor	*Objetivo*	*Actor*	*Objetivo*
Cajero	Procesar ventas Gestionar devoluciones Abrir caja Cerrar caja …	**Administrador del sistema**	Añadir usuarios Modificar usuarios Eliminar usuarios Gestionar seguridad Gestionar tablas …
Jefe de cajas	Controlar productividad cajero Distribuir cajeros en cajas …	**Sistema de Control de Ventas**	Analizar datos de ventas y rendimiento
…	…	…	…

- el actor principal y los objetivos de usuario dependen del límite del sistema



fuente: C. Larman: UML y patrones

- escenario (o *instancia de caso de uso*)
  - es una descripción narrativa de lo que la gente hace y experimenta cuando trata de utilizar una aplicación informática, es decir, una secuencia específica de acciones e interacciones entre los actores y el sistema objeto de estudio.
  - descripción concreta e informal de una sola característica del sistema, desde el punto de vista de un solo actor
  - los analistas y los usuarios escriben y refinan diversos escenarios para comprender mejor lo que debe hacer el sistema
  - identificación de escenarios
    - ¿qué tareas necesita el actor que realice el sistema?
    - ¿qué información consulta el actor? ¿quién crea esos datos? ¿se pueden modificar? ¿quién puede hacerlo?
    - ¿qué cambios externos necesita informar el actor al sistema? ¿cuándo y con qué frecuencia?
    - ¿de qué eventos necesita el actor que le informe el sistema? ¿cuándo y con qué frecuencia

- ejemplos de escenarios
  - Un cliente llega a una caja con artículos para comprar. El cajero utiliza el sistema PDV para introducir el identificador de cada artículo. Cuando ha pasado el último artículo el sistema presenta el total, el cliente paga y el sistema gestiona el pago y presenta el recibo. El cliente se va con el recibo y los artículos.
  - El usuario, previamente autentificado, navega por la tienda online y va marcando los libros que le interesan. El sistema los registra en el carro de la compra del usuario. Cuando termina de comprar el usuario accede a su carro de la compra y procede a realizar el pago. El sistema gestiona el pago solicitando los datos necesarios y accediendo a la pasarela de pago bancario que debe autorizar los datos de la tarjeta. El sistema confirma la venta y muestra al usuario un recibo para que lo guarde o lo imprima.

Un ejemplo en formato “informal” de distintos escenarios de un caso de uso (Larman02, pág. 45):

Gestionar Devoluciones

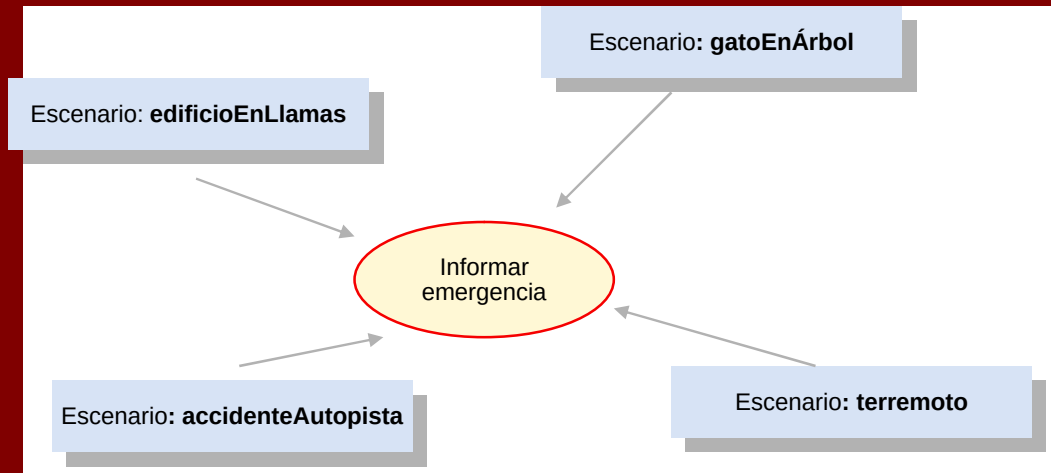
Escenario principal de éxito:

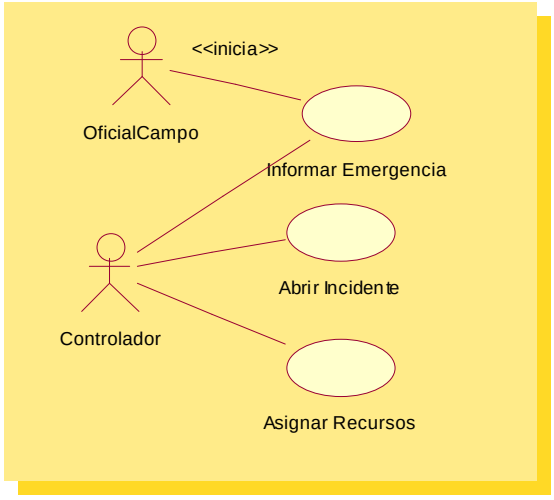
Un cliente llega a una caja con artículos para devolver. El cajero utiliza el PDV para registrar cada uno de los artículos devueltos...

Escenarios alternativos:

1. Si se pagó con tarjeta de crédito, y se rechaza la transacción de reembolso a su cuenta, informar al cliente y pagarle en efectivo.
2. Si el identificador del artículo no se encuentra en el sistema, notificar al Cajero y sugerir la entrada manual del código de identificación (quizás esté alterado).
3. Si el sistema detecta fallos en la comunicación con el sistema de contabilidad externo...

- caso de uso
  - especifica todos los escenarios posibles para una determinada funcionalidad
  - representa una colección de escenarios con éxito y fracaso relacionados, que describe a los actores utilizando un sistema para satisfacer un objetivo.
  - es iniciado por un actor
  - puede interactuar con otros actores
  - representa un flujo de eventos completo a través del sistema, es decir, describe una serie de interacciones relacionadas que resultan de la inicialización del caso de uso.


Escenario: gatoEnÁrbol
Escenario: edificioEnLlamas
Informar emergencia
Escenario: accidenteAutopista
Escenario: terremoto

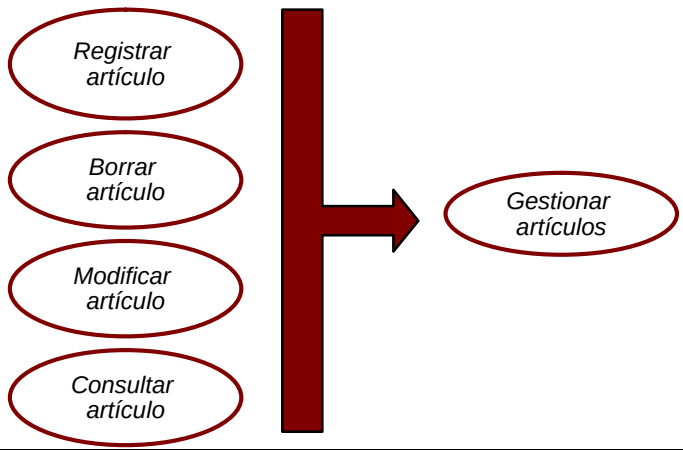

<<inicia>>
OficialCampo
Informar Emergencia
Abrir Incidente
Controlador
Asignar Recursos

- las tareas se pueden agrupar a muchos niveles de granularidad
  - ejemplos:
    - definir estrategia de mercado
    - establecer política de descuentos
    - negociar contrato con proveedor
    - gestionar devoluciones de productos
    - iniciar sesión en el sistema
    - imprimir factura
    - ...
  - todos son casos de uso *a diferentes niveles*, dependiendo de los límites del sistema, actores y objetivos
  - PREGUNTA: ¿a qué nivel y alcance deben expresarse los casos de uso?
  - RESPUESTA: *Para el análisis de requisitos de una aplicación informática, hay que centrarse en los casos de uso al nivel de los procesos de negocio elementales (EBP, Elementary Business Processes)*
  - EBP:
    - tarea realizada por una persona en un lugar, en un instante, como respuesta a un evento del negocio, que añade valor cuantificable para el negocio y deja los datos en un estado consistente. Por ejemplo, Autorizar Crédito, o Solicitar Precio
    - no es un pequeño paso como “eliminar línea de pedido”, o “imprimir factura””
    - no tarda días y múltiples sesiones como “negociar contrato con proveedor”
    - son los caso de uso “base”, pero puede haber otros
  - pueden existir casos de uso que no sean EBP:
    - subtareas de un caso de uso base
    - ejemplo: “*Pago a Crédito”* puede aparecer en varios casos de uso base, por lo que se puede separar en un caso de uso propio conectado a varios casos de uso base

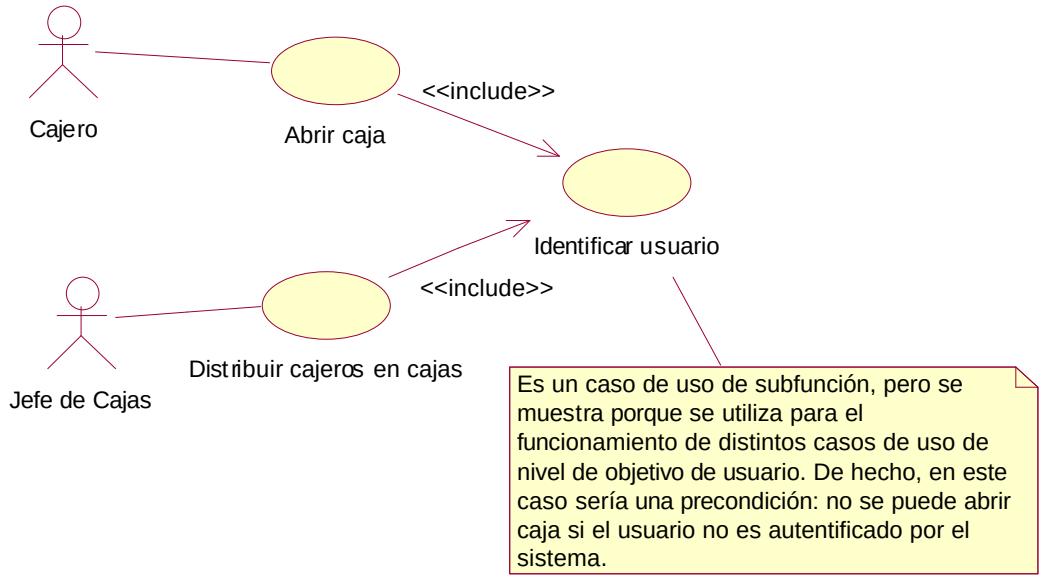
- casos de uso y objetivos
  - los actores tienen objetivos y utilizan las aplicaciones para ayudarles a satisfacerlos
  - por tanto, un caso de uso EBP se denomina caso de uso ***a nivel de objetivo de usuario***, para remarcar que sirve para satisfacer un objetivo de un actor principal
  - procedimiento:
    1. encontrar los objetivos de usuario
    2. definir un caso de uso para cada uno
  - excepción: agrupación de objetivos separados del tipo Altas-Bajas-Modificaciones-Consultas, en casos de uso denominados *Gestionar<X>*

Estos casos de uso, muy habituales en aplicaciones de gestión, se agrupan normalmente en un único caso de uso.

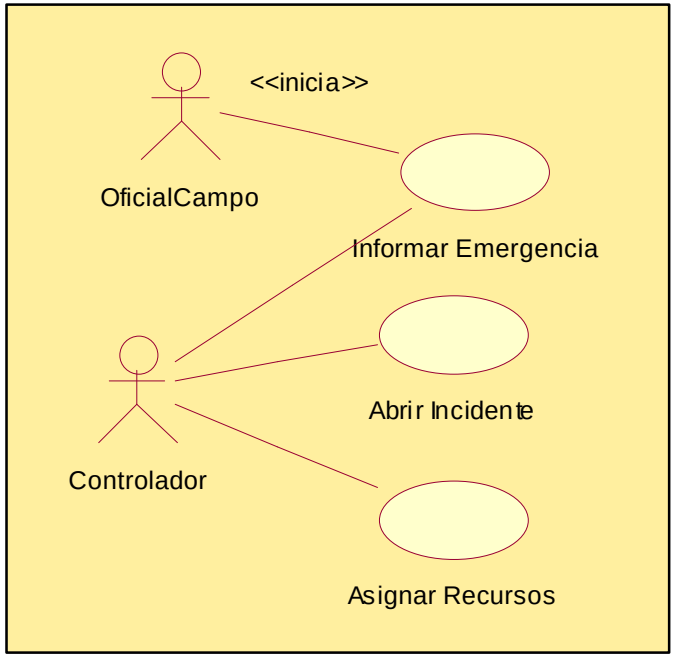
Lógicamente, si un actor está únicamente autorizado a realizar determinadas funciones (por ejemplo, consultar artículos), se mostrarán los casos de uso correspondientes



- objetivos y casos de uso de subfunción
  - objetivo de subfunción: subobjetivos que dan soporte a un objetivo de usuario. Por ejemplo, para cumplir el objetivo *Abrir Caja* el cajero necesita identificarse en el sistema.
  - se representan objetivos de subfunción como casos de uso cuando la subfunción se repite o es una precondición en muchos casos de uso de nivel de objetivos de usuario
  - ejemplo: *Identificar Usuario*



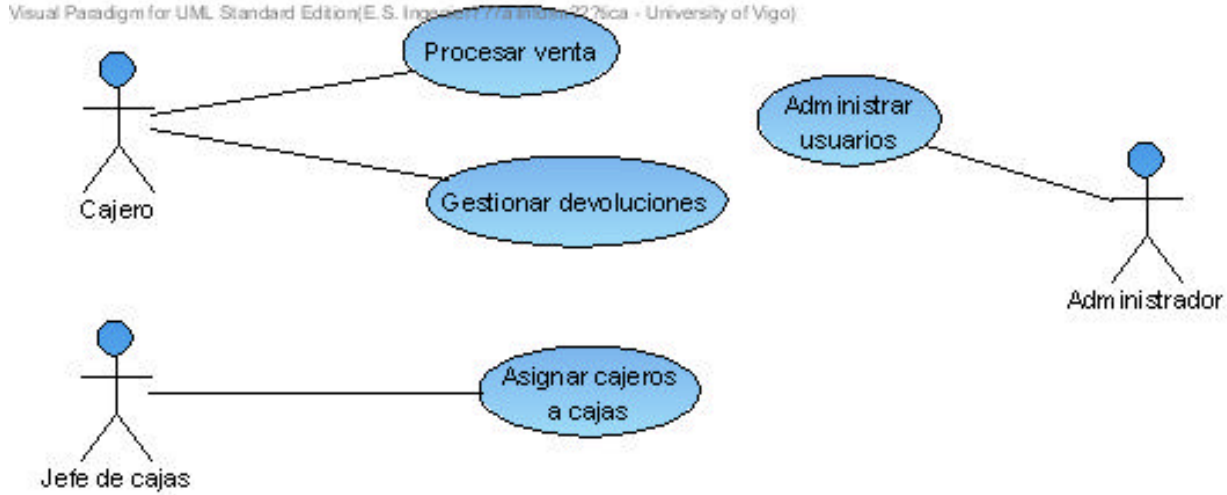
relación entre actores y casos de uso

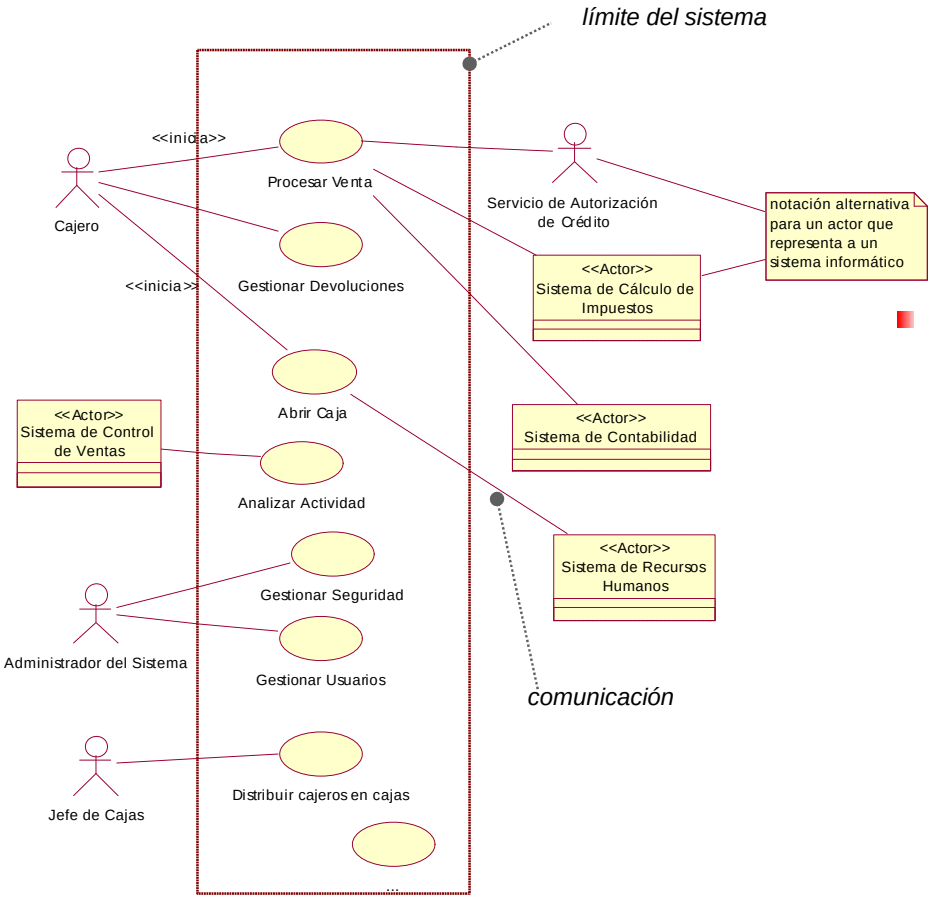


- relaciones de comunicación entre actores y casos de uso
  - representan el flujo de información durante el caso de uso
  - se puede distinguir entre el actor que inicia el caso de uso y los demás actores que intervienen posteriormente
  - los casos de uso, identificados previamente a partir de los objetivos de los actores, se representan mediante óvalos y representan una tarea que el sistema en desarrollo tiene que incorporar
  - Diagrama de Casos de Uso: representa el contexto del sistema:
    - límites del sistema
    - qué permanece fuera del sistema
    - cómo se utiliza el sistema
    - resume el comportamiento de un sistema y sus actores

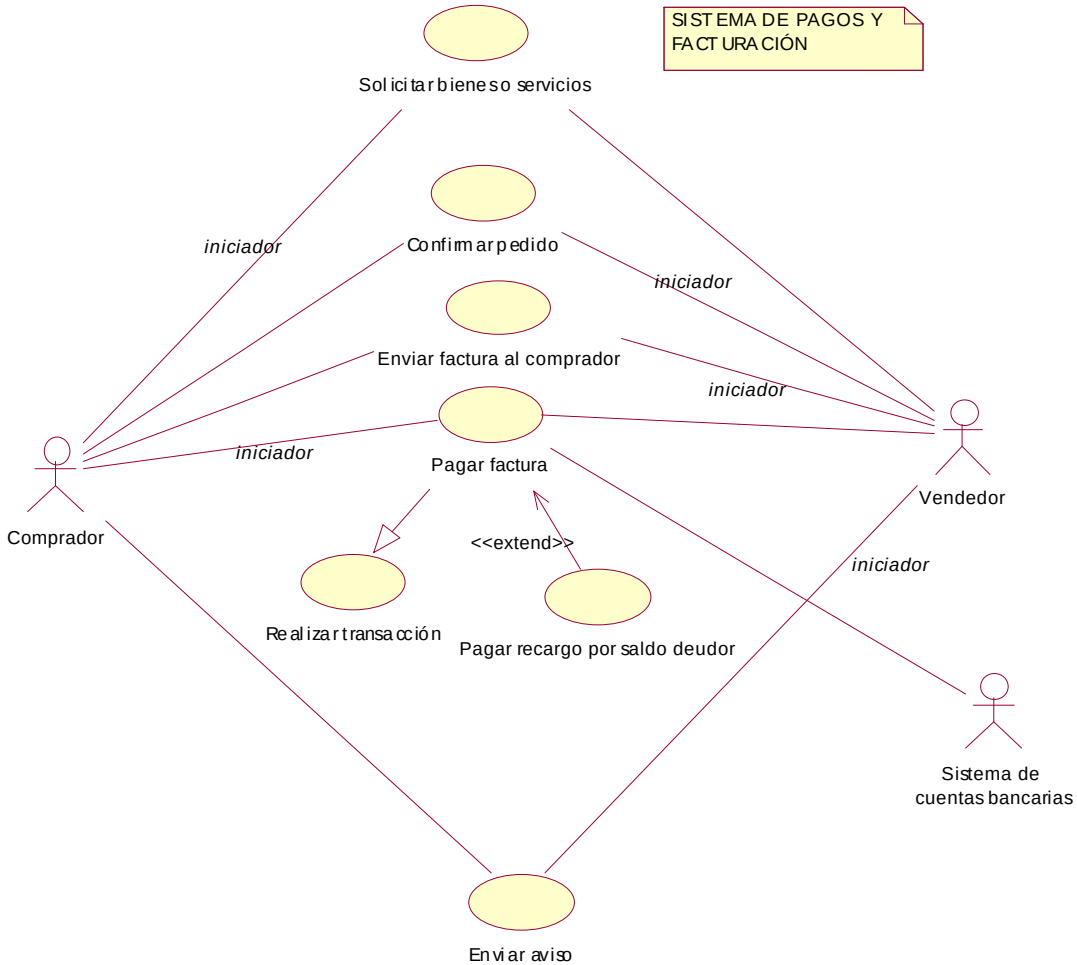
relación entre actores y casos de uso

Actor	Objetivo	Descripción escenarios	Caso de uso
Cajero	Procesar venta	...	**Procesar venta**
Cajero	Gestionar devoluciones	...	**Gestionar devoluciones**
Jefe de cajas	Distribuir cajeros entre las cajas	...	**Asignar cajeros a cajas**
Administrador	Altas de usuarios	...	**Gestionar usuarios**
Administrador	Bajas de usuarios	...	
Administrador	Modificar usuarios	...	





- sugerencias en la realización de diagramas de casos de uso
  - en diagramas de contexto, utilizar únicamente casos de uso de nivel de objetivos de usuario
  - mostrar los actores que representen sistemas informáticos con una notación alternativa a los actores humanos
  - situar los actores humanos a la izquierda y los de apoyo a la derecha
  - lo realmente importante es la descripción de los casos de uso, y no tanto los diagramas


SISTEMA DE PAGOS Y FACTURACIÓN
Solicitar bienes o servicios
Confirmar pedido
Enviar factura al comprador
Pagar factura
Realizar transacción
<<extend>>
Pagar recargo por saldo deudor
Enviar aviso
iniciador
iniciador
iniciador
iniciador
iniciador
Comprador
Vendedor
Sistema de cuentas bancarias

- Proceso Unificado: desarrollo dirigido por casos de uso
  - los requisitos se recogen principalmente en el Modelo de Casos de Uso
  - los casos de uso son parte importante de la planificación de las iteraciones: el trabajo de una iteración se define en parte eligiendo algunos escenarios o casos de uso completos. Por tanto, son una entrda clave para realizar estimaciones
  - las *realizaciones de casos de uso* dirigen el diseño, es decir, el equipo diseña objetos y subsistemas que colaboran para ejecutar o realizar los casos de uso
  - el trabajo con los casos de uso se realiza a lo largo de las diversas iteraciones

Ejemplo de la estrategia del Proceso Unificado sobre el método de desarrollo de los requisitos

Disciplina	Artefacto	Comentarios y nivel de esfuerzo de los requisitos				
		Inicio 1 semana	**Elab 1 4 semanas**	**Elab 2 4 semanas**	**Elab 3 3 semanas**	**Elab 4 3 semanas**
Requisitos	Modelo de Casos de Uso	2 días de taller de requisitos. Se identifican por el nombre la mayoría de los casos de uso y se resumen en un párrafo breve	Cerca del final de la esta iteración, tiene lugar un taller de requisitos de 2 días. Se obtiene un mejor entendimiento y retroalimentación a partir del trabajo de implementación. Entonces se completa el 30% de los casos de uso en detalle	Cerca del final de la esta iteración, tiene lugar un taller de requisitos de 2 días. Se obtiene un mejor entendimiento y retroalimentación a partir del trabajo de implementación. Entonces se completa el 50% de los casos de uso en detalle	Repetir, se completa el 70% de todos los casos de uso en detalle	Repetir con la intención de clarificar y escribir en detalle el 80-90% de los casos de uso. Sólo una pequeña parte de éstos se construyen durante la elaboración; el resto se aborda durante la construcción.
Diseño	Modelo de Diseño	Nada.	Diseño de un pequeño conjunto de requisitos de alto riesgo significativos desde el punto de vista de la arquitectura.	Repetir	Repetir	Repetir. Deberían ahora estabilizarse los aspectos de altor riesgo significativos para la arquitectura.
Implementación	Modelo de implementación (código, etc.)	Nada	Implementar esto.	Repetir. Se construye el 5% del sistema final.	Repetir. Se construye el 10% del sistema final.	Repetir. Se construye el 15% del sistema final.
Gestión del Proyecto	Plan de Desarrollo de Software	Estimación muy imprecisa del esfuerzo total	La estimación comienza a tomar forma	Un poco mejor...	Un poco mejor...	Ahora se pueden establecer racionalmente la duración global del proyecto, los hitos más importantes, estimación del coste y esfuerzo.

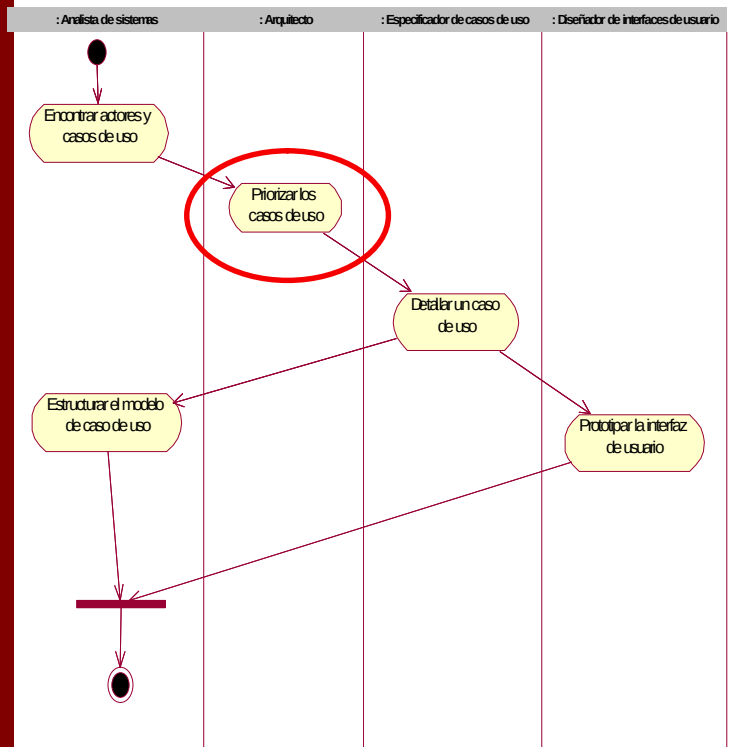
fuente: C. Larman, UML y patrones, 2002

- casos de uso en la fase de inicio
  - fase breve (pocos días)
  - identificación de objetivos y personal involucrado
  - determinar alcance del proyecto
  - elaboración lista actor-objetivo
  - iniciar diagrama de contexto de casos de uso
  - descripción de casos de uso
    - casos de uso importante, complejos o arriesgados se escriben en formato breve
    - entre el 10-20% de los casos de uso que representan las principales funciones o son arriesgados se escriben en formato completo
    - escribir lista de *intereses* y *personal involucrado* para estos casos de uso
  - decidir si se realiza un estudio más profundo (fase de elaboración) o se rechaza el proyecto
  - ejemplo de un Modelo de Casos de Uso en la fase de inicio para un sistema de PDV:

Completo	Informal	Breve
Procesar Venta	Procesar Alquiler	Abrir Caja
Gestionar Devoluciones	Analizar Actividad de Ventas	Cerrar Caja
	Gestionar Seguridad	Gestionar Usuarios
	...	Distribuir Cajeros
		Suspender Operación
		Gestionar Tablas del Sistema
		...

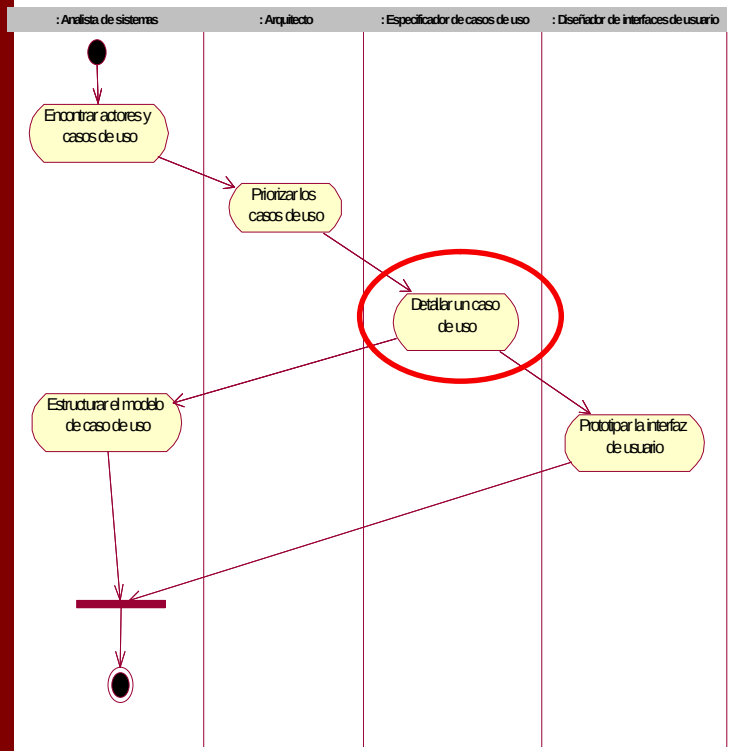
- casos de uso en la elaboración
  - fase de múltiples iteraciones de duración fija
  - se construyen incrementalmente partes del sistema arriesgadas, de alto valor o significativas para la arquitectura
  - se identifican y clarifican la mayoría de los requisitos
  - retroalimentación de las fases de diseño e implementación
  - se pueden realizar talleres de requisitos en cada iteración:
    - no se estudian todos los casos de uso: se priorizan
    - se refinan los requisitos principales, que son inestables en las primeras iteraciones, estabilizándose en las últimas
    - escritura y reescritura de la mayoría de los casos de uso, en formato completo
  - al final de la fase de elaboración
    - quedan descritos en detalle entre el 80 y el 90% de los casos de uso
    - quedan programadas partes del sistema (entre un 10 y un 15% del sistema)
- casos de uso en la construcción
  - fase de múltiples iteraciones de duración fija, centrada en completar el sistema
  - puede ser necesario escribir o reescribir casos de uso menores (incluso puede necesitarse algún taller de requisitos)
  - el grado de cambio de los requisitos es mucho menor que en la elaboración, pues la mayoría de los casos de uso funcionales y no funcionales más importantes deben haberse estabilizado

priorizar casos de uso



- priorización de casos de uso
  - determinar cuáles son necesarios para el desarrollo en las primeras iteraciones y cuáles pueden dejarse para posteriores iteraciones
  - cuestiones a tener en cuenta:
    - casos de uso con dificultad de desarrollo
    - casos de uso imprescindibles para la puesta en marcha del sistema
    - organización del desarrollo incremental
    - disponibilidad de equipo de desarrollo
  - se revisa la priorización con el jefe de proyecto y se utiliza como entrada para la planificación de cada iteración del proyecto

detallar los casos de uso



- objetivo principal: describir su flujo de sucesos en detalle
  - cómo comienza
  - cómo termina
  - cómo interactúan con los actores
- se detalla paso a paso la secuencia de acciones del caso de uso
- se trabaja estrechamente con los usuarios reales de los casos de uso
- resultado: descripción detallada mediante
  - texto
  - diagramas

secciones de la descripción del caso de uso (1)

- **actor principal**: actor que recurre a los servicios del sistema para cumplir un objetivo
- **personal involucrado y lista de intereses**
  - el caso de uso captura *todo y sólo* el comportamiento relacionado con la satisfacción de los intereses del personal involucrado
  - ejemplo:
    - Cajero: quiere entradas precisas, rápidas y sin errores de pago, ya que las pérdidas se deducen de su salario.
    - Vendedor: quiere que las comisiones de ventas estén actualizadas
- **precondiciones**:
  - establecen lo que *siempre debe* cumplirse antes de comenzar un escenario en el caso de uso.
  - no se prueban en el caso de uso, sino que son condiciones que se asumen que son ciertas.
  - normalmente una precondición implica un escenario de otro caso de uso que se ha completado con éxito, como "*inicio de sesión*", o "*validación de usuario*".
  - ejemplo: *el cajero se identifica y el sistema lo autentifica*.
- **postcondiciones**:
  - establecen qué debe cumplirse cuando el caso de uso se completa con éxito (bien el escenario principal de éxito o algún camino alternativo)
  - ejemplo: *se registra la venta. El impuesto se calcula de manera correcta. Se actualizan la Contabilidad y el Inventario. Se registran las comisiones. Se genera el recibo.*

secciones de la descripción del caso de uso (2)

- **escenario principal de éxito (o flujo básico)**
  - describe el camino de éxito típico que satisface los intereses del personal involucrado
  - no suele incluir condiciones o bifurcaciones
  - recoge los pasos, que pueden ser de tres tipos:
    - una interacción entre actores
    - una validación (normalmente a cargo del sistema)
    - un cambio de estado realizado por el sistema (por ejemplo, registrando una venta o modificando un registro de la base de datos)
  - el primer paso indica el evento que desencadena el comienzo del escenario
  - ejemplo:

    1. El Cliente llega a un terminal PDV con mercancías para comprar
    2. El Cajero comienza una nueva venta.
    3. El Cajero introduce el identificador del artículo.
    4. ...

    El Cajero repite los pasos 3-4 hasta que se indique

    5. ...

secciones de la descripción del caso de uso (3)

- **extensiones (o flujos alternativos)**
  - indican todos los otros escenarios o bifurcaciones, tanto de éxito como de fracaso.
  - la combinación del escenario principal y los escenarios de extensión deberían satisfacer *casi* todos los intereses del personal involucrado (algunos intereses se documentan como requisitos no funcionales)
  - ejemplos:

    3a. Identificador no válido
    1. El Sistema señala el error y rechaza la entrada

    3b. Hay muchos artículos de la misma categoría y tener en cuenta una única identidad del artículo no es importante (por ejemplo, 6 cajas de leche de la misma marca).
    1. El Cajero puede introducir el identificador de la categoría del artículo y la cantidad
  - un flujo alternativo tiene dos partes:
    - condición: algo que puede ser detectado por el sistema o el actor (*el Sistema detecta un fallo de comunicación con el sistema de actualización de inventario*)
    - manejo: se puede resumir en un paso o bien incluir una secuencia:

      3-6a. El Cliente le pide al Cajero que elimine un artículo de la compra:
      1. El Cajero introduce el identificador del artículo para eliminarlo de la compra
      2. El Sistema muestra la suma parcial actualizada
  - si una extensión es muy compleja, se puede expresar como un caso de uso aparte
  - pueden incluirse condiciones de extensión que se pueden dar durante cualquiera de los pasos (por ejemplo, *el sistema falla*, o *el Cliente cancela la compra*)

secciones de la descripción del caso de uso (4)

- **requisitos especiales**
  - se recoge cuando un requisito no funcional, atributo de calidad o restricción se relaciona de manera específica con un caso de uso
  - incluye cualidades como rendimiento, fiabilidad y facilidad de uso, y restricciones de diseño (a menudo en dispositivos de entrada/salida) que son obligados o se consideran probables.
  - ejemplo:
    - interfaz de usuario con pantalla táctil en un gran monitor de pantalla plana. El texto debe ser visible a un metro de distancia
    - tiempo de respuesta para la autorización de crédito de 30 segundos al menos en el 90% de los casos
  - en ocasiones resulta conveniente reunir al final todos los requisitos no funcionales en una especificación complementaria

Nombre del caso de uso: Pagar Factura

Actores participantes: Comprador (inicia)
Vendedor
Sistema de cuentas bancarias

Precondición: el comprador ha recibido los bienes y servicios y al menos una factura del sistema. El comprador ahora planifica el pago de las facturas

Flujo de eventos:
Camino básico

1. El comprador invoca al caso de uso para comenzar a hojear las facturas recibidas del sistema. El sistema verifica que el contenido de cada factura es consistente con las confirmaciones de pedido recibidas anteriormente (como parte del caso de uso Confirmar Pedido) y así indicárselo al comprador. La confirmación de pedido describe qué elementos serán enviados, cuándo, dónde y a qué precio.
2. El comprador decide planificar una factura para pagarla por banco, y el sistema genera una petición e pago para transferir el dinero a la cuenta del vendedor. Hay que tener en cuenta que un comprador no puede planificar el pago de la misma factura dos veces [A2].
3. Más tarde, si hay suficiente dinero en la cuenta del comprador, se hace un pago mediante transacción en la fecha planificada. Durante la transacción, el dinero se transfiere de la cuenta del comprador a la del vendedor, como se describe en el caso de uso abstracto Realizar Transacción (que es utilizado por Pagar Factura). Tanto el comprador como el vendedor tienen notificación del resultado de la transacción. El banco recoge unos cargos por la transacción, que se retiran de la cuenta del comprador [A3].
4. La instancia del caso de uso finaliza.

Caminos alternativos

[A2] En el paso 2, el comprador puede, en cambio, pedir al sistema que devuelva un rechazo de factura al vendedor.
[A3] En el paso 3, si no hay suficiente dinero en la cuenta, el caso de uso cancelará el pago y se lo notificará al comprador.

Postcondición: la instancia del caso de uso termina cuando la factura ha sido pagada o cuando el pago de la factura ha sido cancelado y no se ha hecho ninguna transferencia.

Nombre del caso de uso:	InformarEmergencia
Actores participantes:	OficialCampo (inicia) Controlador
Condición inicial:	El OficialCampo activa la función Informar Emergencia en su PDA

Flujo de eventos:

1. El sistema EMERGENCIAS responde presentando un formulario al OficialCampo
2. El OficialCampo cubre el formulario, seleccionando el nivel de emergencia, tipo, ubicación, y una breve descripción de la situación. También describe respuestas posibles a la situación de emergencia. Una vez cubierto el formulario, el OficialCampo lo envía y en ese momento se notifica al Controlador.
3. El Controlador revisa la información recibida y crea un Incidente en la base de datos llamando al caso de uso AbrirIncidente. El Controlador selecciona una respuesta y da un acuse de recibo del informe de emergencia.
4. El OficialCampo recibe el acuse de recibo y la respuesta seleccionada.

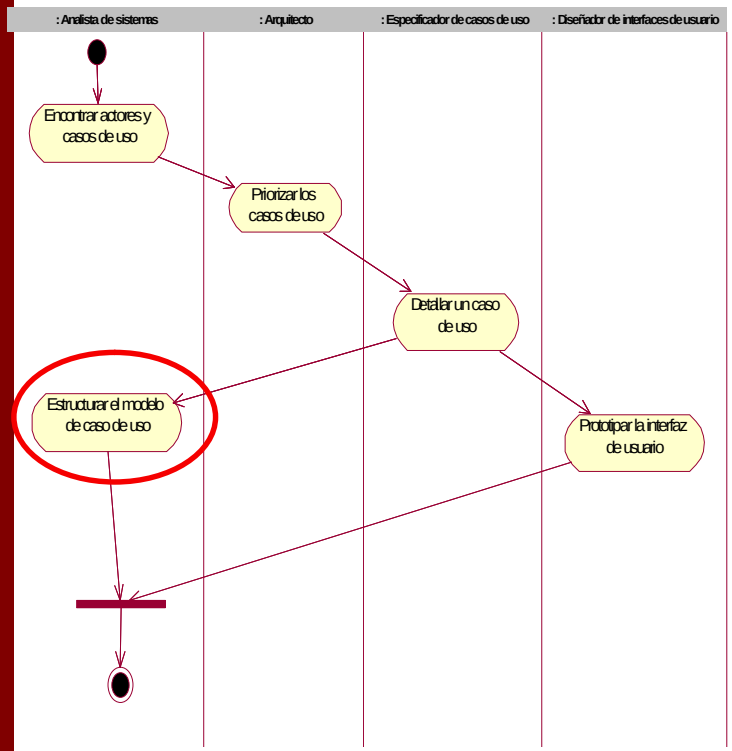
Requerimientos especiales:	Se da acuse de recibo del informe del OficialCampo en menos de 30 segundos. La respuesta seleccionada llega en menos de 30 segundos desde que la envía el Controlador.

descripción del caso de uso

Ubicación		Descripción del caso de uso
Estación *OficialCampo*	1.	El OficialCampo activa la función Informar Emergencia en su PDA.
	2.	El sistema EMERGENCIAS responde presentando un formulario al OficialCampo. El formulario incluye un menú de tipos de emergencia (incendio, accidente, disturbios,...) y campos para introducir la ubicación, descripción del incidente, petición de recursos,...
	3.	El OficialCampo cubre el formulario, y puede también describir respuestas posibles a la situación de emergencia y solicitar recursos específicos. Una vez que ha llenado el formulario, el OficialCampo lo envía oprimiendo el botón “Enviar Informe” y en ese momento se le notifica al Controlador
Estación *Controlador*	4.	Al Controlador se le notifica un nuevo informe de incidente mediante un cuadro de diálogo desplegable. El Controlador revisa la información recibida y crea un Incidente en la base de datos llamando al caso de uso AbrirIncidente. Toda la información contenida en el formulario del OficialCampo se incluye automáticamente en el Incidente. El Controlador selecciona una respuesta asignando recursos al incidente (con el caso de uso AsignarRecursos) y da un acuse de recibo al informe de emergencia enviando un mensaje breve al OficialCampo.
Estación *OficialCampo*	5.	El OficialCampo recibe el acuse de recibo y la respuesta seleccionada.

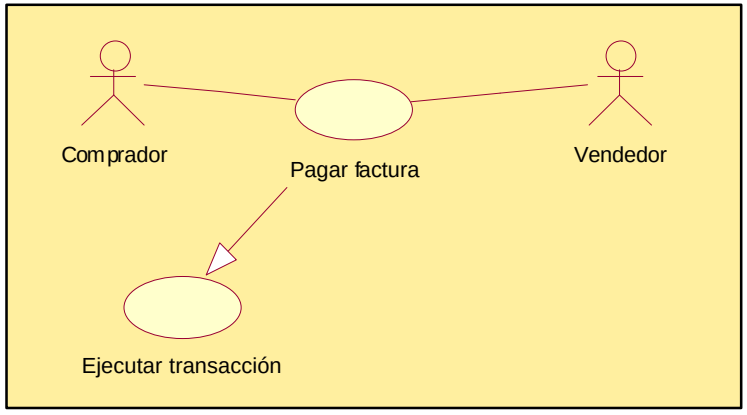
- refinamiento
  - se incorporan detalles al caso de uso
  - se describe el flujo de eventos en mayor detalle
  - se mejora la descripción de las interacciones

estructurar el modelo de casos de uso

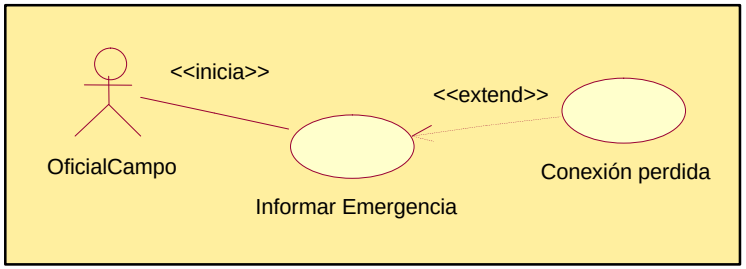


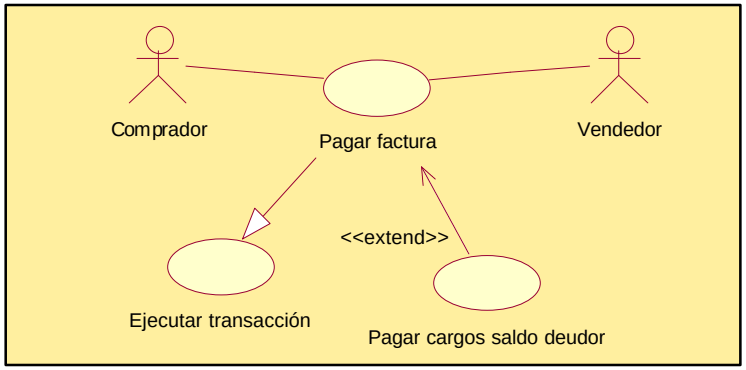
objetivos

- extraer descripciones de funcionalidad (de casos de uso) generales y compartidas que pueden ser utilizadas por casos de uso más específicos
- extraer descripciones de funcionalidad (de casos de uso) adicionales u opcionales que pueden extender casos de uso más específicos (*relaciones de extensión)*
- extraer descripciones de funcionalidad (de casos de uso) adicionales e incondicionales incluidas en la ejecución de casos de uso específicos (*relaciones de inclusión*)



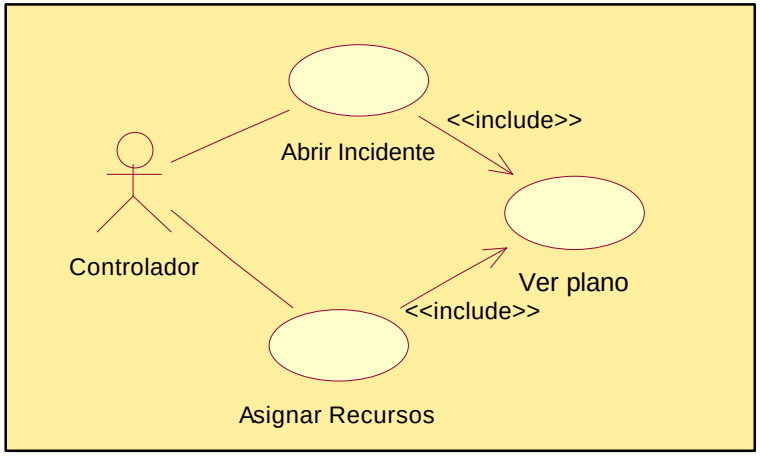
- generalización
  - simplifica la forma de trabajo y la comprensión del modelo de casos de uso
  - permite reutilizar casos de uso "semifabricados" cuando se reúnen casos de uso terminados requeridos por el usuario (*caso de uso concreto)*.
  - el caso de uso "semifabricado" existe solamente para que otros casos de uso lo reutilicen y se les llama *casos de uso abstractos*.
    - no pueden instanciarse por sí mismos
    - una instancia de un caso de uso concreto también exhibe el comportamiento especificado por un caso de uso abstracto que lo (re)utiliza

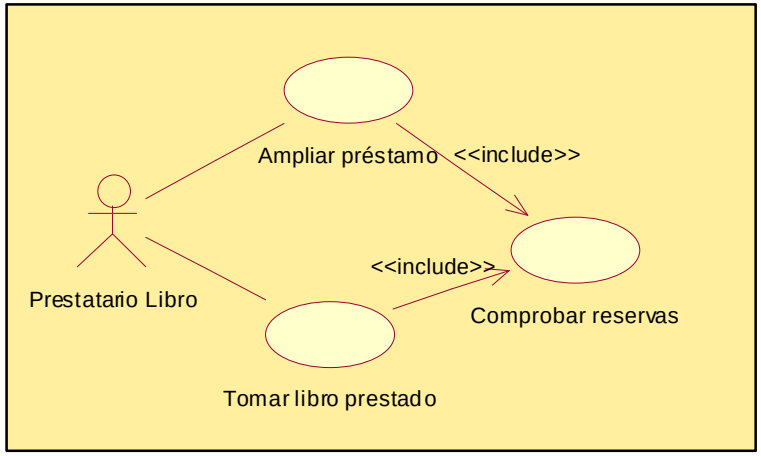


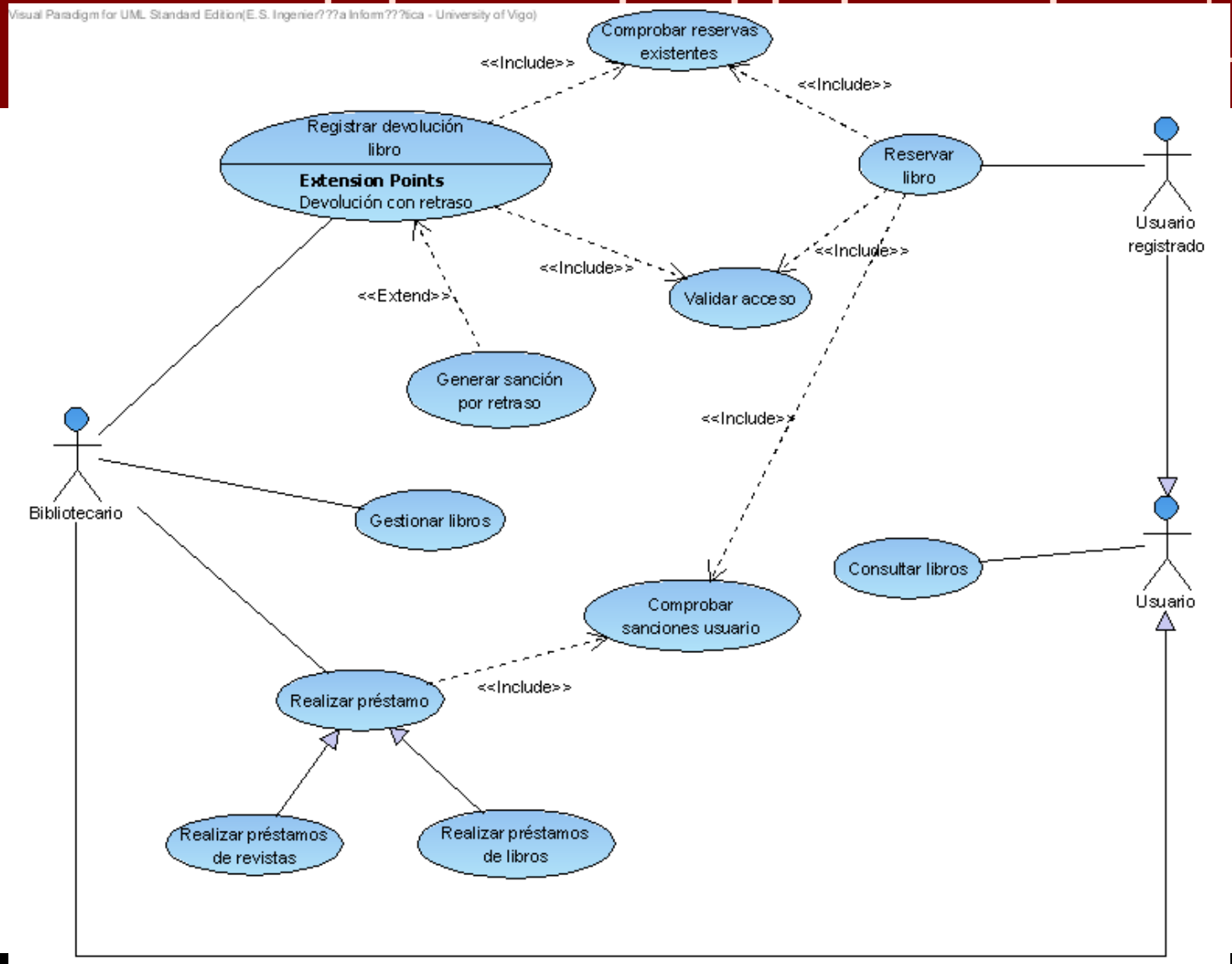


- relaciones de extensión entre casos de uso
  - un caso de uso extiende otro caso de uso si éste puede incluir el comportamiento del primero bajo determinadas condiciones
  - ventajas de separar el flujo excepcional y opcional con respecto al caso de uso básico
    - el caso de uso básico se hace más pequeño y comprensible
    - se distingue entre el caso común y el excepcional, permitiendo que los desarrolladores los traten de forma diferente
  - ambos son casos de uso completos por sí mismos, con condición inicial y final, y comprensibles por el usuario
  - regla general: utilizar relaciones de extensión para comportamientos excepcionales, opcionales o que rara vez ocurren

- relaciones de inclusión entre casos de uso
  - permiten dividir las redundancias y reutilizar casos de uso
  - el comportamiento sólo debe dividirse en casos de uso separados cuando es compartido por dos o más casos de uso
  - no conviene dividir en exceso (especificación confusa)
  - regla general: utilizar relaciones de inclusión para comportamientos que se comparten entre dos o más casos de uso

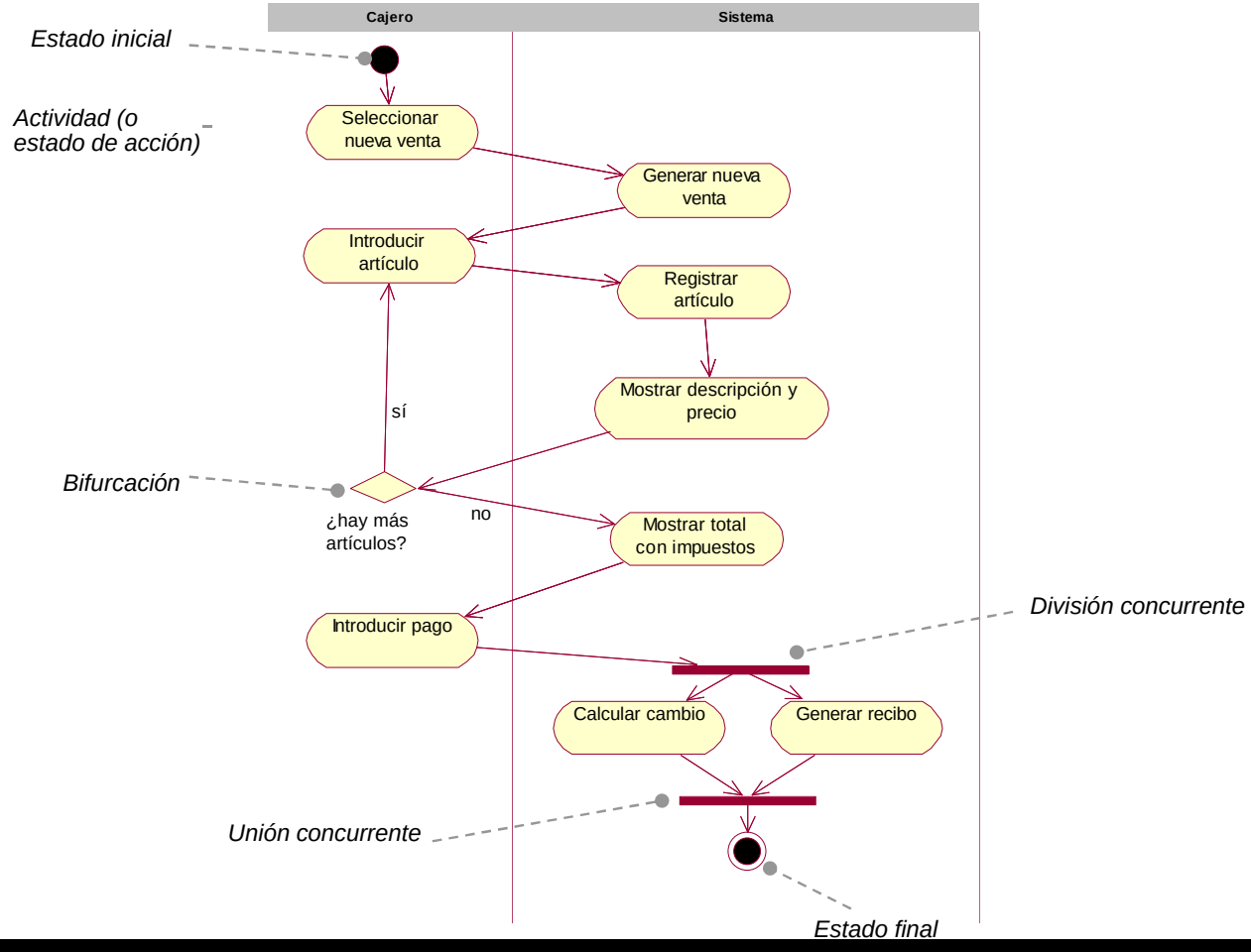





Visual Paradigm for UML Standard Edition(E.S. Ingenier???a Inform???tica - University of Vigo)
Comprobar reservas existentes
<<Include>>
<<Include>>
Registrar devolución libro
Extension Points
Devolución con retraso
Reservar libro
Usuario registrado
<<Include>>
<<Include>>
<<Extend>>
Validar acceso
Generar sanción por retraso
<<Include>>
Bibliotecario
Gestionar libros
Consultar libros
Usuario
Comprobar sanciones usuario
Realizar préstamo
<<Include>>
Realizar préstamos de revistas
Realizar préstamos de libros

- utilización de diagramas
  - puede resultar útil utilizar diagramas para describir un caso de uso cuando existen diferentes estados y transiciones alternativas que dificultan la comprensión de la descripción textual
  - diagramas
    - de actividad: describen las transiciones entre estados con más detalle, como secuencias de acciones.
    - de interacción: describen cómo interactúa una instancia de un caso de uso con la instancia de un actor
  - aconsejable que sean simples, para que sean comprensibles por el usuario

Ejemplo de un Diagrama de Actividad: Escenario de Procesar Venta



Ejemplo de un Diagrama de Actividad: Escenario de Procesar Venta

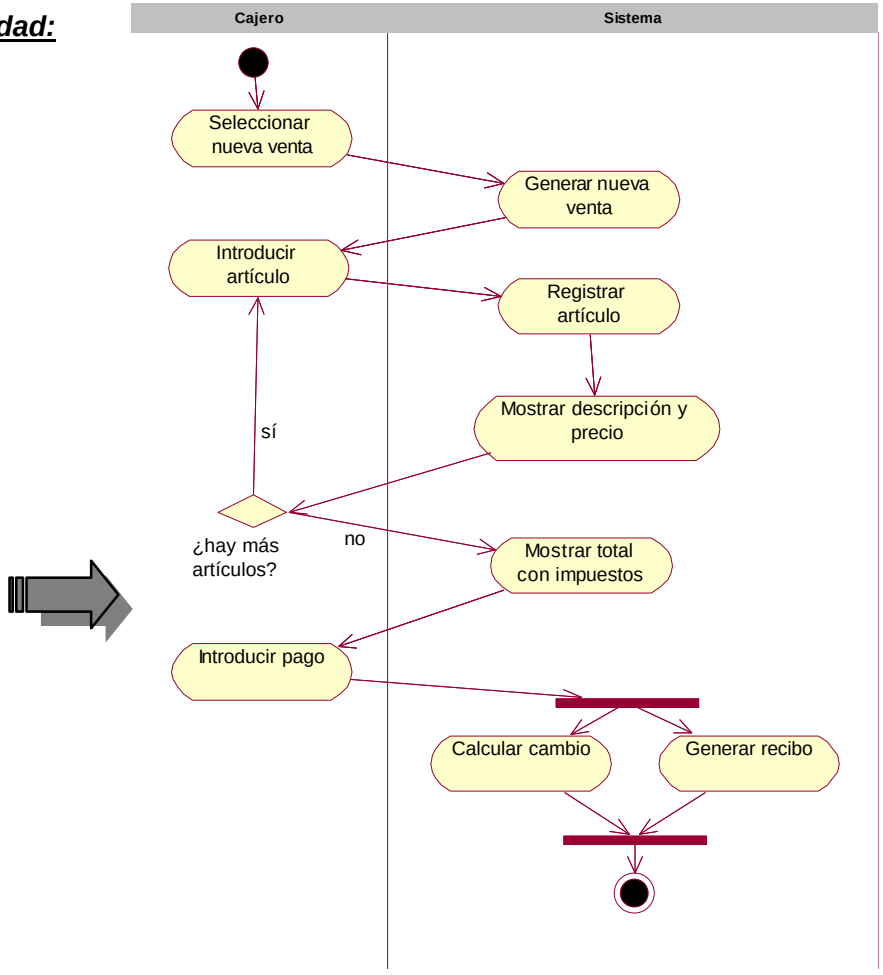
Escenario simple de Procesar Venta para el pago en efectivo.

1. El Cliente llega al terminal PDV
2. El Cajero inicia una nueva venta
3. El Cajero inserta el identificador del artículo
4. El Sistema registra la línea de venta y presenta la descripción del artículo, precio y suma parcial

El Cajero repite los pasos 3 y 4 hasta que se indique

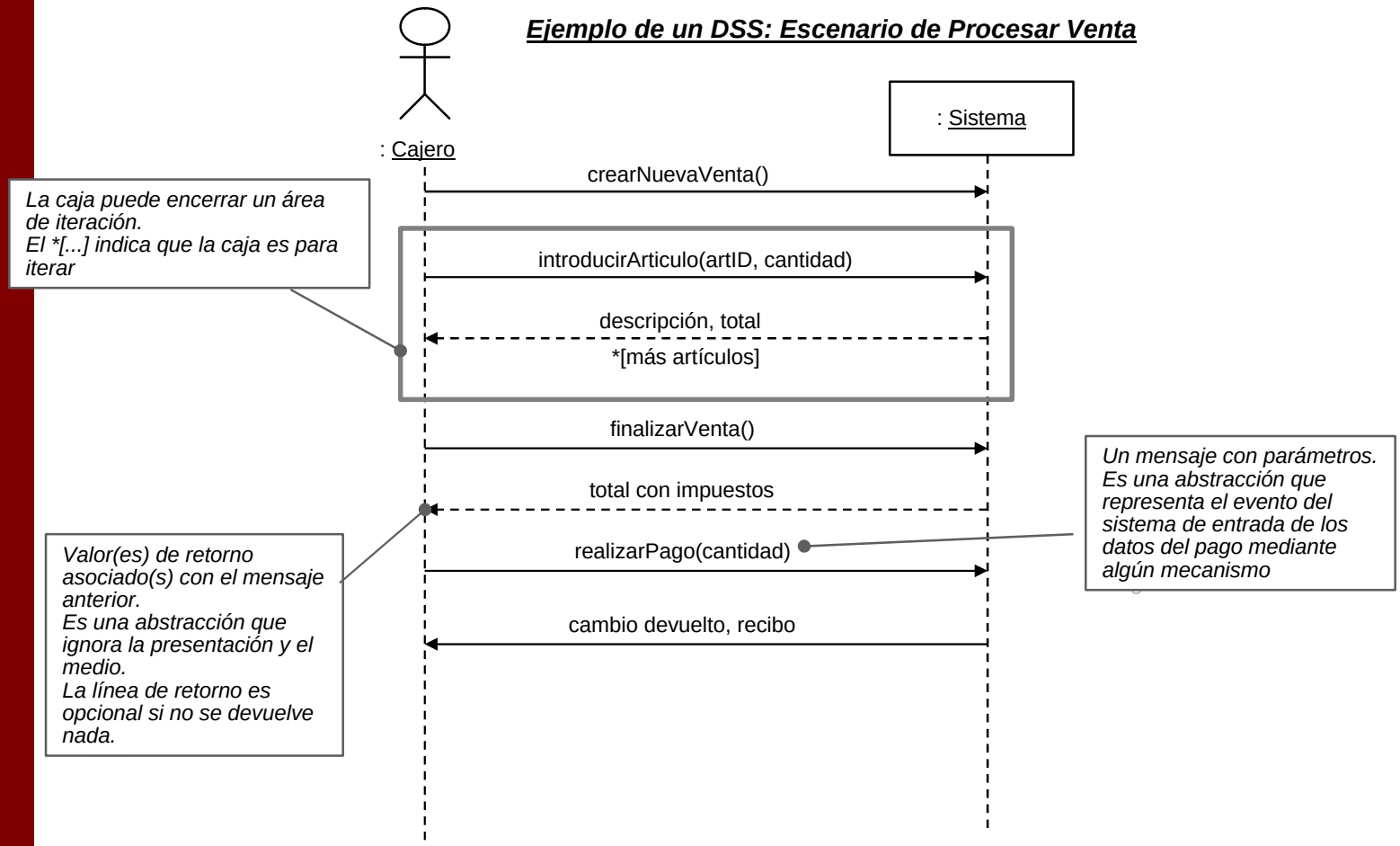
5. El Sistema muestra el total con los impuestos calculados
6. El Cajero le dice al Cliente el total, y pide que le pague
7. El Cliente paga y el Sistema gestiona el pago

...



- diagramas de secuencia del sistema (DSS)
  - diagrama de secuencia:
    - notación de UML que muestra aspectos dinámicos de un sistema, y que se utiliza en distintas fases
    - permite representar las interacciones entre los actores y las operaciones que inician
  - DSS: muestra, para un escenario específico de un caso de uso:
    - los eventos que generan los actores externos
    - el orden de los eventos
    - eventos entre los sistemas
  - los sistemas se tratan como cajas negras
  - debe realizarse un DSS para el escenario principal de éxito del caso de uso, y los escenarios alternativos complejos o frecuentes
  - los DSS en el Proceso Unificado
    - no aparecen explícitamente en el UP
    - fase de inicio: no se suelen realizar en esta fase
    - fase de elaboración: la mayoría se crean en esta fase, para detallar identificar las operaciones y dar soporte a la estimación de tiempo y costes
  - no es necesario crear DSS para todos

Ejemplo de un DSS: Escenario de Procesar Venta



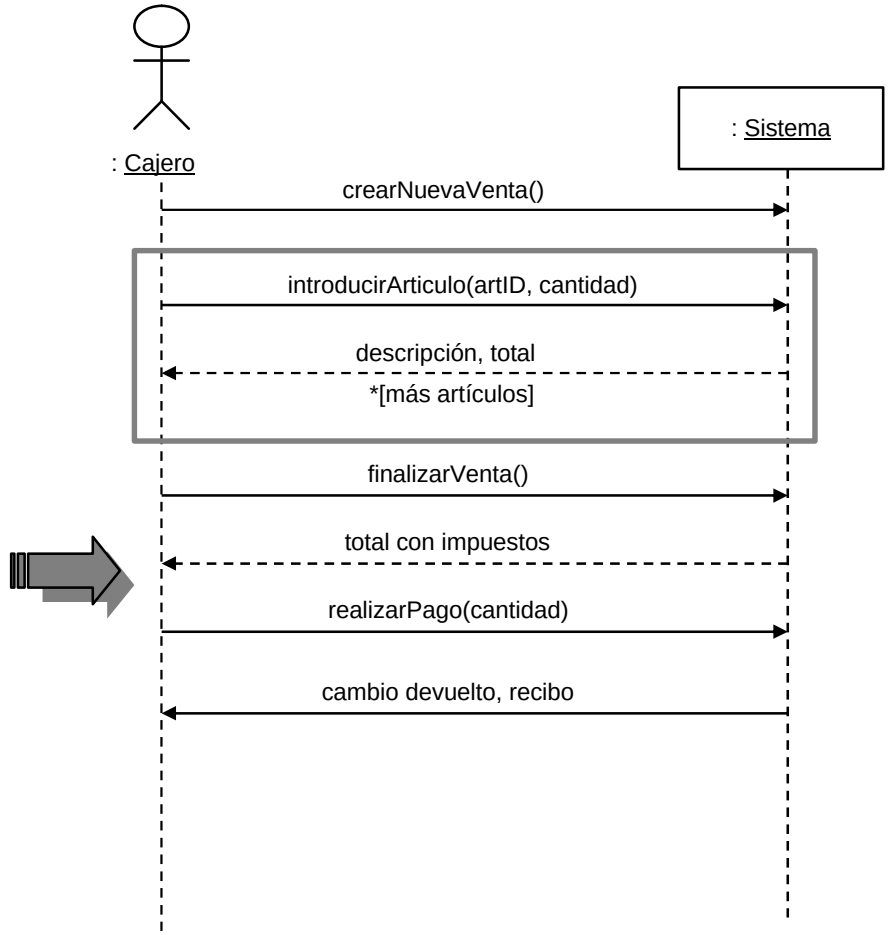
Escenario simple de Procesar Venta para el pago en efectivo.

1. El Cliente llega al terminal PDV
2. El Cajero inicia una nueva venta
3. El Cajero inserta el identificador del artículo
4. El Sistema registra la línea de venta y presenta la descripción del artículo, precio y suma parcial

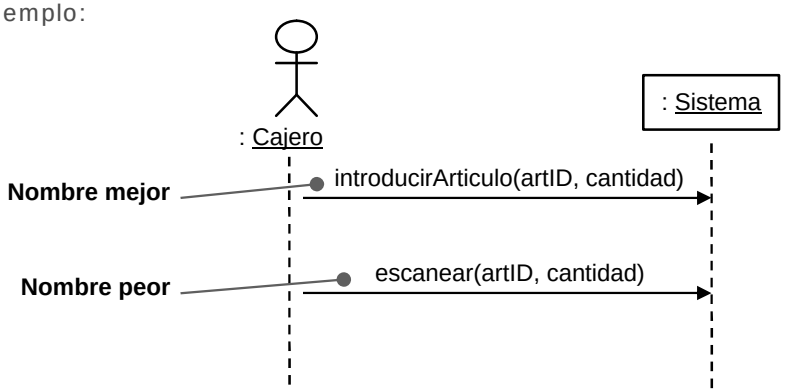
El Cajero repite los pasos 3 y 4 hasta que se indique

5. El Sistema muestra el total con los impuestos calculados
6. El Cajero le dice al Cliente el total, y pide que le pague
7. El Cliente paga y el Sistema gestiona el pago

...

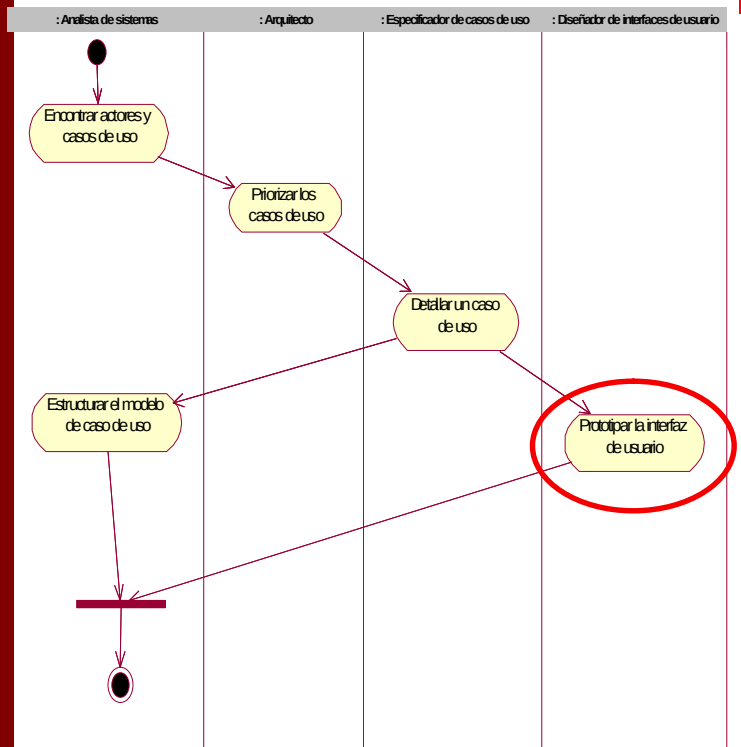


- algunas cuestiones de los DSS
  - se pueden utilizar para ilustrar colaboraciones entre sistemas (en iteraciones posteriores a la de inicio)
  - en ocasiones puede mostrarse el texto (o fragmentos) del caso de uso del escenario junto al diagrama
  - DSS y Glosario: los términos que aparecen en el DSS (operaciones, parámetros, valores de retorno) pueden ser explicados en el Glosario
  - no es necesario crear DSS para todos los escenarios de todos los casos de uso, sino que se crearán sólo para algunos escenarios seleccionados de la iteración actual
  - no se recomienda invertir mucho tiempo en estos diagramas
  - asignación de nombres a eventos y operaciones
    - los eventos (y las operaciones del sistema asociadas) deben expresarse en términos de intenciones del usuario, y no en términos del medio de entrada físico o a nivel de elementos de interfaz
    - debe comenzar con un verbo (añdir, insertar, finalizar, crear,...)
    - ejemplo:

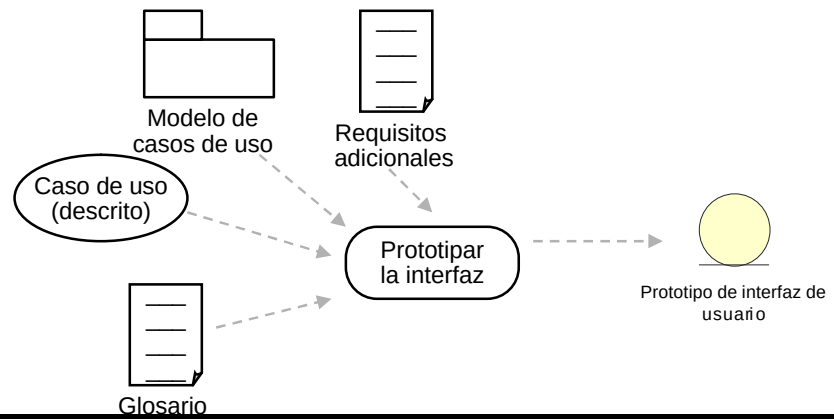


los casos de uso en el desarrollo

- **planificación**
  - antes de realizar estimaciones y planificar todo el proceso del proyecto se necesita tener los casos de uso junto con
    - una idea correcta de qué significa cada uno
    - un correcto entendimiento de quién necesita cada uno y en qué medida lo necesita
    - el conocimiento de qué casos de uso tienen más riesgo
    - un plan del tiempo de implementación de cada caso de uso
  - si existe una entrega de varias versiones,
    - hay que negociar con el cliente qué casos de uso se entrega en cada una considerando cuestiones como
      - necesidad del caso de uso para el cliente
      - tiempo de desarrollo del caso de uso
      - riesgo del caso de uso
    - luego se decide en qué orden se implementan y qué casos de uso pertenecen a cada iteración del sistema
- **aspectos políticos**: realizar casos de uso importantes para aquellas personas con poder para retrasar o cancelar el proyecto
- **aspectos técnicos**: entregar primero los casos de uso con mayor riesgo (puede entrar en conflicto con los criterios anteriores)
- **validación del sistema**: para validar el diseño se pueden tomar uno a uno los casos de uso y comprobar que el sistema permite ejecutarlo



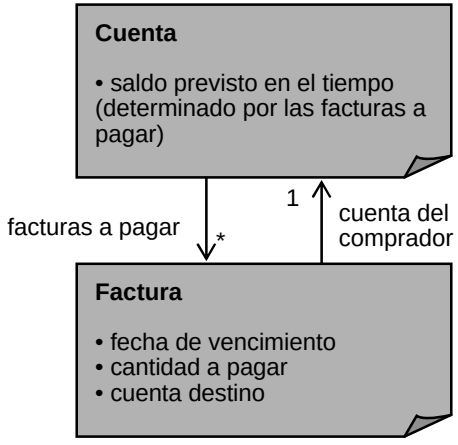
- prototipado de la interfaz
  - diseño lógico de la interfaz: se decide qué se necesita de las interfaces de usuario para habilitar los casos de uso para cada actor
  - diseño físico de la interfaz: se desarrollan prototipos que ilustran cómo pueden utilizar el sistema los usuarios para ejecutar los casos de uso
  - resultado final: conjunto de esquemas de interfaces de usuario y prototipos de interfaces que especifican la apariencia de esas interfaces cara a los actores más importantes



El actor trabajará con elementos de interfaz como Facturas. Factura es por tanto un elemento de la IU. Las facturas tienen fecha de vencimiento, cantidad a pagar y cuenta de destino. Todos estos atributos son necesarios para un actor, que debe decidir si paga o no la factura.

Además, para decidir qué facturas va a pagar, puede querer consultar el saldo de su cuenta (ejemplo de información que un actor necesita cuando ejecuta el caso de uso). Por tanto, la interfaz debe mostrar las facturas según se planifican en el tiempo y cómo afectará el pago planificado de las facturas al saldo.

Por eso, la Cuenta es otro elemento de la IU.



diseño lógico de la interfaz de usuario

- al interactuar con el sistema, los actores utilizarán elementos de interfaces que representan atributos de los casos de uso, y suelen ser términos del glosario (*balance de cuenta, fecha de vencimiento, titular de cuenta,...*)
- diseñador de interfaces
  - recorre todos los casos de uso a los que el actor puede acceder
  - identifica y especifica cada elemento, actor por actor
  - asocia los elementos apropiados de la interfaz de usuario para cada caso de uso
- cuestiones a plantearse para cada actor
  - elementos de interfaz necesarios para posibilitar el caso de uso
  - relación entre esos elementos
  - modo de utilización en los diferentes casos de uso
  - apariencia de los elementos
  - modo de manipulación
- determinar qué elementos de interfaz deben ser accesibles al actor en cada caso de uso
  - clases del dominio, entidades del negocio o unidades de trabajo son adecuadas como elementos de interfaz para cada caso de uso
  - elementos de la IU con la que va a trabajar el actor
  - acciones que puede invocar el actor, y decisiones que puede tomar
  - guía o información que necesita el actor antes de invocar cada acción de los casos de uso
  - información que debe proporcionar el actor al sistema
  - información que debe proporcionar el sistema al actor
  - valores medios de todos los parámetros de entrada y salida (necesario para optimizar la interfaz gráfica)
- se recomienda utilizar hojas adhesivas para representar elementos de interfaz

- diseño físico de la interfaz de usuario
  - pasos
    - se preparan esquemas aproximados de la configuración de elementos de las interfaces
    - se bosquejan elementos adicionales necesarios para combinar varios elementos de interfaces de usuario (carpetas, ventanas, herramientas, controles,...)
    - se construyen prototipos ejecutables de las configuraciones más importantes (por ejemplo, con una herramienta de prototipado rápido)
  - puede haber varios prototipos, uno por actor, para verificar que cada actor puede ejecutar el caso de uso que necesita
  - revisión y validación
    - puede hacerse superficialmente y corregirse después, durante el diseño
    - debe verificarse
      - que cada actor navegue de forma adecuada
      - que proporcione apariencia agradable y una forma consistente de trabajo
      - que cumple con estándares relevantes como el color, tamaño de botones, situación de barras de herramientas,...

identificación de requerimientos no funcionales

- requerimientos no funcionales
  - aspectos del sistema visibles para el usuario, que no stán relacionados de forma directa con el comportamiento funcional del sistema.
  - abarcan diversos aspectos:
    - interfaz de usuario y factores humanos: tipo de interfaz, experiencia,...
    - documentación:documentación requerida, destinatarios, tipo de documentación técnica,...
    - consideraciones de hardware: compatibilidad con otro hardware, existencia de otros sistemas,...
    - características de ejecución: usuarios concurrentes, carga de trabajo,...
    - gestión de errores y excepciones
    - cuestiones de calidad: fiabilidad, disponibilidad, robustez,...
    - modificaciones futuras
    - ambiente físico: condiciones climáticas, exposición a golpes, accidentes,...
    - seguridad
    - recursos consumidor por el sistema
  - priorización de los requisitos

identificación de requerimientos no funcionales

- artefacto de UML: Especificación Complementaria
  - captura otros requisitos, información y restricciones que no se recogen en los casos de uso o en el Glosario.
  - comprende:
    - requisitos FURPS+ : funcionalidad, facilidad de uso, fiabilidad, rendimiento y soporte (*funcionality, usability, reliability, performance, supportability*)
    - informes
    - restricciones de software y hardware (sistemas operativos, plataformas, redes, sistemas de bases de datos,...)
    - restricciones de desarrollo (herramientas de proceso y desarrollo,...)
    - otras restricciones de diseño e implementación
    - cuestiones de internacionalización (monedas, medidas, idiomas,...)
    - documentación (usuario, instalación, administración) y ayuda
    - licencia y cuestiones legales
    - estándares (técnicos, de seguridad, de calidad,...)
    - cuestiones del entorno físico (temperatura, vibraciones,...)
    - cuestiones operacionales (gestión de errores, frecuencia de copias de seguridad,...)
    - reglas del dominio o negocio
  - ver ejemplo en *C. Larman, UML y patrones, págs 80-84*

- glosario
  - lista de los términos relevantes y sus definiciones
  - reduce problemas de comunicación y requisitos ambiguos: evita que un término se utilice de forma distinta por diferentes personas involucradas en el desarrollo
  - debe comenzarse cuanto antes
- el glosario como Diccionario de Datos
  - el diccionario de datos recoge datos sobre los datos (metadatos)
  - fase de inicio: el glosario debe ser un documento sencillo de términos y descripciones
  - fase de elaboración: se amplía a un diccionario de datos, incorporando atributos sobre los términos:
    - alias
    - descripción
    - formato (tipo, longitud, unidad)
    - relaciones con otros elementos
    - rango de valores
    - reglas de validación
  - importante tener en cuenta las unidades (medida, moneda,...)
- puede haber términos compuestos
  - hay términos que incluyen otros elementos
  - ejemplo: Venta puede incluir fecha, lugar, cliente, lugar,...

GLOSARIO

Historial de revisiones

fuente: C. Larman, UML y patrones, 2002

Versión	*Fecha*	*Descripción*	*Autor*
Borrador de Inicio	10 de octubre, 2003	Primer borrador. Se refinará durante la elaboración	Juan Pérez

Definiciones

Término	*Definición e información*	*Alias*
artículo	Un artículo o servicio en venta	
autorización de pago	Validación llevada a cabo por un servicio externo de autorización de pago, que hará o garantizará el pago al vendedor	
solicitud de autorización de pago	Un conjunto de elementos enviados electrónicamente a un servicio de autorización, normalmente como un array de caracteres. Los elementos comprenden: •ID de la tienda •número de cuenta del cliente •cantidad •fecha	
UPC	Código de 12 dígitos que identifica un artículo. Normalmente se representa mediante un código de barras en los artículos.	Código de Producto Universal
…	…	…

el documento de análisis de requerimientos

- documentos de análisis de requerimientos (RAD)
  - en él se documentan los resultados de la actividad de obtención de requerimientos y la actividad de análisis
  - describe totalmente el sistema desde el punto de vista de los requerimientos funcionales y no funcionales
  - sirve como base contractual entre cliente y desarrolladores
  - usuarios del RAD:
    - cliente
    - usuarios
    - administradores del proyecto
    - analistas del sistema
    - diseñadores del sistema
  - debe escribirse cuando el modelo de casos de uso sea estable, es decir, cuando casi no haya modificaciones de requerimientos
  - se actualiza durante el proceso de desarrollo cuando se descubren problemas de especificaciones o cuando cambia el alcance del sistema

el documento de análisis de requerimientos

1. 1. Introducción
    - 1.1. Propósito del sistema
    - 1.2. Alcance del sistema
    - 1.3. Objetivos y criterios de éxito del proyecto
    - 1.4. Definiciones, siglas y abreviaturas
    - 1.5. Referencias
    - 1.6. Panorama
2. Sistema actual
3. Sistema propuesto
    - 3.1. Panorama
    - 3.2. Requerimientos funcionales
    - 3.3. Requerimientos no funcionales
        - 3.3.1. Interfaz de usuario y factores humanos
        - 3.3.2. Documentación
        - 3.3.3. Consideraciones de hardware
        - 3.3.4. Características de rendimiento
        - 3.3.5. Gestión de errores y condiciones extremas
        - 3.3.6. Cuestiones de calidad
        - 3.3.7. Modificaciones al sistema
        - 3.3.8. Ambiente físico
        - 3.3.9. Cuestiones de seguridad
        - 3.3.10. Cuestiones de recursos
        - 3.4. Seudorrequerimientos (requerimientos de implementación)
    - 3.5. Modelos del sistema
        - 3.5.1. Escenarios
        - 3.5.2. Modelo de casos de uso
        - 3.5.3. Modelo de objetos
            - 3.5.3.1. Diccionario de datos
            - 3.5.3.2. Diagramas de clases
        - 3.5.4. Modelos dinámicos
        - 3.5.5. Interfaz de usuario: rutas de navegación y maquetas de pantallas
3.6. Glosario

plantilla de ejemplo de un Documento de Análisis de Requerimientos

Sommerville, I. *Ingeniería de Software*, cap. 5

Larman, C. *UML y patrones*, cap. 6, 7 y 9.

Stevens, P., *Utilización de UML en ingeniería del software con objetos y componentes,* cap. 7 y 8

Bruegge, B., Dutoit, A.H., *Ingeniería del Software Orientado a Objetos*, cap. 4

Jacobson, Rumbaugh y Booch, *El Proceso Unificado de Desarrollo*, cap. 7