

AREA: *Ingeniería de Software*

EJE TEMÁTICO: *Interacción Humano Computadora (HCI)*

TÍTULO: **PATRONES DE INTERACCIÓN APLICADOS A PLANTILLAS DE CASOS DE USO Y DISEÑO DE INTERFACES.**

AUTOR: *Marcuzzi, Cecilia. Becaria CICITCA 2011-2012.*

CATEGORIA: *Estudiante Avanzada de la carrera Licenciatura en Sistemas de Información*

INSTITUCION: *Instituto de Informática – Fac. Cs. Exactas F. y Naturales – Universidad Nacional de San Juan*

MAIL: *cecymar_9@yahoo.com.ar*

Resumen— El presente trabajo ha sido realizado dentro de una beca de investigación para estudiantes avanzados, cuyo objetivo principal fue recopilar y analizar patrones de interacción para el diseño de interfaz, en el marco del proyecto de investigación “Formalización de descripciones de Casos de Uso” (E-883). Primero se analizaron las características de usabilidad en el diseño de una interfaz, dentro de la disciplina Interacción Humano-Computador, luego se identificaron, detallaron y se aplicaron los patrones de interacción y usabilidad a casos de estudio reales, para adquirir experiencia y poder reconocer elementos o características comunes deseables en un diseño de interfaz usable y, a partir del conocimiento obtenido se extendió la plantilla Integradora para Documentar Casos de Uso v1.4 (generada por el grupo de investigación en proyecto anterior). Se incorporaron estos atributos comunes, a fin de que sean tenidos en cuenta en etapas tempranas del desarrollo de software y en los subsecuentes procesos de transformación propuestos por el Desarrollo Dirigido por Modelos .

Palabras Clave— Interacción Humano-Computadora; Modelo de casos de uso; Patrones; Plantillas; Usabilidad.

I. INTRODUCCIÓN

En un mercado o una sociedad cada vez más informatizada y más exigente el usuario común no se conforma simplemente con un sistema que hace lo que él necesita, sino que su conformidad o satisfacción tiene que ver también con otras características adicionales, como la facilidad de uso, comodidad en el uso y aspectos de diseño visual del mismo (Nielsen, 1999).

Según Nielsen (Nielsen, 1999) estos aspectos son conocidos, dentro de la disciplina de Interacción Humano-Computador, como características de ‘usabilidad’ del software y, además, la usabilidad forma parte de los atributos de calidad del software.

En base a estas características, con el objeto brindar al cliente un software que cubra todas sus necesidades y expectativas, y siguiendo los principios ya instaurados en la comunidad, surge la necesidad de identificar patrones de usabilidad para el diseño de interfaz, en etapas tempranas del desarrollo de software. La utilización de los patrones se llevará a cabo por medio de la extensión ó ampliación de la plantilla CUPIDO (Plantilla Integradora para Documentar Casos de Uso) desarrollada en el proyecto E-822: “Modelo de Casos de Uso - un Eje para el Proceso de Desarrollo de Software” (CuProSoft) (Lund et al., 2010). A través de la misma es posible especificar un caso de uso e identificar la forma de interactuar entre el usuario y el sistema.

Los patrones son soluciones a problemas comunes en el diseño. Es buena costumbre implementarlos, ya que ahorran tiempo y mejoran en efectividad y eficiencia la aplicación. Erich Gamma y otros (Gamma et al., 1995) sugieren la siguiente definición de Christopher Alexander: "*Cada patrón de diseño describe un problema que ocurre una y otra vez en nuestro entorno, así como la solución a ese problema, de tal modo que se puede aplicar esta solución un millón de veces, sin hacer lo mismo dos veces*".

Los diseñadores y desarrolladores históricamente no le han dado la suficiente importancia al aspecto de usabilidad dentro del proceso de diseño, porque no existe una guía estandarizada que los ayude en dicho proceso (Ford y Gelderblom, 2003).

Teniendo en cuenta las falencias en el diseño de interfaces desde una perspectiva de usabilidad, se ve la necesidad de presentar una iniciativa que permita minimizar dicha problemática con la definición de un marco de referencia o patrón HCI (Interacción Humano-Computador) para el diseño de interfaces, que sirva de guía a los diseñadores y desarrolladores en el proceso de diseño y construcción de interfaces para el Desarrollo Dirigido por Modelos (MDD).

II. MARCO TEÓRICO

A. *interacción Humano-Computadora*

La Interacción Persona-Ordenador (IPO) más conocida por su nombre en inglés, Human-Computer Interaction (HCI), es el estudio de la relación que existe entre los usuarios humanos y los sistemas informáticos que usan para realizar diversas tareas (Faulkner, 1998).

Una definición de HCI es “*Proveer un entendimiento de la forma en que los usuarios trabajan, las tareas que necesitan ejecutar y la forma en que los sistemas computacionales necesitan ser estructurados para facilitar el logro de dichas tareas*” (Faulkner, 1998).

La ACM, Association for Computer Machinery (Nielsen, 1993), propone “*Es la disciplina relacionada con el diseño, evaluación e implementación de sistemas informáticos interactivos para el uso de seres humanos y con el estudio de los fenómenos más importantes con los que está relacionado*”.

Es decir que los objetos de estudio de esta disciplina son el Usuario, que es la persona que interactúa con un sistema informático y las Interacciones, que son todos los intercambios que suceden entre la persona y la computadora (Baecker y Buxton, 1989).

El enfoque científico de HCI incluye una gran variedad de herramientas y técnicas que ayudan a desarrollar mejores interfaces de usuario.

B. Usabilidad

La usabilidad se define como “*un atributo de calidad que determina la facilidad de la interfaz para ser utilizada*” (Ferré et al., 2001: 22–30) . Se asocia a él estos cinco atributos:

- Aprendizaje — rapidez y facilidad con las que los usuarios pueden realizar trabajo productivo con un sistema que no conocen, junto con la facilidad con la que recuerdan la forma en la que se debe operar con el sistema.

- Eficiencia en el uso — el número de tareas por unidad de tiempo que el usuario puede realizar con el sistema.

- Fiabilidad — también llamada “fiabilidad en el uso”, se refiere al porcentaje de errores cometidos por el usuario en el uso del sistema y el tiempo que se tarda en recuperarse de estos errores.

- Satisfacción — las opiniones subjetivas que se forman los usuarios al utilizar el sistema.

- Errores— el sistema debe tener una baja tasa de errores (Moreno y Sánchez-Segura, 2003: 117–126).

El concepto de usabilidad también se refiere a los métodos para mejorar la facilidad de uso durante el proceso de diseño (Nielsen, 1999).

Otra definición expresa: “*Es el grado en el cual un producto puede ser usado por usuarios determinados para lograr los objetivos específicos con efectividad, eficiencia y satisfacción, en un contexto específico de uso*” (ISO, 1998: 9241-11).

Es muy importante tener en cuenta que se trabaja para seres humanos que quieren realizar una tarea de una forma sencilla y eficaz, frente a una computadora, en un entorno gráfico. La usabilidad ayuda a que esta tarea se lleve a cabo de manera natural, analizando el comportamiento humano y los pasos necesarios para ejecutar la tarea provechosamente.

C. Patrones

La notación formal de los patrones nació con los patrones arquitectónicos de Christopher Alexander “Un patrón es una descripción de un problema, su solución y el contexto de aplicación.; en teoría, indica la manera de utilizarlo en circunstancias diversas” (Alexander, 1980). En los años ‘80 Kent Beck y Ward Cunningham (Beck y Cunningham, 1987) hicieron su aplicación al software.

Es decir que un patrón es una pareja de problema/solución, con un nombre, y que es aplicable a otros contextos con una sugerencia sobre cómo usarlo en situaciones nuevas (Craig, 2004).

El concepto de patrón más ampliamente utilizado en el desarrollo del software es el patrón de diseño. En este contexto, un patrón de diseño es una descripción de las clases y de los objetos que trabajan conjuntamente para resolver un problema concreto. Un patrón de diseño es una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias (Gamma et al., 1995).

Concretamente, los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

D. Clasificación de Patrones

Los patrones analizados se encuentran clasificados en patrones de interacción, de usabilidad y arquitectónicos, según los diferentes autores. Estos últimos no se detallan porque no se ven reflejados en el diseño de la interfaz.

Patrones de interacción

Según Van Welie y H. Trætteberg (Welie y Trætteberg, 2000: 13–16) un patrón de interacción modela un aspecto referente a la interfaz de un Sistema de Información, en función de los requerimientos del usuario, por lo que ayudan a la aplicación o al diseñador a comunicarse mejor con el usuario. Dentro del enfoque HCI se busca que la interacción de los usuarios con el sistema sea lo más natural e intuitiva posible. A través de los patrones de interacción es posible captar la experiencia de los usuarios, minimizando los errores y facilitando la realimentación. De esta forma la interfaz es más intuitiva y los usuarios aprenden a usar el sistema rápidamente.

El concepto de patrones de interacción es cercano a las ideas de Alexander (Alexander et al., 1980), ya que tienen que ver con la representación de la información conforme a las necesidades del usuario, y se traduce en la satisfacción, eficiencia y aceptabilidad que sienten los usuarios al manipular sistemas fáciles de usar. Se exponen los patrones de interacción más conocidos en la Tabla 1.

Tabla 1. Resumen de Patrones de Interacción

PATRONES DE INTERACCIÓN		
Nombre del Patrón	Descripción	Autor
Wizard	Patrón "asistente". Guía al usuario en la realización de una tarea que necesita tomar varias decisiones, mostrarle cuales son los pasos que existen y cuáles han sido realizados a la vez que se le guía a lo largo de la tarea.	(Welie y Trætteberg, 2000: 13–16)
Stepping	Permite a los usuarios ir al paso anterior y siguiente de una tarea para realizar posibles modificaciones.	
Input Error Message	Informa al usuario acerca de entradas de datos incorrectas, dónde se han producido y cómo resolverlas.	
Outgoing Links	Muestra al usuario los enlaces que le llevarán fuera del actual sitio web marcándolos con un icono después de su etiqueta.	

Patrones de usabilidad

La usabilidad es uno de los atributos de calidad claves en el desarrollo de software (Nielsen, 1999).

El concepto de patrón de usabilidad ya ha sido utilizado en la literatura y se define como “una descripción de las soluciones que mejoran los atributos de usabilidad” (Perzel y Kane, 1999). Según (Graham, 2003) y (Tidwell, 1998), un patrón de usabilidad documenta los factores humanos implicados en los elementos de la interfaz para aplicaciones web.

Los aspectos de usabilidad que tratan estos patrones se refieren básicamente a las interfaces de usuario, por lo que se denominan también patrones de interfaz (Casaday, 1997: 289–290) o patrones de diseño de la interacción (Tidwell, 1998). Y, como indican autores como (Welie y Trætteberg, 2000: 13–16), aunque existen varias colecciones de patrones, no ha surgido aún un conjunto ampliamente aceptado de éstos. No parece haber consenso acerca del formato y del enfoque de los patrones de interfaz. En la tabla 2 se presentan los patrones de usabilidad.

Tabla 2. Resumen de Patrones de Usabilidad

PATRONES DE USABILIDAD		
Nombre del Patrón	Descripción	Autor
Sentido de localización	Indica al usuario la ubicación que tiene dentro del sitio web, la cantidad de contenidos y la ruta de información a seguir para llegar a ese punto.	I. Graham (Graham, 2003)
Logotipo del sitio arriba a la izquierda	Permite a los usuarios volver a la página principal (homepage).	
Alert	Muestra un mensaje de alerta al usuario.	J. Tidwell (Tidwell, 1998)
Progress Indication	Muestra un indicador con el progreso real o al menos aproximado de que se está avanzando.	

Web Registration Forms	Ofrece a los usuarios la posibilidad de almacenar su información personal para su posterior uso, sin la necesidad de tener que volver a introducirla.	D. Perzel y D. Kane (Perzel y Kane, 1999)
---------------------------------------	---	---

E. Modelo de casos de uso. Plantillas

El Modelo de Casos de Uso constituye uno de los modelos más utilizados por la industria del Software debido a su simplicidad para definir y describir la funcionalidad de un sistema desde el punto de vista de sus usuarios. En general, un modelo de casos de uso está compuesto por los actores o usuarios y la funcionalidad que el sistema les proveerá.

Jacobson manifiesta en (Jacobson et al.,2000) *“El modelo de casos de uso permite que los desarrolladores del software y los clientes lleguen a un acuerdo sobre los requisitos, es decir, sobre las condiciones y posibilidades que debe cumplir el sistema. El modelo de casos de uso sirve como acuerdo entre clientes y desarrolladores y proporciona la entrada fundamental para el análisis, el diseño y las pruebas.”*

Desde otra perspectiva un tanto informal, se puede decir que un caso de uso describe la forma en que un sistema es empleado por los actores (usuarios del sistema) para alcanzar sus objetivos. Tiene una notación gráfica y está acompañado de una descripción de lo que hace (Larman, 2001: 45–46), esta descripción generalmente se realiza en lenguaje natural. También existen plantillas para describir los casos de uso, con un esquema predefinido, que permiten guiar al analista en la descripción del mismo, reflejando en su estructura los elementos comunes de los Casos de Uso (Balmaceda et al., 2009).

Las plantillas de casos de uso permiten especificar la funcionalidad definida para cada caso de uso. Son documentos estructurados que facilitan el proceso de documentación de un proyecto de software. No existe un estándar para documentar casos de uso, en el proyecto E-822 denominado “Modelo de Casos de Uso - un Eje para el Proceso de Desarrollo de Software” (CuProSoft), se desarrolló una plantilla para documentar y describir casos de uso en etapas tempranas del ciclo de desarrollo, denominada CUPIDo (Casos de Uso, Plantilla Integradora para Documentarlos) (Lund et al., 2010), adaptada a las necesidades y ámbitos de desarrollo de software local y/o regional, en el contexto de los sistemas administrativos, donde la formalización y documentación de sistemas es escasa o nula y cuando existe es netamente informal.

Esta plantilla fue evaluada a través de un experimento controlado (Lund et al., 2011) , y validada en varios trabajos de alumnos de la materia Diseño de Software, entre ellos se cita el diseño de un Notificador de Morosos de libros, para biblioteca (Balmaceda et al., 2009). Desde el año 2010 forma parte de la guía práctica de la materia Diseño de Software de las Licenciaturas en Ciencias de la Computación y Sistemas de Información, de la FCEF-UNSJ.

V. PROPUESTA Y RESULTADOS

La propuesta se basó en incorporar patrones de usabilidad y de interacción, a la última versión de la plantilla CUPIDO, para considerar los aspectos de la interfaz de usuario, en etapas tempranas del desarrollo de software.

Se incorporó un nuevo campo en la plantilla, para permitir identificar los patrones a aplicar en el diseño de la interfaz de ese caso de uso.

Fue necesario indicar cómo esos patrones identificados se incorporan en el flujo de eventos de cada una de las plantillas de los casos de uso del sistema en estudio. Para tal fin se generó una nueva versión de la plantilla CUPIDO, obteniendo como resultado la versión 1.5 de la plantilla, denominada CUPIDO+Pi, reflejando este cambio como se observa en la Fig. 1.

Además del cambio introducido en la planilla, también fue necesario instruir al usuario para incorporar en el flujo de eventos los patrones de interfaz.

Para la aplicación y evaluación de la plantilla propuesta, se tomó un caso de estudio: un “Sistema de Gestión de plantillas para documentación de casos de uso de proyectos de desarrollo de software”. El análisis de este sistema fue realizado en el marco del proyecto E-883/2010 y los casos de uso fueron documentados oportunamente con la plantilla CUPIDO versión 1.4., luego transcritos y adaptados para incorporar los cambios introducidos en la plantilla CUPIDO+Pi.

El sistema consiste en poder gestionar la documentación de casos de uso (con la correspondiente generación, actualización y versionado de las plantillas de casos de uso) para cualquier proyecto de desarrollo de software llevado a cabo por ingenieros de software, gestionando el versionado de la documentación generada, etc.

Proyecto: Sistematización de plantillas de Casos de Uso	Paquete: Carga y actualización de plantillas		
Caso de Uso: 2 -Crear Plantilla de Casos de Uso	Nombre del documento digital: Crear Plantilla de Casos de Uso.doc		
Estado: En revisión	Versión: 2.0	Fecha revisión: 09/05/2012	Hoja: 9/13

Plantilla CUPIDO v1.5

Hoja de Revisión

Fecha	Versión	Descripción	Autor
09/05/2012	2.0	Revisión y modificación de flujo de eventos	CM

Plantilla de Caso de Uso "CUPIDO+Pi"



Nº	Campo	Descripción del campo
1	Actor	<i>Ingeniero de Sistema</i>
2	Tipo de Caso de Uso	<i>Cu.base</i>
3	Breve Descripción	<i>Este caso de uso crea una nueva plantilla de caso de uso de un proyecto determinado</i>
4	Precondiciones	<i>Exista el proyecto</i>
5	Flujo de Eventos	<p><i>El ingeniero de sistemas solicita crear nueva plantilla</i> Activación de patrón Wizard <i>El sistema busca proyecto</i> Activación del patrón Progress indication <i>[si Proyecto existe]</i> <i>El sistema solicita buscar plantilla</i> Activación del patrón Progress indication <i>[Si existe plantilla]</i> <i>El sistema envía a ingeniero mensaje de plantilla existente</i> <i>[Si no existe plantilla]</i> <i>El sistema crea Nueva Plantilla</i> Activación de patrón Wizard <i>El actor ingresa los datos de la plantilla</i> <i>El sistema llama al CU Incorporar actores a la plantilla de CU</i> <i>El sistema llama al CU Cargar Hoja de Revisión</i> <i>El sistema llama al CU Incorporar Flujo de Eventos</i> <i>El sistema busca tipo de Caso de uso asociado</i> <i>El sistema asigna Cu asociado a la plantilla</i> <i>El sistema asigna estado a la plantilla</i> <i>El sistema graba datos de la plantilla</i> <i>El sistema asigna Plantilla Nueva al Proyecto</i> Activación de patrón Web Registration Forms Activación de patrón Input Error Message <i>[Si Crear nueva plantilla es correcto]</i> <i>El sistema informa creación exitosa</i> <i>[Si Crear nueva plantilla es incorrecta]</i> <i>El sistema informa al ingeniero los campos cuyos datos son incorrectos</i></p>
6	Postcondiciones	<i>Plantilla nueva generada</i>
7	Extensión (extend)	
8	Inclusión (include)	<i>Paquete Carga y actualización de plantillas :: CU Cargar Hoja de Revisión, CU Incorporar Actores a la plantilla, CU Incorporar Flujo de Eventos.</i>
9	Requisitos asociados	
10	Consideraciones / Observaciones	

11	Frecuencia de Uso	De tres a cinco veces por día
12	Patrones de Usabilidad	<i>Wizard, Web Registration Forms, Input Error Message, Progress indication</i>

Fig. 1. Plantilla CUPIDo v1.5 – Caso de uso: “Crear Plantilla de Casos de Uso”

La Fig. 1 representa la plantilla versión 1.5 para el caso de uso “Crear plantilla de caso de uso”, con la incorporación de patrones de interfaz propuesta. En este caso, los patrones de interfaz correspondientes a la plantilla son: Wizard, Progress Bar, Web Registration Form e Input Error Message. En el flujo de eventos presentado en la Fig. 1 se puede visualizar la activación de cada uno de los patrones utilizados.

Siguiendo el mismo criterio se analizaron todos los casos de uso del sistema y se incluyeron patrones de interfaz a todas las plantillas ya generadas, a fin de determinar su aplicabilidad y usos, y para inducir y promover el diseño de interfaz del sistema en etapas tempranas del desarrollo de software.

También, durante el curso de la beca, se plantearon todos los prototipos de diseño de cada una de las plantillas confeccionadas, donde se visualizan los patrones implementados. Para el desarrollo de los prototipos se utilizó la aplicación independiente Pencil versión 1.3-4, herramienta libre para el diseño web (pencil).

A continuación, en la Fig. 2 se presenta el prototipo para el caso de uso indicado en la plantilla previa, en el cual se pueden visualizar los patrones de interfaz resaltados.

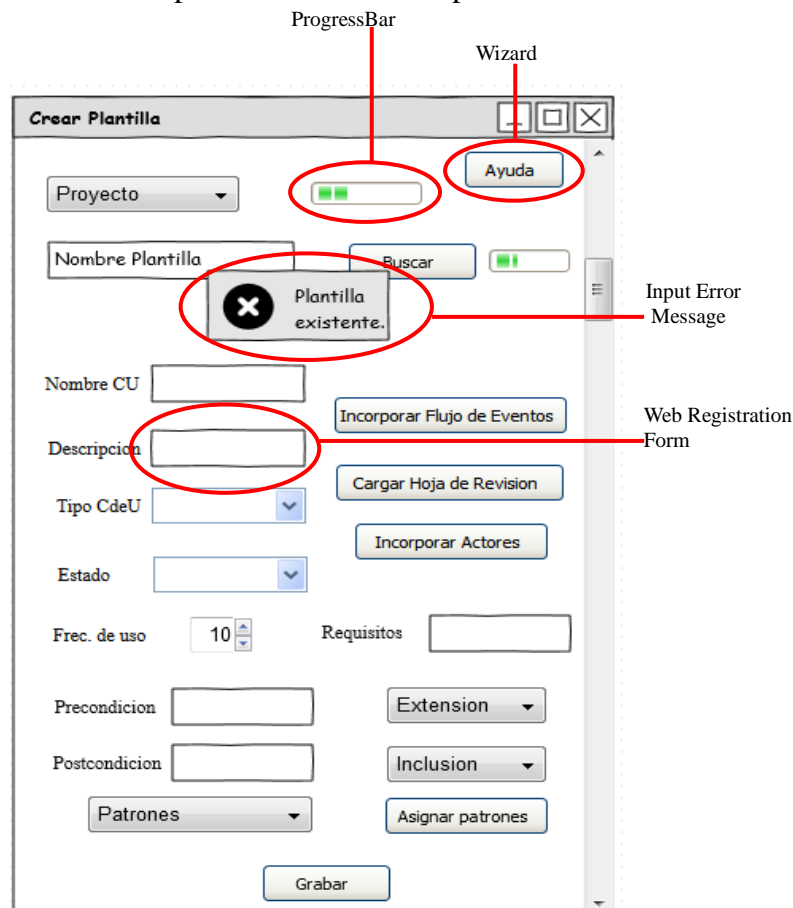


Fig. 2. Pantalla “Asignar Plantilla al proyecto” correspondiente al caso de uso “Crear Plantilla de Casos de Uso”.

IV. CONCLUSIONES

A medida que se fueron identificando los patrones se fue generando una antología, y se analizó y aplicó cada uno de los patrones de interfaz en un caso de estudio real. Así se obtuvo mayor conocimiento de ellos y las bondades de cada uno.

Poder realizar esta aplicación concreta permitió destacar la importancia del uso de patrones de interfaz en etapas tempranas de desarrollo de software, a fin de que el diseñador tenga en cuenta estas características y junto al diseño del sistema pueda generarse el diseño de la interfaz, y minimizar el re-trabajo posterior.

Poder mostrar al cliente o usuario prototipos con la incorporación de patrones de interfaz es un valor agregado y sustancial que, no sólo permitirá a los usuarios vislumbrar el alcance del sistema, sino también mejorar la calidad del sistema a desarrollar, además de familiarizarse con él y minimizar el impacto de implantación de nuevos sistemas.

REFERENCIAS

- J. Nielsen, (1999): “Designing Web Usability”, 1.a ed. Peachpit Press.
- M. I. Lund, C. Ferrarini, L. Aballay, y E. Meni, (2010): ”CUPIDO - Plantilla para Documentar Casos de Uso”, presented at the V Congreso de Tecnología en Educación y Educación en Tecnología, El Calafate, Santa Cruz, Argentina.
- E. Gamma, R. Helm, R. Johnson, y J. Vlissides, (1995) “Design Patterns: Elements of Reusable Object-Oriented Software”.
- Ford G.y Gelderblom H., (2003: 218 – 230): ”The Effects of Culture on Performance Achieved through the use of Human Computer Interaction”, presented at the ACM International Conference Proceeding Series.
- C. Faulkner, (1998) “The essence of human-computer interaction”. Upper Saddle River, NJ, USA: Prentice-Hall, Inc..
- J. Nielsen, (1993): “Usability Engineering” - The Association for Computing Machinery’s Special Interest Group on Computer- Human Interaction (ACM SIGCHI), 1. a ed. Morgan Kaufmann.

R. M. Baecker & W. Buxton, (1989: 118–119): “Readings in Human-Computer Interaction: A multidisciplinary approach” (Morgan Kaufmann Publishers, Inc., 1987), SIGCHI Bull., vol. 21, no. 1.

X. Ferré, N. Juristo, H. Windl, y L. Constantine, (2001: 22–30): “Usability Basics for Software Developers. IEEE Software”, vol. 18 (11),

A. M. Moreno y M. Sánchez-Segura, (2003: 117–126), “Patrones de usabilidad: mejora de la usabilidad del software desde el momento de arquitectónico”.

I. O. for Standardization, (1998): “ISO - International Organization for Standardization” - 9241-11: “Guidance on usability”.

C. Alexander, S. Ishikawa, M. Silverstein, M. Jacobson, S. Angel, y I. Fiksdahl-King, (1980): “Un lenguaje de patrones”, Gustavo Gili.

K. Beck y W. Cunningham, (1987): "Using Pattern Languages for Object-Oriented Programs". OOPSLA'87.

L. Craig, (2004): “UML y Patrones”. Prentice Hall,.

M. Van Welie y H. Trætteberg, (2000: 13–16): “Interaction Patterns in User Interfaces”, in Proc. Seventh Pattern Languages of Programs Conference: PLoP 2000.

I. Graham, (2003): “A pattern language for Web usability”. Addison-Wesley.

J. Tidwell, (1998): “Interaction Design Patterns”. Pattern Languages of Programming”, Washington University, Technical Report TR 98-25.

K. Perzel y D. Kane, (1999): “Usability Patterns for Applications of the World Wide Web”, presented at the PLoP '99 Conference.

F. Buschmann, K. Henney, y D. C. Schmidt, (2007): “Pattern-Oriented Software Architecture: On Patterns and Pattern Languages”. John Wiley & Sons.

G. Casaday, (1997: 289–290): “Notes on a pattern language for interactive usability”, in CHI '97 extended abstracts on Human factors in computing systems: looking to the future, New York, NY, USA.

I. Jacobson, G. Booch, y J. Rumbaugh, (2000): “Chapter 1: The Unified Process: Use-Case Driven, Architecture-Centric, Iterative, and Incremental”, in The Unified Software, 1st ed. Pearson.

C. Larman, (2001: 45–46): “Chapter 6: Use-Case Model: Writing Requirements in Context”, in Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process, 2nd ed., Prentice Hall PTR,.

S. Balmaceda, M.I. Lund y L. Olgúin, (2009): “E-NOTIFICADOR – subsistema de notificación automática a morosos de la FCFN de la UNSJ”, presented at the WICCSI’2009, UNSJ.

M.I.Lund, L.N.Abally, E.Torres, M.Herrera, y E.Ormeño, (2011): “Validación de Usabilidad de una plantilla para documentar Casos de Uso - Estudio Exploratorio”, presented at the XVII Congreso Argentino de Ciencias de la Computación – Workshop Ingeniería de Software (CACIC-WIS), La Plata.

D. S. Frankel, J. Parodi, y R. Soley, Eds., (2004): “The Mda Journal: Model Driven Architecture Straight From The Masters”. Meghan Kiffer Pr.

www.pencil.evolus.vn/en-US/Home.aspx