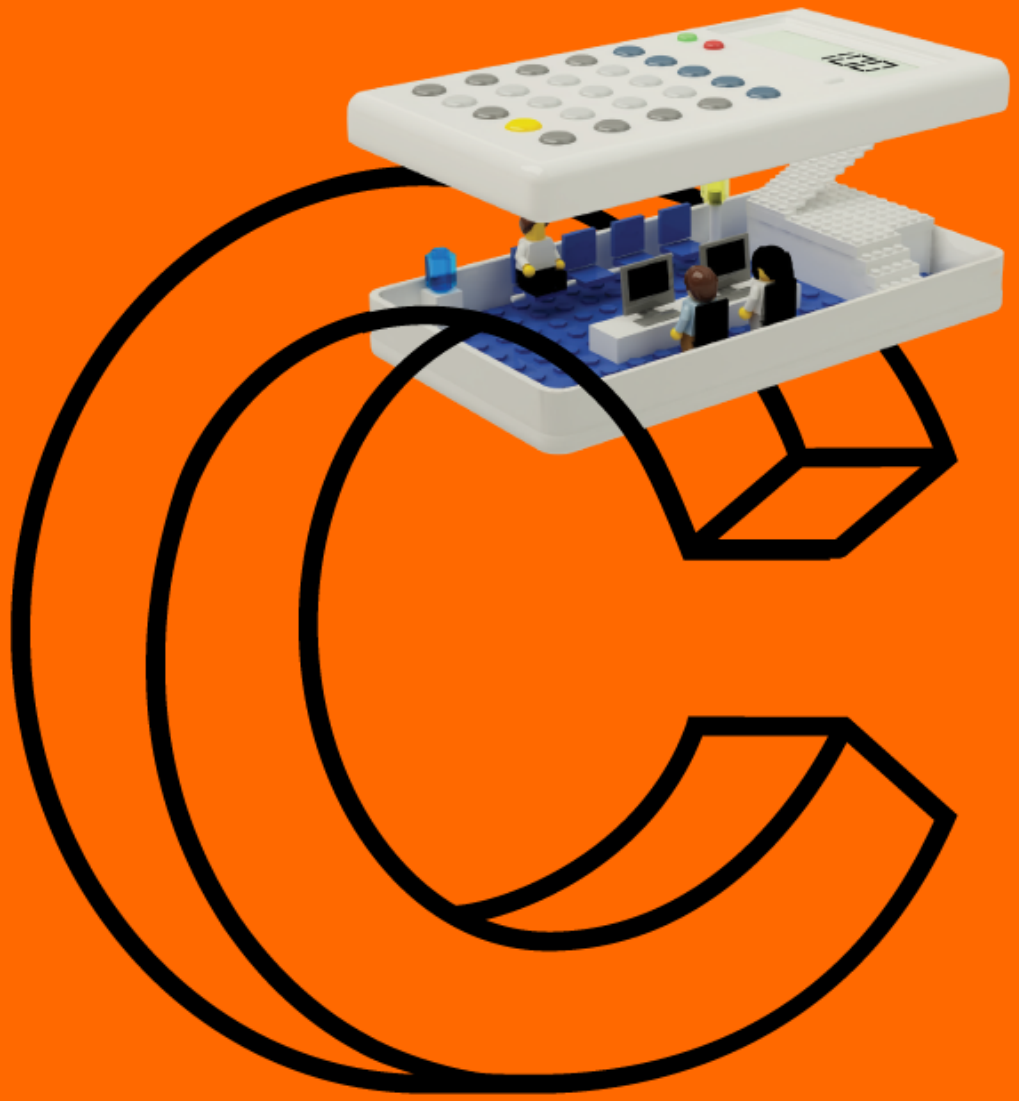
Ingeniería de Requerimientos de Software

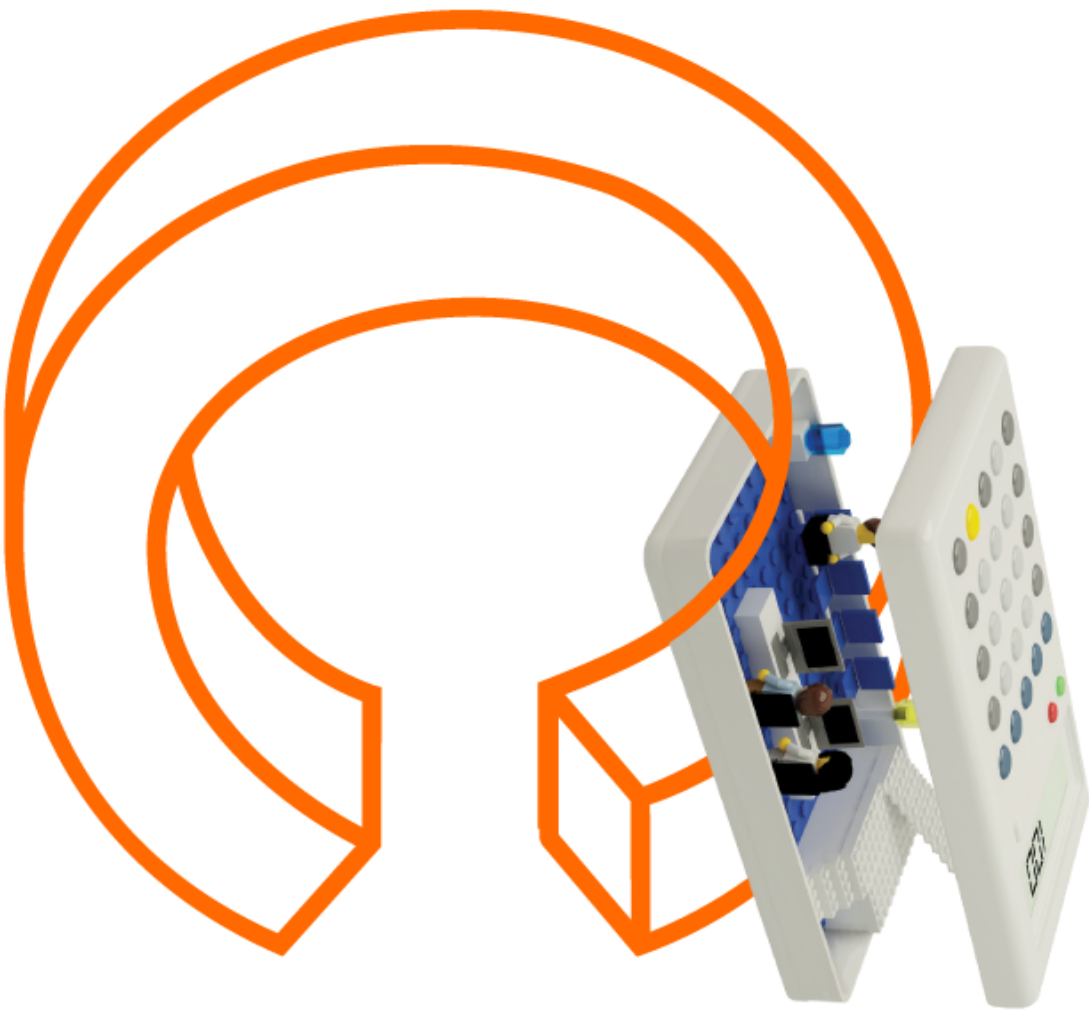


idat

Mg. Daniel Arias, **PMP™** y **Scaled Scrum Master™**

Modelo de Casos de Uso de Sistemas

Semana 6



Semana 6: Modelo de Casos de Uso de Sistemas

Logro de aprendizaje

Comprender el problema

- Atender solicitudes de partes interesadas
- Encontrar actores y casos de uso
- Relaciones entre caso de uso
  - Asociación
  - Generalización
  - Include
  - Extend

Actividades

Actividad 06:

- Elabora el Modelo de Casos de Uso de Sistema.

Guía IR-06:

- Modelo de Casos de Uso de Sistema.

Aprendemos:

¿Qué aprendimos la clase pasada?

- Disciplina de Requerimientos
- Objetivos de la Disciplina de Requerimientos
- Flujo de Trabajo de la Disciplina Requerimientos
- Artefactos de la captura de requisitos
- Actores y Casos de Uso de Sistemas
- Matriz de Requerimientos

Aprendemos:

1.- Introducción

Uno de los primeros procesos que se realizan en un proyecto de construcción de software es la **especificación de requisitos**. Los objetivos de este proceso son **identificar, validar y documentar los requisitos de software**; es decir determinar las características que deberán tener el sistema o las restricciones que deberá cumplir para que sea aceptado por el cliente y los futuros usuarios del sistema de software.

Aprendemos:

1.- Introducción

El producto final es el documento de especificación de requisitos de software y en este se señala -con el detalle adecuado-, lo que el **usuario necesita del sistema de software**. Es por ello, que el documento de requisitos de software se considera como un **contrato entre el cliente y el equipo de desarrollo** del sistema.

Actualmente, el desarrollo orientado a objetos y el uso de UML se han incrementado. Es por ese motivo que el empleo de casos de uso se esta imponiendo frente a otras técnicas de especificación de requisitos.

Lamentablemente, la bibliografía existente muestra muchas formas de aplicar los casos de uso y no son pocas las veces en que su empleo es incorrecto. Algunas de las causas son:

- **Mala interpretación** del estándar UML y
- **Secuencia incorrecta de actividades** para la creación de casos de uso.

Aprendemos:

2.- UML y la Especificación de Requisitos

Para determinar la funcionalidad de un sistema a desarrollar, UML señala el uso de dos elementos: el **actor** y el **caso de uso**.

Actor

Caso de Uso

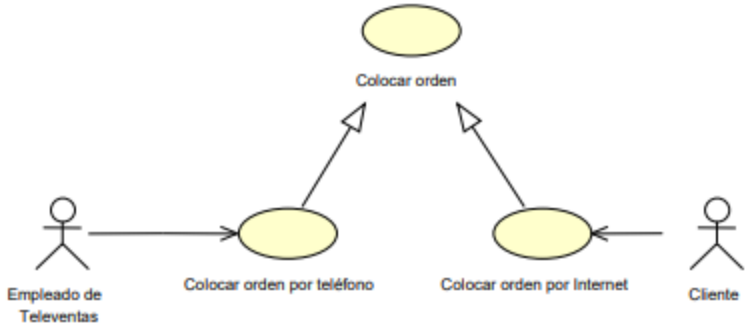
El actor representa una **entidad** externa que **interactúa con el sistema**. Las entidades externas podrían ser personas u otros sistemas. Es importante resaltar que los **actores son abstracciones de papeles o roles** y no necesariamente tienen una correspondencia directa con personas.

A diferencia del actor, el caso de uso hace referencia al sistema a construir, detallando su comportamiento, el cual se traduce en resultados que pueden ser observados por el actor. Los casos de uso **describen las cosas que los actores quieren que el sistema haga**, por lo que un caso de uso debería ser una tarea completa desde la perspectiva del actor.

Aprendemos:

2.- UML y la Especificación de Requisitos (cont.)

Los actores y los casos de uso forman un modelo al que se le denomina “modelo de casos de uso”. Dicho modelo muestra el comportamiento del sistema desde la perspectiva del usuario y servirá como producto de entrada para el análisis y diseño del sistema.



UML especifica que para representar gráficamente la relación entre un actor y caso de uso se debe trazar una línea que los una a la que se le denomina “relación de comunicación”. Además, UML señala que los casos de uso pueden tener relaciones entre sí. Los tipos de relaciones que pueden existir son:

- Include
- Extends y
- Generalización

Aprendemos:

2.- UML y la Especificación de Requisitos (cont.)

2.1. Caso de Uso Vs. Requerimiento Funcional

Existe una correspondencia directa entre ambos, la diferencia radica en la manera en que describen la necesidad de funcionalidad.

- Los Requerimientos Funcionales se describen desde la perspectiva del usuario o cliente del proyecto.
- Los CUS se describen desde la perspectiva de la arquitectura del sistema.

Aprendemos:

3.- Actividades para la Especificación de Requisitos con Casos de Uso

Los resultados de la especificación de requisitos son dos productos: el catalogo de requisitos y el documento de especificación de requisitos de software. El primero de ellos contiene la lista de requisitos de software clasificada por tipo y prioridad; y el segundo de ellos, especifica el comportamiento del sistema a un grado de detalle mayor al del catalogo de requisitos.

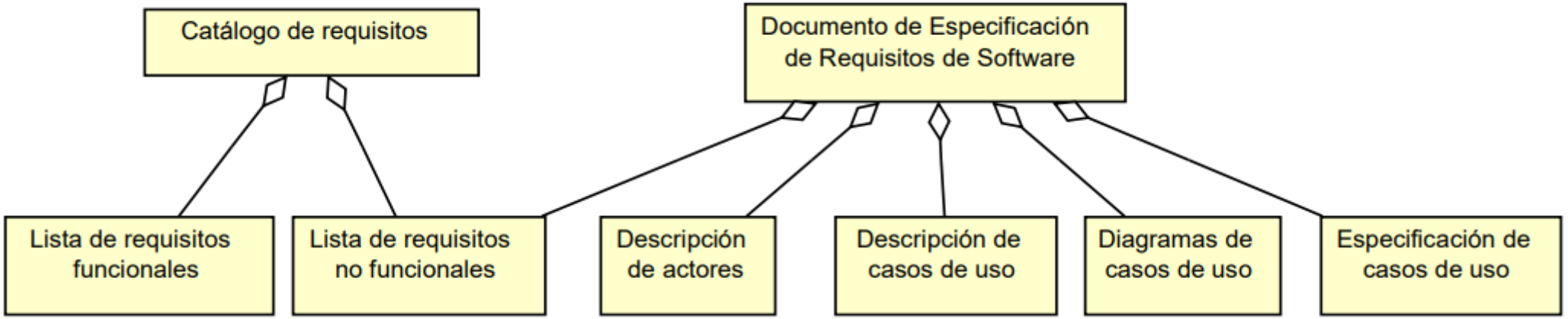


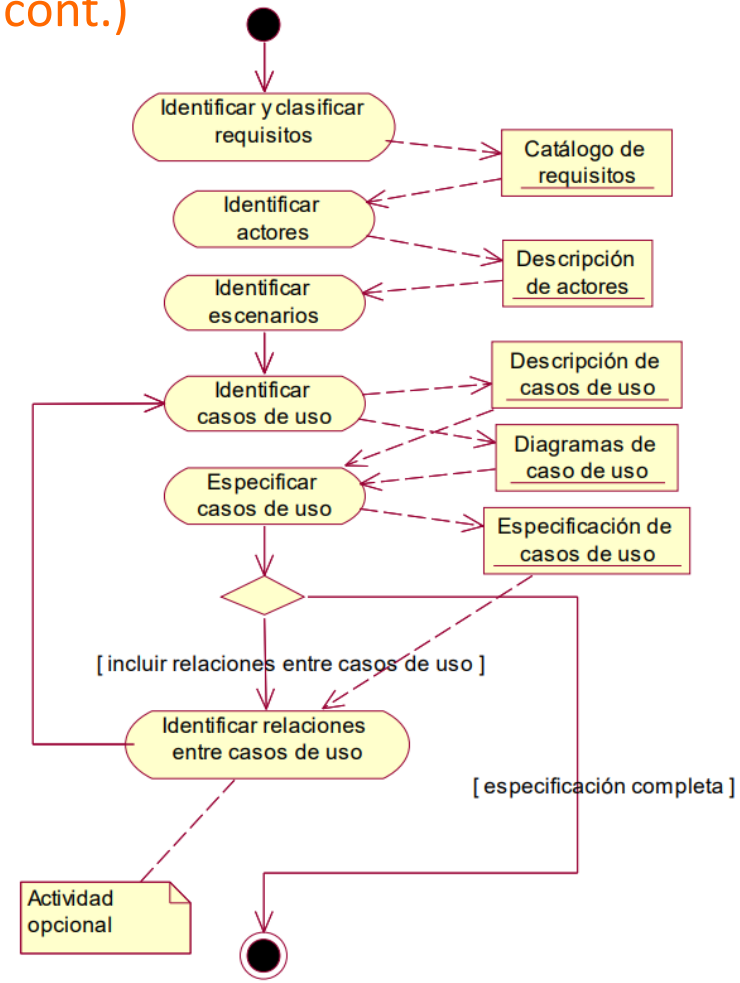
Diagrama de clases que representan los productos de la especificación de requisitos

Aprendemos:

3.- Actividades para la Especificación de Requisitos con Casos de Uso (cont.)

Las actividades para la especificación de requisitos de software usando casos de uso son las siguientes:

1. Identificar y clasificar requisitos
2. Identificar actores
3. Identificar escenarios
4. Identificar casos de uso
5. Especificar casos de uso
6. Identificar relaciones entre casos de uso.



Aprendemos:

Actividad 1: Identificar y clasificar requisitos

Esta actividad es el punto de partida para las siguientes actividades del proceso de obtención de requisitos y se refiere a la **identificación de los requisitos** del sistema de software a desarrollar.

En esta actividad, deberemos responder a los siguientes cuestionamientos: ¿qué le permitirá hacer, el sistema de software al usuario? y ¿el cliente o usuario me solicita alguna restricción para construir el sistema de software?

Contestando a esas preguntas realizar una lista que contendrá los requisitos del sistema.

Luego de haber obtenido la lista, estos deberán ser clasificados en dos grupos:

- Requisitos funcionales y
- Requisitos no funcionales.

Aprendemos:

Actividad 1: Identificar y clasificar requisitos (cont.)

Ejemplo de Requisitos funcionales y no funcionales

Requisitos Funcionales	Requisitos No Funcionales
1. El sistema permitirá registrar los clientes de la empresa.	3. La interfaz de usuario del sistema se implementará sobre un navegador Chrome Versión 74.0 o posterior
2. El sistema permitirá a los usuarios realizar una búsqueda de los clientes por DNI, nombre o apellido.	4. El sistema deberá soportar al menos 20 transacciones por segundo
	5. El sistema permitirá que los nuevos usuarios se familiaricen con su uso en menos de 15 minutos.

Luego, estos requisitos se clasificarán según su importancia, obteniéndose, de esta manera, una lista que contendrá los requisitos clasificados por dos criterios:

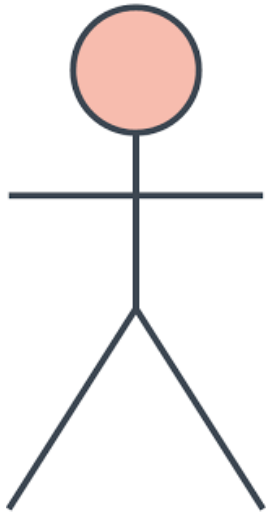
- Tipo de requisito (funcional y no funcional) e
- Importancia.

A esta lista se le conoce como catalogo de requisitos.

Aprendemos:

Actividad 2: Identificar actores

Luego de haber identificado los requisitos funcionales y no funcionales se procederá a identificar los actores del sistema. Para encontrar actores del sistema se puede buscar en las categorías de personas, otro software, dispositivos de hardware o redes de computadoras.



Para un sistema de biblioteca, los actores podrían ser: bibliotecario y cliente (si es que hay módulos de consulta de libros). En el caso de un sistema de ventas, los actores podrían ser: el cliente (si se realiza ventas por Internet), el vendedor y el sistema de facturación.

En un sistema, un usuario del sistema puede actuar como muchos actores; por ejemplo, en un banco, Juan Pérez podría ser cliente y operador dependiendo el momento y el uso que haga del sistema.

Aprendemos:

Actividad 3: Identificar escenarios

Un escenario, según Bruegge “es una descripción concreta, enfocada e informal de una sola característica del sistema desde el punto de vista de un solo actor”; es decir, un escenario muestra la secuencia de pasos que se produce cuando un actor interactúa con el sistema en una situación específica y un tiempo determinado.

Un ejemplo de escenario para un sistema de biblioteca es el siguiente: “Juan Pérez se conecta al sistema de la Biblioteca Nacional a través de Internet. Juan Pérez selecciona realizar búsqueda y cuando aparece el formulario ingresa en título de libros la frase ‘especificación de requisitos’. El sistema encuentra un único libro y lo muestra, el libro de la biblioteca es ‘Especificación de Requisitos de Software’ de Alan Davis y código B 73-825”

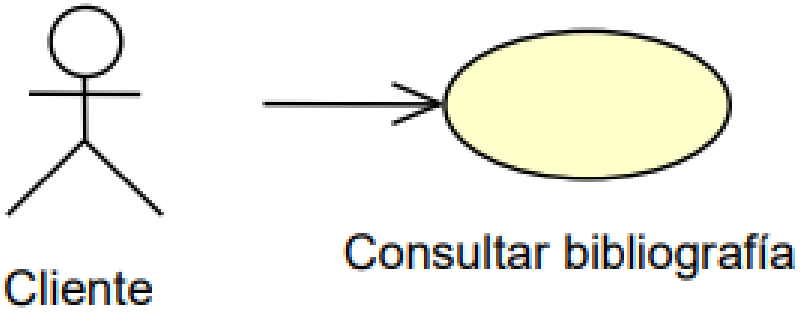
Cabe resaltar que no es necesario documentar los escenarios de manera formal. Esto quiere decir que carece de importancia la creación de documentos que describan todos los escenarios posibles del sistema, ya que su propósito es servir en la identificación de los casos de uso del sistema.

Aprendemos:

Actividad 4: Identificar casos de uso

La **diferencia** entre escenarios y casos de uso **radica en que un escenario es una instancia de un caso de uso**. El caso de uso es el que especifica todos los escenarios posibles para una parte de funcionalidad dada; es decir, todos los escenarios similares se agrupan en un solo caso de uso.

Por ejemplo, en el sistema de biblioteca las consultas de bibliografía por Internet por parte de Juan Pérez y cualquier otro cliente se puede agrupar en un solo caso de uso, al que se le puede denominar “Consultar bibliografía”.



Aprendemos:

Actividad 5: Especificar de casos de uso

Luego de haber identificado los casos de uso, se tienen que indicar, detalladamente, la forma en la que el actor interactúa con el sistema. Esto se determina mediante la especificación y documentación de cada caso de uso.

RUP precisa que la especificación de cada caso de uso debe contener lo siguiente:

1. Precondiciones, que señalan los estados en que debe estar el sistema para que se pueda ejecutar el caso de uso.
2. Flujo básico, que señala la secuencia de pasos que se va a producir en la mayoría de las veces en que ese caso de uso se ejecute.
3. Flujos alternativos, que contienen las secuencias de pasos que se producirán como alternativas al flujo básico del caso de uso; es decir, especifican los pasos que se producirán en situaciones excepcionales.
4. Postcondiciones, que señalan el estado en que el sistema quedara luego de haberse ejecutado el caso de uso.

Aprendemos:

Actividad 5: Especificar de casos de uso (cont.)

Es importante resaltar que durante esta actividad se pueden producir las siguientes situaciones que deben tenerse en cuenta y que no deben ser motivo de preocupación:

1. Un caso de uso que debe partirse en dos casos de uso.
2. Un caso de uso que debe eliminarse, ya que debería formar parte de otro caso de uso.
3. Dos casos de uso que deben formar uno solo.

1		**Proyecto:**	Sistema de Finanzas Itsae
2	**Caso de Uso**	**CU-07**	**Generar Contrato**
3	**Nivel:**	Avanzado	
4	**Destinatario:**	Desarrolladores	
5	**Descripción:**	Permite generar el contrato des estudiante, permitiendo al cajero generar sus respectivos cobros y poder entregar el contrato.	
6	**Autores:**	Geovanny Beltrán Evelyn Pasquel Pedro Abad Daniel Guamán	
7	**Actores:**	Cajero/a	
8	**Requerimientos:**	Gestionar Matricula , Gestionar ítem de cobro	
9	**Verificado por:**	Cajero , Tnlg.Pedro Abad	
10	**Precondiciones:**	Generar ítem de cobro	
11	**Postcondiciones:**	-	
12	**Flujo Principal:**	**Paso**	**Acción**
		1	Solicito buscar contrato
		2	Buscar contrato
		3	Genera el cobro de matricula
		4	Realiza cobro
		5	Muestra el contrato
		6	Finalizar
13	**Flujo Secundario:**	-	-
14	**Excepciones:**	Pasos	Acción
15	**Relaciones:**	2	CU Buscar contrato,CU Realizar cobro
16	**Prioridad:**	Indispensable	
17	**Clases:**		
18	**Tablas:**	Contrato, Matricula, Detalle Contrato, Datos alumno	
19	**Datos:**	Datos alumno, Datos contrato, Datos matricula.	
20	**Métodos:**	Buscar contrato (), Genera cobro (), Mostrar contrato ().	
21	**Interfase:**		

Aprendemos:

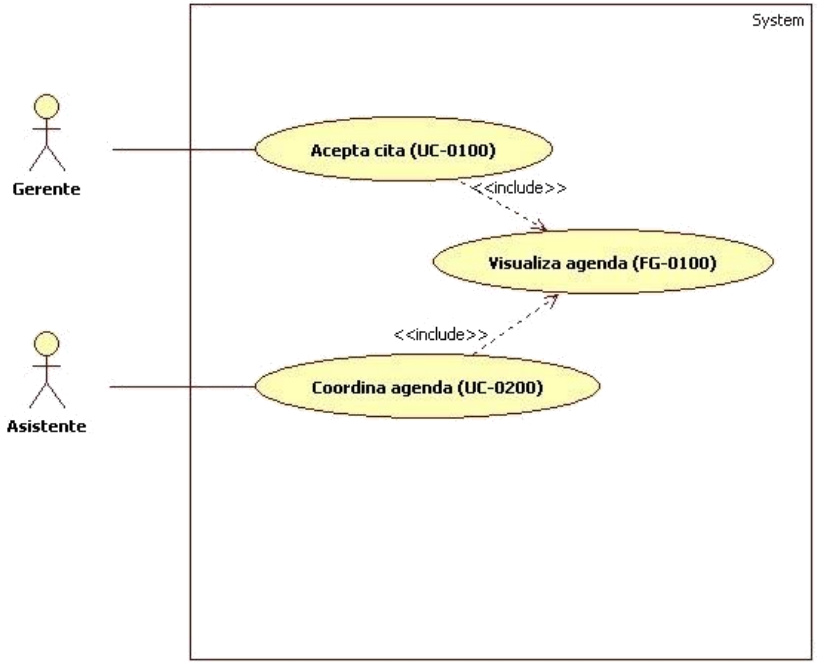
Actividad 6: Identificar relaciones entre casos de uso (opcional).

En esta actividad se identifican, en base a las especificaciones de casos de uso, las relaciones “include”, “extends” y “generalization” entre casos de uso. Es importante resaltar que esta actividad es opcional.

Aprendemos:

Actividad 6: Identificar relaciones entre casos de uso (cont).

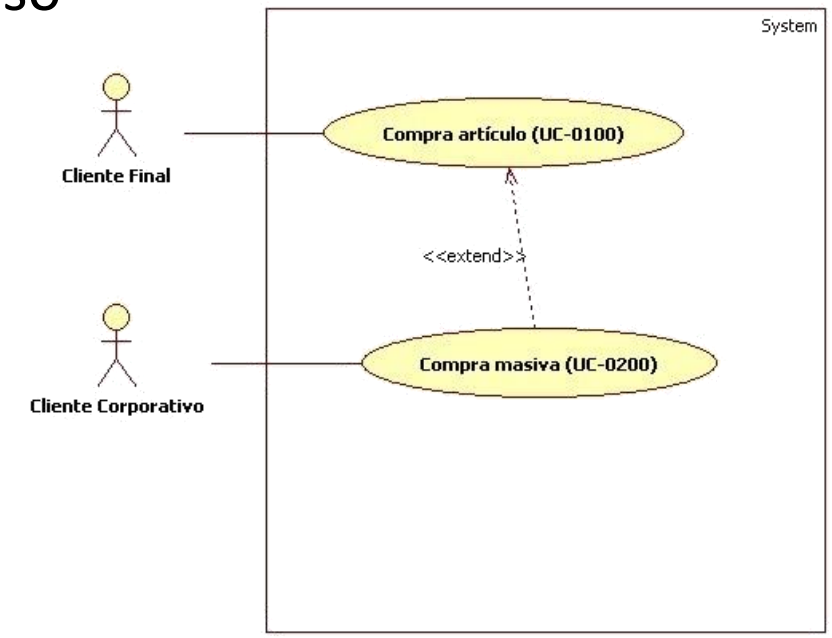
- **include** relación que define un caso de uso obligatorio, pero que también puede ser usado por otros casos de uso.
  - Se establece cuando el caso de uso base necesita incluir obligatoriamente la secuencia de acciones descritas por el caso de uso incluido.
  - Indica que el comportamiento del caso de uso incluido está explícitamente insertado dentro del comportamiento definido por el caso de uso base.
  - El caso de uso base es el que conoce la asociación entre ambos y el caso de uso incluido, no necesita conocer cuáles casos de uso lo incluyen.



Aprendemos:

Actividad 6: Identificar relaciones entre casos de uso (cont).

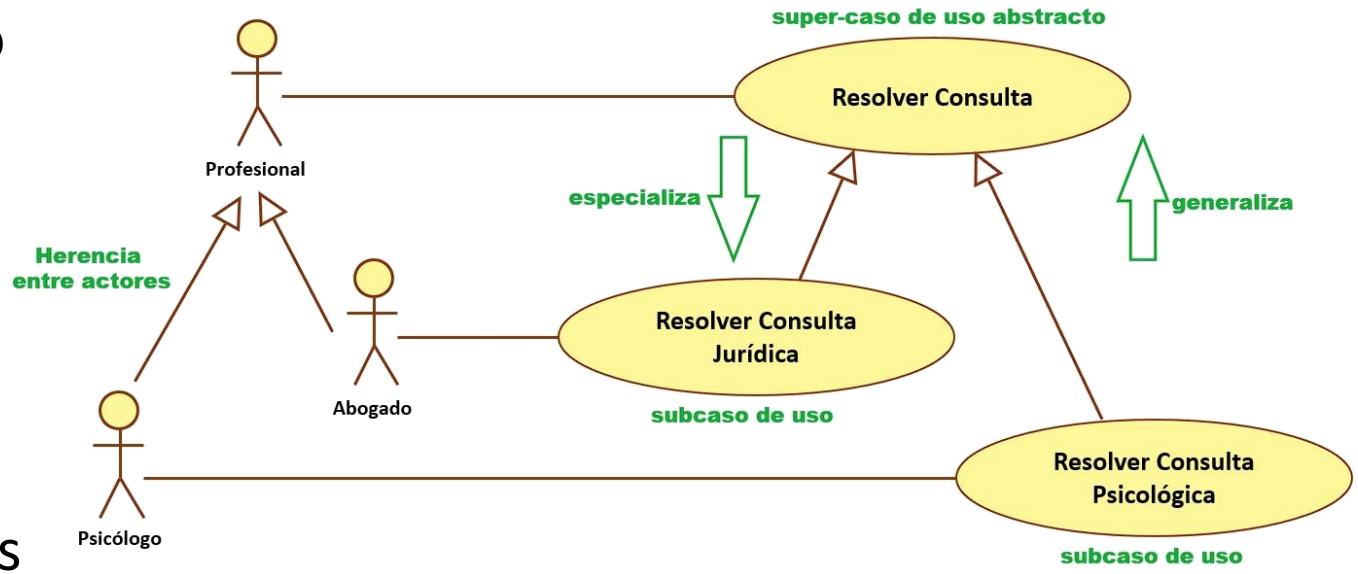
- **extend** relación que define un curso alterno opcional (dependiendo de una condición) de otro caso de uso.
  - Conecta un caso de uso extendido a un caso de uso base.
  - El caso de uso extendido encapsula comportamiento opcional del caso de uso base.
  - El caso de uso extendido es a menudo abstracto, pero no necesariamente tiene que serlo.
  - Su ejecución es opcional.



Aprendemos:

Actividad 6: Identificar relaciones entre casos de uso (cont).

- **generalization** relación entre casos de uso, que es análoga a la relación de “herencia” entre clases.
  - Se utiliza cuando el caso de uso padre debe ser sub clasificado en uno o más casos de uso hijos.
  - El caso de uso hijo hereda la estructura, comportamiento y las relaciones del padre.
  - Este tipo de relación también es utilizado entre actores.



Aprendemos:

4.- Errores Comunes en la Especificación de Requisitos usando Casos de Uso

En cada una de las actividades especificadas anteriormente se producen y generan errores. A continuación, se incluyen algunos de los mas frecuentes:

1. Errores en la identificación de actores
2. Errores en la identificación de casos de uso
3. Errores en la especificación de los casos de uso
4. Errores en el uso de las relaciones entre casos de uso

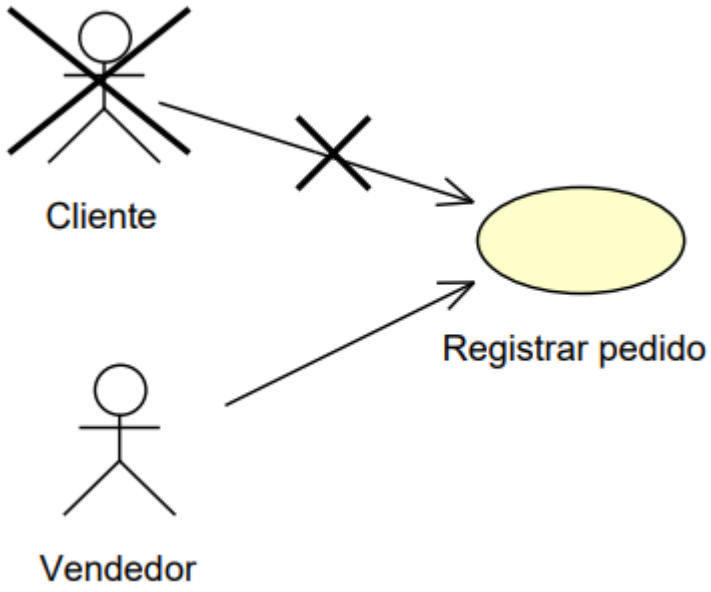
Aprendemos:

4.- Errores Comunes en la Especificación de Requisitos usando Casos de Uso

4.1 Errores en la identificación de actores

Los errores introducidos en esta etapa se deben principalmente a no comprender quienes son los actores del sistema.

En algunos casos se incluyen actores que realmente no lo son; por ejemplo, en un sistema en el que se realizan pedidos de productos, se considera al cliente como un actor. Realmente quien ingresa los pedidos en el sistema es el vendedor y no el cliente, por lo tanto, el vendedor sería el actor del sistema.

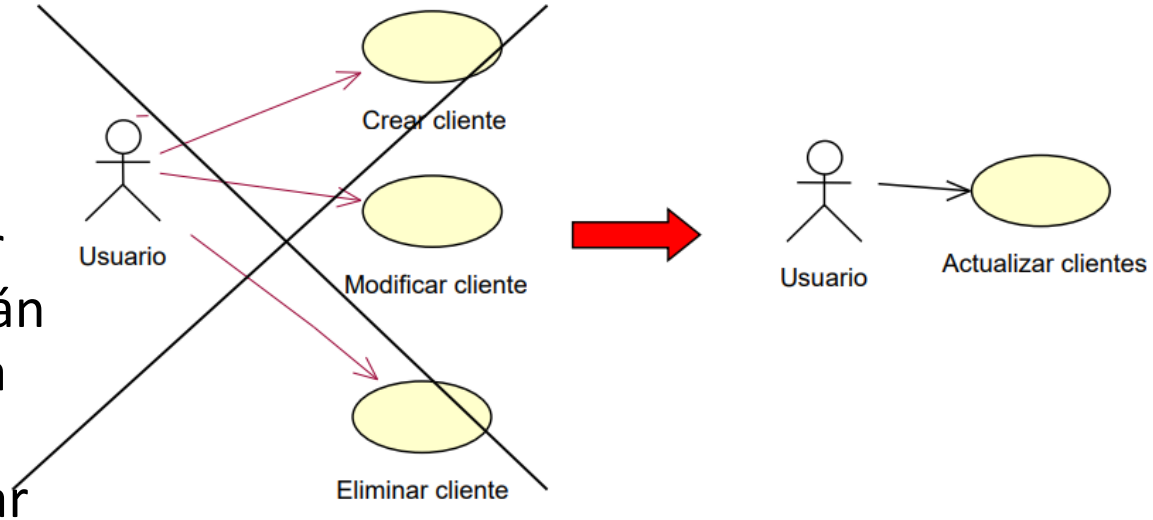


(*) En el ejemplo, si el sistema permitiera Registrar pedidos por Internet, el cliente sí sería un actor del sistema.

Aprendemos:

4.- Errores Comunes en la Especificación de Requisitos usando Casos de Uso
4.2 Errores en la identificación de casos de uso

Un error muy extendido, y que es cometido en la mayoría de la bibliografía sobre casos de uso, es considerar las opciones del menú o funciones del sistema como casos de uso (revisar el libro de Larman).



Kurt Bittner señala que los CU deben mostrar lo que el usuario necesita del sistema y no mostrar las funciones u opciones del menú que permitirán realizar lo solicitado; por ejemplo, en un sistema donde se debe almacenar la información de los clientes, lo que al usuario le importa es actualizar la información de clientes. Esta actividad la podrá realizar accediendo a las opciones del menú agregar, modificar y eliminar clientes; por lo tanto, la funcionalidad del sistema será representada con el caso de uso “Actualizar cliente”

Aprendemos:

4.- Errores Comunes en la Especificación de Requisitos usando Casos de Uso

4.3 Errores en la especificación de los casos de uso

La técnica de CU se debe utilizar para la especificación de requisitos, no para el diseño. Los errores que se producen en esta actividad se deben a la inclusión de cuestiones de diseño en la especificación de CU. Algunos de los errores que se cometen son los siguientes:

- Introducir palabras que se refieran a componentes de ventanas como: botones, listas desplegables, opciones de menú, etc. En la especificación de CU debe incluirse la información que será ingresada o será mostrada, pero no que componente de la ventana se va a utilizar para mostrar dicha información, sino se estaría realizando el diseño de pantallas en el proceso de especificación de requisitos, lo cual sería incorrecto.
- Mencionar elementos correspondientes al diseño de algoritmos o de base de datos en la especificación de CU; por ejemplo, “grabar en la tabla clientes en la base de datos” u “ordena los datos con el algoritmo de la burbuja” son oraciones que no deben incluirse en una especificación; ya que son elementos que se determinan en la etapa de diseño.

Aprendemos:

4.- Errores Comunes en la Especificación de Requisitos usando Casos de Uso
4.3 Errores en la especificación de los casos de uso (cont.)

- Otro error es incluir “etc” o “así sucesivamente” cuando se indica la información que se debe ingresar o mostrar.

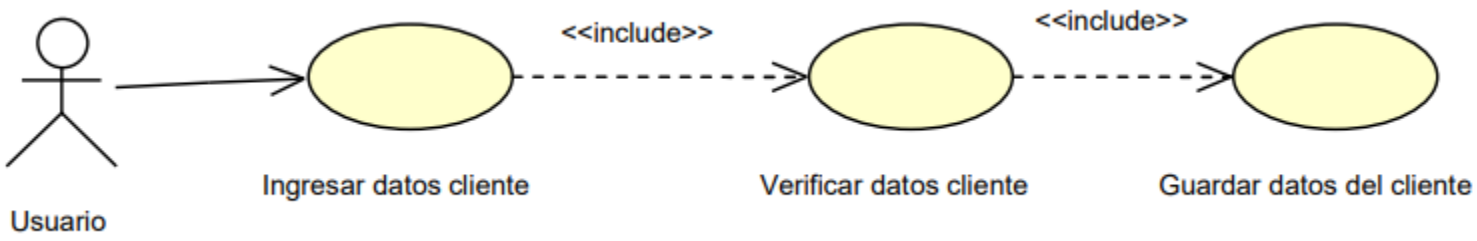
La especificación de CU debe contener información exacta y precisa que permita realizar una buena estimación del esfuerzo requerido para realizar las etapas de análisis, diseño y codificación. Si la información no es exacta, se pueden producir retrasos debido a modificaciones de la base de datos, cambios en el diseño de las pantallas o en el código fuente producto de las especificaciones tardías de requisitos.

Aprendemos:

4.- Errores Comunes en la Especificación de Requisitos usando Casos de Uso
4.2 Errores en el uso de las relaciones entre casos de uso

Los errores que se producen al incluir relaciones entre los casos de uso se deben principalmente a confundir los casos de uso con los procesos de los diagramas de flujo de datos (DFD). Es por eso que se ven diagramas de casos de uso que parecen DFDs.

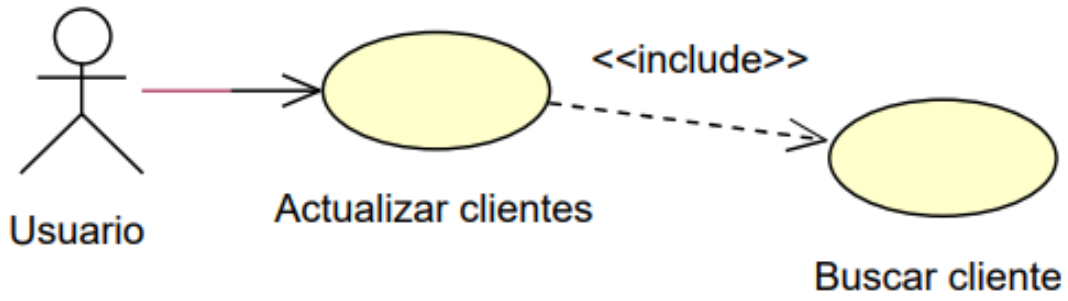
Para evitar cometer este error, se aconseja que no haya mas de dos niveles de relaciones de tipo “include” o “extend”.



Aprendemos:

4.- Errores Comunes en la Especificación de Requisitos usando Casos de Uso
4.2 Errores en el uso de las relaciones entre casos de uso (cont.)

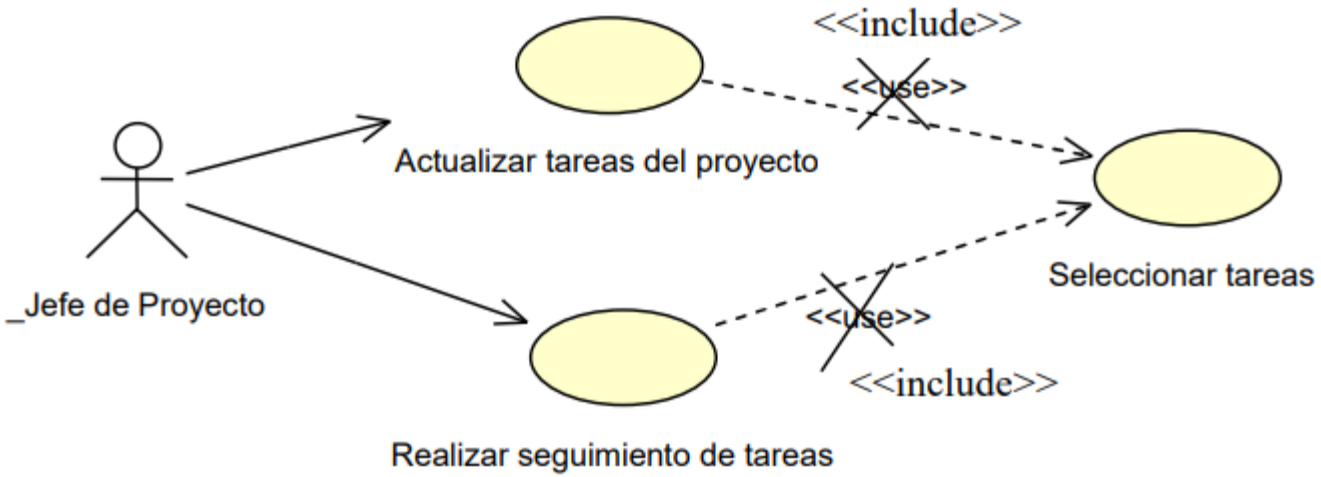
Otro error frecuente es crear un caso de uso que es incluido por un solo caso de uso; por ejemplo, la figura muestra el caso de uso “Buscar Cliente”, el cual es incluido sólo por el caso de uso “Actualizar clientes”. Se debe tener en cuenta que los casos de uso incluidos deben obtenerse luego de haber realizado las especificaciones de los casos de uso, ya que en ese momento es que se determinaran cuales son los pasos que se repiten entre los diferentes casos de uso y es allí donde se determinan las relaciones de tipo “include”.



Aprendemos:

4.- Errores Comunes en la Especificación de Requisitos usando Casos de Uso
4.2 Errores en el uso de las relaciones entre casos de uso (cont.)

Se puede encontrar bibliografía en la que se emplea la relación “use” entre casos de uso. Se debe tener en cuenta que dicha relación corresponde a versiones anteriores a UML versión 1.3, por lo que su utilización debe evitarse



Aprendemos:

UML y la Especificación de Requisitos

¿Dudas?

¿Miedos?

¿Temores?

Notas

Aprendemos:

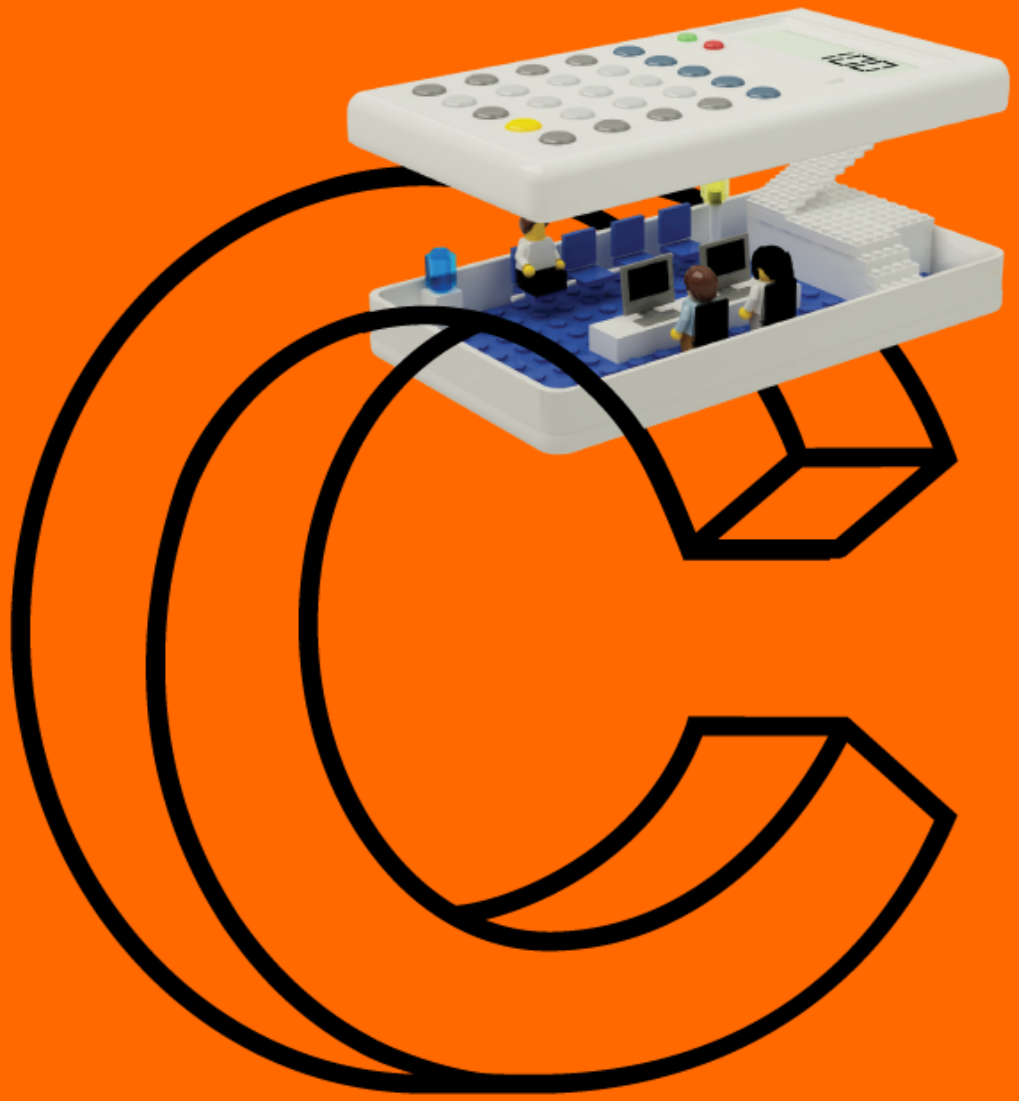
Bibliografía

1. Bittner, K., Why Use Cases Are Not Functions, http://www.therationaledge.com, USA, 2000.
2. Brown, W.J., AntiPatterns: Refactoring Software, Architectures, and Projects in Crisis, John Wiley & Sons, USA, 1998.
3. Bruegge B., Allen, S., Ingeniería de Software Orientado a Objetos, Addison-Wesley, USA, 2002.
4. Gamma, E., Design Patterns: elements of reusable object-oriented software, Addison-Wesley, USA, 1995.
5. Heumann, J., Introduction to Business Modeling Using the Unified Modeling Language, The Rational Edge, March 2001, USA.
6. Larman, C., UML y Patrones, Introducción al Análisis y Diseño Orientado a Objetos, México, Prentice Hall Hispanoamericana, 1999
7. Ministerio de Administraciones Públicas, Análisis del Sistema de Información-Métrica Versión 3, España, 2000.
8. Object Management Group, OMG Unified Modeling Language, http://www.uml.org, USA, 1999
9. Rational Software, Rational Unified Process version 2001A.04.00.13, USA, 2001.
10. Schneider, G., Winters, J.P., Applying Use Cases, Second Edition, Addison-Wesley, Massachussets, USA, 2001.
11. Yourdon E., Análisis Estructurado Moderno, Prentice Hall Hispanoamericana, México, 1989.

Aplicamos lo aprendido: Asignación para la clase siguiente

- Revisión de los trabajos (avance)

Gracias



idat