

# Un Profile UML para el Modelo de Negocio

**Paola Martellotto**

Universidad Nacional de Río Cuarto  
Facultad de Ciencias Exactas, Fco-Qcas y Naturales - Dpto. de Computación  
Río Cuarto, Córdoba, Argentina, C.P. 5800  
paola@dc.exa.unrc.edu.ar

**Marcela E. Daniele**

Universidad Nacional de Río Cuarto  
Facultad de Ciencias Exactas, Fco-Qcas y Naturales - Dpto. de Computación  
Río Cuarto, Córdoba, Argentina, C.P. 5800  
marcela@dc.exa.unrc.edu.ar

## Abstract

This paper proposes a UML profile for the business modeled supported by RUP [2]. Through the extension mechanisms new stereotypes, tagged values and constraints are defined that allow to represent the artifacts of the business model, their relations and constraints. This UML Profile for the business modeled one is defined on the basis of the generic model of the business model proposed in previous works [6][7][8], which is a reconstruction well organized of the original model presented by its authors in RUP, that besides to allow to visualize and to specify of way complete and organized the artifacts that compose the business model and existing relations between such, it express of way needs the constraints associated to these artifacts. With this extension of UML, a language more specific is described than aid to the software engineers to model and to represent the concepts of the specific domain of the structure and the dynamics of a business.

**Keywords:** Business Modeling, Generic Model, UML Profile, Stereotypes, Constraints, Tagged Values.

## Resumen

Este trabajo propone un profile UML para el modelado de negocio soportado por RUP[2]. A través de los mecanismos de extensión de UML se definen nuevos estereotipos, restricciones y valores etiquetados que permiten representar los artefactos del modelo de negocio, sus relaciones y restricciones. Este profile UML se basa en el Modelo Genérico del Modelo de Negocio propuesto en trabajos anteriores [6] [7] [8], el cual es una reconstrucción del modelo original presentado por sus autores en RUP, que además de permitir visualizar y especificar de manera completa y organizada los artefactos que componen el modelo de negocio y las relaciones existentes entre los mismos, expresa de manera precisa las restricciones asociadas a dichos artefactos. Con esta extensión de UML para definir profiles, se describe un lenguaje más específico que ayuda a los ingenieros de software a modelar y representar los conceptos del dominio específico al modelado de la estructura y dinámica de un negocio.

**Palabras Claves:** Modelado de Negocio, Modelo Genérico, Profile UML, Estereotipos, Restricciones, Valores Etiquetados.

## 1 Introducción

El desarrollo de software nunca ha sido una tarea fácil y ha dado lugar a la creación de una gran cantidad de metodologías de desarrollo de software y técnicas de modelado, atendiendo la necesidad de simplificar dicha complejidad. En la actualidad, las metodologías, técnicas, y herramientas para el desarrollo de software que han despertado mayor interés, son principalmente aquellas que se basan en la descripción de un modelo de negocio completo y preciso, que sirva de base sólida para la obtención de un software de alta calidad, que permita aumentar la productividad, evitar tareas repetitivas, y por tanto, permitir a las organizaciones ser más competitivas.

El Proceso Unificado [1] es una metodología de desarrollo de software que define el modelo de negocio como una actividad necesaria y fundamental para establecer una clara abstracción de la organización. Sobre la base de la propuesta original de RUP (Rational Unified Process) [2], se construyó un Modelo Genérico del Modelo de Negocio[8] como una reconstrucción del modelo original, que permite visualizar con mayor claridad y especificar con mayor precisión los artefactos que componen el modelo de negocio y sus relaciones principales. El objetivo principal de este modelo genérico es facilitar al desarrollador de software el conocimiento, la comprensión y el desarrollo del modelado del negocio, a partir de reglas y relaciones definidas entre los artefactos que conforman el mismo, brindando la posibilidad de construir un modelo suficiente y correcto para el conocimiento que se requiere del contexto. El modelo genérico permite además, colaborar en el análisis y control de un modelo de negocio construido previamente.

Los perfiles UML constituyen el principal mecanismo que este lenguaje ofrece para extender su sintaxis y su semántica y adaptar los modelos UML a un dominio concreto de aplicación. Este trabajo propone un perfil UML para el Modelo Genérico del Modelo de Negocio anteriormente propuesto en [8], que es soportado por el Modelado de Negocio del RUP.

Existe una gran cantidad de perfiles UML definidos en la actualidad, algunos estandarizados por la OMG [10] o en proceso de serlo, y otros que están puestos a disposición para ser utilizados a pesar de no representar estándares. Algunos trabajos consultados pueden verse en [9], [14], [15] y [16].

Este trabajo está organizado por la sección 2 donde se introduce la noción de los mecanismos de extensibilidad y de perfiles UML. En la sección 3 se presenta la definición del Perfil UML del modelo de negocio a partir del Modelo Genérico del Modelo de Negocio. Y por último se exponen las conclusiones.

## 2 Mecanismos de extensibilidad de UML

UML es el lenguaje de modelado gráfico estándar [3] fácil de entender y de usar con un cierto nivel de formalismo, es de propósito general y está soportado por diversas herramientas. Puede ser aplicado en diferentes dominios usando sus poderosos mecanismos de extensibilidad, permitiendo a los usuarios extender la semántica de lenguaje para adaptarse a tales dominios específicos.

Los mecanismos de extensión de UML son: estereotipos, valores etiquetados y restricciones. Los estereotipos extienden el vocabulario de UML, permitiendo crear nuevos tipos de bloques

de construcción que se derivan de los existentes, pero son específicos a un problema. Los estereotipos están definidos por un nombre y pueden tener restricciones y valores etiquetados asociados. Las restricciones extienden la semántica de un bloque de construcción de UML, permitiendo añadir nuevas reglas o modificar las existentes. Las restricciones imponen condiciones sobre los elementos del metamodelo que han sido estereotipados. Pueden expresarse tanto en lenguaje natural como en algún lenguaje formal o semiformal. Los valores etiquetados extienden las propiedades de un bloque de construcción de UML, permitiendo crear nueva información en la especificación de ese elemento. Los valores etiquetados son definidos como un meta-atributo adicional de la metaclass del metamodelo extendido por el profile.

Un profile UML es un paquete estereotipado «profile» que utiliza estereotipos, valores etiquetados y restricciones, y extiende el metamodelo UML para ser usado en un dominio específico.

## **2.1 Construcción de un Profile UML**

Según la propuesta de Fuentes&Vallecito[13] los pasos necesarios para construir un profile UML:

1. Disponer de la definición del metamodelo del dominio de la aplicación que se pretende modelar con un profile UML.
2. Definir el profile. En el paquete «profile» definir un estereotipo por cada uno de los elementos del metamodelo que se desea incluir en el profile. Estos estereotipos tendrán el mismo nombre que los elementos del metamodelo, estableciéndose de esta forma una relación entre el metamodelo y el profile. Los elementos del metamodelo UML extendido son las clases, asociaciones, atributos, operaciones, paquetes, etc.
3. Definir como valores etiquetados de los elementos del profile a los atributos que aparecen en el metamodelo. Incluir la definición de sus tipos y sus posibles valores iniciales.
4. Definir las restricciones del profile a partir de las restricciones del dominio. Por ejemplo, las multiplicidades de las asociaciones que aparecen en el metamodelo del dominio, o las propias reglas de negocio de la aplicación deben traducirse en la definición de las correspondientes restricciones.

Una vez que se ha definido el profile UML, la forma de utilizarlo en una aplicación concreta se representa mediante una relación de dependencia estereotipada «apply», entre el paquete UML que contiene la aplicación y los paquetes que definen el profile UML.

## **3 Profile UML del Modelo Genérico del Modelo de Negocio**

### **3.1 Modelo Genérico del Modelo de Negocio**

El Modelo Genérico del Modelo de Negocio [8] es una reconstrucción del modelo original de RUP que simplifica la obtención de una solución al problema de modelado de negocio, ya que resume en un único diagrama de clases todos los artefactos que componen el modelo de negocio, y las relaciones y restricciones impuestas entre ellos permiten construir un modelo correcto, a partir de su verificación durante el proceso de construcción del modelo de negocio. Por otro lado, estas reglas también sirven para verificar la corrección de un modelo previamente construido.

La Figura 1 muestra el Modelo Genérico del Modelo de Negocio representado con un diagrama de clases de UML. La Tabla 1 indica la referencia a las siglas usadas para cada artefacto.

**Tabla 1. Referencias de los Artefactos del Modelo de Negocio**

Modelos	Documentos
⟨MCUN, Modelo de Casos de Uso de Negocio⟩	⟨GN, Glosario de Negocio⟩
⟨MAN, Modelo de Análisis de Negocio⟩	⟨VFO, Valoración del Fin de la Organización⟩
<b>Elementos de Modelos</b>	⟨VN, Visión del Negocio⟩
⟨CUN, Caso de Uso del Negocio⟩	⟨DAN, Documento Arquitectura del Negocio⟩
⟨AN, Actor del Negocio⟩	⟨ESN, Especificación Suplementaria del Negocio⟩
⟨RCUN, Realización de Caso de Uso de Negocio⟩	⟨RN, Reglas del Negocio⟩
⟨SN, Sistema de Negocio⟩	⟨ON, Objetivo del Negocio⟩
⟨EN, Entidad de Negocio⟩	
⟨WN, Worker de Negocio⟩	
⟨EvN, Evento de Negocio⟩	

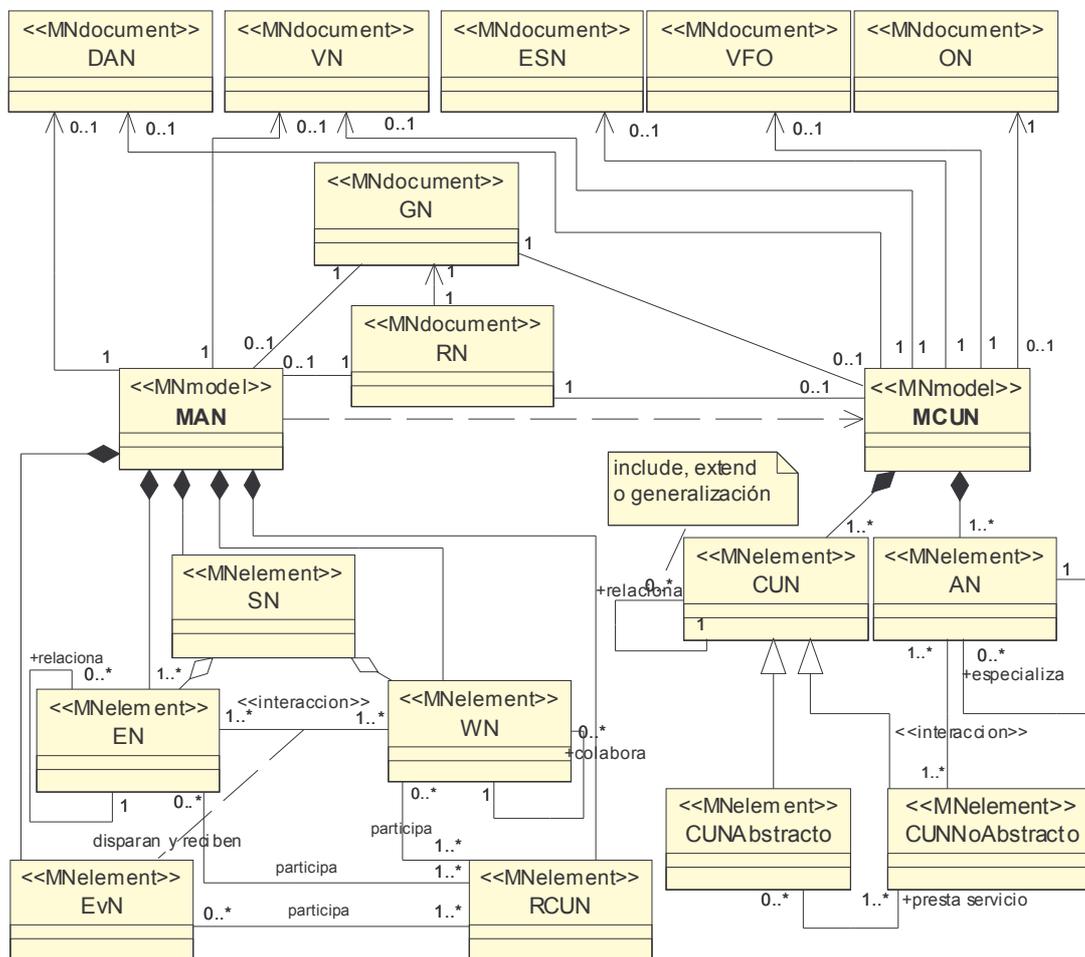


Figura 1 Modelo Genérico del Modelo de Negocio

### 3.2 Profile UML para el Modelo de Negocio

Para la construcción del profile UML del Modelo Genérico del Modelo de Negocio (Figura 2) se definen tres nuevos estereotipos básicos: MNmodel, MNelement, MNdocument, que representan los tipos de las clases del modelo genérico. Además, se define un estereotipo por cada clase del modelo conceptual de la Figura 1, con el mismo nombre de la clase original y

como una especialización de alguno de los estereotipos básicos.

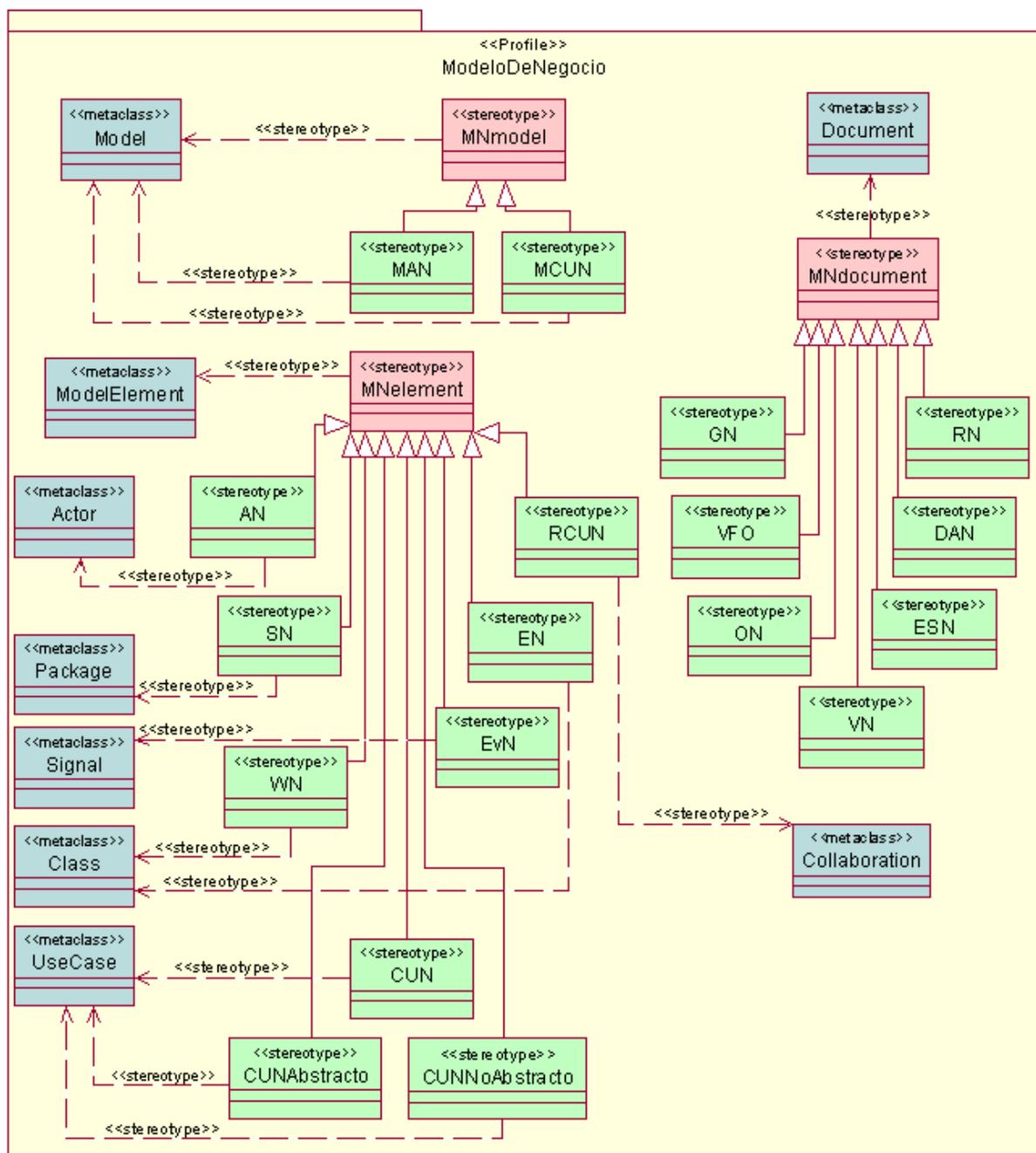


Figura 2 Profile UML del Modelo Genérico del Modelo de Negocio

**Tabla 3. Correspondencia entre la meta-clase de UML, clases del Modelo Genérico y estereotipos asociados**

Metaclase de UML	Clase del Modelo Genérico	Estereotipo	Metaclase de UML	Clase del Modelo Genérico	Estereotipo
Model	MAN	MNmodel	Class	EN	MNelement
Model	MCUN	MNmodel	Collaboration	RCUN	MNelement
UseCase	CUN	MNelement	Document	GN	MNdocument
UseCase	CUNAbstracto	MNelement	Document	VFO	MNdocument
UseCase	CUNNoAbstracto	MNelement	Document	ON	MNdocument
Actor	AN	MNelement	Document	VN	MNdocument
Package	SN	MNelement	Document	ESN	MNdocument

Class	WN	MNelement	Document	DAN	MNdocument
Signal	EvN	MNelement	Document	RN	MNdocument

### 3.2.1 Definición de los valores etiquetados de las clases del profile

Las clases pueden contener valores etiquetados y se especifican por un nombre y un valor. De las tres clases básicas MNmodel, MNelement y MNdocument heredan todas las demás clases que conforman el modelo genérico del modelo de negocio. El valor etiquetado para cada una de estas clases y heredado por todas las demás, es expresado como una tupla como se muestra en la Figura 3 y es descrita en detalle en la Tabla 4.

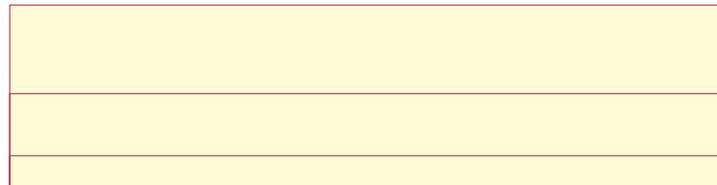


Figura 3 Valor etiquetado de las clases básicas MNmodel, MNelement, MNdocument

**Tabla 4. Descripción de la tupla que define el valor etiquetado**

<i>artefactos_rel</i>	<p><math>\langle s1[m,n], s2[m,n], \dots, sn[m,n] \rangle</math> siglas de los artefactos relacionados, su multiplicidad y navegabilidad.  <math>s_i</math> es la sigla que identifica un artefacto relacionado. Por cada <math>s_i</math> se puede indicar multiplicidad y navegabilidad con <b>m</b> y <b>n</b>.  <b>m</b> = es el intervalo que indica la multiplicidad.  <b>n</b> = si existe la navegabilidad en un sentido, <b>n</b> indica la sigla del artefacto hacia el cual se manifiesta dicha navegabilidad.</p>
<i>orden</i>	<p>Indica el orden de creación del artefacto respecto de los demás.  <b>1</b> = corresponden a esta categoría los artefactos cuya creación no depende de ningún otro. <b>2</b> = dependen de artefactos de orden 1. <b>3</b> = dependen de artefactos de orden 2. <b>4</b> = dependen de artefactos de orden 3</p>
<i>artefactos_dep</i>	<p><math>\langle s1, s2, \dots, sn \rangle</math> siglas de los artefactos de los cuales depende, <math>orden \neq 1</math>.</p>

### 3.2.2 Definición de estereotipos, valores etiquetados y restricciones de las clases del profile

En la siguiente descripción se expone la definición del estereotipo, restricciones y valor etiquetado de los principales artefactos del modelo genérico del modelo de negocio. Las principales restricciones son formalizadas en OCL.

#### Clase: MCUN

Estereotipo: <<MNmodel>>

Nombre de la clase: MCUN

#### Valor Etiquetado:

esp\_artefacto =  $\langle MCUN, \langle CUN[1..*], AN[1..*], MAN[0..1, MCUN], DAN[0..1], ESN[0..1, ESN], ON[0..1], VFO[0..1], VN[0..1], GN[1, GN], RN[1] \rangle, 2, \langle RN, GN, ON \rangle \rangle$

#### Restricciones:

- Los artefactos que componen el MCUN son CUN y AN.

context MCUN inv:

```
self.contents → forAll ( c |
    (c.oclIsKindOf(UseCase) and c.stereotype.name = "CUN")
    or (c.oclIsKindOf(Actor) and c.stereotype.name = "AN"))
```

- Un CUNNoAbstracto se relaciona al menos a un AN y puede relacionarse a otros CUN.

context CUNNoAbstracto inv:

```
self.associations → forAll ( a |
```

```
a.allConnections → forAll ( r |
  (r.type.ocllsKindOf(UseCase) and r.stereotype.name = "CUN" → size() ≥ 0)
  or (r.type.ocllsKindOf(Actor) and r.stereotype.name = "AN" → size() ≥ 1)))
```

- **Un CUNAbstracto debe estar relacionado al menos a un CUNNoAbstracto**

context CUNAbstracto inv:

```
self.associations → forAll ( c |
  c.allConnections → forAll ( r |
    r.type.ocllsKindOf(UseCase) and r.stereotype.name = "CUNNoAbstracto")) → size() ≥ 1
```

- **Un CUN no puede estar relacionado a sí mismo.**

context CUN inv:

```
self.relaciona → forAll ( c | c <> self )
```

- **Un CUNAbstracto no esta relacionado a ningún Actor.**

context CUNAbstracto inv:

```
self.associations → forAll ( a |
  a.allConnections → forAll ( r |
    r.type.ocllsKindOf(Actor) and r.stereotype.name = "AN")) → isEmpty
```

- **Un AN no puede estar relacionado a sí mismo.**

context AN inv:

```
self.especializa → forAll ( c | c <> self )
```

- **Un MCUN se relaciona exactamente a un ON, GN, RN, y puede estar relacionado o no a un DAN, VN, ESN, VFO**

context MCUN inv:

```
self.associations → forAll ( a |
  a.allConnections → forAll ( r |
    (r.type.ocllsKindOf(Document) and r.stereotype.name = "ON" → size() = 1)
    or (r.type.ocllsKindOf(Document) and r.stereotype.name = "GN" → size() = 1)
    or (r.type.ocllsKindOf(Document) and r.stereotype.name = "RN" → size() = 1)
    or (r.type.ocllsKindOf(Document) and r.stereotype.name = "DAN" → size() ≥ 0 and size() ≤ 1)
    or (r.type.ocllsKindOf(Document) and r.stereotype.name = "VN" → size() ≥ 0 and size() ≤ 1)
    or (r.type.ocllsKindOf(Document) and r.stereotype.name = "ESN" → size() ≥ 0 and size() ≤ 1)
    or (r.type.ocllsKindOf(Document) and r.stereotype.name = "VFO" → size() ≥ 0 and size() ≤ 1)))
```

## Clase: MAN

Estereotipo: <<MNmodel>>

Nombre de la clase: MAN

### Valor Etiquetado:

```
esp_artefacto=(MAN,<EN[1..*] SN[0..*], WN[0..*], RCUN[0..*], EvN[0..*], MCUN[1,MCUN],
DAN[0..1], VN[0..1] , GN[1,GN] ,RN[1]>,3, <MCUN>)
```

### Restricciones:

- **Los artefactos que componen el MAN son EN, SN, WN, EvN, y RCUN.**

context MAN inv:

```
self.contents → forAll ( c |
  (c.ocllsKindOf(ModelElement) and c.stereotype.name = "EN")
  or (c.ocllsKindOf(ModelElement) and c.stereotype.name = "SN")
  or (c.ocllsKindOf(ModelElement) and c.stereotype.name = "WN")
  or (c.ocllsKindOf(ModelElement) and c.stereotype.name = "EvN")
  or (c.ocllsKindOf(ModelElement) and c.stereotype.name = "RCUN"))
```

- **Cada EN debe participar en al menos una RCUN.**

context EN inv:

```
self.participa → exists ( r |
  r.ocllsKindOf(Collaboration) and r.stereotype.name = "RCUN" → size() ≥ 1 )
```

- **Cada WN debe participar en al menos una RCUN.**  
context WN inv:  
self.participa → exists ( r |  
r.oclIsKindOf(Class) and r.stereotype.name = “WN” → size() ≥ 1 )
- **Cada EvN debe participar en al menos una RCUN.**  
context EvN inv:  
self.participa → exists ( r |  
r.oclIsKindOf(Signal) and r.stereotype.name = “EvN” → size() ≥ 1 )
- **Un MAN se relaciona exactamente a un GN, RN, y puede estar relacionado o no a un DAN, VN**  
context MAN inv:  
self.associations → forAll ( a |  
a.allConnections → forAll ( r |  
(r.type.oclIsKindOf (Document) and r.stereotype.name = “GN” → size() = 1)  
or (r.type.oclIsKindOf (Document) and r.stereotype.name = “RN” → size() = 1)  
or (r.type.oclIsKindOf (Document) and r.stereotype.name = “DAN” → size() ≥ 0 and size() ≤ 1)  
or (r.type.oclIsKindOf (Document) and r.stereotype.name = “VN” → size() ≥ 0 and size() ≤ 1)))

### Clase: CUN

Estereotipo: <<MNelement>>

Nombre de la clase: CUN

**Valor Etiquetado:**

esp\_artefacto=(GN, <MCUN[0..1,GN], MAN[0..1,GN], RN[0..1,GN], DAN[0..1,GN]>,1)

### Clase: GN

Estereotipo: <<MNdocument>>

Nombre de la clase: GN

**Valor Etiquetado:**

esp\_artefacto=(GN, <MCUN[0..1,GN], MAN[0..1,GN], RN[0..1,GN], DAN[0..1,GN]>,1)

### Clase: RN

Estereotipo: <<MNdocument>>

Nombre de la clase: RN

**Valor Etiquetado:**

esp\_artefacto=(RN, <MCUN[0..1], MAN[0..1], GN[0..1,GN]>, 1)

### Clase: ON

Estereotipo: <<MNdocument>>

Nombre de la clase: ON

**Valor Etiquetado:**

esp\_artefacto=(ON, <MCUN[0..1], DAN[0..1], VN[0..1]>, 1)

### Clase: AN

Estereotipo: <<MNelement>>

Nombre de la clase: AN

**Valor Etiquetado:**

esp\_artefacto=(AN, <MCUN[1], CUN[1..\*]> ,4, <MCUN>)

### Clase: EN

Estereotipo: <<MNelement>>

Nombre de la clase: EN

**Valor Etiquetado:**

esp\_artefacto=(EN, <MAN[1],WN[1..\*], SN[0..1]> ,4, <MAN>)

**Clase: WN**

Estereotipo: <<MNelement>>

Nombre de la clase: WN

**Valor Etiquetado:**

esp\_artefacto=(WN, <MAN[1],EN[1..\*], SN[0..1]> ,4, <MAN>)

**Clase: EvN**

Estereotipo: <<MNelement>>

Nombre de la clase: EvN

**Valor Etiquetado:**

esp\_artefacto=(EvN, <MAN[1],RCUN[1..\*]> ,4, <MAN>)

**Clase: RCUN**

Estereotipo: <<MNelement>>

Nombre de la clase: RCUN

**Valor Etiquetado:**

esp\_artefacto=(RCUN, <MAN[1],EvN[1..\*]> ,4, <MAN>)

**Clase: SN**

Estereotipo: <<MNelement>>

Nombre de la clase: SN

**Valor Etiquetado:**

esp\_artefacto=(SN, <MAN[1],EN[1..\*], WN[0..\*]> ,4, <MAN>)

## 4 Conclusiones

En este trabajo se presentó una profile UML para el modelo de negocio soportado por RUP[2], en base al modelo genérico del modelo de negocio propuesto anteriormente en [8].

Siguiendo una metodología de desarrollo evolutiva, iterativa e incremental, las diferentes etapas que deben ejecutar los ingenieros de software para el desarrollo de un sistema, transitan en torno a las funcionalidades identificadas en la etapa de captura de requerimientos y son refinadas en cada iteración. Bajo esta premisa, es fundamental poseer una correcta y detallada definición de la estructura y la dinámica de la organización que tales requisitos del sistema deberán soportar.

El profile UML para el modelado del negocio, ofrece un interesante aporte al desarrollo de software puesto que ayuda a realizar y obtener un modelo consistente de la estructura y dinámica del negocio en la que se pretende implantar un sistema de software. Y con el conocimiento de que los profiles UML permiten la construcción de modelos correctos a partir de un lenguaje específico, que ayuda a los ingenieros de software a organizar y representar conceptos del dominio específico que se quiere modelar.

## Referencias Bibliográficas

[1] Ivar Jacobson, Grady Booch, James Rumbaugh. “The Unified Software Development Process”. Addison Wesley, 1999.

- [2] Rational Unified Process. <http://www.rational.com/rup/>
- [3] Grady Booch, James Rumbaugh, Ivar Jacobson. "The Unified Modeling Language". Addison Wesley, 1999.
- [4] Eriksson, H.E., Penker, M.: Business Modeling with UML. Business Patterns at Work. John. Wiley & Sons, Inc. 2000. ISBN: 0-471-29551-5.
- [5] Baum Gabriel, Daniele Marcela, Martellotto Paola, Novaira María M Propuesta de un modelo genérico para el modelo de negocio. en el VI Workshop de Investigadores en Ciencias de la Computación (WICC 2004), 215-219, Universidad Nacional del Comahue, Neuquén (Argentina), 20 y 21 de Mayo, 2004.
- [6] Baum G., Daniele M., Martellotto P. & Novaira M. (2004). Un Modelo Genérico para el Modelo de Negocio. X Congreso Argentino de Ciencias de la Computación, CACIC'2004: ISBN 987-9495-58-6, Universidad Nacional de La Matanza, Octubre de 2004.
- [7] Daniele M., Martellotto P, Baum G. Formalización del Modelo Genérico del Modelo de Negocio. Anales de la 34 JAIIO (Jornadas Argentinas de Informática e Investigación Operativa) ISSN1666 1141, en el Simposio ASIS (Simposio Argentino en Sistemas de Información), Septiembre 2005, Rosario, Argentina.
- [8] Daniele M., Martellotto P, Baum G. Generic Model of the Business Model and its Formalization in Object-Z. Capítulo del libro "Verification, Validation and Testing in Software Engineering", Editado por Aristides Dasso y Ana Funes, publicado por Idea Group. <http://www.idea-group.com>. ISBN: Hard cover: 1-59140-851-2. Soft cover: 1-59140-852-0. E-book: 1-59140-853-9. Jul 2006.
- [9] Pedro Sinogas, André Vasconcelos, Artur Caetano, João Neves, Ricardo Mendes, José Tribolet. Business Processes Extensions to UML Profile for Business Modeling. 2001. <http://www.inesc-id.pt/pt/indicadores/Ficheiros/894.pdf>
- [10] OMG (Object Management Group). <http://www.omg.org/>
- [11] Unified Modeling Language: Superstructure and Infrastructure Specification. Version 2.0. Formal/05-07-04. Formal/05-07-05. <http://www.uml.org/>
- [12] OCL 2.0 Specification. Formal 06-05-01. [www.omg.org/docs/formal/06-05-01.pdf](http://www.omg.org/docs/formal/06-05-01.pdf).
- [13] Fuentes L. y Vallecito A., 2003. Una Introducción a los Perfiles UML. Univ. de Málaga.
- [14] UML Profile for Patterns Specification. February 2004. Version 1.0. formal/04-02-04. An Adopted Specification of the Object Management Group, Inc.
- [15] SPEM-Software Process Engineering. Metamodel Specification. Nov. 2002. Version 1.0. formal/02-11-14. An Adopted Specification of the Object Management Group, Inc.
- [16] Johnston Simon. Rational UML Profile for business modeling. Jul 2004. <http://www-128.ibm.com/developerworks/rational/library/5167.html>.