

Reglas heurísticas para la generación de casos de uso a partir de esquemas preconceptuales.

Generating Use Cases Based on Heuristic Rules from Preconceptual Schemes.

Carlos Mario Zapata Jaramillo*
Heidy Joana Villa Buitrago**
John Jairo Chaverra Mojica***



Tipo de artículo: Resultado de Investigación.

Recibido: 8 de julio, 2014

Aceptado: 20 de septiembre, 2014

Resumen

Los casos de uso son herramientas de gran utilidad durante el desarrollo de aplicaciones de software, ya que sirven para determinar los requisitos funcionales del sistema. Existen diferentes trabajos sobre la generación automática de casos de uso partiendo de descripciones en lenguaje natural o controlado. Sin embargo, algunos de estos trabajos se enfocan en la solución, la cual no se tiene clara en las etapas iniciales del desarrollo del software. Por otra parte, se requiere la intervención intensiva de los analistas, lo que conlleva subjetividades que limitan la automatización del proceso y aumentan el tiempo de desarrollo. En este artículo se definen reglas heurísticas para generar casos de uso a partir de esquemas preconceptuales; dichos esquemas parten del dominio del sistema, lo que implica una mejora en la plenitud de los diagramas y la consistencia entre ellos, además de una reducción en el tiempo de elaboración de los casos de uso.

Palabras clave: casos de uso, esquemas preconceptuales, requisitos funcionales.

Abstract

Use cases are useful tools to be used during the development of software applications, since they help to determine the system functional requirements. Some work has been devoted to the automated generation of use cases from either natural or controlled language. However, some of it is focused on the solution, which is not clear in the initial stages of software development. Moreover, intensive intervention from the analysts is required, which leads to subjectivity, limiting the automation of the process and increasing development time. In this paper we define a set of heuristic rules for generating use cases from pre-conceptual schemas; such schemas are based on the system domain, which leads to better completeness of the diagrams and consistency among diagrams, and reduces the use case elaboration time.

Keywords: use cases, preconceptual schemes, functional requirements, software application.

* Ph.D. en Ingeniería. Profesor titular de la Universidad Nacional de Colombia. Líder del grupo de investigación en Lenguajes Computacionales. cmzapata@unal.edu.co

** Estudiante de Maestría en Ingeniería de Sistemas. Universidad Nacional de Colombia. hjvillab@unal.edu.co

*** Magíster en Ingeniería de Sistemas. Universidad Nacional de Colombia. jjchaver@unal.edu.co

Introducción

El diagrama de casos de uso muestra la iteración del usuario con el sistema, lo cual proporciona la base de los requisitos funcionales de la aplicación de software, es decir, la forma como debe operar el sistema y responder ante diferentes acciones del usuario.

Varios autores se enfocan en la generación automática o semiautomática de los diagramas de casos de uso (Ben Achour, 1998; 1999; Pastor et al., 2003; 2004; Shishkov et al., 2002; Liu y Dong, 2003; Liu et al., 2004), tomando como punto de partida el lenguaje natural o un lenguaje controlado. De esta manera se busca plasmar completamente los requisitos de los interesados. Sin embargo, estos autores parten de una especificación del sistema enfocada en la solución y no en el dominio, causando así limitaciones en el proceso de automatización.

En las etapas iniciales del desarrollo de software, los analistas cuentan con una definición del dominio que entregan los interesados. Por ello, para obtener una descripción de la solución se requiere la intervención de los analistas, lo que conlleva a subjetividades y errores de interpretación; además, se incrementa el tiempo de elaboración de la aplicación de software.

En este artículo se definen reglas heurísticas que permiten obtener automáticamente los casos de uso a partir de esquemas preconceptuales (Zapata et al., 2006). Estos esquemas, por ser cercanos al interesado, proporcionan una descripción del dominio con mayor plenitud, lo que contribuye a eliminar problemas de interpretación, reducir los tiempos de elaboración y garantizar que todos los requisitos funcionales del interesado se plasmen en los casos de uso. Las reglas heurísticas se ejemplifican con un estudio de laboratorio.

Este artículo se organiza de la siguiente manera: en la sección 2 se expone la metodología, en la

sección 3 se define el marco teórico que reúne los conceptos del dominio, en la sección 4 se realiza un análisis crítico del trabajo enfocado en la obtención de diagramas de casos de uso, en la sección 5 se definen las reglas heurísticas para la generación de casos de uso desde esquemas preconceptuales, en la sección 6 se presenta un estudio de laboratorio para ejemplificar las reglas heurísticas, finalmente, en la sección 7 se presentan las conclusiones y el trabajo futuro.

Metodología

Análisis de un estudio de laboratorio que se enfoca en dar solución a los problemas de inventario presentados en una empresa fabricante de ropa. Este estudio de laboratorio se usa para ejemplificar las reglas heurísticas que permiten generar automáticamente casos de uso a partir de esquemas preconceptuales y un rastreo bibliográfico de reglas heurísticas.

Esquemas preconceptuales

Según Zapata, los esquemas preconceptuales son modelos que permiten la representación gráfica de la terminología de un dominio para facilitar su traducción a diferentes esquemas conceptuales. En la Figura 1 se muestran los diferentes elementos que conforman un esquema preconceptual, cuya descripción es la siguiente (Zapata et al., 2006):

- Concepto: es un sustantivo o un sintagma nominal del discurso del interesado.
- Nota: permite especificar los diferentes valores que se pueden asignar a los conceptos.
- Relación estructural: es una relación permanente entre los conceptos. Se asocia con los verbos “es” y “tiene”.
- Relación dinámica: se asocia con los verbos de actividad.
- Condicional: es una relación de causalidad que indica las restricciones que los conceptos deben cumplir.

- Implicación: sirve para unir relaciones dinámicas o para unir condicionales con relaciones dinámicas, estableciendo entre ellas una relación causa-efecto.
- Conexión: permite enlazar los conceptos con las relaciones y las relaciones con los conceptos. Un tipo especial de conexión permite enlazar los conceptos y las notas.

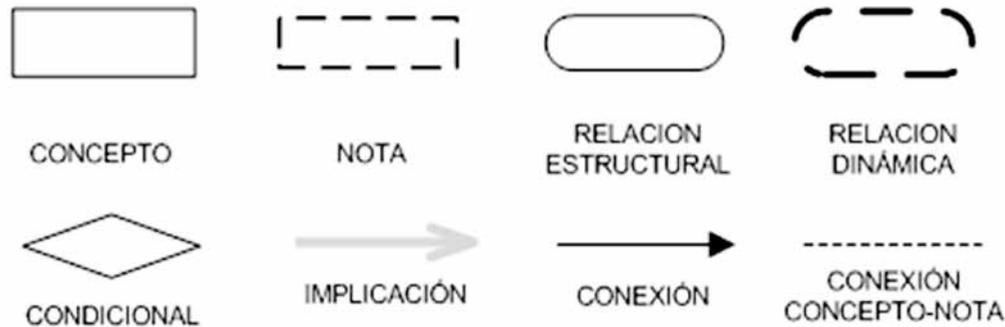


Figura 1. Esquemas preconceptuales

Fuente: Zapata et al. (2006)

Diagramas de casos de uso

En la especificación de la superestructura del Unified Modeling Language UML (OMG, 2007) el diagrama de casos de uso se define como el “diagrama que muestra las relaciones entre los actores y el sistema. El modelo de casos de uso describe los requisitos funcionales del sistema en términos de las secuencias de acciones, incluyendo las variantes que el sistema u otra entidad pueda realizar interactuando con actores del sistema”. A continuación se presentan los elementos de la especificación del diagrama de casos de uso (OMG, 2007; Fowler, 2004):

- Casos de uso: son las especificaciones de los conjuntos de acciones que realiza el actor sobre el sistema, es decir, son los pasos que describen de principio a fin un proceso.
- Actores: son los roles que los usuarios desempeñan respecto del sistema y que emplean los casos de uso. Los actores pueden ser usuarios humanos y otros sistemas, que se comunican con el sistema que se requiere.
- Relaciones: son interacciones que se pueden presentar entre casos de uso, entre actores y casos de uso y entre los actores. Las relaciones pueden ser de cuatro tipos:
 - Asociación: se establece entre los actores y los casos de uso.
 - <<include>>: se presenta cuando una instancia del caso de uso origen incluye también el comportamiento descrito por el caso de uso destino; es decir, un caso de uso incluido describe un objetivo de bajo nivel de un caso de uso base (Cockburn, 2000).
 - <<extends>>: ocurre cuando el caso de uso origen extiende el comportamiento del caso de uso destino; en otras palabras, el caso de uso base invoca el caso de uso a extender base bajo ciertas condiciones (Cockburn, 2000).
 - <<inheritance>>: se presenta cuando un caso de uso hereda la especificación del caso de uso origen y posiblemente la modifica y/o amplía. Este tipo de relación también se presenta entre los actores.
 - Escenario: es una secuencia específica de acciones que ilustra un comportamiento. Los escenarios se usan para ilustrar una interacción o la ejecución de una instancia de un caso de uso.

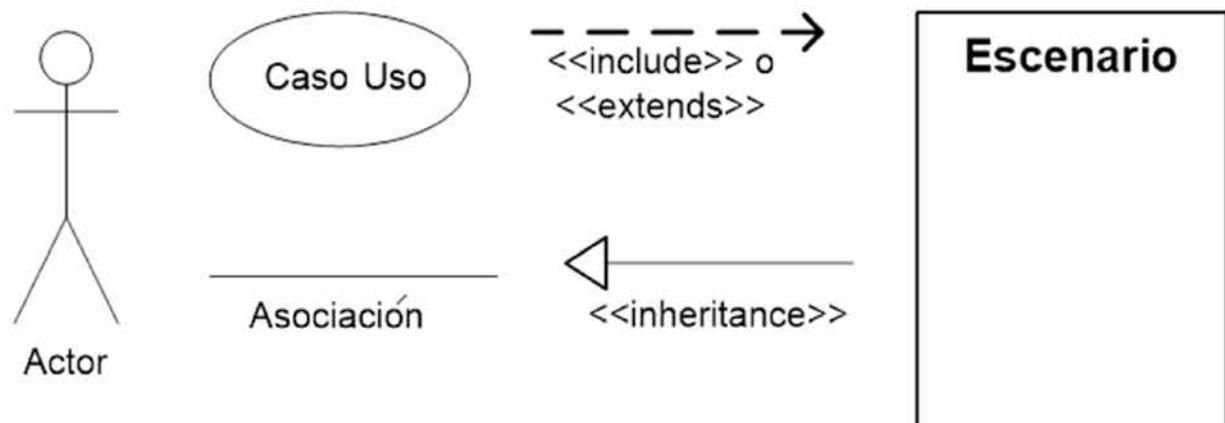


Figura 2. Diagrama de casos de uso
Fuente: Elaboración propia (2014)

Antecedentes

Varios autores especifican reglas heurísticas que abordan la generación automática o semiautomática de casos de uso a partir de diferentes tipos de diagramas o descripciones del sistema (Zapata y Tamayo, 2007; Dong, 2003; Liu et al., 2004; López et al., 2000).

Dado que los casos de uso describen los requisitos funcionales del sistema, su elaboración suele partir de descripciones orientadas a la solución, causando así problemas de plenitud en los diagramas y subjetividad surgida en el análisis. Al tomar como base los esquemas preconceptuales se eliminan estos problemas, pues al partir del dominio del problema, el interesado puede validar la información desde el inicio del desarrollo de software y, por medio de las reglas heurísticas, se apoya la plenitud y consistencia en los diagramas de casos de uso.

Por otra parte, Chaverra (2011) agrega un mayor nivel de detalle a las relaciones dinámicas, al incluir las denominadas “especificaciones” que caracterizan el comportamiento de la relación dinámica en términos de algunas operaciones

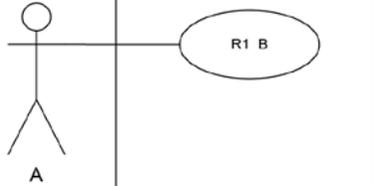
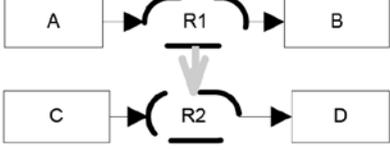
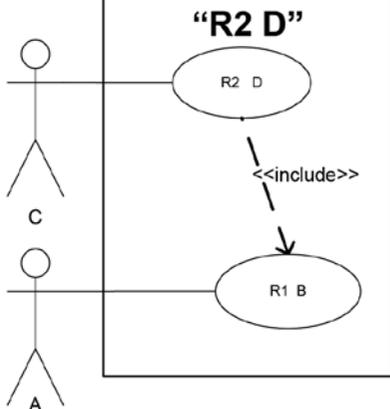
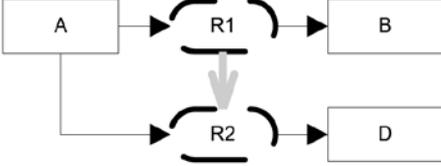
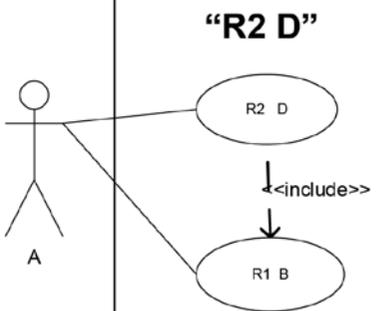
elementales (insertar, editar, borrar, seleccionar y buscar). En este artículo se definen reglas heurísticas para obtener la especificación de los casos de uso a partir de la especificación de las relaciones dinámicas, dando un valor agregado sobre otras propuestas, pues se puede realizar la validación de las especificaciones desde fases tempranas del desarrollo de software, cuando se está levantando la información del dominio del problema.

Reglas heurísticas

Partiendo del esquema preconceptual, el cual engloba el dominio del problema del interesado, en este artículo se definen reglas heurísticas para generar diagramas de casos de uso. Así se obtienen los requisitos funcionales del sistema, con una mirada más completa del dominio y un diagrama con mayor consistencia respecto de las necesidades del interesado.

A continuación, en la Tabla 1, se enmarcan las reglas heurísticas con la notación definida en las Figuras 1 y 2.

Tabla 1. Reglas heurísticas para la generación de los casos de uso a partir de esquemas preconceptuales

Numeración	Esquema preconceptual	Diagrama de casos de uso
1.		
2.		
3.		
4.		
5.		

Fuente: Elaboración propia (2014)

En la Tabla 2 se enmarcan las reglas para definir la especificación del caso de uso (R1 B) definido anteriormente en la regla número 2.

Tabla 2. Reglas para definir la especificación del caso de uso (R1 B)

Caso de uso	C + Código consecutivo. Relación dinámica (R1) + Concepto destino (B)
Versión	Se genera un código consecutivo empezando en uno.
Fecha	Fecha actual.
Autor	Quien lo está realizando.
Fuente	Esquemas preconceptuales.
Propósito	Caso de uso.
Objetivo	Con este caso de uso se pretende + Caso de uso (R1 B) + Nombre del proyecto.
Resumen	Relaciones dinámicas de la especificación.
Actores	A + Código consecutivo. Nombre del concepto identificado como actor (A).
Precondición	Datos que se debieron ingresar con anterioridad para poder seleccionar, agregar, listar o insertar.
Secuencia de interacciones	Flujo normal: se da a partir de la especificación de la relación dinámica (R1), la cual se asocia con el caso de uso que se está describiendo. Dicha especificación se divide en dos secciones ; en la primera división se enmarcan todas las acciones que realiza el actor y en la segunda división, se enmarcan todas las respuestas que entrega el sistema. El flujo normal del caso de uso es la secuencia de estas acciones en el orden que aparecen en la especificación de la relación dinámica. Flujo alternativo: existe un flujo alternativo cuando dentro de la especificación de la relación dinámica hay un condicional entendiéndose como flujo alternativo la secuencia de acciones que se deben realizar al seguir por la rama negativa del condicional.
Frecuencia	Los casos de uso de tipo primario tienen una frecuencia de 1 como mínimo. Los casos de uso de tipo secundario se ejecutará n el número de veces necesarias para realizar determinada acción.
Tipo	Primario: p or defecto , todos los casos de uso se tomaran como tipo primario. Secundario: todos los casos de uso que hagan parte de un <i>extends</i> o de un <i>include</i> .
Postcondiciones	Resultado de las acciones que se realizan dentro de la especificación de la relación dinámica. Nombre del concepto destino (B) + relación dinámica en pasado (R1).
Gráfico	<p>El gráfico muestra un actor representado por un símbolo de palo humano con la letra 'A' debajo. Una línea horizontal conecta el actor con un recuadro rectangular que contiene un óvalo con el texto 'R1 B' dentro. Encima del recuadro, el texto 'R1 B' está encerrado en comillas.</p>

Fuente: Elaboración propia (2014)

Estudio de laboratorio

El ejemplo que se analiza en este estudio de laboratorio se enfoca en dar solución a los problemas de inventario presentados en una empresa fabricante de ropa. Este estudio de laboratorio se usa para ejemplificar las reglas heurísticas que permiten generar automáticamente casos de uso a partir de esquemas preconceptuales.

HVB Ltda. es una empresa innovadora, con grandes ideas. Desde sus inicios tuvo como meta buscar la expansión de la empresa, conforme al incremento de las ganancias. Sin embargo, no se desarrolló un plan estructurado para alcanzar esta meta.

La empresa cuenta con unas instalaciones donde se confecciona la ropa y se distribuye, además de una tienda limítrofe donde se venden las prendas. En esta tienda trabajan dos vendedores, quienes se encargan de registrar las ventas, los descuentos y atender los clientes. En la fábrica se encuentran laborando el gerente, el supervisor de ventas y el supervisor de producción.

El supervisor de producción registra la compra de insumos, materia prima y outsourcing; de cada compra se debe registrar el nombre del producto, cantidad, fecha, descripción, valor y nombre del proveedor. De dicho proveedor se debe conocer su nombre, NIT, identificación, dirección y teléfono de contacto.

La ropa que fabrica HVB Ltda. se puede vender al detal o al por mayor. Se tiene la restricción

que solo a distribuidores se puede vender al por mayor. Las prendas se clasifican por grupos, los cuales tienen un código que los identifica, un nombre, código de barras, fecha de puesta en venta, destino y cantidad.

Cuando los vendedores realizan una venta, deben ingresar en el sistema el código, el detalle y el descuento, si lo tiene. En el detalle se debe seleccionar la prenda, ingresar la cantidad para calcular el subtotal y, posteriormente, el total; adicionalmente, deben registrar el cliente con su identificación, nombre y teléfono.

Los descuentos en las prendas solamente se dan en ciertas circunstancias: en el momento que las prendas tienen más de tres meses en el mercado, se les realiza un descuento del 20%; así mismo, en el aniversario de la empresa todas las prendas tienen un descuento del 50%. Estos descuentos se realizan con el objetivo de velar por la novedad en las prendas.

Las ventas realizadas a distribuidores siempre se hacen al por mayor. De los distribuidores se conoce su dirección, código, NIT, nombre y teléfono. Cuando se realiza una venta de este tipo, el supervisor la registra con su respectivo detalle y los datos del distribuidor que les compró. En la Figura 3 se presenta el esquema preconceptual que ilustra el dominio descrito en el caso de estudio.

En la Tabla 3 se presentan el esquema heurísticas definidas con anterioridad, y la respectiva especificación de dicho caso de uso. del esquema preconceptual al aplicar las reglas

Tabla 3. Esquema preconceptual al aplicar las reglas heurísticas

No.	Esquema preconceptual	Diagrama de caso de uso	Especificación caso de uso
1.			

Fuente: Elaboración propia (2014)

A continuación, en las Tablas 4 y 5, se definen las especificaciones de los casos de uso, respectivamente tomados como ejemplo para el estudio de laboratorio.

Tabla 4. Especificación caso de uso registrar venta

Caso de uso	<i>CU0: Registrar venta</i>		
Versión	<i>1.0</i>	<i>Fecha</i>	<i>Fecha actual</i>
Autor	<i>Heidy Joana Villa Buitrago</i>		
Fuente	<i>Esquema preconceptual</i>		
Propósito	<i>Registrar venta</i>		
Objetivo	<i>Con este caso de uso se pretende registrar ventas del proyectode inventario HVB Ltda.</i>		
Resumen			
Actores	<i>A0: Supervisor de Ventas</i>		
Precondición	<i>Que se ingresen previamente compradores. Que se ingrese previamente grupo de prendas.</i>		
Secuencia de interacciones	Supervisor de Ventas	SISTEMA	
1		Muestra interfaz para ingresar datos de venta.	
2	Ingresa datos de la venta		
3	Agrega detalle		
4	Selecciona la prenda	Véase caso de uso ingresar grupo prenda	
5	Selecciona comprador	Véase caso de uso ingresar comprador	
6	Da <i>click</i> en el botón insertar venta	Inserta la venta	
7	Da <i>click</i> en el botón insertar detalle	Inserta detalle	
8	Da <i>click</i> sobre el botón registrar	Registra venta	
9		Muestra interfaz principal	
Demora			
Frecuencia	<i>Cada vez que se registre una venta.</i>		
Tipo	<i>Primario</i>		
Postcondiciones	<i>Venta registrada</i>		
Gráfico	<pre> graph TD subgraph "Registrar Venta" CU0([CU0: Registrar_Venta]) CU1([CU1: Ingresar_Grupo_Prenda]) CU0 -.-> <<include>> CU1 end A0((A0: Vendedor)) --- CU0 A1((A1: Supervisor_Producción)) --- CU1 </pre>		

Fuente: Elaboración Propia (2014)

Tabla 5. Especificación caso de uso registrar compra

Caso de uso	<i>CU3: Registrar compra</i>		
Versión	<i>1.0</i>	Fecha	<i>Fecha actual</i>
Autor	<i>Heidy Joana Villa Buitrago</i>		
Fuente	<i>Esquema preconceptual</i>		
Propósito	<i>Registrar compra</i>		
Objetivo	<i>Con este caso de uso se pretende registrar compras del proyecto de Inventario HVB Ltda.</i>		
Resumen			
Actores	<i>A1: Supervisor de Producción</i>		
Precondición	<i>Que se ingresen previamente los proveedores.</i>		
Secuencia de interacciones	Supervisor de Producción	SISTEMA	
1		Muestra interfaz para ingresar datos de venta	
2	Ingresa datos de la compra		
	Selecciona proveedor	Véase caso de uso ingresar proveedor	
	Agrega detalle		
	Da <i>click</i> en el botón insertar compra	Inserta la compra	
	Da <i>click</i> en el botón insertar detalle	Inserta detalle	
	Da <i>click</i> en el botón registrar	Registra compra	
		Muestra interfaz principal	
Demora			
Frecuencia	<i>Cada vez que se registre una compra.</i>		
Tipo	<i>Primario</i>		
Postcondiciones	<i>Compra registrada.</i>		
Gráfico	<p>The diagram is titled "Registrar Compra". It features an actor labeled "A1: Supervisor_producción" on the left. A solid line connects the actor to an oval use case labeled "CU3: Registrar Compra". A dashed line with an open arrowhead points from a lower oval use case labeled "CU4: Ingresar Proveedor" up to "CU3: Registrar Compra", with the text "«extends»" written next to the arrow.</p>		

Fuente: Elaboración propia (2014)

Conclusiones

Con las reglas heurísticas que se definieron en este artículo se logró construir casos de uso con una mayor completitud y consistencia, dado que, al partir de los esquemas preconceptuales en los que se encuentra plasmado el dominio del problema y no la solución, se tiene mayor cercanía a las necesidades del interesado y, además, se reduce el tiempo de elaboración, reduciendo el ciclo de vida del software.

En los casos de uso obtenidos a partir de los esquemas preconceptuales, se da una visión amplia del dominio del sistema. Por ende, los requisitos funcionales se definen a cabalidad, asegurando el desarrollo de un producto de software que cumpla con todas las necesidades del interesado.

Como se puede identificar en el estudio de laboratorio que se ilustra en este artículo, algunos elementos de la especificación del caso de uso no se definieron, ya que aún no existen reglas heurísticas que generen automáticamente dichos elementos. Esto se debe a que en el alcance de este proyecto no se tenían contemplados esos elementos y, consecuentemente, se propone realizar un estudio futuro que analice los elementos necesarios para definir las reglas heurísticas faltantes que referencien estos elementos.

Referencias

Ben, A. C. (1998). Guiding the construction of textual case use specifications. *Data & Knowledge Engineering*, 25(2), 125-160.

Chaverra, J. (2011). *Generación automática de prototipos funcionales a partir de esquemas preconceptuales*. Tesis de Grado. Universidad Nacional de Colombia, Medellín.

Cockburn, A. (2000). *Characterizing People as Non-Linear, First-Order Components in Software Development*. Presented at the 4th International Multi-Conference on Systems, Cybernetics and Informatics, Orlando.

Dong, J. (2003). *Representing the Applications and Compositions of Desing Patterns in UML*. *Proceedings of the 2003. ACM Symposium on Applied Computing*, Melbourne.

Fowler, M. (2004). *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Boston: Addison Wesley.

Liu, D. (2003). *Automating Transition from Use Cases to Class Model*. M.Sc. Thesis, University of Calgary.

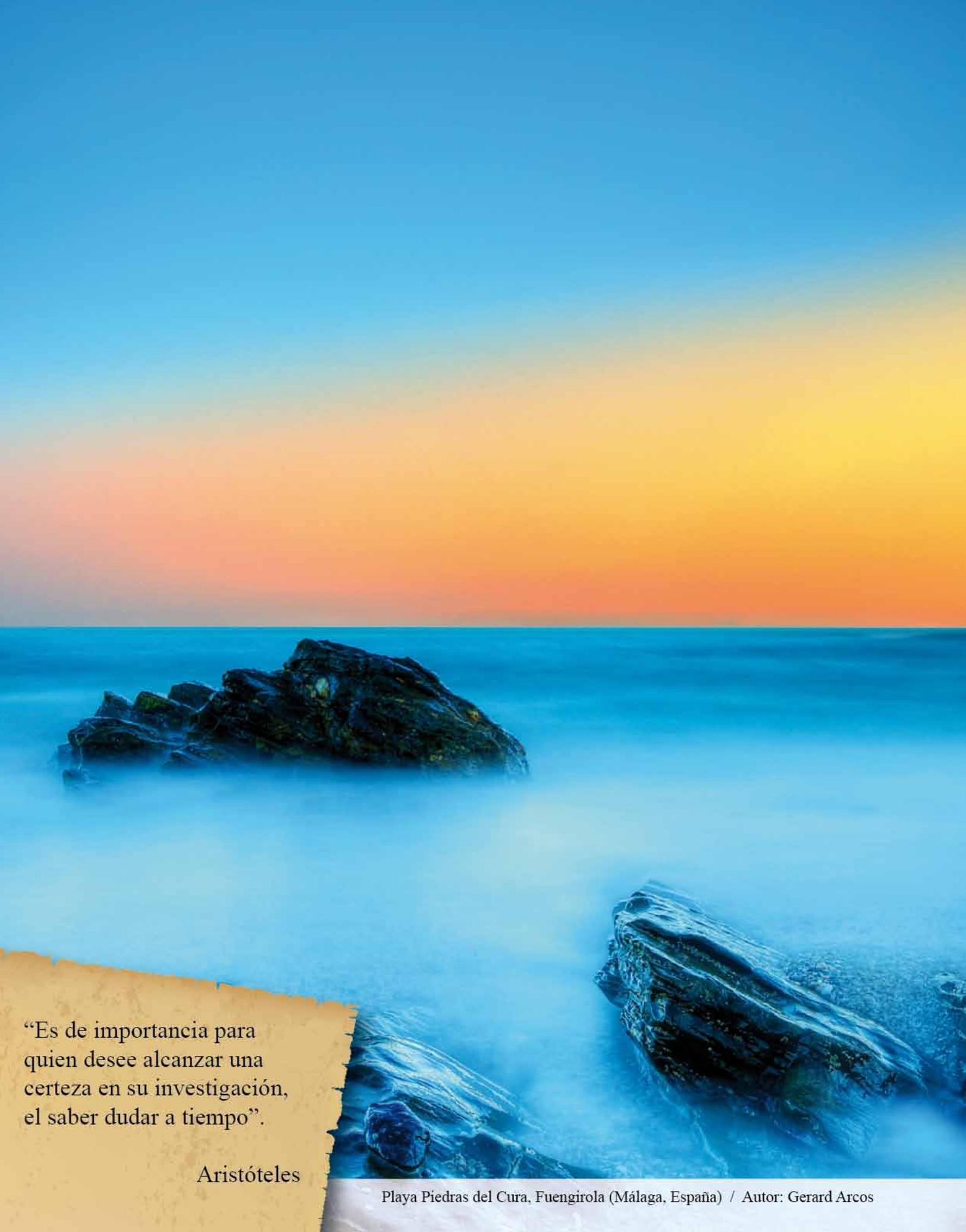
Liu, D., Subramaniam, K., Eberlei, A. & Far, B. (2004). Natural Language Requirements Analysis and Class Model Generation Using UCDA. *Lecture Notes on Artificial Intelligence*, 30C(29), 295-304.

López, O., Laguna, M. & Márques, J. M. (2000). *Generación automática de casos de uso para desarrollo de software basado en reutilización*. V Jornadas Ingeniería de Software y Bases de Datos (JISBD 2000), Valladolid.

OMG – Object Management Group (2007). Recuperado de <http://www.omg.org/>

Pastor, O., Díaz, I., Losavio, F. & Matteo, A. (2004). Specification Pattern of Use. *Information and Management*, 41, 961-975.

- Pastor, O., Díaz, I. & Moreno, L. (2003). *Traducción de casos de uso en patrones de interacción de instancias: una aproximación lingüística*. Memorias de las terceras Jornadas Iberoamericanas de Ingeniería de Software e Ingeniería del Conocimiento. Valdivia.
- Shishkov, B., Xie, Z., Lui, K. & Dietz, J. (2002). *Using Norm Analysis to Derive Use Case from Business Processes*. Proceedings of the 5th Workshop on Organizations semiotics, Delft.
- Zapata, C., Gelbukh, A. & Arango, F. (2006). Pre-conceptual Schemas: a Conceptual-graph-like Knowledge Representation for Requirements Elicitation, *Lecture Notes in Computer Sciences*, 42(93), 17-27.
- Zapata, C., Gelbukh, A. & Arango, F. (2008). UNlencep: obtención automática de diagramas UML a partir de lenguaje controlado. *Dyna*, 74, (153), 223-236.
- Zapata, C. & Tamayo, P. (2007). Generación del diagrama de casos de uso a partir del lenguaje natural o controlado: una revisión crítica. *Dyna*, 76, (159), 193-203.
- Zapata, C., Tamayo, P. & Arango, F. (2007). Conversión de esquemas preconceptuales a diagramas de casos de uso empleando ATOM3. *Dyna*, 74, (153), 237-251.



“Es de importancia para
quien desee alcanzar una
certeza en su investigación,
el saber dudar a tiempo”.

Aristóteles