

# REQUERIMIENTOS

## DE SOFTWARE POR CASO DE USO

## ESTRUCTURA DE CONTENIDOS

	Pág.
Introducción.....	3
Mapa de contenido.....	4
1. Generalidades.....	5
1.1. Introducción al UML.....	5
1.2. Casos de uso.....	6
1.3. Notación de casos de uso.....	7
1.4. Empleo de los casos de uso.....	10
2. Documentación de casos de uso.....	11
Glosario.....	14
Bibliografía.....	16
Control del documento.....	17

## INTRODUCCIÓN

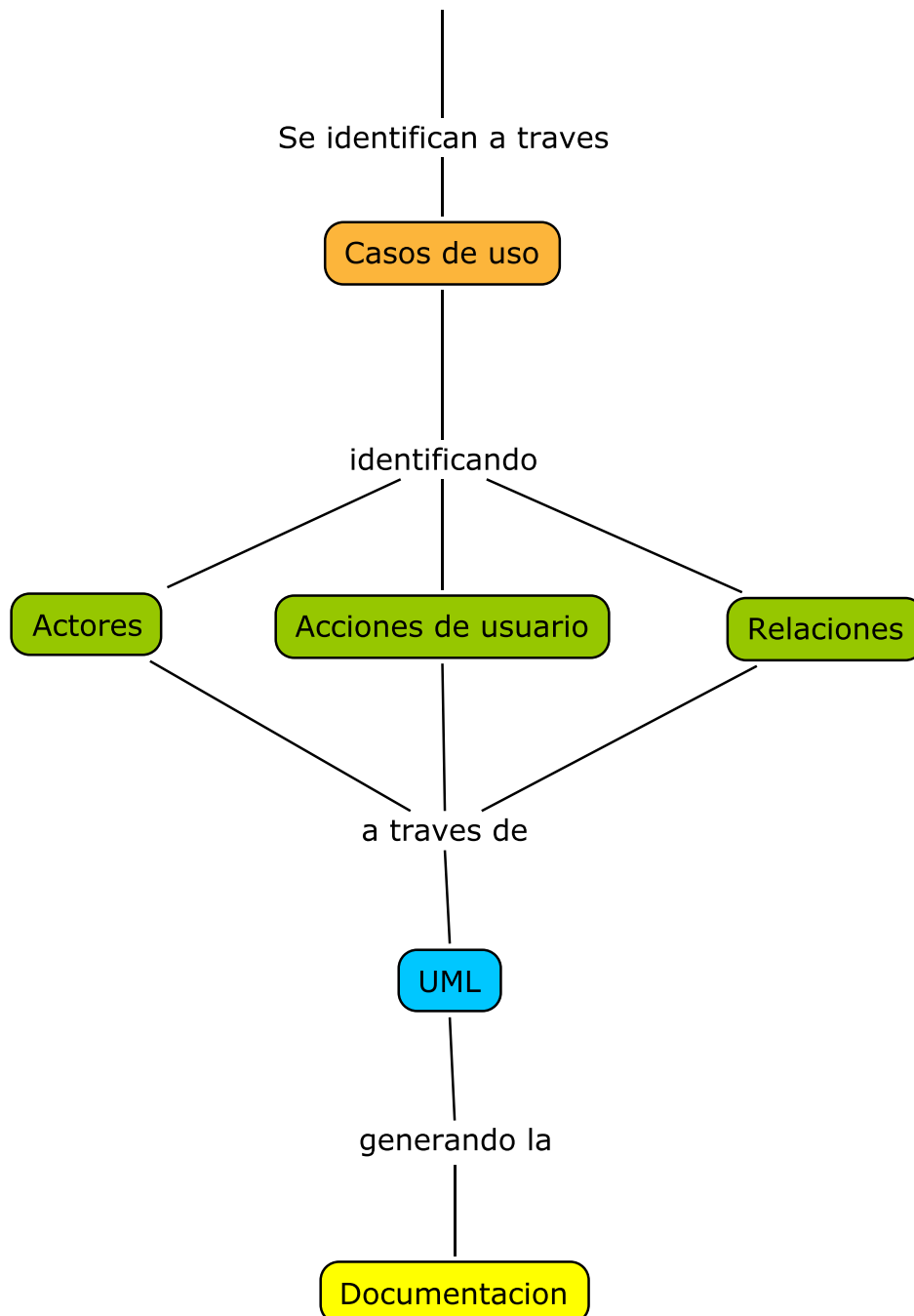
Una de las principales actividades para los analistas es determinar los requerimientos funcionales de software. Para ello existe distintos modelos que permiten identificar y documentar los requerimientos funcionales, uno de ellos y de mayor utilización en el mundo son los casos de uso en **UML (Lenguaje Unificado de Modelado)**.



Un caso de uso es un requerimiento funcional del software a desarrollar, como por ejemplo para “crear usuario”. Para que el analista pueda modelar de manera más sencilla el UML (Lenguaje Unificado de Modelado) permite a través de diagramas identificar, analizar y dar una mayor comprensión de los requerimientos funcionales que contendrá el software a diseñar. El presente material de estudio describe una introducción al UML y a los casos de uso, al igual que la utilización de una herramienta software para su elaboración y documentación.

## MAPA DE CONTENIDO

# REQUERIMIENTOS DE SOFTWARE POR CASO DE USO



## DESARROLLO DE CONTENIDOS

### 1. Generalidades

#### 1.1. Introducción al UML

El lenguaje unificado de modelado (UML) nació de la unión de los métodos análisis y diseño orientado a objetos a finales de la década de 1980 principios de la década de 1990. Unifica los métodos de los siguientes autores Booch, Rumbaugh y Jacobson. ***El UML es un lenguaje de modelado y no corresponde a un método de desarrollo de software o una técnica para el desarrollo de software. Es totalmente gráfico y en él se expresan los diseños del software desde la identificación de requerimientos hasta el diseño de interfaces y clases.*** A través de él se pueden esquematizar la estructura y el comportamiento del software.

Dentro de la esquematización de la estructura se encuentran los **diagramas de casos de uso**. El proceso unificado de desarrollo de software obtenido mediante la unión de los tres autores (Booch, Jacobson, Rumbaugh), está centrado en los casos de uso, de ahí la importancia del lenguaje unificado de modelado que complementa, no sólo este modelo de proceso de desarrollo de software sino todos los modelos que quieran utilizarlo.

El UML también permite documentar todos los procesos de desarrollo del software, se comporta como un lenguaje común entre el cliente y el analista, donde a través de gráficos se representa lo que finalmente el cliente requiere.



**Figura 1.** Lenguaje unificado de modelado

En definitiva el UML es una herramienta para analistas funcionales (los que definen, lo que el programa debe hacer) y analistas programadores (los que construyen a través de programas esas funcionalidades), es un estándar en la industria del software aprobado por la ISO/IEC 19501:2005 Information Technology Open Distributed Processing Unified Modeling Language, que lo usan las empresas de medianos o grandes equipos de

analistas y desarrolladores cuyo objetivo es analizar, planificar y documentar todo un proyecto de software.

La última versión del UML se encuentra en la versión 2.5 en la **norma ISO/IEC 19501:2005**.

## 1.2. Casos de uso

Los casos de uso describen las funcionalidades de los sistemas desde la visión del usuario. Se utilizan para mostrar la funcionalidad que el sistema que se está diseñando y que ofrecerá y que usuarios se comunicaran con el sistema para usar dicha funcionalidad. Es una técnica que consiste en hacer un diagrama con su respectiva notación.

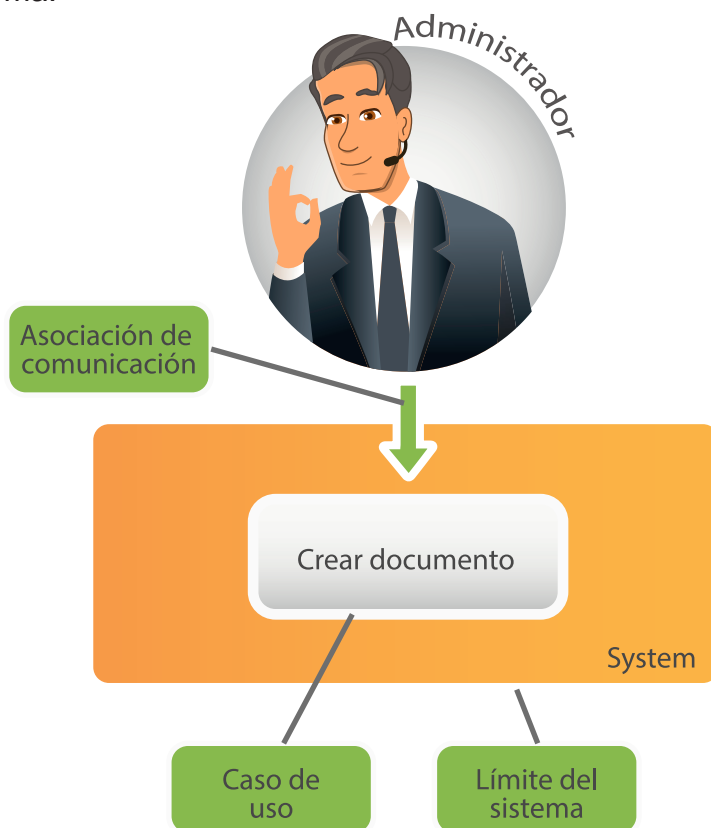


**Figura 2.** *Ejemplo de diagramas de casos de uso*

También se denomina como el modelo de requisitos y complementa los requerimientos de un sistema software a desarrollar. Los casos de uso describen la funcionalidad que ofrecerá el sistema desde la perspectiva del usuario definiendo el ámbito del sistema y el conocimiento por parte del diseñador de lo que el usuario requiere. La descripción del caso de uso ofrece una interacción entre el usuario y el sistema. Estas descripciones están en forma de resumen o en forma más detallada en donde esta interacción se describe paso a paso.

### 1.3. Notación de los casos de uso

De acuerdo con el estándar en la norma ISO/IEC 19501: 2005 los diagramas de caso de uso muestran tres aspectos del sistema: los actores, los casos de uso y los límites del sistema o subsistema.



**Figura 3.** Notación de los casos de uso

Los actores representan el rol que los usuarios, dispositivos u otros sistemas realizan cuando se comunican con el caso de uso particular del sistema.

Al momento de determinar los casos de uso se hace necesario identificar el actor. El actor es el nombre que se le da al usuario cuando desempeña ese papel. En figura anterior el actor es el Administrador, es un usuario que desempeña el rol de administrador. Los actores llevan a cabo los casos de uso. Un actor puede realizar varios casos de uso y un caso de uso puede ser realizado por varios actores.

Para identificar un actor solo es necesario visualizar los roles que interactuarán en el sistema.

Un ejemplo es el siguiente:

“Se requiere que el vendedor genere la factura de venta y la envíe a través del sistema externo quien procesara los datos de la misma...”

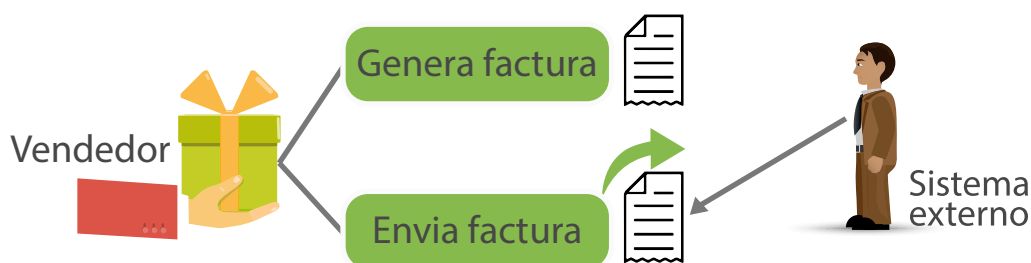
Del párrafo anterior se determinan los siguientes actores:



**Figura 4. Actores**

Los vínculos o el tipo de comunicación entre los actores y los casos de uso establecen la asociación directa con el caso de uso. Se dice el actor usa o ejecuta el caso de uso.

En el caso anterior el vendedor **genera** la factura y la **envía**. Los casos de uso se identifican a través de los verbos o acciones que el actor realiza o realizará, la acción identificada es genera y envía y se representa de la siguiente manera:



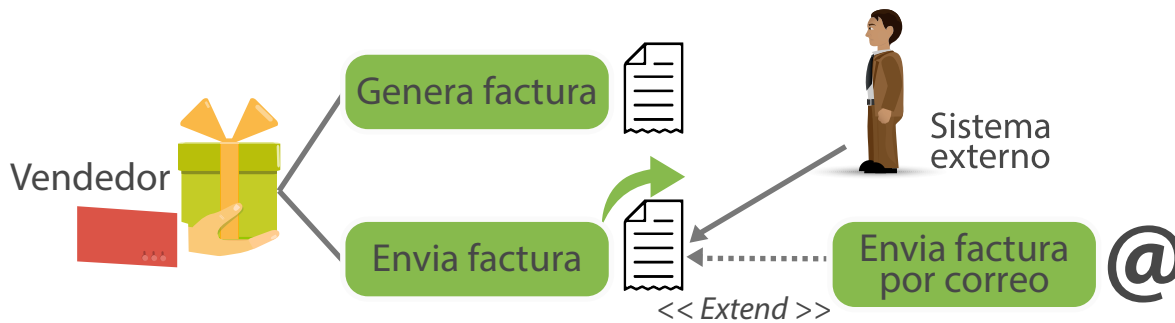
**Figura 5. Casos de uso**

**NOTA:** no necesariamente todos los actores son seres humanos, es decir también pueden corresponder un sistema externo que requiera información del sistema actual.

Otro tipo de relaciones entre casos de uso que se utilizan comúnmente son las relaciones uses y extends.



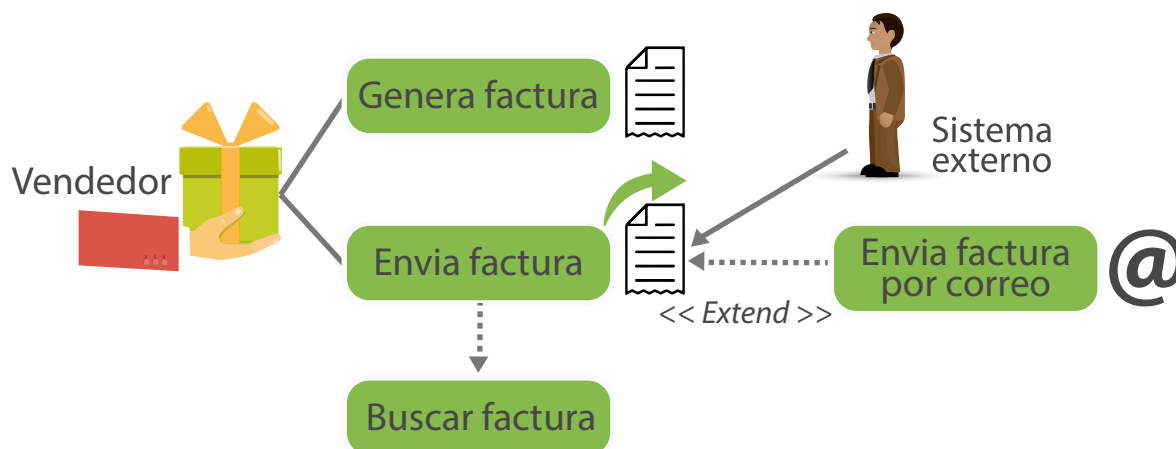
Se utiliza la relación extends cuando se tiene un caso de uso que es muy parecido a otro pero que hace un poco más. Por ejemplo en el caso anterior se puede ampliar el caso de uso envía factura por envía factura por correo electrónico. Ambos casos establecen la misma acción la diferencia es que el caso de uso además de enviar la factura al sistema externo también envía un correo electrónico con la factura y se representaría de la siguiente manera:



**Figura 6.** Utilización de extend

La anterior figura se lee así: el caso de uso enviar factura por correo extiende del caso enviar factura.

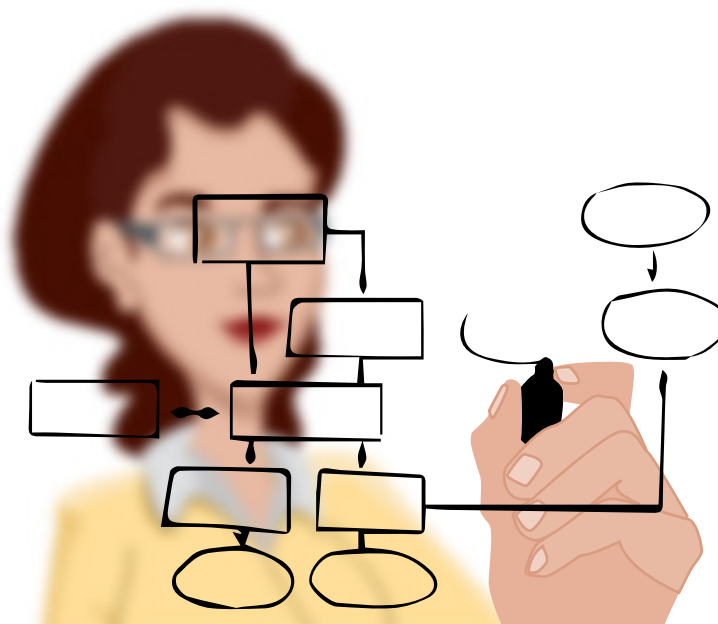
La relación **uses** o incluye se da cuando una porción del comportamiento o acción es similar en más de un caso de uso y no se requiere copiar la descripción de la conducta. En el caso anterior se amplía con el siguiente requerimiento funcional, antes de que el usuario envíe la factura puede realizar una búsqueda de facturas para seleccionarla y enviarla. Lo anterior se vería así:



**Figura 7.** Utilización de uses o include

La diferencia entre **include** o **uses** y **extends** es su intención. Ambos implican la factorización de comportamientos comunes de varios casos de uso dejando un solo caso de uso incluido o extendido.

#### 1.4. Empleo de los casos de uso



Los casos de uso son una herramienta esencial para la captura de requerimientos funcionales, planificación, control de proyectos. La identificación de los casos de uso es una de las tareas principales del analista de sistemas, de hecho es lo primero que debe realizarse.

La identificación de la mayoría de los casos de uso se realizaría en las etapas iniciales del proceso, sin embargo es común que a medida que avanza el proyecto se van identificando otros casos de uso. Todo caso de uso es un requerimiento potencial.

Se recomienda listar y analizar los casos de uso primero y luego realizar un poco de modelado, esto ayuda a descubrir el modelado conceptual junto con los usuarios.

Los analistas emplean casos de uso con diferentes números de casos de uso. El autor Ivar Jacobson, dice que en un proyecto de 10 personas-año el esperaría unos 20 casos de uso sin contarlas relaciones include y extend. Sin embargo en proyectos similares se tienen más de 100 casos de uso. Demasiados casos de uso pueden ser abrumadores pero se recomienda tener el menor número de casos de uso sin perder de vista el cronograma del proyecto.

## 2. Documentación de casos de uso

Es importante el analista a medida que va identificando los requerimientos vaya documentando los mismos. Para ello se recomienda utilizar la siguiente plantilla.

<b>Nombre del caso de uso</b>	
Actor	
Pre-condiciones	
Post-condiciones	
Flujo Básico	En ítem se profundiza más adelante.
Flujo Alternativo	En ítem se profundiza más adelante.

**Tabla 1.** *Plantilla para documentar casos de uso*

Los analistas se pueden apoyar en software para generar los casos de uso y documentarlos.

Para el desarrollo de este contenido se presenta el software **Star UML**, el cual incluye todos los diagramas de estructura y comportamiento del UML, adicionalmente en su última versión incluye la elaboración de diagramas de entidad relación y soporta la ingeniería inversa.

Para descargar la versión gratuita de este software el aprendiz debe ingresar a la página **web staruml.io** y en la parte superior derecha se selecciona la opción **V1**.

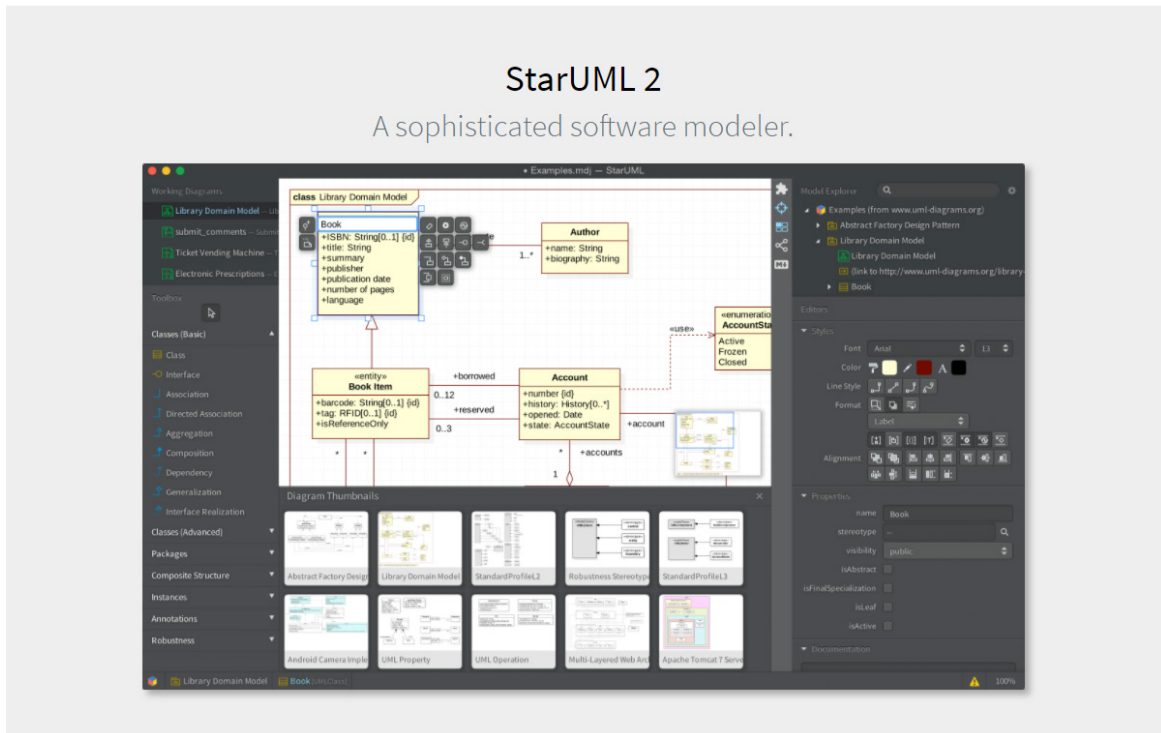


Figura 8. Página de descarga de Star UML

Una vez seleccionada la opción **V1**, aparecerá la siguiente página <http://staruml.sourceforge.net/v1/>. Al interior de la página ubique la opción de descarga correspondiente

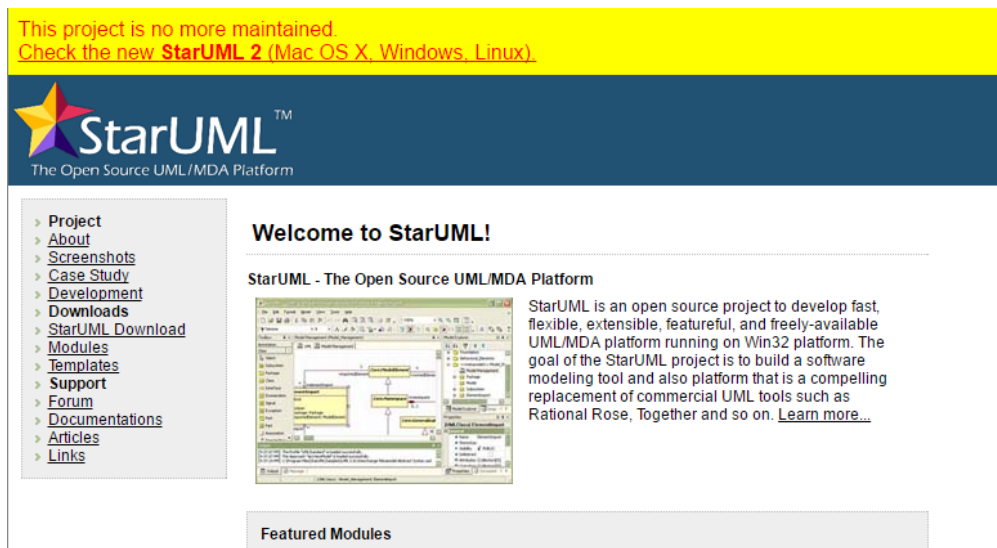


Figura 9. Página de descarga de Star UML

Esta aplicación puede ser usada en sistemas operativos Windows, Linux y Mac, sin embargo, la versión gratuita viene solo para Windows.

Para comprender un poco más la herramienta y generar el primer documento de requerimientos por casos de uso se propone estudiar el siguiente caso y ver los tutoriales propuestos.

Se retoma el caso del **vendedor viajero**.

“Se requería que el vendedor genera la factura de venta y la envíe a través del sistema externo. El vendedor puede realizar búsquedas de las facturas, adicionalmente puede consultar cuanto ha vendido y cuanto hace falta para completar la meta por último el vendedor listar los clientes de su ruta correspondiente en el viaje...”

## GLOSARIO

**ISO:** (International Organization for Standardization - Organización Internacional para la Estandarización). Su nombre ISO significa “igual” en griego. Fue fundada en el año 1946 y unifica a más de cien países. Se encarga de crear estándares o normas internacionales.

**Procedimiento:** forma específica de llevar a cabo una actividad. En muchos casos los procedimientos se expresan en documentos que contienen el objeto y el campo de aplicación de una actividad; que debe hacerse y quien debe hacerlo; cuando, donde y como se debe llevar a cabo; que materiales, equipos y documentos deben utilizarse; y como debe controlarse y registrarse.

**Proceso:** conjunto de recursos y actividades interrelacionados que transforman elementos de entrada en elementos de salida. Los recursos pueden incluir personal, finanzas, instalaciones, equipos, técnicas y métodos.

**Proyecto:** es una serie de actividades encaminadas a la consecución de un objetivo, con un principio y final claramente definidos. La diferencia fundamental con los procesos y procedimientos estriba en la no repetitividad de los proyectos.

**Requerimiento:** un requisito es una necesidad documentada sobre el contenido, forma o funcionalidad de un producto o servicio. Se usa en un sentido formal en la ingeniería de sistemas, ingeniería de software e ingeniería de requisitos.

**RUP:** el Proceso Unificado no es simplemente un proceso, sino un marco de trabajo extensible que puede ser adaptado a organizaciones o proyectos específicos. De la misma forma, el Proceso Unificado de Rational, también es un marco de trabajo extensible, por lo que muchas veces resulta imposible decir si un refinamiento particular del proceso ha sido derivado del Proceso Unificado o del RUP. Por dicho motivo, los dos nombres suelen utilizarse para referirse a un mismo concepto.

**Sistema:** estructura organizativa, procedimientos, procesos y recursos necesarios para implantar una gestión determinada, como por ejemplo la gestión de la calidad, la gestión del medio ambiente o la gestión de la prevención de riesgos laborales. Normalmente están basados en una norma de reconocimiento internacional que tiene como finalidad servir de herramienta de gestión en el aseguramiento de los procesos.

**Subprocesos:** son partes bien definidas en un proceso. Su identificación puede resultar útil para aislar los problemas que pueden presentarse y posibilitar diferentes tratamientos dentro de un mismo proceso.

**Sinergia:** comúnmente, refleja un fenómeno por el cual actúan en conjunto varios factores, o varias influencias, observándose así un efecto, además del que hubiera podido esperarse operando independientemente, dado por la concausalidad a los efectos en cada uno. En estas situaciones, se crea un efecto extra debido a la acción conjunta o solapada, que ninguno de los sistemas hubiera podido generar en caso de accionar aisladamente.

**UML:** el lenguaje unificado de modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un “plano” del sistema (modelo), incluyendo aspectos conceptuales tales como procesos, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados.

## BIBLIOGRAFÍA

Bennet, Simon., McRobb, Steve., Farmer, Ray. (2006). *Análisis y diseño orientado a objetos de sistemas usando UML*. McGrawHill: Madrid.

Fowler, Martin., Kendall, Scott. (1999). UML gota a gota. Addison Wesley: México.

International Organization for Standardization. (2017). Recuperado de <https://www.iso.org>



## CONTROL DEL DOCUMENTO

<p><b>CONSTRUCCIÓN OBJETO DE APRENDIZAJE</b></p> 	REQUERIMIENTOS DE SOFTWARE POR CASO DE USO	
	Centro Industrial de Mantenimiento Integral - CIMI Regional Santander	
	<b>Líder línea de producción:</b>	Santiago Lozada Garcés
	<b>Asesores pedagógicos:</b>	Rosa Elvia Quintero Guasca Claudia Milena Hernández Naranjo
	<b>Experto temático:</b>	Edward Jose Beltran Lozano
	<b>Diseño multimedia:</b>	Oscar Julian Marquez Sanabria
	<b>Programador:</b>	Francisco José Lizcano Reyes
	<b>Producción de audio:</b>	Víctor Hugo Tabares Carreño

Este material puede ser distribuido, copiado y exhibido por terceros si se muestra en los créditos. No se puede obtener ningún beneficio comercial y las obras derivadas tienen que estar bajo los mismos términos de la licencia que el trabajo original.



UML. Todos los derechos reservados.

StarUML. Todos los derechos reservados.

