

# CONSTRUCCION DE SISTEMAS I 2008

Trabajo de desarrollo:

## Proceso Unificado y Modelo de Análisis



Día de cursada: Martes

Turno: noche

Aula: 48

Docentes Cañellas, Liliana.

Gatelli, Renata.

Sanzone, Alejandra.

Alumno: Di Ludovico, Maximiliano.

## Índice de temas

El Proceso Unificado y El Modelo de Análisis .....	3
El Proceso Unificado Dirigido por Casos de Uso .....	4
Ejemplo de Casos de Uso .....	5
Proceso Unificado Centrado en la Arquitectura .....	6
El Proceso Unificado es Iterativo e Incremental .....	7
Conclusiones.....	7
Modelo de Análisis .....	8
Creación del Modelo de Análisis a partir de los Casos de Uso.....	8
La Realización de un Caso de Uso en el Modelo de Análisis .....	9
Descripción del Flujo de Sucesos de una realización de Casos de Uso ...	11
Breve comparación del Modelo de Análisis de Casos de Uso con el Modelo de Análisis .....	12
Conclusiones.....	12
Bibliografía Consultada: .....	13
Enunciado del problema.....	14
Diagrama de Casos de Uso .....	14
Descripción del Caso de Uso:.....	15
Modelo de análisis Registrar Control de Sistema de Facturación .....	16
Diagrama de clases de análisis.....	16
Prototipo de interfaz asociado.....	17
Diagrama de Colaboración de Análisis .....	19

## El Proceso Unificado y El Modelo de Análisis

En la ingeniería de software el objetivo es construir un producto software o mejorar uno ya existente. Un proceso es efectivo si proporciona normas para el desarrollo eficiente de software de buena calidad.

Es necesario un proceso que sirva como guía para todos los participantes ya sea el cliente, el usuario y el desarrollador.

Por cliente entendemos que es una persona, organización o grupo de personas que encarga la construcción del sistema, ya sea empezando desde cero, o mediante el refinamiento de versiones sucesivas mientras que el usuario es la persona que interactúa con el sistema y el desarrollador será el participante en un flujo de trabajo y durante su desarrollo, se identifican las necesidades de los usuarios y de los clientes como requisitos.

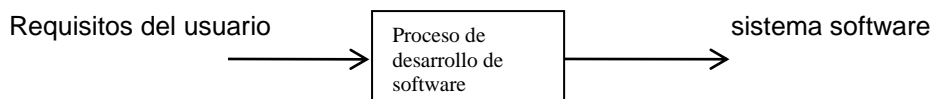
El proceso unificado utiliza el lenguaje unificado de modelado (UML), para preparar todos los esquemas de un sistema software; es un lenguaje que produce dibujos comparables en sus objetivos a los esquemas que se utilizan desde hace mucho tiempo en otras disciplinas técnicas. El proceso pone en práctica el basar gran parte del proyecto de desarrollo en componentes reutilizables, es decir, en piezas de software con una interfaz bien definida (una colección de operaciones que son utilizadas para especificar un servicio de una clase o de un componente).

Hoy en día, la tendencia actual en el software lleva a la construcción de sistemas más grandes y más complejos. Esto es debido en gran parte al hecho de que los computadores son cada vez más potentes y los usuarios esperan más de ellos. También se ha visto afectada por el creciente uso de Internet para el intercambio de todo tipo de información ya sea de texto, fotos, diagramas y de la multimedia y crece a medida que vemos como pueden mejorarse los productos de una versión a otra. También este sistema debe contar con un grado alto de rendimiento y velocidad. El tiempo de salida al mercado es otro factor importante en el proceso de desarrollo de software.

Por lo tanto, la demanda de un potente y complejo producto con estas características hace que se convierta en un problema para los desarrolladores a menos que se renueven los métodos utilizados.

El problema del software se reduce a la dificultad que afrontan los desarrolladores para coordinar las múltiples cadenas de trabajo de un gran proyecto software. La comunidad de desarrolladores necesita una forma coordinada de trabajo. Necesita un proceso que integre las múltiples facetas del desarrollo.

El proceso Unificado es un proceso de desarrollo de software; es el conjunto de actividades necesarias para transformar los requisitos (condición o capacidad que debe cumplir un sistema) de los usuarios en un sistema de software.



El proceso de software es mas simple que un proceso; es un marco de trabajo (un patrón de la arquitectura que proporciona una plantilla extensible para aplicaciones dentro de un dominio específico) que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tamaños de proyectos, etc.

Este marco de trabajo sirve para que los desarrolladores puedan construir herramientas que soporten la automatización del sistema entero.

El proceso unificado esta basado en componentes (parte física y reemplazable de un sistema), lo cual quiere decir que el sistema software (código máquina, los ejecutables y código fuente) en construcción está formado por componentes software interconectados a través de interfaces bien definidas.

Para tal fin, el proceso unificado de desarrollo se basa en tres principios para entender la estructura y dinámica de la organización en la cual el sistema será desplegado, mejorar el objeto (unidad atómica que encapsula estado y comportamiento) organizacional, asegurar a clientes y usuarios un común entendimiento del objeto y derivar los requisitos para soportar la organización objeto; estos son las dirigidas por casos de uso, centrado en la arquitectura y el iterativo e incremental.

## ***El Proceso Unificado Dirigido por Casos de Uso***

Un sistema software se realiza para dar servicio a los usuarios. Por lo tanto, para construir un sistema con éxito debemos conocer lo que los futuros usuarios necesitan y desean, El papel de los desarrolladores es el de ayudar a los usuarios y clientes a comunicar sus necesidades.

El usuario representa alguien o algo que es el que va a interactuar con el sistema que estamos desarrollando. Un ejemplo de interacción seria una persona que utiliza un cajero automático. El usuario inserta la tarjeta, responde a las preguntas que le hace la máquina en la pantalla, y recibe una suma de dinero.

Una interacción como esta es un caso de uso. Un caso de uso es una descripción de un conjunto de secuencias de acciones, incluyendo variaciones que un sistema lleva a cabo y que conduce a un resultado observable de interés para un actor determinado. Es una especificación del comportamiento de una identidad y su situación con los agentes externos. Un caso de uso comprende un inicio que es cuándo y qué actor lo produce, un fin que es lo que produce y qué valor devuelve, una interacción actor-caso de uso que son los mensajes que intercambian ambos, un objetivo que es lo que se quiere llevar a cabo, una cronología y origen de las interacciones, y contempla situaciones opcionales que son los escenarios alternativos que se presentan en el caso de uso.

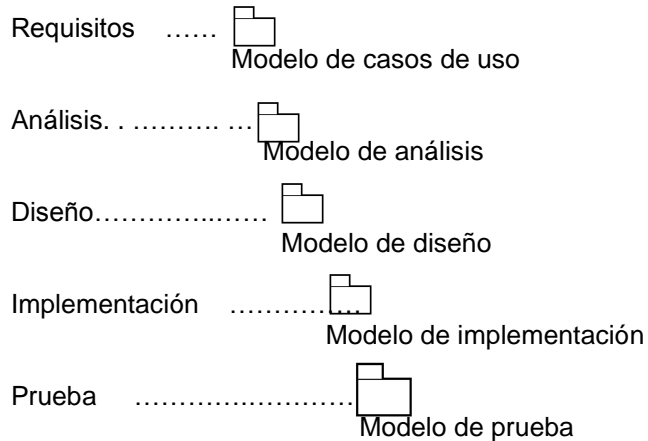
Un sistema puede tener muchos tipos de usuarios. Cada tipo de usuario se representa por un actor. Un mismo usuario puede interpretar varios papeles como actores distintos Podemos encontrar y especificar todos los actores examinando a los usuarios que utilizaran el sistema y a otros sistemas que deben interactuar con él. El nombre del actor describe el papel desempeñado. Los actores utilizan el sistema interactuando con los casos de uso y para obtener algún resultado de valor para un actor. A medida que definimos lo que hacen los actores y lo que hacen los casos de uso, trazaremos una clara separación entre las responsabilidades de los actores y las del sistema. Esta separación no ayuda a delimitar el alcance del sistema. Los casos de uso definen el comportamiento deseado en el sistema, y para comunicar este comportamiento entre las personas involucradas en el sistema.

Todos los actores y casos de uso del sistema forman un modelo de casos de uso que se representa mediante un diagrama de casos de uso el cual describe parte del modelo y muestra un conjunto de casos de uso y actores con una asociación entre cada par actor/caso de uso que interactúan.

Un caso de uso es un fragmento de funcionalidad del sistema que proporciona al usuario un resultado importante; los casos de uso representan los requisitos funcionales (requisitos que especifican una acción que debe ser capaz de realizar el sistema sin restricciones físicas); son una herramienta para especificar los requisitos del sistema. También guían su diseño,

implementación, y prueba; esto es, guían el proceso de desarrollo.

### Flujos de Trabajo y Modelos de Proceso Unificado



La figura muestra la serie de flujos de trabajo y modelos de proceso unificado. Los desarrolladores comienzan capturando los requisitos del cliente en la forma de casos de uso en el modelo de casos de uso. La captura de los requisitos tiene dos objetivos: encontrarlos verdaderos requisitos y representarlos de un modo adecuado a los usuarios, clientes y desarrolladores. Entendemos por verdaderos requisitos a aquellos que cuando se implementen se añadirán el valor esperado para los usuarios. Con respecto a “representarlos de un modo adecuado” quiere decir que la descripción obtenida de los requisitos debe ser comprensible para los usuarios y los clientes.

Después se analizan y diseñan el sistema para cumplir los casos de uso, creando en primer lugar un modelo de análisis, después uno de diseño y después otro de implementación y finalmente preparan el modelo de prueba que les permite verificar que el sistema proporciona la funcionalidad descrita en los casos de uso.

De este modo, los casos de uso dirigen el proceso de desarrollo. “Dirigido por casos de uso” quiere decir que el proceso de desarrollo sigue un flujo de trabajo iniciado por los casos de uso para su realización.

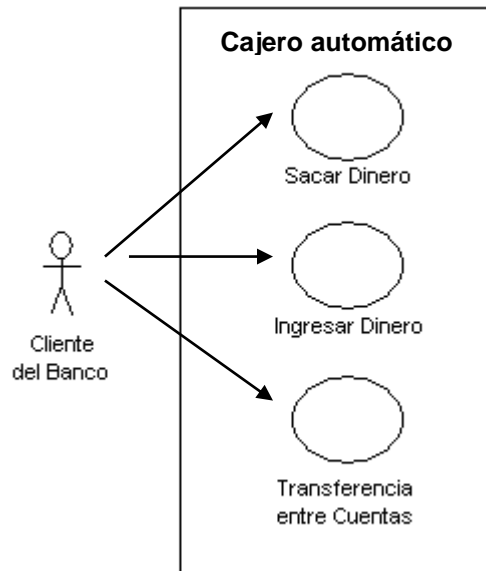
También los casos de uso se utilizan como punto de partida para escribir el manual de usuario. Ya que cada caso de uso describe una forma de utilizar el sistema, son el punto de partida ideal para explicar como puede el usuario interactuar con el sistema.

El modelo de casos de uso esta compuesto por todos los actores y todos los casos de uso del sistema; representan los requisitos funcionales y nos ayuda a delimitar el sistema definiendo todo lo que debe hacer para todos los usuarios. El modelo de casos de uso se describe fundamentalmente mediante el lenguaje natural.

### Ejemplo de Casos de Uso

El actor Cliente de Banco utiliza un sistema de cajero automático para retirar o ingresar dinero o para transferir dinero entre cuentas. Esto se representa por los tres casos de uso que se muestran en la figura y que poseen las asociaciones con el actor para indicar su interacción. Los casos de uso tienen una representación gráfica denominada diagramas de casos de uso. En estos diagramas, los actores se representan en forma de pequeños personajes y los casos de

uso se representan por elipses contenidas dentro de un rectángulo que representa al sistema. La participación de los actores en los casos de uso se indica por una flecha entre el actor y el caso de uso que apunta en la dirección en la que fluye la información. Cada caso de uso puede estar definido por un texto que lo describe, una secuencia de pasos ejecutados dentro del caso de uso, condiciones pre-post para que el caso de uso comience o termine.



La secuencia de acciones para un camino a través del caso de uso realizada durante la operación sacar dinero es:

El cliente del banco se identifica.

El cliente del banco elige de que cuenta sacar dinero y especifica que cantidad.

El sistema deduce la cantidad de la cuenta y entrega el dinero.

## ***Proceso Unificado Centrado en la Arquitectura***

Los casos de uso también ayudan a idear la arquitectura que es la pieza clave que permite comprender el sistema, organizar el desarrollo y hacer evolucionar el software.

La arquitectura en un sistema software se describe mediante diferentes vistas del sistema en construcción. Las vistas muestran los diferentes aspectos del sistema que son modelados. Mediante los diagramas podemos describir el contenido de una vista.

El concepto de arquitectura de software incluye los aspectos estáticos y dinámicos más significativos del sistema. La estructura se describe con los diagramas de clases, componentes y despliegue. El comportamiento dinámico del sistema se describe con diagramas de estados, secuencias, colaboración y actividades.

La arquitectura surge de la necesidad de la empresa, como la perciben los usuarios y se refleja en los casos de uso.

La relación entre los casos de uso y la arquitectura es la interacción entre estos ya que en los casos de uso deben encajar en la arquitectura que se lleve a cabo y la arquitectura debe permitir el desarrollo de todos los casos de uso requeridos, ahora y en el futuro. La arquitectura proporciona una estructura sobre la cual guiar las interacciones mientras que los casos de uso

definen los objetivos y dirigen el trabajo de cada iteración.

## ***El Proceso Unificado es Iterativo e Incremental***

El desarrollo de un producto software comercial supone un gran esfuerzo que puede durar entre varios meses hasta posiblemente un año o más. Es práctico dividir el trabajo en partes mas pequeñas. Cada parte es una interacción que resulta en un incremento, es decir, es una parte pequeña y manejable del sistema. Cada interacción resultara, al menos, en una nueva construcción de forma que se añada un incremento al sistema.

Las interacciones hacen referencia a pasos en el flujo de trabajo, y los incrementos, al crecimiento del producto. En cada interacción los desarrolladores identifican y especifican los caso de uso mas relevantes, crean un diseño utilizando la arquitectura seleccionada como guía, implementan el diseño mediante componentes, y verifica que los componentes satisfacen los casos de uso.

## **Conclusiones**

El proceso Unificado es un proceso de desarrollo de software; es el conjunto de actividades necesarias para transformar los requisitos de los usuarios en un sistema de software. Para ello, el proceso unificado utiliza todas las ventajas de la utilización del lenguaje de modelado unificado y va mas allá del análisis y diseño que proporciona una familia de técnicas que soportan el ciclo de desarrollo de software. El resultado es un proceso controlado en su totalidad y basado en componentes dirigidos por los casos de uso que representan los requisitos funcionales y nos ayudan a delimitar el sistema; es centrado en la arquitectura ya que proporciona una estructura sobre la cual guiar las interacciones, y su desarrollo es interactivo e incremental de tal modo que se atenúen los riesgos y se planteen las cuestiones en el instante en que se está capacitado para resolverlas y permite a los desarrolladores construir herramientas que soporten la automatización del proceso entero y la integración del trabajo a lo largo del desarrollo de un sistema de software



## Modelo de Análisis

El propósito fundamental del análisis es analizar los requisitos con una mayor profundidad, es decir, ofrece una especificación mas precisa de los requisitos. En consecuencia, en el análisis podemos razonar mas sobre los aspectos internos del sistema.

Durante el análisis, analizamos los requisitos refinándolos (utilizar un lenguaje mas formal para apuntar a detalles relativos) y estructurándolos (basada en clases y paquetes). Podemos estructurar los requisitos de manera que nos facilite su comprensión, su preparación, su modificación, y en general, su mantenimiento. El objetivo es conseguir una comprensión mas precisa de los requisitos y una descripción de los mismos de forma que sea mas fácil de mantener y que nos ayude a estructurar el sistema entero, incluyendo su arquitectura.

Existe una trazabilidad directa entre esas distintas estructuras, de forma que podemos hacer la trazabilidad de diferentes opciones (en diferentes niveles de detalles) del mismo requisito. Esta trazabilidad se define entre casos de uso y realizaciones de casos de uso en el modelo de análisis.

El lenguaje de análisis se basa en un modelo de objetos conceptuales, que llamamos modelo de análisis, el mismo se describe utilizando un lenguaje de los desarrolladores, y puede por lo tanto, introducir un mayor formalismo y ser utilizado para razonar sobre los funcionamientos internos del sistema.

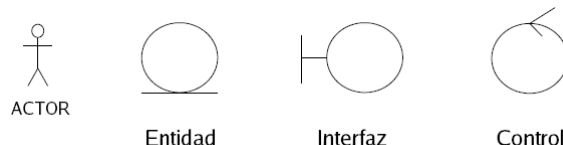
El modelo de análisis nos ofrece un mayor poder expresivo y una mayor formalización; también nos ayuda a estructurar los requisitos y nos proporciona una estructura centrada en el mantenimiento, en aspectos tales como flexibilidad ante los cambios y la reutilización de componentes. Se trata de preservar esta estructura a medida que se va dando forma al sistema y se toman decisiones sobre su diseño e implementación.

El modelo de análisis es una primera aproximación al modelo de diseño, ya que podrá ser mantenible a los cambios de los requisitos e incluirá elementos que podrán ser reutilizables. Cada elemento del modelo de análisis es trazable a partir de elementos del modelo de diseño que lo realizan. También se utiliza para crear un sistema robusto (capacidad de un sistema de adaptarse al cambio) y flexible que emplea la reutilización de manera considerable. El modelo de análisis es diferente de el modelo de diseño en que es un modelo conceptual, en lugar de ser un esquema de implementación. Su utilidad radica en que permite una apreciación global conceptual del sistema y puede contener las clases y paquetes de análisis, las realizaciones de los casos de uso, las relaciones y los diagramas.

### ***Creación del Modelo de Análisis a partir de los Casos de Uso***

El modelo crece incrementalmente a medida que se analizan mas y mas casos de uso. En cada interacción, se elige un conjunto de casos de uso y se reflejan en el modelo de análisis.

El modelo de análisis se representa por medio del diagrama de clases el cual esta compuesto por clases e interfaces y las reglas de interacciones entre las mismas.











La clase interfaz se utiliza para modelar la interacción entre el sistema y sus actores, como entrada de información permite que el actor proporcione datos (cajas de texto, menús desplegables, etc.) o solicite servicios (pulsando botones, etc.); como salida de información presenta datos al actor (listados, texto, etc.). Se puede conectar con un actor o una clase de control.



La clase de control representa coordinación, secuenciamiento; crear, modificar y eliminar información, realizar procesos y cálculos de otros objetos, permiten buscar información concreta de una clase conociendo alguno de los valores de sus atributos. No pueden conectarse directamente con los actores.

La clase entidad se utiliza para modelar información que tiene vida larga, sólo se deben conectar con las clases de control.

### Reglas de Robustez

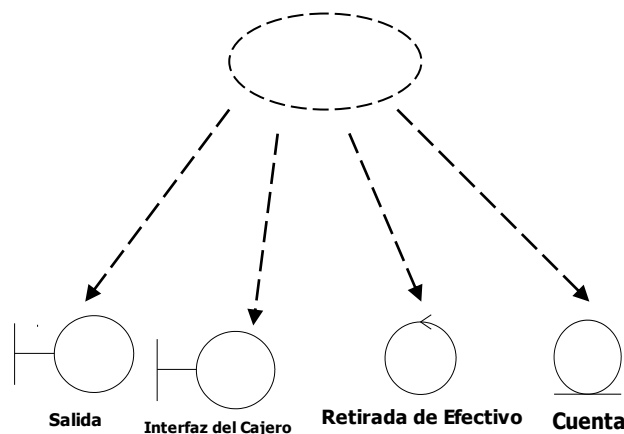
				
ACTOR	ACTOR			
NO	NO	SI	NO	
		NO	NO	SI
		SI	NO	SI
		NO	SI	SI

La presente figura muestra el tipo de conexiones entre clases, interfaces y actores explicadas en el párrafo anterior.

## La Realización de un Caso de Uso en el Modelo de Análisis

A continuación se describe un ejemplo para el caso de uso Sacar Dinero mediante un diagrama de colaboración que es un diagrama de interacción que enfatiza la organización estructural de los objetos que envían y reciben mensajes, es decir, que muestran las interacciones organizadas alrededor de instancias y de enlaces entre ellas.

### Modelo de Análisis



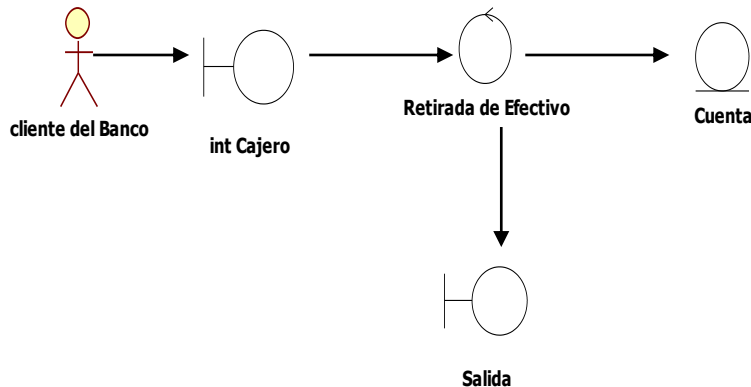
**Diagrama de Clases de Análisis**

Las clases de análisis que participan en la realización sacar Dinero. Salida e Interfaz del Cajero son clases de interfaz, Retirada de Efectivo es una clase de control y Cuenta es una clase entidad.

La notación para la realización de caso de uso o colaboración es una elipse con una línea de trazo fino discontinuo.

La estructura del sistema, se modela en un diagrama de clases de análisis que es un tipo de diagrama estático que describe la estructura de un sistema. Los diagramas de clases de análisis van a tener como elementos conceptos significativos de una aplicación, incluyendo conceptos del mundo real, conceptos abstractos y conceptos de implementación. Los diagramas de clase se utilizan para mostrar las clases y sus relaciones, pero también pueden utilizarse para mostrar subsistemas e interfaces.

### Diagrama de clases de Análisis



La realización del caso de uso Retirada de efectivo implica a la clase Interfaz de cajero y a la clase de entidad Cuenta. La ejecución del caso de uso comienza con un objeto de la interfaz del cajero. Después el trabajo continúa en el objeto de control que coordina la mayor parte del caso de uso en cuestión. El objeto Retirada de Efectivo le pide a la clase salida que entregue el dinero.

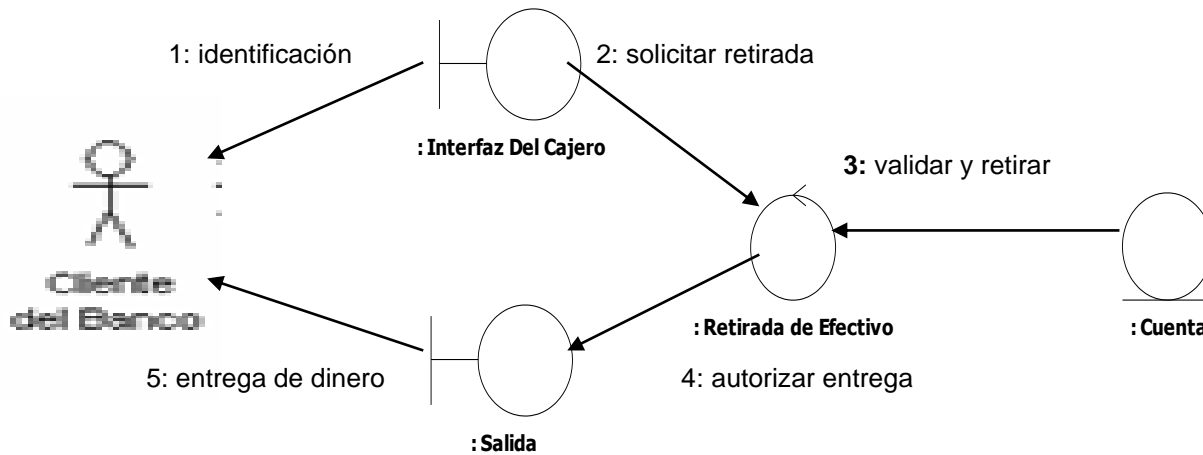
El siguiente esquema es un diagrama de colaboración que se utiliza para describir como una sociedad de objetos de análisis lleva a cabo la realización del caso de uso Sacar Dinero. El diagrama de colaboración es un diagrama de interacción que enfoca las interacciones y los enlaces entre un grupo de objetos colaboradores. Este diagrama se enfoca en el espacio y muestra como los objetos, sus enlaces y los mensajes son enviados entre ellos.

El diagrama muestra como el control pasa de un objeto a otro a medida que se lleva a cabo el caso de uso, y los mensajes que se envían entre los objetos. Un mensaje de un objeto dispara al objeto receptor para que tome el control y lleve a cabo una de las responsabilidades de su clase.

El nombre del mensaje indica el motivo por el cual el objeto realiza la llamada en su iteración con el objeto invocado.

Como complemento del diagrama de colaboración, los desarrolladores pueden usar también texto para explicar como interactúan los objetos para llevar a cabo el flujo de eventos del caso de uso.

### Diagrama de Colaboración



### Descripción del Flujo de Sucesos de una realización de Casos de Uso

Describimos a continuación la realización del caso de uso Sacar Dinero en términos de objetos y actores y sus iteraciones.

1. Un cliente del banco decide sacar dinero y activa el objeto interfaz del cajero. El cliente del banco se identifica y especifica la cantidad a retirar y la cuenta del cual se va a sustraer el dinero.
2. La interfaz del cajero verifica la identidad del cliente y solicita al objeto Retirada de efectivo que lleve a cabo la transacción.
3. Si la identidad del cliente es valida, se solicita al objeto Retirada efectivo que confirme si el cliente tiene derecho a sacar la cantidad especificada de la Cuenta. El objeto retirada de Efectivo lo confirma solicitando al objeto cuenta que valide la petición y si la petición es valida, que reste la cantidad.
4. Después el objeto retirada de efectivo autoriza la salida a que entregue al cliente la cantidad solicitada.
5. Entonces es cuando el cliente recibe la cantidad de dinero solicitada.

Se analizo cada caso de uso y se identificaron todos los roles (es decir, el comportamiento específico de una entidad que participa en un contexto en particular) que participan en cada realización de casos de uso. Las responsabilidades de una clase son la recopilación de todos los roles que cumple en todas las realizaciones de casos de uso. Juntándolas y eliminando repeticiones entre roles, se obtiene una especificación de todas las responsabilidades y atributos de la clase.

## **Breve comparación del Modelo de Análisis de Casos de Uso con el Modelo de Análisis**

<b>Modelo de Casos de Uso</b>	<b>Modelo de análisis</b>
Descrito con el lenguaje del cliente.	Descrito con el lenguaje del desarrollador.
Vista externa del sistema.	Vista interna del sistema.
Estructurado por los casos de uso; proporciona la estructura a la vista externa.	Estructurado por clases y paquetes ;proporciona la estructura a la vista interna del sistema
Utilizado fundamentalmente como contrato entre el cliente y los desarrolladores sobre que debería y que no debería hacer el sistema.	Utilizado fundamentalmente por los desarrolladores para comprender como debería darse forma al sistema, es decir, como debería ser diseñado e implementado.
Puede contener redundancias, inconsistencias, etc. entre requisitos.	No debería contener redundancias, inconsistencias, etc. entre requisitos.
Captura la funcionalidad del sistema, incluida la funcionalidad significativa para la arquitectura	Esboza como llevar a cabo la funcionalidad dentro del sistema, incluida la funcionalidad significativa para la arquitectura; sirve como una primera aproximación al diseño.
Define casos de uso que se analizaran con profundidad en el modelo de análisis.	Define realizaciones de casos de uso, y cada una representa el análisis de un caso de uso del modelo de casos de uso.

## **Conclusiones**

Este modelo es usado para representar la estructura global del sistema, describe la realización de casos de uso y sirve como una abstracción del Modelo de Diseño. El modelo de análisis nos permite especificar la estructura o el comportamiento de un sistema. Este modelo es un primer intento por definir los conceptos claves que describen el sistema. Para representar el Modelo de Análisis se pueden emplear los diagramas de clase de análisis y diagramas de colaboración.

El modelo de análisis debe representar las relaciones de manera tal que puedan identificarse las operaciones y que pueda desarrollarse un buen diseño de intercambio de mensajes.

Cada uno de los elementos del modelo de análisis proporciona la información necesaria para crear modelos de diseño que se requieren para una especificación completa de diseño. El diseño deberá implementar todos los requisitos explícitos del modelo de análisis.

## Bibliografía Consultada:

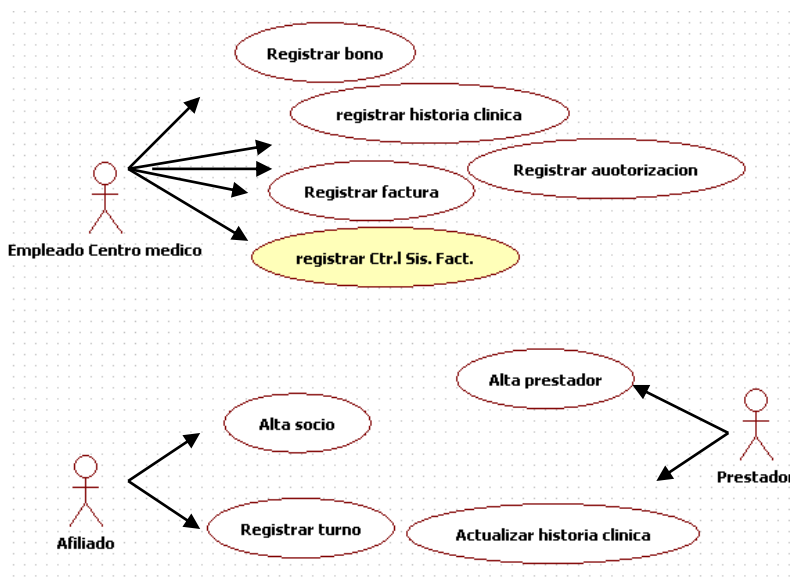
- Análisis y Diseño de Sistemas, Hernando Castillo García Factoría de Software.
- Lenguaje de Modelación Unificado y Proceso Unificado Rational,
- Ejemplo De Uso de Patrones de Análisis, Guillermo Pantaleo ITM.
- Fundamentos de Modelado OO en UML, Patricio Letelier T, Universidad de Valencia, España.
- Unified Modeling Language, Francisco Escamilla, Ingeniería de Software, UML.
- Use Case Driven Object Modeling with UML, Doug Rosenberg Ingeniería del Software.
- Modelamiento Visual y UML, Emilio Palacios Universidad Peruana de Ciencias Aplicadas.
- Desarrollo de Software Utilizando la Metodología RUP, Juan Juliá Igual Universidad Politécnica de Valencia. España.
- Análisis de Sistemas Software, Mischia, Adrián Ingeniería de Software.
- Construcción del Sistema de Información, Ministerio de Administraciones Públicas.
- Ingeniería del Software Un Enfoque Práctico, Roger S. Pressman.
- UML y patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado. Pearson Educación Madrid.
- The Unified Software Development Process, Jacobson Ivar Booch Grady Rumbaugh James.
- Casos de Uso Requerimientos UML, Santiago Ceria.
- Análisis con Casos de Uso. Diez sugerencias prácticas Santiago Ceria, Pragma Consultores.
- Análisis y Diseño Orientado a Objetos, UNLAM – Construcción de Sistemas.
- Análisis y Diseño de Sistemas, Gilber Basilio Robles.
- Site [www.Wikipedia.org.ar](http://www.Wikipedia.org.ar).

En el presente trabajo se detalla el diagrama de casos de uso, trazo grueso, trazo fino y el Modelo de Análisis para el caso de un sistema de control de facturación, así como los límites, las interacciones entre clases interfaz, control y entidad. La descripción del modelo de análisis sirve como una primera entrada al diseño del sistema y este modelo se realiza en lenguaje de los desarrolladores

## Enunciado del problema

La obra social “Susalud” ha decidido implementar un nuevo sistema de control de facturación debido a que su método actual no les brinda el nivel de control que el directorio requiere. Susalud contrata prestadores de servicios médicos, farmacéuticos y odontólogos. El afiliado concurre con su carnet y un bono, que debió comprar previamente en Susalud, al prestador que, según el diagnostico, recetara medicamentos, estudios, análisis, etc. Cada quince días los prestadores envían las facturas correspondientes a sus servicios a Susalud junto con los bonos que el afiliado debió dejar al momento de ser atendido. El afiliado debe solicitar autorización en Susalud antes de realizar un estudio, un análisis y en el caso de compra de medicamentos de alta especialidad. Las facturas se controlan contra los contratos firmados con los prestadores, los bonos y las historias clínicas de los afiliados, en estas últimas figuran los diagnósticos, las prácticas que se realizó y las autorizaciones. La historia clínica se actualiza con cada práctica recibida y con cada autorización solicitada.

## Diagrama de Casos de Uso



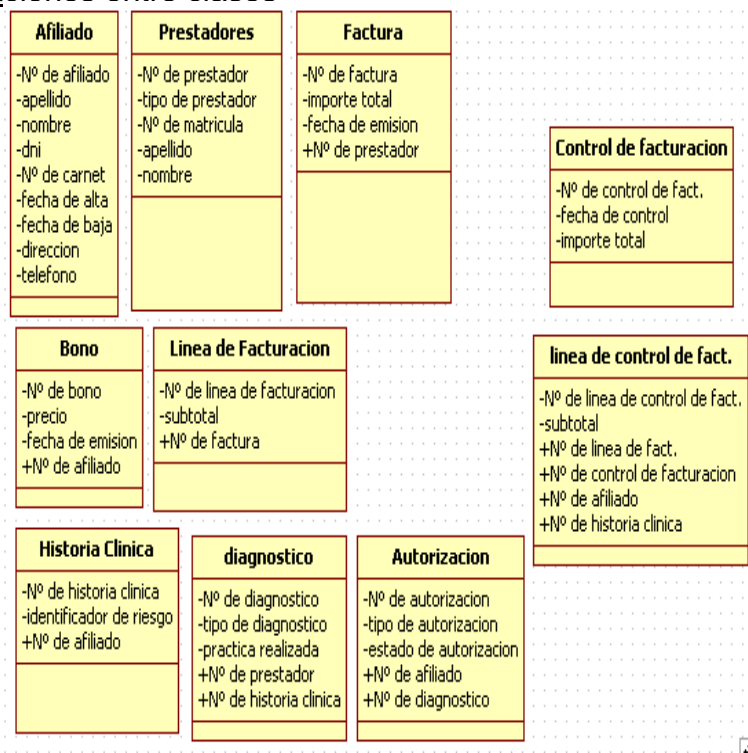
## Descripción del Caso de Uso:

<b>Caso de uso:</b> Registrar control de Sistema de Facturación.	
<b>Actor:</b> el empleado del centro medico.	
<b>Precondición:</b> debe haberse ejecutado los casos de uso registrar factura y registrar bono.	
<b>Trazo grueso:</b> El empleado del centro medico ingresa el número de factura y confirma los datos del prestador, la historia clínica, las autorizaciones y los bonos del afiliado; el empleado del centro medico registra el control de la factura y el sistema emite un número de control de facturación.	
<b>Trazo fino</b>	
<b>Flujo normal de los eventos</b>	
1-El empleado del centro medico selecciona "Realizar control del Sistema de Facturación".	
2-El sistema muestra la interfaz correspondiente con la fecha.	
3-El empleado del centro medico ingresa el numero de factura.	
4-El sistema muestra el número del prestador y carga los números de afiliados, la historia clínica, las autorizaciones y los bonos.	4.1-El número ingresado es erróneo y se vuelve al punto tres.
5-Por cada afiliado: 5.1-El sistema solicita seleccionar un numero de afiliado. 5.2-El empleado del centro medico selecciona un numero de afiliado y muestra el numero de afiliado, la fecha de alta y de baja, el apellido y nombre, el dni, la dirección, el teléfono y el numero de carnet. 5.3-El empleado del centro medico selecciona la pestaña de historia clínica y el sistema muestra el numero de historia clínica, el tipo de diagnostico, el estudio realizado y las autorizaciones. 5.4-El empleado del centro medico selecciona la pestaña bonos y el sistema muestra el número de bono, el precio y fecha de emisión. FIN PARA	
6-El empleado del centro medico selecciona confirmar.	6.1-En cualquier momento el empleado del centro medico detecta una irregularidad y selecciona denunciar.
7-El sistema registra el control de facturación y emite un número de control de facturación.	
<b>Poscondición:</b> El sistema registra el control del sistema de facturación y emite un número de control de facturación.	

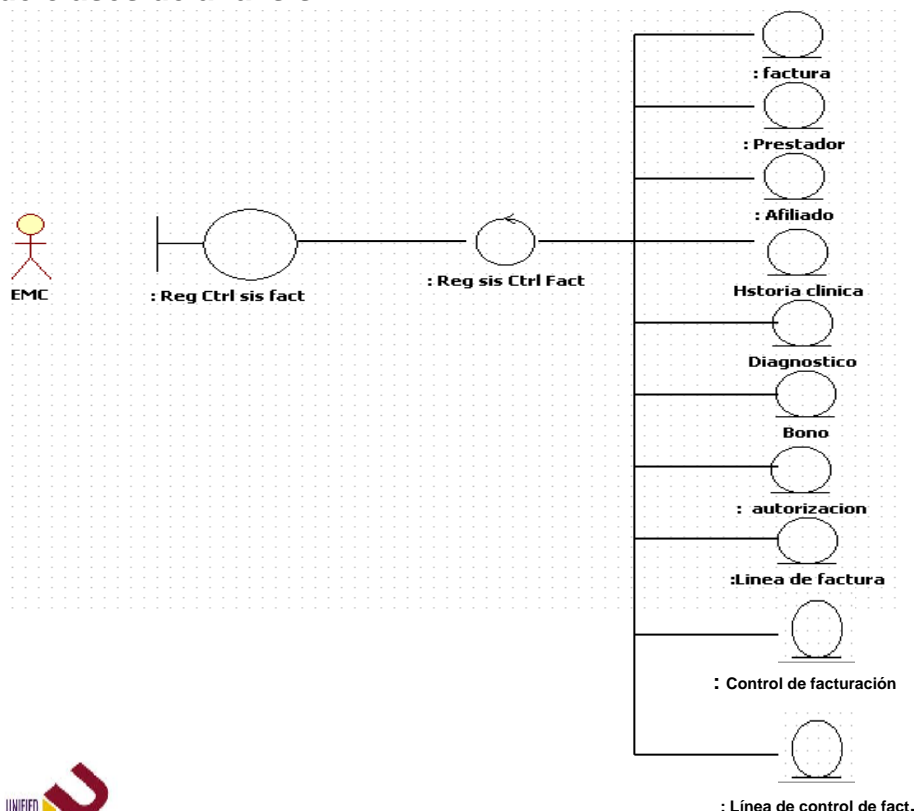


## Modelo de análisis Registrar Control de Sistema de Facturación

### Bosquejo de relaciones entre clases



### Diagrama de clases de análisis



## Prototipo de interfaz asociado

Sistema de Control de Facturación

Nº Factura  Fecha

Prestadores **Afiliado** Historia Clínica Bonos

Nº Prestador  Tipo de Prestador

Nº Matrícula

Apellido

Nº Control de Facturación

Sistema de Control de Facturación

Nº Factura  Fecha

Prestadores **Afiliado** Historia Clínica Bonos

*Datos del Afiliado*

Nº Afiliado  Fecha Alta  Fecha Baja

Apellido  Nombre  D.N.I.

Dirección  Teléfono  Nº Curnet

Nº Control de Facturación

Sistema de Control de Facturación

Nº Factura  Fecha

Pestadores Afiliado **Historia Clínica** Bonos

*Datos de la Historia Clínica*

Nº de Historia

Tipo de Diagnóstico	Práctica Realizada	Autorización
<input type="text"/>		

Nº Control de Facturación

Confirmar Denunciar

Sistema de Control de Facturación

Nº Factura  Fecha

Pestadores Afiliado Historia Clínica **Bonos**

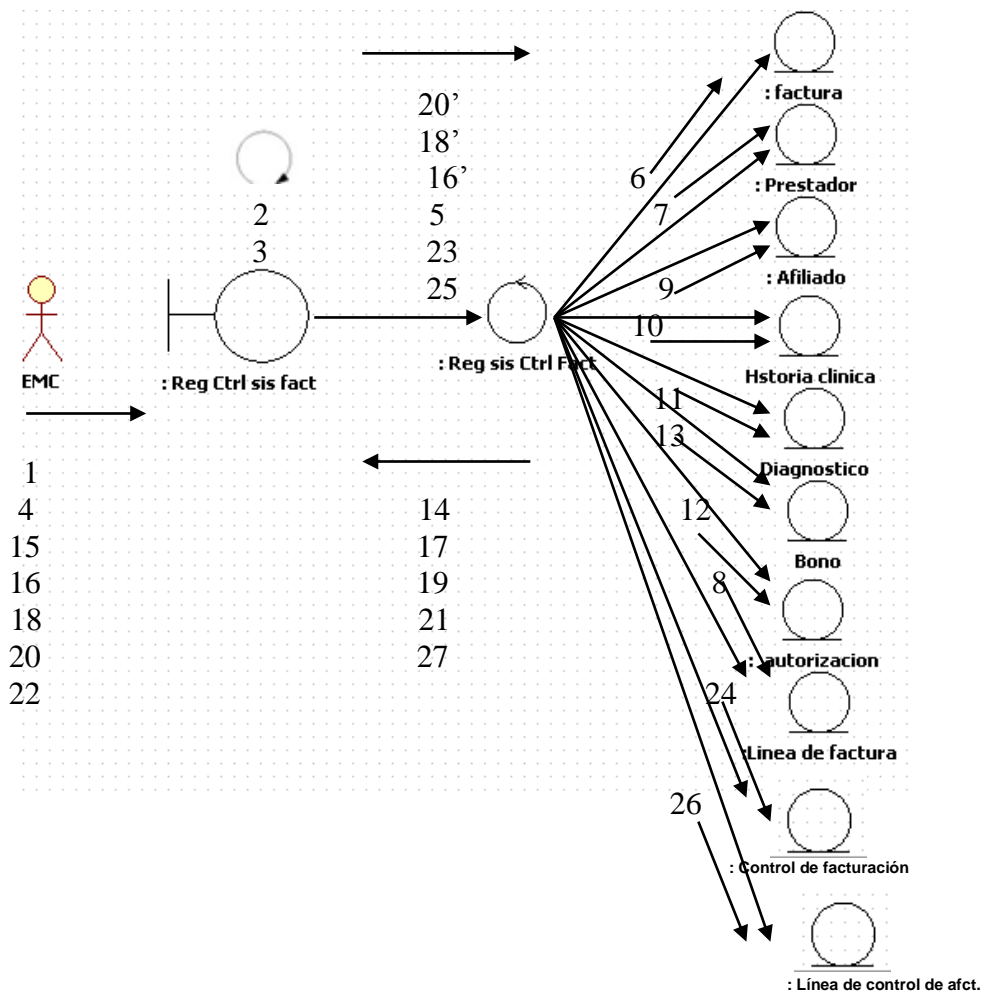
*Datos de los Bonos*

Nº Bono	Precio	Fecha de Emisión
<input type="text"/>		

Nº Control de Facturación

Confirmar Denunciar

## Diagrama de Colaboración de Análisis



1. Seleccionar realizar control de facturación.
2. Obtener la fecha.
3. Mostrar interfaz.
4. Ingresar el número de factura.
5. Buscar los datos de la factura, buscar los datos del prestador, los datos de la línea de factura, los datos del afiliado, datos de la historia clínica, el diagnostico, las autorizaciones y los bonos.
6. Obtener los datos de la factura.
7. Obtener datos del prestador.
8. Obtener los datos de la línea de facturación.
9. Obtener los datos del afiliado.
10. Obtener datos de la historia clínica.
11. Obtener datos del diagnóstico.
12. Obtener datos de la autorización.
13. Obtener datos de los bonos.
14. Mostrar numero de prestador, tipo de prestador, numero de matricula y apellido.
15. Seleccionar la pestaña de Afiliados.
16. Seleccionar número de afiliado.
- 16' Enviar solicitud del numero de cliente.
17. Mostrar numero de afiliado, fecha de alta, fecha de baja, apellido, nombre, dni, dirección, teléfono y numero de carnet.

18. Seleccionar la pestaña de historia clínica.
- 18' Enviar solicitud de datos de historia clínica.
19. Mostrar numero de historia clínica, tipo de diagnostico, práctica realizada y las autorizaciones.
20. Seleccionar la pestaña de bonos.
- 20' enviar solicitud de datos de bonos.
21. Mostrar número de bono, precio y fecha de emisión.
22. Seleccionar confirmar control de sistema de facturación.
23. Registrar control de facturación.
24. Registrar control de facturación.
25. registrar línea de control de facturación.
26. Registrar línea de control de facturación.
27. Mostrar número de control de facturación.