



MÁSTER EN INGENIERÍA INFORMÁTICA Y DE TELECOMUNICACIÓN

Trabajo Fin de Máster

Soporte CASE para el Análisis de la Arquitectura de la Información en Entornos de Desarrollo Web

Estudiante

Luis Alberto Rojas Pino

luisalberto.rojas@estudiante.uam.es

Director

José Antonio Macías Iglesias

j.macias@uam.es

Madrid, Septiembre de 2011

Agradecimientos

El llegar a escribir estas palabras de agradecimiento me ha significado como costo estar sin mi hijo Gaspar, así que, el primer agradecimiento se lo hago él, espero que el tiempo y el amor sean mis aliados para que un día perdone y comprenda mis decisiones.

Tengo a muchas personas a quien agradecer, principalmente a mi profesor tutor José Antonio Macías, que desde el principio ha sido mi pilar y guía en todo esta etapa de estudio del máster, he admirado su entrega e integridad tanto en el ámbito personal como profesional, y me he sentido afortunado de tener la posibilidad de trabajar junto a él. Agradezco su infinita paciencia e intención de formar y enseñar.

A mi familia, en donde he encontrado refugio incluso en la distancia. A mis hermanas Elizabeth y Cynthia que me han entregado amor y ánimo a través de sus constantes muestras de preocupación y cariño. A mis padres Mario y María por luchar siempre para estar unidos, que incluso en esta distancia física en que nos encontramos jamás los he sentido lejos.

A Rafael Parada y Mercedes Gentil por la ayuda y compañía desinteresada, en ustedes he aprendido a primero amar y después preguntar.

A mis amigos en España, Iván, Enrique, José Luis y Diego, que han sido mi familia en mi estadía fuera de mi país.

La realización del presente trabajo ha sido posible gracias a la subvención del proyecto CCG10-UAM/TIC-5772, cofinanciado por la DGUI de la Comunidad de Madrid y la Universidad Autónoma de Madrid, y del proyecto TIN2008-02081/TIN, financiado por el Programa Nacional de Tecnologías Informáticas.

Resumen

Hoy en día, los equipos de desarrollo de aplicaciones Web interactivas están formados por profesionales de diversas áreas, en donde el Arquitecto de la Información trabaja en aspectos relacionados con el modelado conceptual y en la concreción de aspectos no funcionales de un sitio Web, tratando de encontrar un puente que una los conocimientos conceptuales que plantean los usuarios, dentro del espacio del problema, y la información de diseño que luego necesitarán los analistas e ingenieros software, los cuales se encargarán de diseñar la interfaz Web final.

Sin embargo, los roles del Arquitecto de la Información y del Ingeniero del Software no suelen coincidir, ya que el Arquitecto de la Información puede tener un perfil menos técnico y más orientado a tareas conceptuales, lo cual dificulta la continuidad entre la información de análisis conceptual de la Arquitectura de la Información y la información de análisis y diseño que necesitan los analistas e ingenieros software. Si esta continuidad se pudiera hacer de forma automática, el tiempo y esfuerzo de realización del proyecto se reduciría, permitiendo a cada experto concentrarse en su trabajo en función de sus conocimientos y minimizando el tiempo de trasiego de información entre uno y otro profesional.

En general, las herramientas más comúnmente utilizadas por el Arquitecto de la Información, intentan conectar el *output* del Arquitecto de la Información y el *input* que espera el analista e ingeniero software, a través de la generación en distintos formatos de imágenes y HTML. Sin embargo, esta solución elimina detalles semánticos importantes relativos al análisis, y dificulta la interoperabilidad en el desarrollo de aplicaciones Web. Además, ninguna de estas herramientas permite obtener de forma automática o semi-automática información de análisis y diseño para analistas e ingenieros software, a partir de la información de análisis conceptual del Arquitecto de la Información, la cual pueda dar continuidad al resto de fases del ciclo de vida del proyecto.

Con el objetivo de resolver estos inconvenientes, se ha planteado un formalismo que permite al Arquitecto de la Información representar información conceptual de contenidos y transformarla, mediante reglas específicas, en diagramas de clases UML que den soporte al análisis y diseño de los aspectos funcionales de un sitio Web. Dicho formalismo se ha materializado mediante la construcción de una herramienta CASE (Computer Aided Software Engineering) denominada InterArch, fácil de usar y de aprender por el Arquitecto de la Información. La solución propuesta está basada en el EUD o paradigma del Desarrollo por el Usuario Final, y en paradigmas relacionados como el EUSE (End-User Software Engineering), cuyo objetivo es involucrar a usuarios no expertos en computación en el ciclo de vida del desarrollo software. También se ha realizado una evaluación de la herramienta con usuarios reales, en donde se obtuvieron resultados satisfactorios respecto al grado de usabilidad de la herramienta, lo que además ha permitido corroborar las hipótesis de partida en las que está basada la investigación.

Abstract

Today, development teams of Web applications are made up of professionals from diverse areas, where the Information Architect works on aspects related to conceptual modeling and the realization of non-functional aspects of a Web site, trying to find a bridge between the conceptual knowledge proposed by users in the problem domain and the design information that will need analysts and software engineers, who will be at the end responsible for designing the Web interface.

However, roles of these professionals do not often coincide, due to the profile less technical and more task-oriented that the Information Architect owns. This makes it difficult the continuity between the conceptual analysis of information from the Information Architecture, and the design and analysis information needed by analysts and software engineers. If this continuity could be done automatically, time and effort of the project would be reduced, allowing to each expert to focus on their work based on their knowledge and minimizing the time of transferring information from one professional and another.

Generally, most commonly tools used by the Information Architect hardly try to connect the output of the Information Architect and the input expected by analyst and software engineer through of the generation in different image formats and HTML code. However, this solution eliminates important semantic details and makes it difficult to improve interoperability in the development of the Web application. Moreover, none of these tools allows obtain automatic or semi-automatic analysis and design information, intended for analysts and software engineers, from the conceptual analysis of Information Architect, which allows continuity of the project's lifecycle phases.

In order to solve these problems, it has been proposed a formalism that allows the Information Architect to represent conceptual information content, and so transforming it through specific rules into UML diagrams that support the analysis and design of the functional aspects of a Web site. This formalism has been realized by building a CASE (Computer Aided Software Engineering) tool called InterArch, which has been proved to be easy to use and fast to learn by the Information Architect. The proposed solution is based on the EUD paradigm (End User Development) and other related paradigms such as EUSE (End-User Software Engineering). The main objective of such paradigms is to involve non-expert-in-computing users in the development lifecycle. Also, an evaluation with real users has been carried out, where satisfactory results have been reached regarding the degree of usability of the tool. Obtained results helped corroborate the initial hypotheses stated in this research work.

Índice de Contenido

Agradecimientos.....	I
Resumen	III
Abstract	IV
Capítulo 1: Introducción.....	1
1.1. Definición del Problema	1
1.2. Hipótesis de Partida	2
1.3. Solución Propuesta.....	2
1.3.1. Objetivos Principales	3
1.3.2. Ventajas y Desventajas	4
1.3.3. Aspectos no Cubiertos y Limitaciones.....	5
1.4. Estructura del Trabajo.....	6
Capítulo 2: Estado del Arte	9
2.1. Introducción a la Arquitectura de la Información	9
2.2. Componentes de la Arquitectura de la Información.....	12
2.3. Entregables de la Arquitectura de la Información.....	13
2.4. Herramientas de Apoyo al Arquitecto de la Información	18
2.4.1. Herramientas de Apoyo a las Tareas del Arquitecto de la Información.....	18
2.4.2. Herramientas Software para la Elaboración de Prototipos.....	20
2.4.3. Análisis de Herramientas de Apoyo.....	25
Capítulo 3: Solución Propuesta.....	29
3.1. Lineamientos Generales.....	29
3.2. Diseño Arquitectónico de Solución Propuesta.....	30
3.3. Descripción de la Herramienta CASE InterArch	32
3.3.1. Descripción de las Tecnologías Utilizadas.....	37
3.4. Reglas de Transformación	39
3.5. Caso de Estudio.....	45
Capítulo 4: Experimentación	55
4.1. Técnicas de Evaluación Utilizadas.....	55
4.1.1. Descripción de la Tarea Propuesta para Evaluar la Herramienta InterArch.....	57
4.2. Resultados Obtenidos.....	60
4.2.1. Resultados de la Sesión Experimental	60
4.2.2. Resultados y Análisis de la Usabilidad	65
4.2.3. Fiabilidad de la Evaluación.....	72
4.2.4. Análisis Multivariable.....	72
4.3. Discusión de los Resultados.....	75
Capítulo 5: Conclusiones y Trabajo Futuro.....	79

5.1. Conclusiones	79
5.2. Trabajo Futuro	81
Bibliografía.....	85
Anexo A: Cuestionario de Usabilidad.....	89

Índice de Figuras

Figura 1: Sitio público de una empresa de consultoría (Morville & Rosenfeld, 2006)..	13
Figura 2: Maqueta de la página principal de un sitio Web de tarjetas de felicitación (Morville & Rosenfeld, 2006)	14
Figura 3: Hoja de cálculo para inventario de contenidos (Rise, 2005).....	16
Figura 4: Modelo de contenido general (Sachs, 2008).....	17
Figura 5: Matriz de metadatos para 3Com (Morville & Rosenfeld, 2006)	18
Figura 6: Detalle arquitectónico de la herramienta CASE InterArch.....	30
Figura 7: Interfaz de usuario de InterArch dividida en tres partes: A, B y C.....	32
Figura 8: Iconos de elementos de contenido	33
Figura 9: Iconos de elementos de enlace	33
Figura 11: Manipulación para enlazar elementos de contenido en el entorno de trabajo	34
Figura 12: Manipulación en bloque de los elementos en el diagrama.....	35
Figura 13: Creación de elementos visuales heredados	35
Figura 14: Paleta de colores de formato de línea.....	36
Figura 15: Formatos para almacenar el diagrama	37
Figura 16: Tecnologías utilizadas en la implementación de InterArch.....	38
Figura 17: Elementos relacionados de forma jerárquica utilizando InterArch.....	40
Figura 18: Diagrama de clases UML a partir de la transformación del diagrama conceptual de la Figura 17.....	42
Figura 19: Módulo de Reglas de Configuración de InterArch	43
Figura 20: Proceso de transformación de diagramas AI a diagramas UML.	45
Figura 21: Maqueta de productos en una tienda on-line	46
Figura 22: Diagrama de clases UML resultante del análisis de la AI del ejemplo propuesto	51
Figura 23: Adaptación del diagrama de clases UML con el módulo de Reglas de Configuración de InterArch.....	52
Figura 24: Adaptación del diagrama de clases UML con ArgoUML	53
Figura 25: Maqueta de registro de anuncios para productos de segunda mano.	58
Figura 26: Modelo de contenido de registro de anuncios para productos de segunda mano	59
Figura 27: Diagrama de clases UML de registro de anuncios para productos de segunda mano	60
Figura 28: Tiempo promedio empleado por los usuario en las tareas.....	64
Figura 29: Gráficos con las frecuencias para la categoría <i>Utilidad</i>	66
Figura 30: Gráficos con las frecuencias para la categoría <i>Facilidad de Uso</i>	69
Figura 31: Gráficos con las frecuencias para la categoría <i>Facilidad de Aprendizaje</i>	70
Figura 32: Gráficos con las frecuencias para la categoría <i>Satisfacción</i>	71

Índice

Figura 33: Grafico de frecuencia por variable.....	73
Figura 34: Correlación entre las variables <i>Facilidad de uso</i> y <i>Facilidad de aprendizaje</i>	74
Figura 35: Correlación entre las variables <i>Satisfacción</i> y <i>Utilidad</i>	75

Índice de Tablas

Tabla 1: Reseña y evolución sobre la Arquitectura de la Información (Erlin, Yunura, & Azizah, 2008)	10
Tabla 2: Herramientas de apoyo a las tareas del Arquitecto de la Información.....	20
Tabla 3: Herramientas de escritorio para la elaboración de prototipos	23
Tabla 4: Herramientas en línea para la elaboración de prototipos	25
Tabla 5: Ventajas y desventajas de herramientas software para la elaboración de prototipos.....	27

Capítulo 1: Introducción

En el presente capítulo, se plantean los fundamentos principales que dan soporte a la investigación llevada a cabo. En la sección 1.1, se detalla la problemática abordada. En la sección 1.2, se señalan las hipótesis de partida que conducen el trabajo en todas sus fases, las cuales también son ratificadas a lo largo de la presente investigación. En la sección 1.3, se propone una solución conforme a la problemática planteada y las hipótesis descritas. Conjuntamente, se indican los objetivos principales que persigue el trabajo, como también sus ventajas y desventajas, aspectos no cubiertos y limitaciones.

1.1. *Definición del Problema*

En nuestra época tecnológica actual, la Arquitectura de la Información (AI) es un paradigma creciente que se está incorporando paulatinamente en la mayoría de los proyectos Software. La AI se define como la ciencia de la estructuración, organización y gestión de la información donde el arte del etiquetado, el hallazgo y la facilidad de uso son también importantes (Erlin et al., 2008). La AI cubre indistintamente aspectos tales como el diseño de sitios Web, interfaces de dispositivos móviles, interfaces de máquinas dispensadoras, interfaces de juegos electrónicos, etc. Su principal objetivo es facilitar al máximo los procesos de comprensión y asimilación de la información, así como las tareas que ejecutan los usuarios en un espacio de información definido (Morville & Rosenfeld, 2006). La forma en que las personas interactúan con los entornos ricos en información digital está directamente influenciada por la Arquitectura de Información (Elaine, 2002).

El Arquitecto de la Información es la persona quien lleva a cabo las tareas de descripción y estructuración de la información con el fin de que el usuario pueda recuperarla, manejarla y comprenderla de forma sencilla (Yusef & Francisco, 2003). El Arquitecto de la Información trabaja sobre las fases tempranas del Software interactivo, principalmente en aplicaciones Web, tratando de encontrar un puente que una los conocimientos conceptuales que plantean los usuarios, dentro del espacio del problema, y la información de diseño que luego necesitarán los ingenieros de la solución (Ingenieros del Software), que se encargarán de diseñar la interfaz Web final.

Sin embargo, muchas veces, los roles del Arquitecto de la Información y del Ingeniero del Software no suelen coincidir, ya que el Arquitecto de la Información puede tener un perfil menos técnico y más orientado a tareas conceptuales o relacionadas con la ergonomía, lo cual hace necesario cierto grado de interoperabilidad y consonancia entre el *output* que genera el Arquitecto de la Información y el *input* que espera el Ingeniero del Software para comenzar con el diseño de la aplicación Web interactiva. Si esta continuidad se pudiera hacer de forma automática, el tiempo y esfuerzo de realización del proyecto se reduciría, permitiendo a cada experto concentrarse en su trabajo en función de sus conocimientos y minimizando el tiempo de trasiego de información entre uno y otro profesional (Macías & Castells, 2007).

La automatización de las tareas comentadas anteriormente implica el uso de herramientas específicas (Pérez & Codina, 2010). No obstante, las herramientas existentes más

comunes no disponen de funcionalidades que permitan, de forma directa o automática, conectar el *output* del Arquitecto de la Información y el *input* que espera el Ingeniero del Software. Esto se intenta resolver generando distintos formatos exportables de imágenes y HTML. No obstante, esta solución hace desaparecer detalles semánticos importantes relativos al análisis, y dificulta la interoperabilidad y seguimiento si se utilizan herramientas posteriores más precisas.

Por otro lado, ninguna de las herramientas existentes, generan de forma automática o semi-automática información que permita obtener, a partir de los contenidos de información, los diagramas de clase y elementos de contenido que puedan ser utilizados por analistas o ingenieros software para dar continuidad al resto de fases del ciclo de vida del proyecto.

1.2. *Hipótesis de Partida*

El presente trabajo se apoya en las siguientes hipótesis de partida:

- **H_1 :** Es posible generar información de análisis y diseño procesable por analistas e ingenieros software a partir de una representación inicial conceptual de los contenidos por parte del Arquitecto de la Información.
- **H_2 :** Es posible sistematizar el proceso descrito en la hipótesis **H_1** mediante la construcción de una herramienta fácil de usar y de aprender por el Arquitecto de la Información, que permita generar automáticamente información de análisis y diseño para analistas e ingenieros software a partir de las descripciones conceptuales que el Arquitecto de la Información crea, uniendo puentes entre las fases más iniciales del proyecto, dentro del dominio del problema, y las fases previas al desarrollo, más cercanas al dominio de la solución.

1.3. *Solución Propuesta*

De acuerdo a la definición del problema expuesta en la sección 1.1 y las hipótesis de partida enumeradas en la sección 1.2, se plantea como solución idear una representación que permita al Arquitecto de la Información representar información conceptual de contenidos, y que ésta sea transformable, mediante reglas específicas, en diagramas UML que den soporte al análisis y diseño de los aspectos funcionales para gestionar dichos contenidos. La solución propuesta en este trabajo se basa en dos pilares fundamentales:

1. El primero, consiste en idear un formalismo que permita representar el conocimiento proveniente de las definiciones conceptuales del Arquitecto de la Información, así como su posterior transformación, utilizando reglas explícitas, en información de análisis y diseño para el Ingeniero del Software. Este formalismo permitirá corroborar la hipótesis **H_1** .
2. La segunda parte, consiste en idear una herramienta CASE de soporte que permita validar dicho formalismo, la cual será asimismo evaluada en cuanto a su usabilidad, permitiendo así corroborar la hipótesis **H_2** .

La solución propuesta está basada en el EUD o paradigma del Desarrollo por el Usuario Final (del inglés End-User Development) (Klann, 2003; Macías & Castells, 2007). El EUD incentiva la creación de artefactos software que permitan a usuarios no expertos en informática crear o personalizar aplicaciones fácilmente. Además, el trabajo se acerca a paradigmas relacionados como el EUSE (End-User Software Engineering), cuyo objetivo es involucrar a usuarios no expertos en computación, como sucede muchas veces con los Arquitectos de la Información, en aspectos concretos del desarrollo de un proyecto informático relacionados con el análisis y desarrollo de Software (Lieberman et al., 2006).

1.3.1. Objetivos Principales

El objetivo de este trabajo es abordar la problemática mencionada en la definición del problema (Sección 1.1) y unir puentes entre las tareas del Arquitecto de la Información y las del Ingeniero del Software, proponiendo para ello una herramienta CASE que permita a los expertos en el dominio del problema concentrarse en el análisis de contenidos y navegación, generando la herramienta de forma automática las clases de contenido para el Ingeniero del Software en formato UML y concretando además, de forma inmediata, elementos del dominio de la solución.

La investigación plantea los siguientes objetivos principales:

- Proporcionar facilidades al Arquitecto de la Información para elaborar sus productos de forma habitual y concentrarse en sus tareas de análisis conceptual dentro del dominio del problema.
- Proporcionar información de análisis y diseño para analistas e ingenieros software, de forma automática, que garantice la generación y gestión del conocimiento dentro del dominio de la solución.
- Validar la generación de información de análisis y diseño mediante la construcción de una herramienta CASE que satisfaga los requisitos relacionados con la utilidad, satisfacción, facilidad de uso y aprendizaje.
- Aumentar la automatización e interoperabilidad entre las definiciones conceptuales de los contenidos de una aplicación Web (Arquitectura de la Información) y los elementos de análisis y diseño procesables por analistas e ingenieros software.
- Minimizar en términos de coste, tiempo y esfuerzo, el desarrollo de aplicaciones Web interactivas.
- Aumentar la automatización e interoperabilidad en el desarrollo de aplicaciones Web interactivas.

Para la consecución de los objetivos principales, se llevarán a cabo una serie hitos a lo largo del trabajo:

- Estudiar y analizar los diferentes tipos de modelos de contenido elaborados por los Arquitectos de la Información (estructuras de elementos de contenido de un sitio Web).
- Identificar las características de los elementos visuales, utilizados por los Arquitectos de la Información, que tiene su correspondencia con los diagramas UML diseñados por analistas e ingenieros software, ideando un formalismo que permita lograr dichas representaciones y transformaciones entre distintos modelos.
- Desarrollar un prototipo avanzado de la herramienta CASE de soporte, tomando en cuenta las consideraciones del análisis competitivo y comparativo realizado sobre otras herramientas del contexto.
- Realizar un test con usuarios para evaluar la usabilidad de la herramienta CASE construida, de forma que se puedan corroborar finalmente las hipótesis de partida de este trabajo.

1.3.2. Ventajas y Desventajas

Conforme a la solución propuesta y a los objetivos planteados, a continuación se presentan algunas de las ventajas más importantes que esta investigación promueve:

1. *Facilita a usuarios, no expertos en computación, concretar aspectos del desarrollo de un proyecto informático.*

La herramienta CASE es fácil de usar y aprender, y permite a Arquitectos de la Información elaborar sus productos de forma habitual. Adicionalmente, incorpora una capa de interpretación, capaz de generar información útil para el desarrollo de un proyecto informático, relacionado con el análisis y desarrollo de Software.

2. *Aumenta la eficacia de las actividades del Arquitecto de la Información*

La herramienta CASE provee un entorno de trabajo visual para el Arquitecto de la Información, la cual cuenta con elementos visuales para elaborar diagramas en un entorno funcional para manipular e interactuar con los elementos visuales a través de distintas opciones de formato, configuración y edición. Esto permite al Arquitecto de la Información concentrarse en el análisis de contenidos y en los aspectos relevantes de sus actividades.

3. *Permite una continuidad entre la información de análisis conceptual de la AI y la información de análisis y diseño que necesitan los analistas informáticos.*

La herramienta CASE permite convertir la información de análisis conceptual de la Arquitectura de la Información en información análisis y diseño para analistas e ingenieros software, facilitando de esta forma la continuidad de la información entre *output* del Arquitecto de la Información y *input* esperado por analistas e ingenieros software para llevar a cabo el modelado de la aplicación.

4. *Formalismo para representar información no funcional y transformarla en descripciones UML.*

La investigación propone además un formalismo para representar el conocimiento de las definiciones conceptuales del Arquitecto de la Información (*modelos de contenidos*), el cual se lleva a cabo mediante la propuesta de un conjunto de reglas de transformación que se aplican a la estructura y jerarquía de los diagramas de *modelado de contenido* elaborados por el Arquitecto de la Información. Lo anterior también incluye su posterior transformación en información de análisis y diseño para el Ingeniero del Software. Esta información es customizable, lo cual permite al Ingeniero del Software ajustar la generación de esta información.

1.3.3. Aspectos no Cubiertos y Limitaciones

En esta sección se indican algunos de los aspectos no cubiertos y limitaciones del trabajo que, si bien no han sido implementados por razones de extensión y tiempo de realización del trabajo, serán tenidos en cuenta para posibles ampliaciones y trabajo futuro:

1. *No se proporciona soporte para todos los productos entregables de la Arquitectura de la Información*

La herramienta CASE se concentra en los *modelos de contenidos* elaborados por la Arquitectura de la Información. Esto sólo corresponde a uno de los productos entregables de la Arquitectura de la Información, no considerándose de momento otros tales como los *blueprints* (planos), los *wireframes* (maquetas) y los vocabularios controlados. Esto es debido a que, principalmente, no se ha encontrado una correspondencia directa que permita generar información de forma automática con otros productos requeridos por Analistas o Ingenieros del Software. Sin embargo, el tratamiento de los *modelos de contenidos* corresponde a un avance importante en cuanto a la automatización e interoperabilidad entre las definiciones conceptuales de los contenidos de una aplicación Web y los elementos de análisis y diseño procesables por analistas e ingenieros software.

2. *El formalismo propuesto no permite la incorporación ni modificación de las reglas de transformación.*

El formalismo propuesto para representar el conocimiento de las definiciones conceptuales del Arquitecto de la Información corresponde a un conjunto estático de reglas de transformación que se traduce en criterios fijos que son aplicados a todos los *modelos de contenidos* elaborados por Arquitectos de la Información. Sin embargo, aún cuando esta investigación no incluye la opción de incorporar ni modificar estas reglas de transformación, el conjunto actual permiten un variado catálogo de transformaciones que permite al Ingeniero del Software continuar el trabajo con otras herramientas más avanzadas de modelado mediante la exportación del diagrama resultante.

3. *No incluye módulo para visualizar diagramas de clases UML*

La herramienta CASE genera los diagramas de clases UML en formato XMI, este archivo requiere ser importado por una herramienta de diagramación que soporte el formato de representación XMI. Actualmente, no se cuenta con un módulo integrado dentro de la herramienta que permita realizar la visualización de los diagramas de clases UML. No obstante, el soporte de diagramas de clases UML en formato XMI está ampliamente extendido, y son muchas las aplicaciones que lo soportan sin problemas. Ejemplos de tales herramientas son ArgoUML, StarUML, BOUML, Visual Paradigm, Circa y Mia-Generation, entre otras, que cuentan con un entorno de visualización UML avanzado. En este sentido, la visualización de clases UML es un problema que no se ha tratado en este trabajo, ya que el objetivo principal ha sido concentrarse más en la parte formal e ingenieril del proceso mediante la definición del formalismo y la validación mediante la herramienta CASE correspondiente que da soporte al proceso de creación y transformación de modelos.

4. *No se provee ingeniería inversa para la generación de modelos de contenido.*

La herramienta CASE no cuenta con una funcionalidad para generar *modelos de contenido* a partir de otras fuentes de información, como podría ser de la lectura y análisis de los diagramas de clases UML elaborados por analistas e ingenieros software. Aún cuando la herramienta no cuenta con esta generación automática de información, la herramienta provee un entorno de trabajo eficiente para que Arquitectos de la Información puedan crear sus *modelos de contenido* de forma normal. No obstante, la ingeniería inversa es un aspecto importante a considerar en futuras mejoras de la herramienta, la cual esta considera dentro de los trabajos futuros de la investigación.

1.4. ***Estructura del Trabajo***

El trabajo llevado a cabo se ha estructurado de la siguiente manera:

- En el Capítulo 1 se han presentado los fundamentos principales de la investigación. Se ha descrito la definición del problema que afronta la investigación, junto con la solución propuesta conforme a la problemática e hipótesis planteadas. Además, se indicaron los objetivos, ventajas y desventajas que presenta la solución propuesta. Finalmente, se han descrito los aspectos no cubiertos y limitaciones de la investigación.
- En el Capítulo 2 se presentan los principales elementos conceptuales de la Arquitectura de la Información, aportando diferentes definiciones, importancia y los límites con las disciplinas relacionadas. También, se presentan los componentes de la Arquitectura de la Información (organización, etiquetado, navegación y búsqueda). En último lugar, se describen las diferentes herramientas más comúnmente utilizadas por el Arquitecto de la Información, junto con un análisis de las herramientas descritas que permita comparar las soluciones existentes con la elaborada en este trabajo.

Introducción

- En el Capítulo 3 se presenta en detalle la herramienta y el formalismo ideado. Se plantean los componentes principales de la arquitectura propuesta, describiendo los distintos procesos que son responsable de la gestión y transformación de modelos de contenido dentro de un entorno visual orientado al Arquitecto de la Información. Finalmente, a modo de ejemplo, se presenta un caso de estudio para describir en detalle el funcionamiento de la herramienta.
- En el Capítulo 4 se describe la experimentación que se realizó para medir la usabilidad de la herramienta. Por un lado, se introducen las diferentes técnicas de evaluación seleccionadas para evaluar la usabilidad de la herramienta. Por otro lado, se da una descripción detallada de todo el proceso experimental llevado a cabo, seguido de un análisis y discusión sobre los resultados obtenidos.
- En el Capítulo 5 se describen las conclusiones relacionadas con la investigación presentada en este trabajo. También, se presentan las actividades que se proponen como trabajo futuro así como mejoras propuestas, las cuales se han redactado en base a los resultados obtenidos de la evaluación de la herramienta y de las observaciones realizadas por los usuarios en el cuestionario de usabilidad.

Capítulo 2: Estado del Arte

En este capítulo se presentan los elementos conceptuales de la Arquitectura de la Información, así como las diferentes herramientas empleadas por Arquitectos de la Información. En la sección 2.1, se introduce el término de Arquitectura de la Información, indicando su importancia, atributos y límites. En la sección 2.2, se describen los componentes en los que se basa la Arquitectura de la Información. En la sección 2.3, se detallan las diferentes representaciones visuales que desarrollan los Arquitectos de la Información. En la sección 2.4, se realiza una revisión de las distintas herramientas más comúnmente utilizadas por el Arquitecto de la Información. Finalmente, se efectúa un análisis de las herramientas descritas, reportando las conclusiones resultantes, ventajas y desventajas en cada caso.

2.1. *Introducción a la Arquitectura de la Información*

Desde muy antiguo, los humanos se vienen enfrentando a la necesidad de organizar y poner a su disposición grandes volúmenes de información, de tal forma que sea posible acceder a ella y consumirla de una manera eficiente en cada caso. Hoy en día, con el surgimiento de la Internet y la masificación de información en los sitios Web, nos enfrentamos a la misma necesidad pero con el desafío de satisfacerla de un modo distinto. Se ha restado la exigencia de los espacios físicos para su almacenamiento y disposición, y se ha trasladado a nuevos espacios virtuales, lo que ha exigido nuevos paradigmas para su organización, estructuración, navegación y disposición. Ante este nuevo escenario, surge la disciplina de la Arquitectura de la Información como respuesta a la necesidad diseñar, analizar, organizar, disponer y estructurar la información en espacios virtuales.

Erlin et al., realizaron en 2008 un estudio sobre la reseña y evolución de la AI en la era de la información (Erlin et al., 2008), en el cual se elaboro una tabla con un análisis de los diferentes atributos que se han utilizado para definir la AI. En la Tabla 1, se pueden observar los distintos atributos empleados por diversos autores para señalar qué es y en qué consiste la AI. Entre los atributos más comúnmente aceptados se encuentran *estructurar, organizar, etiquetar, arte y ciencia*, promovidos principalmente por Morville & Rosenfeld en su famoso libro “Information Architecture for the World Wide Web” (Morville & Rosenfeld, 2006). Otros atributos conocidos son *negocios/empresa*, utilizado por Subramaniam y Garrett, el atributo *encontrar/uso*, utilizado por Garrett, IAI y Morville & Rosenfeld, y el atributo *gestión*, utilizado por Evernden, Morrogh y NCER. Otros atributos que han sido utilizados para definir la AI se corresponden con *mapa de alto nivel, infraestructura, teoría, principios, navegación, indexado, modelo, concepto, ambiente, diseño de procesos, colección de datos, intercambio, estándares, propósitos, acción, grupo y terminología*. Como se puede observar, existe una gran variedad de atributos empleados para intentar dar a conocer y describir la AI, lo cual dificulta poder contar con una definición exacta de la AI. A este hecho, habría que agregarle que la AI es una disciplina nueva con una gran cantidad de conceptos y áreas

relacionadas, lo que dificulta aún más la tarea de realizar una descripción completa y exhaustiva de la AI.

Definición	Atributos de definición									
	Año	Estructurar	Organizar	Etiquetar	Arte	Ciencia	Negocios / Empresa	Encontrar / uso	Gestión	Otros
Morville & Rosenfeld	2006	X	X	X	X	X		X	X	
IAI	2008		X	X	X	X		X		
Dickson & Wetherbe	1989						X			Mapa alto nivel
Carter	1999						X			Infraestructura
Evernden	2003	X							X	Teoría, principios
Subramaniam	2004	X	X	X		X				Navegación, indexado
Wikipedia	2008					X				Modelo, concepto
Morrogh	2008								X	Ambiente, diseño de procesos
NCER	2005		X						X	Colección de datos, intercambio, estándares
Garrett	2002				X			X		Propósitos, acción
Barker	2005									Grupo, navegación, terminología

Tabla 1: Reseña y evolución sobre la Arquitectura de la Información (**Erlin, Yunura, & Azizah, 2008**)

Para hacer frente a la dificultad de poder definir la AI, existe una forma eficiente que ayuda a obtener una mejor definición de las cosas, que es a través de la identificación de sus diferentes límites. En el caso particular de la AI, es posible trazar algunos límites entre la Arquitectura de la Información y el resto de disciplinas relacionadas (Morville & Rosenfeld, 2006), que se describen a continuación, las cuales permiten identificar algunas labores que difieren con las del Arquitecto de la Información:

- *Diseño gráfico:* Comprende los aspectos de comunicación visual, desde el diseño de los logos y las identidades corporativas a la disposición de cada página.

- *Diseño de interacción:* Se centraliza en el comportamiento que tienen los usuarios al realizar las tareas y los procesos que se encuentran a nivel de la interfaz en el software y en los sistemas de información. Concentrándose en ayudar a los usuarios a conseguir sus objetivos y completar tareas.
- *Ingeniería de usabilidad:* Se enfoca en aplicar el rigor del método científico a la investigación, pruebas y análisis de usuarios. Frecuentemente, se dedica a probar todos los aspectos de la experiencia de usuario, incluyendo la Arquitectura de la Información y el diseño gráfico.
- *Diseño de la experiencia:* Es un término que engloba la Arquitectura de la Información, la ingeniería de usabilidad, el diseño gráfico y el diseño de interacción como componentes de la experiencia de usuario.
- *Desarrollo de software:* No se suele confundir el desarrollo de software con la Arquitectura de la Información, pero estos dos campos están muy relacionados entre sí. Los Arquitectos de la Información dependen de los desarrolladores para llevar sus ideas a la práctica. Los desarrolladores permiten entender qué es y qué no es posible.
- *Gestión del contenido:* La Arquitectura de la Información describe una foto fija o una vista espacial del sistema de información, mientras que la gestión del contenido describe una vista temporal al mostrar cómo la información debe fluir hacia, alrededor y desde ese mismo sistema con el tiempo.
- *Gestión del conocimiento:* Su objetivo es desarrollar herramientas, políticas e incentivos para animar a la gente a compartir lo que sabe, y la Arquitectura de la Información se centran en hacer más accesible lo que ya ha sido capturado.

Aunque es posible especificar algunas diferencias, como hemos visto antes, entre la Arquitectura de la Información y las disciplinas relacionadas, existen algunas actividades comunes que requieren de una colaboración interdisciplinaria para poder ser llevadas a término de forma exitosa. Es precisamente esta interacción entre actividades de la AI y el resto de disciplinas relacionadas lo que permite producir sitios Web de calidad.

Además de presentar el concepto de la AI y concretar algunas de sus diferencias con respecto a otras disciplinas relacionadas, también se hace necesario identificar la importancia de la Arquitectura de la Información en el contexto de una organización. Para conocer esta importancia, se deben considerar los siguientes costes (Morville & Rosenfeld, 2006):

- *El coste de encontrar información:* Se debe considerar cuánto le cuesta a una compañía que un trabajador pierda cierta cantidad de tiempo para encontrar una determinada información.

- *El coste de no encontrar información:* El no encontrar la información puede llevar a la organización a tomar decisiones equivocadas, así como la pérdida de alguna oportunidad de negocio dado que el cliente no pueda encontrar el producto que estaba buscando.
- *El coste de la educación:* El coste de capacitar a los clientes para informar respecto a nuevos productos y/o servicios.
- *El coste de construcción:* El coste de rediseñar nuevamente el sitio Web dado que el actual no facilita la buscabilidad.

Estos costes sólo corresponden a algunos a considerar en el momento de calcular el ROI (retorno de la inversión) en Arquitectura de la Información; el desafío está precisamente en comprender cuáles son y comunicarlos tan clara y directamente como sea posible.

2.2. Componentes de la Arquitectura de la Información

Una vez ya presentado el término Arquitectura de la Información a través de sus límites, importancia y atributos, en esta sección se presentan una serie de sistemas que ayudan a definir y estructurar la AI. A estos sistemas se les denomina componentes de la Arquitectura de la Información:

a) Sistema de Organización

El sistema de organización permite dividir y clasificar un ámbito concreto en grupos de entidades, utilizando criterios similares entre las entidades que la conforman y acorde a cómo las personas comprenden este ámbito. Los Arquitectos de la Información deben organizar la información para que las personas puedan encontrar las respuestas correctas a sus preguntas respecto a sus necesidades de información.

b) Sistema de Etiquetado

Se utilizan etiquetas para representar fragmentos de mayor tamaño de información en los sitios Web. Por ejemplo, *Contacto* puede ser una etiqueta que representa una parte de la información, como puede ser un nombre de contacto, una dirección, un teléfono, un fax, o un correo electrónico. El objetivo de una etiqueta es comunicar información de manera eficiente. Los sitios Web utilizan dos tipos de formatos para etiquetar: el icono y el textual. Los sistemas de etiquetado con formato icono utilizan imágenes como etiquetas para representar contenidos, y los sistemas de etiquetado con formato textual utilizan términos o palabras de un lenguaje como etiquetas para representar contenidos.

c) Sistema de Navegación

El sistema de navegación se corresponde con estructuras arquitectónicas que ordenan y agrupan los contenidos de una página Web bajo categorías que forman una clasificación (Gutierrez, 2010). El sistema de navegación se utiliza para trazar el rumbo, determinar la posición y para encontrar el camino de regreso. El sistema de navegación tiene

semejanzas con el sistema de organización, pues los dos sistemas coinciden en ordenar y agrupar los contenidos de un sitio Web bajo unas categorías que forman una clasificación. Sin embargo, el sistema de navegación difiere del sistema de organización, ya que el primero permite identificar las relaciones entre los contenidos del sitio Web y de la página del sitio Web que en un momento determinado el usuario esté visitando.

2.3. *Entregables de la Arquitectura de la Información*

Los Arquitectos de la Información dependen de las representaciones visuales para comunicar lo que realmente hacen. Entre los esquemas utilizados por los Arquitectos de Información están los planos (*blueprints*) y maquetas (*wireframes*). Ambos, se centran principalmente en la estructura del contenido de un sitio Web más que en su contenido semántico. Planos y maquetas permiten representar la estructura, el movimiento, el flujo y las relaciones entre el contenido. Otros entregables de la AI corresponden a los modelos de contenidos, asignación e inventario de contenidos y los vocabularios controlado. A continuación se da una descripción de dichos productos:

a) Planos (*Blueprints*)

Los planos muestran las relaciones entre las páginas y otros componentes de datos, y se pueden utilizar para describir los sistemas de organización, navegación y etiquetado. Los planos se representan a través de cajas relacionadas con flechas y textos, con la finalidad de mostrar cómo los elementos del sitio Web se agrupan y cómo se vinculan o se relacionan entre sí (Shiple, 1998).

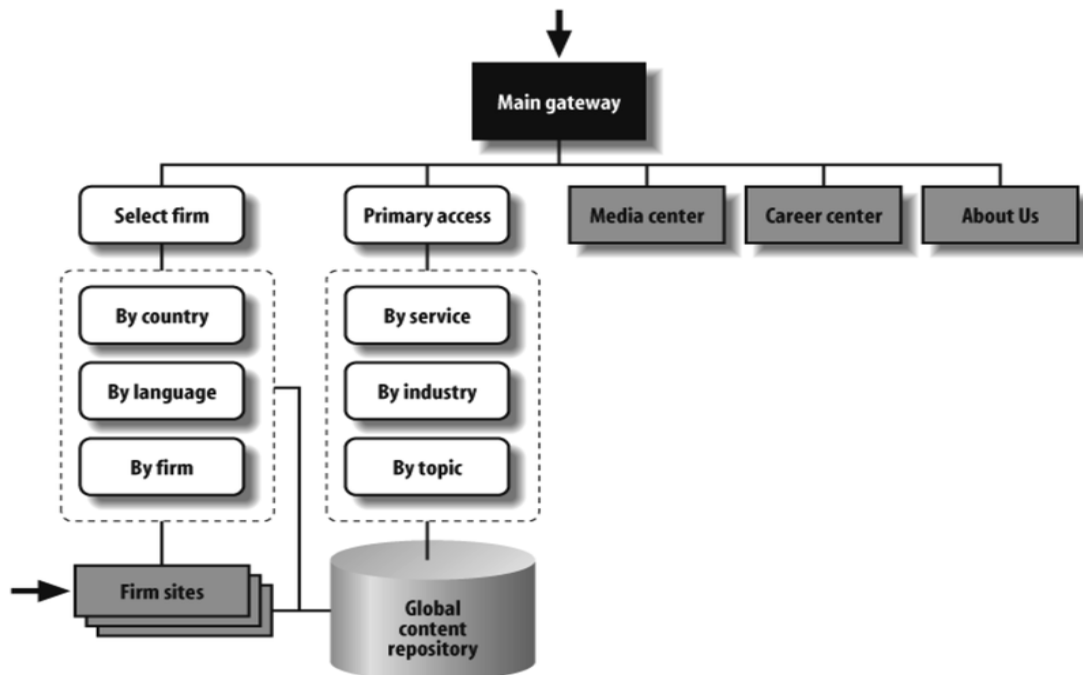


Figura 1: Sitio público de una empresa de consultoría (Morville & Rosenfeld, 2006)

Durante la fase de diseño, los planos son útiles para explorar los primeros enfoques y esquemas de organización. Estos planos se pueden realizar a mano, pero también se

pueden usar diagramas (ir a la sección 2.4.2 para obtener mayor detalle de herramientas de diagramas para el Arquitecto de la Información).

La Figura 1 corresponde a un plano que muestra gráficamente cómo la información está organizada en un sitio Web de una empresa de consultoría. Como se puede observar, existen niveles principales (Select firm, Primary access, Media center y About Us) que muestran como están organizados los contenidos del sitio Web y los niveles secundarios como By country, By language y By firm muestran información de navegación relacionada con uno de los elementos principales (Select firm). También, se muestran aspectos relacionados con el modo de acceso al repositorio de contenido global que tienen las distintas firmas de la consultora, esto se pone de manifiesto en la incorporación de los elementos Firm sites y Global content repository en el plano.

b) Maquetas (Wireframes)

Las maquetas permiten describir cómo una página individual o plantilla deben verse desde una perspectiva arquitectónica. Elaborar estas maquetas conlleva generar plantillas de contenido jerarquizado caracterizadas por la simpleza, sin la inclusión de aspectos visuales como, por ejemplo, colores, familias tipográficas, logos, fotos, etcétera. De hecho, la idea es que la maqueta se realice en escala de grises. En general, se puede decir que las maquetas representan la intersección entre la estructura de la Información de un sitio Web, su diseño visual y la información subyacente (Gutierrez, 2010).

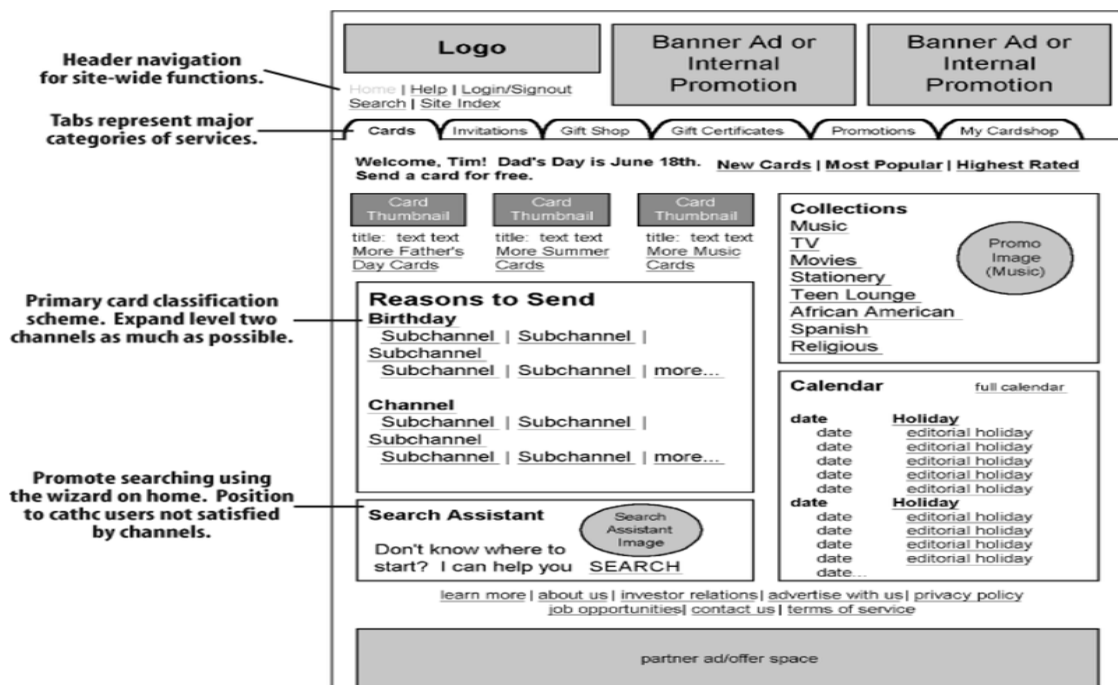


Figura 2: Maqueta de la página principal de un sitio Web de tarjetas de felicitación (Morville & Rosenfeld, 2006)

Las maquetas ayudan a que el Arquitecto de la Información pueda decidir cómo agrupar los componentes de contenido, cómo ordenarlos y qué grupos de componentes tienen prioridad. La Figura 2, muestra una maqueta de la página principal de un sitio Web de tarjetas de felicitación, en donde el Arquitecto de la Información ha determinado que *Razones para enviar* (Reasons to Send) es de una prioridad más alta que el *Asistente de búsqueda* (Search Assistant). Esta prioridad se pone de manifiesto por la posición prominente del contenido y el uso de un tipo de letra más grande para su partida. También, se han definido el tipo de navegación que contendrá el encabezado y la ubicación de la información del logo y de los anunciantes del sitio Web.

- **Tipos de Maquetas**

Las maquetas pueden ser construidas de muchas formas y tamaños. El nivel de fidelidad puede variar para adaptarse a los propósitos del Arquitecto de Información, y de acuerdo a su complejidad se clasifican en diferentes niveles: maquetas de fidelidad baja, media y alta. Las maquetas de *fidelidad baja* son las más simples, y sobre ellas se realiza un bosquejo en blanco y negro para determinar en términos generales los espacios que ocupará cada elemento en el sitio Web; no hay elementos gráficos ni contenido real. Las maquetas de *fidelidad media* permiten introducir algunos aspectos de contenido, diseño y navegación del sitio Web. Y por último, las maquetas de *fidelidad alta*, son representaciones avanzadas que incluyen mayor detalle visual, como imágenes, colores, tipografías, texto e iconos a utilizar, las cuales presentan una aproximación cercana a lo que finalmente será el sitio Web final. Las maquetas de fidelidad alta tienen las siguientes ventajas y desventajas:

Ventajas:

- La incorporación del contenido, el color e imágenes en las maquetas del sitio Web, ayudan a mejorar la comunicación entre los clientes y el equipo de trabajo.
- Mediante la simulación del ancho y el tamaño de fuente del sitio Web, la maqueta obliga a reconocer las limitaciones de una página HTML.

Desventajas:

- Una mayor fidelidad requiere un mayor esfuerzo. Se necesita mucho tiempo para diseñar una maqueta detallada. Esto puede ralentizar el proceso y aumentar los costes.
- Al integrar los elementos visuales y de contenido en un diseño estructurado, el enfoque puede cambiar prematuramente desde Arquitectura de la Información a aspectos relacionados con la interfaz y el diseño visual de un sitio Web.

A pesar de todo, considerando ventajas y desventajas, las maquetas son una herramienta eficiente para la comunicación y colaboración durante el proceso de diseño de la Arquitectura de la Información.

c) Asignación e Inventario de Contenidos

El proceso de asignación de contenido implica la combinación del contenido existente en trozos de contenido que son útiles para su inclusión en el sitio Web. Un trozo de contenido no es necesariamente una oración, un párrafo o una página. Más bien, es la parte más fina de los contenidos que merece o requiere un tratamiento individual.

Un subproducto del proceso de asignación de contenido es un inventario de contenidos que describe el contenido disponible y dónde se puede encontrar. Los sitios Web voluminosos pueden requerir una solución de gestión de contenido con tecnología de base de datos para manejar grandes colecciones de contenido, mientras que para sitios Web simples es posible que sólo se necesite una hoja de cálculo. La Figura 3 muestra una hoja de cálculo para un inventario de contenidos creada por Sarah Rise de Seneb Consulting (Rise, 2005), en la cual los contenidos se encuentran categorizados de acuerdo a los diferentes niveles de las páginas Web, y la dirección URL de su ubicación:

1	Project Name	Final Deliverable	Example																
2	Page ID Number	Level 1 Page Title	Level 2 Page Title	Level 3 Page Title	Level 4 Page Title	Level 5 Page Title	Level 6 Page Title	Level 7 Page Title	Current URL	Format	Content Type	3rd Party Content	Login Required	Template	Page Title	Local Nav	Metadata Element	Metadata Element	
3																			
4																			
5		Yellow Fill: wiki page							Green Text: Cross link (represented elsewhere on spreadsheet or on site)										
6		Rose Fill: form							Red Text: outstanding questions										
7		Purple Fill: AIIA site																	
8																			
9																			
10																			
11		Initiatives							http://www.aifa.org/pg/initiatives.php										
12		volunteer@aifa.org																	
13		Public Initiatives																	
14		Education Curriculum							http://aifa.org/pg/curriculum.php										
15		What is the Educational Curriculum?																	
16		Benefits																	
17		Get Involved																	
18		Volunteer now							http://aifa.org/pg/how_it_works.php										
19		IA Curriculum page							http://www.aifa.org/projects/wiki/index.php?page=IACurriculum										
20		IA Education mailing list							http://lists.ibiblio.org/mailman/listinfo/aifa-education										
21		University of Denver, College of Education							http://www.du.edu/~whing/LS4700IA/syllabus.htm										
22		University of Michigan, School of Information							http://semanticsstudies.com/publications/iasj.htm										
23		UCLA, School of Information Studies							http://is.geis.ucla.edu/courses/279/syllabus.htm										
24		University of Alberta							http://www.ualberta.ca/~liven/34outline.htm										
25		University of Baltimore							http://raven.ubalt.edu/classes/ia630.185_p03/syllabus.htm										
26		Pratt Institute							http://www.designweenie.com/manual/viaci/										
27		Translating Information Architecture							http://aifa.org/translations/										
28		Metrics for IA							http://www.aifa.org/projects/wiki/index.php?page=MetricsForIA										
29		IA Local Groups							http://www.aifa.org/projects/wiki/index.php?page=LocalGroups										
30		Member Initiatives																	
31		Job Board							http://www.aifa.org/projects/wiki/index.php?page=JobBoard										
32		Member Directory							http://www.aifa.org/projects/wiki/index.php?page=MemberDirectory										
33		IA Tools							http://www.aifa.org/projects/wiki/index.php?page=ToolsProject										
34		Get Involved																	
35		Learn More																	

Figura 3: Hoja de cálculo para inventario de contenidos (Rise, 2005)

d) Modelos de Contenido

Los modelos de contenido son *micro* arquitecturas de información formadas por pequeños trozos de contenido relacionados entre sí. Los modelos de contenido se basan en conjuntos coherentes de elementos y conexiones lógicas entre ellos. Un ejemplo de modelo de contenido puede ser una receta de cocina, donde sus elementos se corresponden con una lista de ingredientes, el título de la receta, y los pasos para cocinar, los cuales se utilizan para crear el modelo de contenido e identificar oportunidades para la reutilización del contenido y controlar la granularidad de la información (Sachs, 2008).

La Figura 4 muestra un ejemplo de modelo de que describe el contenido general para una revista de publicaciones. Este modelo de contenido contiene un elemento principal General content, el cual está compuesto por cinco elementos descendientes

(Object ID, Textual, Reference, Publishing, Categorisation y Discovery), donde cada uno de estos elementos se descompone a su vez en elementos descendientes que ayudan a describirlos. Por ejemplo, el elemento secundario Publishing se describe en base a sus seis elementos que descienden de él (Content Owner, Content Author, Last Update, Publish date, Review Date y Retire Date), los cuales permiten indicar las diferentes fechas y los tipos de contenidos que contiene la publicación.

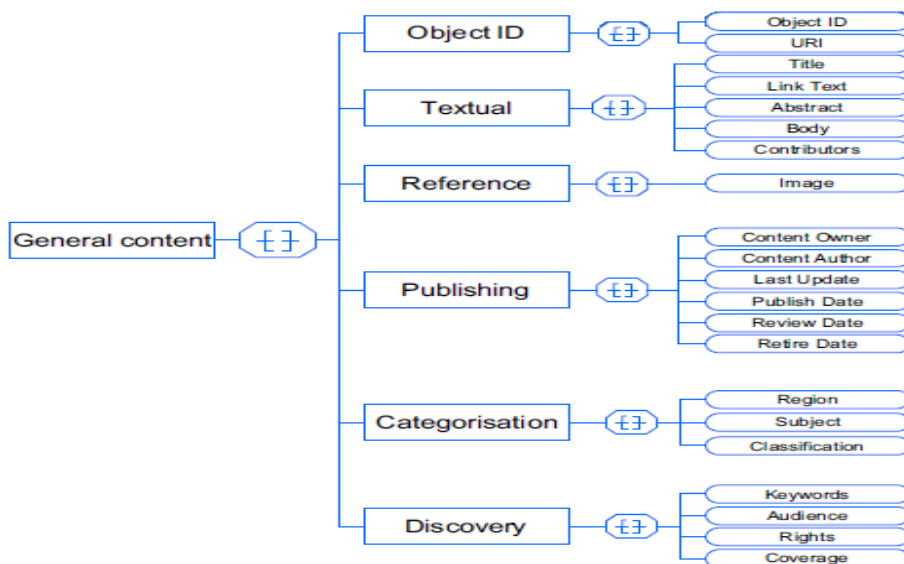


Figura 4: Modelo de contenido general (Sachs, 2008)

e) Vocabulario Controlado

La comunicación más efectiva se produce cuando todas las partes involucradas están de acuerdo sobre el significado de los términos utilizados. En consecuencia, encontrar las palabras adecuadas para transmitir el mensaje del sitio Web puede ser una de las partes más difíciles de su desarrollo. La concreción de lo anterior, es lo que se conoce como vocabulario controlado, el cual comprende una lista o índice de términos que establece relaciones unívocas y precisas entre ellos, así como con los conceptos representados (Hassan & Núñez, 2005).

Hay dos tipos principales de productos de trabajo asociados al desarrollo de vocabularios controlados. Uno de estos corresponde a las matrices de metadatos que facilitan la discusión sobre la priorización de los vocabularios, y el otro es una aplicación que permita gestionar los términos del vocabulario y las relaciones. La Figura 5, corresponde a una matriz de metadatos elaborada para 3Com. Como se puede observar, esta matriz contiene los diferentes vocabularios empleados en el sitio Web, en donde se realiza una descripción por cada uno de los términos empleados y se muestran ejemplos de sus posibles usos, además de una categorización del nivel de dificultad que representan en su mantenimiento.

Vocabulary	Description	Examples	Maintenance
Subject	Terms that describe networking	Home networking; servers	Difficult
Product type	Types of products that 3Com sells	Hubs; modems	Moderate
Product name	Names of products that 3Com sells	PC Digital WebCam	Difficult
Product brand	Brands of products that 3Com sells	HomeConnect; SuperStack	Easy
Technology	Types of technologies associated with products	ISDN; Broadband; Frame relay	Moderate
Protocols	Types of standards and protocols associated with products	TCP/IP; Ethernet	Moderate
Hardware	Types of devices that products are used in	PDA; Wireless phone; Internet appliances; PC	Moderate
Geographic location: region	Name of geographic region	Europe; APR	Easy
Geographic location: country	Name of country	Germany; Czech Republic	Easy
Language	Name of language	German; Czech	Easy
Technology applications	Names of applications for technologies	Call center; e-business	Moderate
Industries	Types of industries that 3Com works with	Healthcare; government	Easy
Audiences	Kinds of audiences the 3Com site attracts	Consumers; First-time visitors; media	Easy
Customer group: workplace	Type of workplace that customers work in	Home; office	Moderate
Customer group: business	Size or scale of business that customers work in	Small business; large enterprise; service provider	Moderate
Roles	Type of role that people have in their business	IT manager; consultant	Moderate
Document type	Purpose of content object	Form; instructions; guide	Easy

Figura 5: Matriz de metadatos para 3Com (Morville & Rosenfeld, 2006)

2.4. Herramientas de Apoyo al Arquitecto de la Información

Cada profesional cuenta con herramientas de apoyo para llevar a cabo de forma eficiente su labor. En el caso del Arquitecto de la Información, éste cuenta con una variada gama de herramientas para la elaboración de prototipos y tareas específicas que llevan a cabo en las distintas etapas de desarrollo de la Arquitectura de la Información.

En esta sección se realiza una revisión de las herramientas comúnmente usadas por el Arquitecto de la Información y una descripción detallada de las herramientas para la elaboración de prototipos. En la sección 2.4.1, se ha confeccionado una lista de herramientas de apoyo de acuerdo a las tareas del Arquitecto de la Información. En la sección 2.4.2, se describen una serie herramientas de software para la elaboración de prototipos, tanto para su uso en escritorio como en línea, basado en el trabajo de Pérez y Codina (Pérez & Codina, 2010). Finalmente, en la sección 2.4.3 se ofrece un análisis de las herramientas descritas para el Arquitecto de la Información.

2.4.1. Herramientas de Apoyo a las Tareas del Arquitecto de la Información

En esta sección se presentan una serie de herramientas comunes de apoyo al Arquitecto de la Información agrupadas de acuerdo a las diferentes tareas que desempeñan. Esta información está basada en el estudio realizado por Morville et al. (Morville et al., 2009).

La Tabla 1, muestra una lista con las diferentes herramientas de apoyo a las tareas del Arquitecto de la Información. Como se puede observar, la información está categorizada de acuerdo a los productos de referencia de los Arquitectos de la Información, por ejemplo, *Clasificación Automática*, *Motores de Búsqueda*, *Herramientas de Gestión de Tesoros*, *Plataforma de Conocimiento para Empresas*, *Sistemas de Gestión de Contenidos*, *Investigación de Usuarios y Análisis Web / Seguimiento*. Para ello, se realiza una descripción general por cada una de estas

categorías, y se listan las diferentes herramientas de apoyo con que cuenta cada categoría.

Categoría	Descripción	Herramientas de Apoyo
Clasificación Automática	Software que utiliza reglas definidas por humanos o algoritmos de reconocimiento de patrones para asignar automáticamente los metadatos de vocabulario controlado a los documentos. Esto es equivalente a la asignación de documentos a categorías dentro de una taxonomía.	<ul style="list-style-type: none"> • Interwoven Metatagger, http://www.interwoven.com/products/content_intelligence/index.html • Entrieva de SemioTagger, http://www.entrieva.com/entrieva/semiotagger.htm • Vivísimo de Clustering Engine, http://vivisimo.com/html/vce • Autonomía IDOL Server, http://www.autonomy.com/content/Products/IDOL/index.en.html
Motores de Búsqueda	Software que proporciona la indexación de texto completo y capacidades de búsqueda.	<ul style="list-style-type: none"> • Endeca Information Access Platform, http://endeca.com • Google Enterprise Solutions, http://www.google.com/enterprise • Fast, http://www.fastsearch.com • Autonomy, http://autonomy.com
Herramientas de Gestión de Tesoros	Herramientas que proporcionan apoyo para el desarrollo y la gestión de vocabularios controlados y tesauros.	<ul style="list-style-type: none"> • MultiTes, http://www.multites.com • Factiva Synaptica, http://www.factiva.com/products/taxonomy/synaptica.asp • Lexico, http://www.pmei.com/lexico.html • WebChoir, http://www.webchoir.com • Term Tree, http://www.termtree.com.au • DataHarmony, http://www.dataharmony.com
Plataforma de Conocimiento para Empresas	Herramientas que proveen soluciones empresariales integradas completamente.	<ul style="list-style-type: none"> • Microsoft SharePoint, http://www.microsoft.com/sharepoint/portals/erver.asp • Bea's AquaLogic, http://bea.com • Oracle, http://www.oracle.com/technology/products/ias/portal/index.html • IBM's WebSphere, http://www.ibm.com/websphere/portal
Sistemas de Gestión de Contenidos	Software que gestiona el flujo de trabajo desde la autoría de contenido a la edición de la publicación.	<p>Para empresas:</p> <ul style="list-style-type: none"> • Interwoven, http://www.interwoven.com • Vignette, http://www.vignette.com • Microsoft Content Management Server, http://www.microsoft.com/cmserver • Stellent, http://www.stellent.com <p>Para uso personal y trabajo en grupo;</p> <ul style="list-style-type: none"> • WordPress, http://wordpress.org • Movable Type,

Análisis Web / Seguimiento	Software que analiza el uso y rendimiento estadístico de los sitios Web, proporcionando métricas valiosas sobre el comportamiento del usuario y sus características.	http://www.sixapart.com/movabletype <ul style="list-style-type: none"> • Drupal, http://drupal.org • Plone, http://plone.org • SocialText, http://www.socialtext.com
	Software que apoya la investigación sobre el usuario, incluido el <i>cardsorting</i> en línea y las pruebas de usabilidad a distancia.	<ul style="list-style-type: none"> • WebTrends, http://www.webtrends.com • Google Analytics, http://www.google.com/analytics • Omniture, http://www.omniture.com/products/web_analytics • CoreMetrics, http://www.coremetrics.com • Mint, http://www.haveamint.com
Investigación de Usuarios		<ul style="list-style-type: none"> • MindCanvas, http://www.themindcanvas.com • Morae, http://www.techsmith.com/morae.asp • Macromedia Captivate, http://www.adobe.com/products/captivate • Ethnio, http://www.ethnio.com • xSort, http://www.ipragma.com/xsort

Tabla 2: Herramientas de apoyo a las tareas del Arquitecto de la Información

2.4.2. Herramientas Software para la Elaboración de Prototipos

En la sección anterior, se han presentado diversas herramientas de apoyo agrupadas de acuerdo a las tareas más comunes llevadas a cabo por el Arquitecto de la Información. Sin embargo, no se han introducido en concreto las herramientas software empleadas comúnmente por estos profesionales para confeccionar blueprints (planos), wireframes (maquetas) y modelos de contenido. Estas herramientas se describen en esta sección de forma detallada, ya que están más relacionadas con el trabajo presentado. En concreto, en esta sección se describirá cada una de estas herramientas software por separado con el objetivo de indicar de qué forma representan el conocimiento proveniente de las definiciones conceptuales del Arquitecto de la Información, y las funcionalidades con que cuentan para dar una continuidad a la información de análisis y diseño que necesitan los analistas e ingenieros software. Dada la variedad existente, dichas herramientas se han categorizado en dos grupos diferentes: el primero dedicado a herramientas de escritorio, y el segundo a herramientas en línea.

a) Herramientas de Escritorio:

El primer grupo de soluciones descritas está formado por un conjunto de herramientas de escritorio utilizadas comúnmente por los profesionales dedicados al diseño de interacción y a la Arquitectura de la Información. Las herramientas descritas corresponden a Axure, Visio Profesional, OmniGraffle, Denim, Conceptdraw Pro, Smartdraw y Pencil Project. En la Tabla 3 se presenta en detalle cada una de estas herramientas, dando además una descripción general sobre ellas y comentando sus funcionalidades, librerías, formatos aceptados y sitios Web donde encontrarlas.

Herramienta	Sitio Web	Descripción	Funcionalidades	Librerías y Formatos
Axure	www.axure.com	Es junto a Visio de Microsoft, una de las herramientas de prototipado más utilizadas por los Arquitectos de la Información. Se trata de una solución de escritorio en inglés y de pago, diseñada específicamente para el prototipado, en Windows y Apple, de entornos Web (wireframes y blueprints).	Diseño de interacción similar al ofrecido en las herramientas de Office de PowerPoint y Visio. Permite trabajo en equipo y compartir información con los clientes y partes interesadas. Proporciona características para el diseño en la fidelidad adecuada para diferentes tipos de proyectos. Permite realizar anotaciones y notas a pie de página, la edición colaborativa con un sistema de control de versiones y la creación de prototipos dinámicos y navegables.	Incorpora una librería estándar formada por una veintena de elementos gráficos para el prototipado Web que permiten la gestión y edición de sus elementos y la incorporación de nuevos componentes gráficos externos. Axure permite la exportación a formatos como Html, Word, Cvs, algunos formatos de imagen (Bitmap, PNG, JPG o GIF).
Visio Profesional	office.microsoft.com	La herramienta Visio es uno de los programas informáticos, de uso exclusivo para Windows y es uno de los más versátiles y comúnmente utilizado en la elaboración de diagramas.	Las herramientas que lo componen permiten realizar diagramas de oficinas, diagramas de bases de datos, diagramas de flujo de programas, UML. Permite visualizar, analizar y comunicar los sistemas, recursos y procesos, y los datos en que se basan.	Ofrece una librería estándar formada por una cincuentena de elementos gráficos para el diseño del prototipado Web. Permite la exportación a formatos como PDF, HTML, SGV, TIFF, JPEG, GIF, PNG o Visio; e importa formatos como HTML, SGV, TIFF, JPEG, GIF, PNG o Visio. Adicionalmente, permite incluir anotaciones y notas a pie de página, la edición colaborativa de prototipado y la creación de prototipos dinámicos.

OmniGraffle	<p>www.omnigroup.com/applications/OmniGraffle</p>	<p>OmniGraffle es una de las herramientas de prototipado más utilizadas en el entorno de Apple.</p>	<p>Ofrece la posibilidad de incluir anotaciones y notas a pie de página, prototipos dinámicos y la edición colaborativa de prototipado.</p>	<p>Esta herramienta permite la exportación a diferentes formatos como OmniGraffle, PDF, TIFF, PNG, JPEG, EPS, HTML, OmniOutliner, SVG, PICT, Photoshop, BMP o Visio; y la importación de formatos como OmniGraffle, Visio, DOT, Xcode o EOModeler.</p>
Denim	<p>dub.washington.edu:2007/denim</p>	<p>Denim es una herramienta de prototipado gratuita, de código abierto y sólo disponible en lengua inglesa para Windows, Unix o Mac OS X.</p>	<p>Permite desarrollar, además de prototipado, diagramas de flujo y storyboards. Permite la exportación de las propuestas de prototipado a HTML, o a su propio formato Denim. Permite la creación de prototipos dinámicos y navegables.</p>	<p>Incluye una librería reducida de elementos gráficos para el prototipado Web.</p>
Conceptdraw Pro	<p>www.conceptdraw.com</p>	<p>ConceptDraw Pro es una aplicación comercial de escritorio en lengua inglesa y muy versátil para Windows y Apple.</p>	<p>Incorpora el Site Mapper Wizard, que genera de forma automática el mapa del sitio de una Web ya existente.</p>	<p>Esta herramienta, ofrece una librería estándar de más de ciento veinte elementos gráficos para el prototipado Web. Permite la exportación a formatos como PDF, HTML, Xml, PowerPoint, Macromedia Flash, los principales formatos gráficos, SVG, o Conceptdraw; y la importación de los formatos Xml, los principales formatos gráficos, PowerPoint, BMP, Visio o Conceptdraw. Permite las notas a pie de página, la edición colaborativa y la creación de prototipos dinámicos.</p>

SmartDraw	www.smartdraw.com	Smartdraw es una herramienta en inglés y sólo para Windows, diseñada para la realización de tareas gráficas en el contexto de las organizaciones que incluye también recursos para el prototipado de páginas Web.	Permite anotaciones y notas a pie de página, la edición colaborativa de prototipado y la creación de prototipos dinámicos e interactivos.	Esta herramienta incorpora una librería estándar que incluye un extenso conjunto de elementos gráficos para el prototipado Web. Permite la exportación a diferentes formatos como PDF, Word, PowerPoint, Excel, Word Perfect, Formatos gráficos, HTML, Metalife, o Smartdraw; pero sólo importa los formatos Visio o Smartdraw.
Pencil Project	pencil.evolus.vn	Es una solución informática en lengua inglesa, gratuita y de código abierto disponible para ordenadores con sistema operativo Windows o GNU/Linux.	Permite la creación de prototipos dinámicos y anotaciones.	Esta herramienta ofrece una librería formada por más de cincuenta elementos gráficos para el prototipado Web, que admiten una edición posterior y la incorporación de nuevos elementos gráficos externos. Permite la exportación a diferentes formatos como HTML, Png, Open office, Word, PDF o Pencil

Tabla 3: Herramientas de escritorio para la elaboración de prototipos

b) Herramientas en Línea

El segundo y último grupo corresponde a aquellas herramientas que se ejecutan a través de un navegador y sin necesidad de instalar un software adicional. Para algunas empresas y profesionales esto es una ventaja porque evita la instalación y el mantenimiento de programas en los ordenadores de la empresa. Además, de forma automática, los datos y documentos creados quedan guardados en servidores de Internet.

En el conjunto de herramientas en línea más comúnmente utilizadas por Arquitectos de la Información que se van a describir se incluyen: Mockflow, iPlotz, Pidoco, Lovely Chart, MockingBird y Lumzy. En la Tabla 4 se presentan en detalle cada una de estas herramientas, dando además una descripción general sobre ellas y comentando sus funcionalidades, librerías, formatos aceptados y sitios Web donde encontrarlas.

Herramienta	Sitio Web	Descripción	Funcionalidades	Librerías y Formatos
MockFlow	www.mockflow.com	Mockflow es una aplicación para el diseño de interfaces de usuario y prototipado de sitios Web. Es una de las aplicaciones que proporciona un mayor número de elementos de diseño, tanto de tipo estándar como para móviles.	Proporciona funcionalidades de prototipado para control de versiones, colaboración en tiempo real, anotaciones, diseño basado en esquemas, importación de esquemas predefinidos o generación de mapas del Web, entre otras. Adicionalmente, dispone de una aplicación gemela de escritorio para trabajar en modo offline.	Incorpora una amplia librería para maquetas, administrador de imágenes y opciones para importar elementos adicionales. Permite exportar a formatos de PPT, PDF o PNG.
	www.iplotz.com	Esta herramienta tiene similares características con la herramienta Mockflow.	Provee funciones específicas de gestión de proyectos y de edición colaborativa.	Incorpora librerías para maquetas y proporciona exportación a HTML y opciones de exportación a png, jpg o pdf.
Pidoco	https://pidoco.com	Permite crear wireframes y prototipos interactivos, recoger información y realizar pruebas de usabilidad para involucrar a las principales partes interesadas y los usuarios finales durante el diseño de la interfaz de usuario.	Permite crear prototipos y maquetas interactivos, con una producción colaborativa. También permite realizar test de usabilidad remoto.	Provee librerías para prototipos y maquetas. Permite exportaciones a Word y HTML, y la posibilidad de dos vistas diferentes del prototipado.
Lumzy	https://www.lumzy.com	Herramienta Web que permite crear prototipos. Ofrece una versión gratuita.	Permite la colaboración en tiempo real. Incorpora un editor exclusivo para imágenes. Provee un chat para la edición colaborativa.	Incorpora librerías para prototipos. Permite exportar a JPEG, PDF y PNG

MockingBird	https://gomockingbird.com	Herramienta Web que permite crear bosquejos de aplicaciones Web. Ofrece una versión gratuita. Proporciona un variado número de elementos y facilidad de uso.	Incorpora la posibilidad de dos vistas diferentes del prototipado. Además de opciones de trabajo colaborativo y el poder llevar a cabo test de usabilidad en modo remoto. Permite el proceso de compartir con colaboradores y usuarios.	Incorpora librerías para planos y maquetas. Permite exportar a PDF y PNG
Lovely Chart	www.lovelycharts.com	LovelyChart es una aplicación para elaborar diagramas de todo tipo, tal como, mapas de sitios, procesos de negocios, maquetas, entre otras. Cuenta con una versión para escritorio y una versión gratuita en línea. Posee opciones de alineación, estilos de conexión.	Provee funcionalidades para generar diagramas en forma automática a partir de archivos de texto.	Cuenta con librerías para diagramas de flujo, mapas de sitio, redes, maquetas y BPMN. También permite crear librerías propias. En cuanto a los formatos, esta permite exportar a JPG, PNG y PDF.

Tabla 4: Herramientas en línea para la elaboración de prototipos

2.4.3. Análisis de Herramientas de Apoyo

Las soluciones analizadas están formadas por un conjunto de herramientas de escritorio y en línea utilizadas en general por los profesionales dedicados al diseño de la interacción y a la Arquitectura de la Información, las cuales incorporan librerías formadas por una veintena de elementos gráficos para prototipos de sitios Web que permiten la gestión y edición de sus elementos y la incorporación de nuevos componentes gráficos externos. Adicionalmente, permiten incluir anotaciones y notas de pie de página, la edición colaborativa del prototipado y la creación de prototipos dinámicos. En la Tabla 5, se presenta un análisis de las ventajas y desventajas de cada una de las herramientas descritas en la sección 2.4.2:

	Ventajas	Desventajas
Axure	Destaca el diseño de interacción similar al ofrecido en las herramientas de Office de PowerPoint y Visio. Es especialista en maquetas de distintos niveles de fidelidad, y provee facilidades para realizar notas a pie de página, la edición colaborativa, prototipos dinámicos y navegables. Además, las librerías de esta herramienta se pueden ampliar con la incorporación de nuevos componentes gráficos.	Esta herramienta sólo puede ser utilizada en los sistemas operativos Windows y Mac, con una versión en inglés y de pago. Tampoco cuenta con una automatización entre las definiciones conceptuales de los contenidos de una aplicación Web y los elementos de análisis y diseño procesables por analistas e ingenieros software.
Visio Profesional	La herramienta Visio es uno de los programas informáticos más versátiles, provee un conjunto amplio de funcionalidades, formatos y plantillas, así como la edición colaborativa y dinámica de prototipos. Además, las librerías de esta herramienta se pueden ampliar con la incorporación de nuevos componentes gráficos.	La herramienta presenta una baja operatividad, debido a que sólo se puede utilizar en el sistema operativo Windows, y no permite una continuidad entre la información de análisis conceptual de la AI y la información de análisis y diseño que necesitan los analistas informáticos.
OmniGraffle	Esta herramienta ofrece una importante biblioteca de plantillas gráficas, vídeos tutoriales, un espacio de soporte al usuario y de foros de ayuda. También, permite la compatibilidad con la herramienta Visio Profesional, a través de la importación y exportación de archivos.	Es de uso exclusivo para entornos de Apple y en inglés, y no dispone de funcionalidades que permitan, de forma directa o automática, conectar el <i>output</i> del Arquitecto de la Información y el <i>input</i> que espera el Ingeniero del Software.
Denim	Esta herramienta es de uso muy intuitivo y sencillo, la cual permite diseñar en diferentes niveles de refinamiento. Además, está disponible en forma gratuita, de código abierto y para los sistemas operativos Windows, Unix y Mac.	Esta herramienta presenta escasas funcionalidades para la elaboración de diagramas y una reducida librería de elementos gráficos, y no permite la importación de nuevos elementos gráficos. Tampoco cuenta con opciones de reconocimiento de correspondencias entre los modelos de contenidos de AI y los diagramas de clases UML.
ConceptDraw Pro	Esta herramienta es muy versátil, y ofrece una amplia librería de elementos gráficos. También, permite la edición colaborativa y dinámica de prototipos.	Esta herramienta presenta una baja operatividad al ofrecer sólo una versión en inglés, y no permite representar información no funcional y transformarla en descripciones UML.
Smartdraw	Esta herramienta ofrece variadas opciones de formatos para la exportación de archivos. También, permite la compatibilidad con Visio Profesional y SmartDraw, y provee un conjunto amplio de funcionalidades para elaboración de diagramas.	Esta herramienta no genera información de análisis y diseño procesable por analistas e ingenieros software a partir de la representación inicial conceptual de los contenidos por parte del Arquitecto de la Información. Ofrece sólo una versión en inglés y para Windows.

Pencil Project	Esta herramienta es gratuita y de código abierto. Además, es muy flexible y ofrece una amplia librería de objetos gráficos.	Esta herramienta presenta una baja operatividad al ofrecer sólo una en inglés, y no cuenta con una automatización entre las definiciones conceptuales de los contenidos de una aplicación Web y los elementos de análisis y diseño procesables por analistas e ingenieros software.
	Esta herramienta ofrece una interfaz muy intuitiva y fácil de usar, es una de las aplicaciones que proporciona el mayor número de elementos de diseño, tanto de tipo estándar como para móviles. También, provee un completo conjunto de funcionalidades para prototipos. Además de contar con una versión gratuita sin restricción de funcionalidades.	Esta herramienta no permite una continuidad entre la información de análisis conceptual de la AI y la información de análisis y diseño que necesitan los analistas informáticos.
Mockflow	Esta herramienta ofrece un amplio conjunto de elementos de diseño para móviles y maquetas dinámicas. Provee funciones específicas para gestión de proyectos y de edición colaborativa, y cuenta con una alta operatividad al permitir su uso en distintos entornos de trabajos.	Esta herramienta tiene cierta complejidad en el uso, al ofrecer un mayor número de funciones, y no dispone de funcionalidades que permitan, de forma directa o automática, conectar el <i>output</i> del Arquitecto de la Información y el <i>input</i> que espera el Ingeniero del Software.
	Esta herramienta provee niveles altos de eficacia en la exportación a Word y HTML. Además, incluye opciones para realizar el test de usabilidad en modo remoto.	Esta herramienta no genera información de análisis y diseño procesable por analistas e ingenieros software a partir de la representación inicial conceptual de los contenidos por parte del Arquitecto de la Información.
iPlotz	Esta herramienta provee un grado alto de versatilidad comparada a las de escritorio. Provee facilidades de uso en la edición de los elementos e incluye una versión gratuita.	Esta herramienta ofrece un conjunto limitado de elementos visuales, y no permite representar información no funcional y transformarla en descripciones UML.
	Esta herramienta permite llevar a cabo el test de usabilidad en modo remoto. Provee altos niveles de facilidad de uso y una versión gratuita.	Esta herramienta ofrece un conjunto limitado de funcionalidades para elaborar diagramas, y no cuenta con una automatización entre las definiciones conceptuales de los contenidos de una aplicación Web y los elementos de análisis y diseño procesables por analistas e ingenieros software.
Pidoco	Esta herramienta provee altos niveles de facilidad de uso. Incorpora un editor exclusivo para imágenes y un chat para la edición colaborativa. Además provee una versión gratuita.	Esta herramienta ofrece un conjunto limitado de funcionalidades y elementos gráficos para elaborar diagramas. Tampoco permite una continuidad entre la información de análisis conceptual de la AI y la información de análisis y diseño que necesitan los analistas informáticos.
Lovely Chart		
MockingBird		
Lumzy		

Tabla 5: Ventajas y desventajas de herramientas software para la elaboración de prototipos

En general, las herramientas en línea existentes suelen ser menos expresivas y completas que las soluciones de escritorio. De hecho, sólo las herramientas Mockflow, iPlotz y Pidoco presentan funcionalidades comparables con las herramientas de escritorio, destacando las características de edición colaborativa, test de usabilidad en modo remoto y elementos gráficos de calidad. En términos generales, respecto a las características de operatividad y funcionalidad, las herramientas que más destacan son iPlotz y Smartdraw. No obstante, en la mayoría de estas herramientas existen dificultades para conectar el *output* del Arquitecto de la Información y el *input* que espera el Ingeniero del Software. Esto se intenta resolver generando distintos formatos exportables de imágenes y HTML, lo que hace desaparecer detalles semánticos importantes relativos al análisis, y dificulta la interoperabilidad y seguimiento si se utilizan herramientas posteriores más específicas para la ingeniería del software.

Por otro lado, ninguna de las herramientas existentes, ni siquiera las comentadas anteriormente, generan de forma automática o semi-automática información que permita obtener, a partir de los contenidos de información, los diagramas de clase y elementos de contenido que puedan ser utilizados por analistas o ingenieros software para dar continuidad al resto de fases del ciclo de vida del proyecto. Es por esto que, en el siguiente capítulo, se presenta y describe una propuesta para dar solución a estos problemas encontrados en el estado del arte actual.

Capítulo 3: Solución Propuesta

En este capítulo se describe el formalismo ideado y la implementación, mediante una herramienta CASE, que se propone en este trabajo para validar la solución propuesta, lo que permitirá corroborar la hipótesis H_1 enunciada con anterioridad. En la sección 3.1, se plantean los lineamientos generales de la propuesta. En la sección 3.2, se describen los componentes arquitecturales principales de la propuesta. En la sección 3.3, se describe la herramienta CASE InterArch. En la sección 3.4, se detalla el conjunto de reglas de transformación, que dan lugar al formalismo ideado para representar el conocimiento proveniente de las definiciones conceptuales del Arquitecto de la Información. Finalmente, la sección 3.5 describe un caso de estudio, que permite describir en profundidad el funcionamiento de InterArch.

3.1. *Lineamientos Generales*

A modo general, es difícil estipular los límites operacionales de la Arquitectura de la Información, lo que hace necesario muchas veces el uso de diversos tipos de herramientas y estándares. No obstante, es posible hacer un resumen de los productos más comunes que el Arquitecto de la Información debe crear para el análisis de la Arquitectura de la Información de una aplicación Web interactiva. De todos estos productos, como ya se ha comentado, los más importantes se corresponden con los blueprints (planos), los wireframes (bosquejos o maquetas), los modelos de contenido y los vocabularios controlados (Erlin et al., 2008; Morville & Rosenfeld, 2006), los cuales han sido descritos en la sección 2.3 del capítulo 2. Estos productos representan un conocimiento importante para los diferentes profesionales que participan en los proyectos de construcción de sitios Web, y se hace indispensable compartirlos en diferentes formatos y plataformas para su posterior utilización por los demás profesionales que integran el equipo de trabajo. Sin embargo, algunos de estos productos, en especial los modelos de contenido, son especialmente trascendentales para analistas e ingenieros software, ya que marcan una modelización explícita de aspectos no funcionales de la aplicación en forma de contenidos de información que deben ser procesados, posteriormente, junto con los elementos funcionales de la aplicación Web. Por otro lado, los modelos de contenido son susceptibles de un tratamiento automático que permita, a partir de ellos, generar de forma automática los diagramas de clase y elementos de contenido que definirán la aplicación en el dominio de la solución. Esta es la razón por la cual el presente trabajo se concentra en estos elementos esenciales de cara a una automatización de las salidas del proceso de análisis de la Arquitectura de la Información.

Para que la generación de información automática sea posible, se plantea el diseño, desarrollo y evaluación de una herramienta CASE denominada InterArch (Rojas & Macías, 2011a, 2011b). Esta herramienta está basada en dos principios esenciales. Primero, dado que habitualmente el Arquitecto de la Información posee un perfil menos técnico y más orientado a aspectos de diseño y organización de la información, InterArch le permitirá concentrarse en sus tareas de análisis conceptual dentro del

dominio del problema. Esto facilitará que el profesional de la información pueda elaborar sus productos de forma habitual. Segundo, en base al análisis inicial realizado por el Arquitecto de la Información, InterArch generará la información UML para analistas e ingenieros software de forma automática, concretando los elementos que tienen su correspondencia con los diagramas de clase y elementos de contenido que utilizan los profesionales del software. Para ello, dicha información se generará en un formato textual y transportable en XMI (XMI, 1999), el cual es el estándar definido por la OMG para el intercambio de diagramas UML, de forma que sea procesable por cualquier herramienta CASE existente con objeto de dar continuidad al resto de fases y actividades del proyecto.

3.2. *Diseño Arquitectónico de Solución Propuesta*

La herramienta de autor InterArch está compuesta principalmente por una serie de procesos que se encargan de la gestión y transformación de los modelos en un entorno visual orientado al Arquitecto de la Información. Como se especifica en la Figura 6, estos procesos comprenden el modelado visual de los elementos conceptuales requeridos por el profesional de la información, la transformación del modelo visual en un modelo intermedio, y la generación textual transportable del análisis de la información en diagramas UML. Dichos procesos tienen por objetivo tomar como entrada el diseño visual de diagramas del Arquitecto de la Información y generar como salida diagramas UML para el analista e ingeniero software.

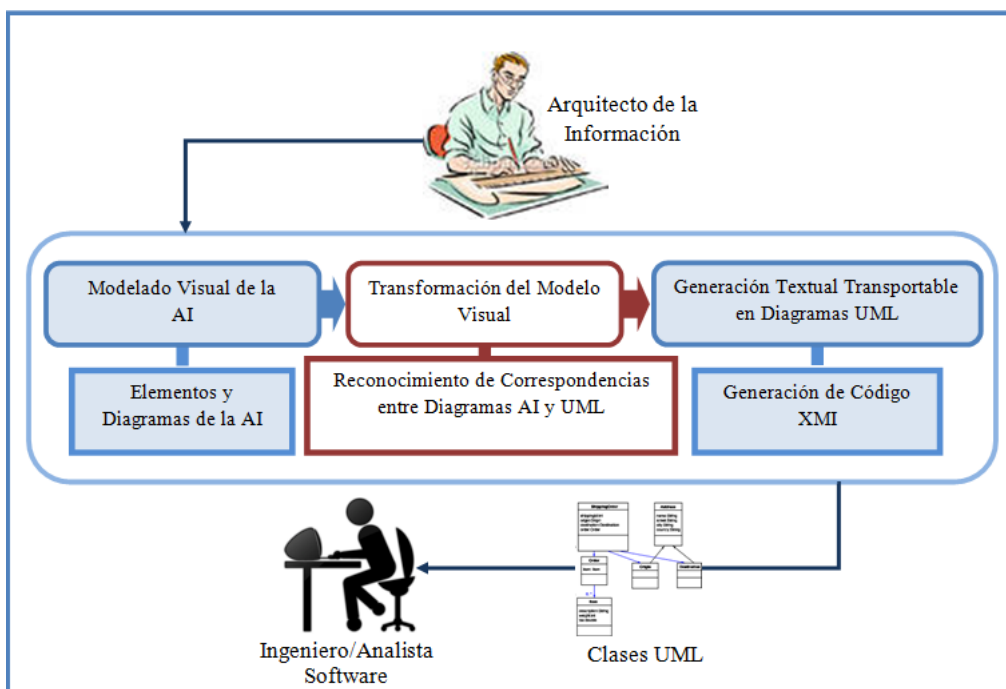


Figura 6: Detalle arquitectónico de la herramienta CASE InterArch

La idea principal que subyace detrás de estos componentes arquitecturales es permitir al Arquitecto de la Información trabajar en la elaboración visual de sus diagramas de forma transparente, pero incorporando por debajo una capa de interpretación capaz de

reconocer las distintas correspondencias entre estos diagramas de análisis de la información y las clases UML requeridas por analistas e ingenieros software. La transformación del modelado visual se realiza en base a una serie de reglas de relación y asociación que se aplican al modelo conceptual inicial realizado por el Arquitecto de la Información, generando como elemento de salida un conjunto de diagramas UML en formato transportable XMI.

A continuación, se explicará en detalle cada uno de los componentes arquitecturales de la herramienta InterArch.

- *Modelado Visual de la Arquitectura de la Información*

El modelado visual-conceptual de los elementos arquitectónicos de la información se lleva a cabo mediante la interfaz de usuario principal de la herramienta InterArch. Esta interfaz es el entorno principal de trabajo del Arquitecto de la Información, y consta de elementos visuales para elaborar diagramas en un entorno funcional que permite manipular e interactuar con los elementos visuales a través de distintas opciones de formato y edición. En la Figura 7 se muestra esta interfaz de usuario, cuyas partes principales están marcadas con letras mayúsculas (A, B y C) en la Figura 7, y que serán explicadas con detalle más adelante.

- *Transformación del Modelo Visual*

La transformación del modelo visual comprende la identificación de cada elemento visual elaborado por el Arquitecto de la Información para componer, posteriormente, los diagramas UML utilizados por el analista e ingeniero software. Esto se lleva a cabo a través de reglas de transformación que actúan en función de la asociación y relación de los elementos visuales tanto de forma individual como grupal. Hay dos tipos de reglas de transformación:

- Las Reglas de Jerarquía que generan clases de análisis de acuerdo a un conjunto de criterios respecto a la estructura y relación entre los elementos visuales elaborados por el Arquitecto de la Información.
- Las Reglas de Configuración que permiten especificar la información sobre el tipo y nivel de dependencia entre relaciones, además de la navegabilidad a partir de los esquemas conceptuales elaborados por el Arquitecto de la Información. Algunas de estas Reglas de Configuración toman valores por defecto en la generación de clases, los cuales pueden ser modificados por analistas e ingenieros para adaptar, con mínimo esfuerzo, la información de diseño correspondiente.

Para mayor información respecto a ambas reglas de transformación, ir a la sección 3.4 del presente capítulo.

- *Generación Textual Transportable de Diagramas UML*

Como paso final, InterArch genera los diagramas de clases de contenido UML en formato XMI a partir del análisis de la información llevado a cabo por el Arquitecto de la Información, condicionado además por las reglas de transformación descritas anteriormente. El XMI proporciona un estándar de facto que permite la edición y personalización de diagramas UML, por parte de analistas e ingenieros, para ser incorporados y reutilizados como documentación del proyecto software. El fin es poder continuar con el análisis y diseño del proyecto en curso y conjugar dichos diagramas con la parte funcional de la aplicación Web interactiva que se desea crear mediante otro tipo de herramientas CASE utilizadas durante el ciclo de vida.

3.3. Descripción de la Herramienta CASE InterArch

La solución propuesta, se basa en los elementos esenciales expuestos en la descripción del diseño arquitectónico de la herramienta CASE InterArch (Ver Figura 6). En general, InterArch permite manipular, formatear y relacionar elementos visuales de contenido para la elaboración de diagramas de la Arquitectura de la Información, lo que permite al profesional realizar modelados visuales de la Arquitectura de la Información a través de la manipulación de los elementos visuales. Estas y otras funcionalidades de la herramienta, se presentan de forma detallada en esta sección, algunas de las cuales corresponden a las opciones de formato y edición de los componentes, el entorno de trabajo, los elementos visuales y los diferentes estilos para su manipulación:

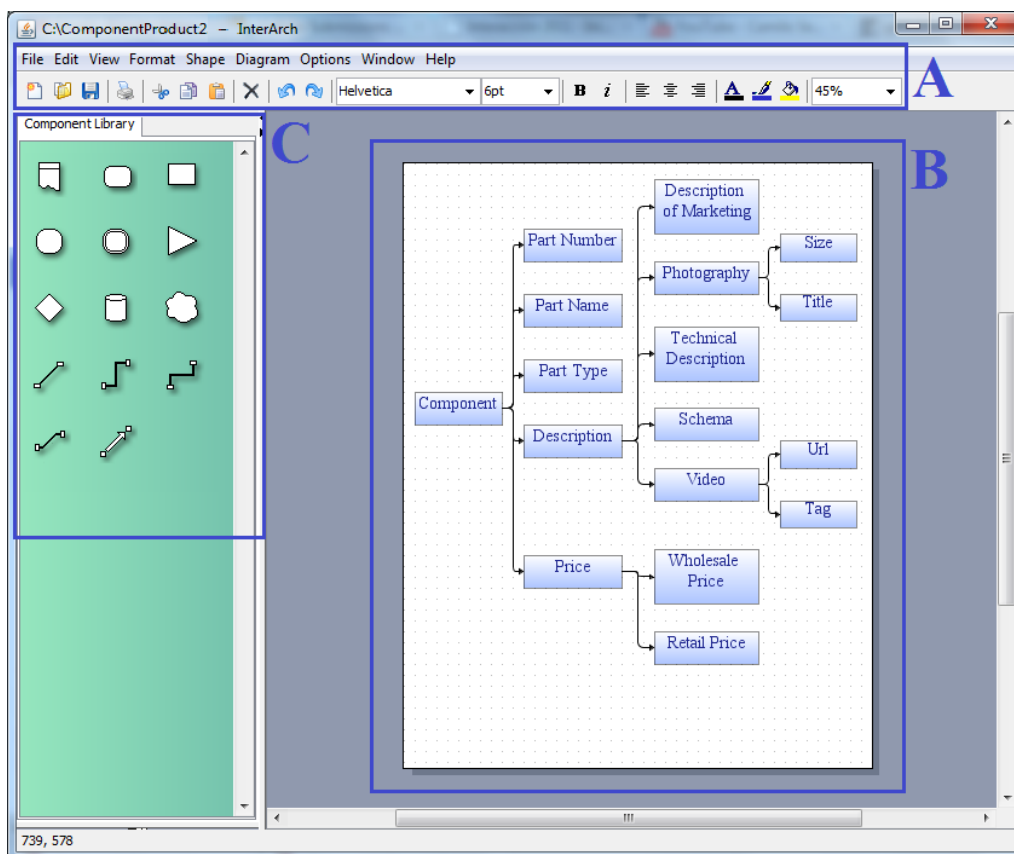


Figura 7: Interfaz de usuario de InterArch dividida en tres partes: A, B y C

a) Elementos Visuales para Diagramas

En la parte C de la Figura 7, se muestran los iconos de trabajo de los distintos elementos visuales que permiten enriquecer la interpretación visual de los diagramas elaborados por el Arquitecto de la Información. En cuanto a su tipología, existen dos tipos de elementos principales para el modelado visual:

- *Los elementos para la interpretación de contenido del modelado visual-conceptual de la AI.* Estos corresponden a elementos que permiten la manipulación e interpretación de las entidades de contenido del modelado visual-conceptual de la AI. En la Figura 8, se muestran los elementos de contenido con que cuenta la herramienta InterArch. Si bien existen elementos de contenido con diferentes formas y estilos visuales, la finalidad de cada uno de ellos es la misma, es decir, permiten representar los elementos de contenido y datos en forma de diagrama. La idea principal de estos elementos de contenido es permitir al Arquitecto de la Información definir los objetos visuales que representarán a las entidades de contenido a su gusto. Para su manipulación, estos elementos deben ser arrastrarlos al entorno de trabajo de la herramienta InterArch.

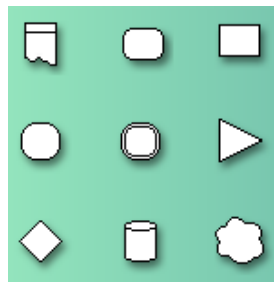


Figura 8: Iconos de elementos de contenido

- *Los elementos de enlace que permiten definir los tipos de asociaciones y relaciones entre los elementos de contenido.* En la Figura 9, se muestran los elementos de enlace con que cuenta la herramienta InterArch. Si bien existen elementos de enlaces con diferentes formas y estilos visuales, la finalidad de cada uno de ellos es la misma, es decir, tienen como objetivo permitir al Arquitecto de la Información crear relaciones entre los diferentes elementos de contenido, lo que da lugar a definir el tipo de jerarquía de los diagramas. Para manipular estos elementos de enlace, se necesita identificar el elemento de contenido de origen y el elemento de contenido destino.

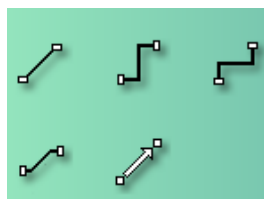


Figura 9: Iconos de elementos de enlace

La clasificación de los elementos visuales tiene como objetivo facilitar el tratamiento adecuado en la aplicación de reglas de transformación, para así obtener una generación adecuada de diagrama de clases UML.

b) Entorno de Trabajo Visual

Como ya se ha mencionado anteriormente, el entorno de trabajo visual corresponde principalmente a la parte B de la Figura 7. Este es el entorno de trabajo principal que permite manipular y relacionar los distintos elementos visuales que dispone la herramienta. En el ejemplo que aparece en la Figura 7, se pueden apreciar las relaciones representadas entre los elementos de contenido, las cuales describen la estructuración y precios de los componentes de cada producto en una tienda on-line.

Manipulación de elementos visuales: Para manipular los elementos visuales dentro del entorno de trabajo de la herramienta InterArch, basta con arrastrar los elementos visuales hacia el entorno de trabajo para comenzar a manipular y configurar las distintas opciones con que cuenta cada elemento visual. A continuación se presentan algunos de los tipos de manipulaciones con que cuenta la herramienta InterArch:

- El enlace de los elementos de contenido se logra a base de seleccionar el elemento de contenido de inicio y arrastrarlo hacia el elemento de contenido de fin. Esto genera de forma automática un elemento de enlace entre los elementos de contenido. En la Figura 10, se puede apreciar dos elementos de contenido (*Componente* y *Price*), los cuales están conectados mediante un elemento de enlace.

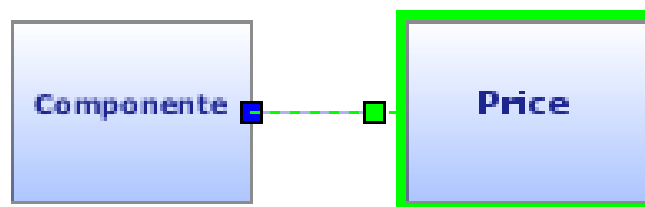


Figura 10: Manipulación para enlazar elementos de contenido en el entorno de trabajo

- Los elementos del diagrama pueden ser agrupados y manipulados en bloque. Este tipo de manipulación se logra a través de la selección de los elementos visuales, con lo cual se habilita la opción para ser agrupados o simplemente manipular en bloque todos los elementos visuales seleccionados. En la Figura 11, se muestra la manipulación en bloque de los elementos de contenido *Component* y *Price*, en la cual se intenta modificar la posición de estos elementos dentro del entorno de trabajo de la herramienta InterArch.

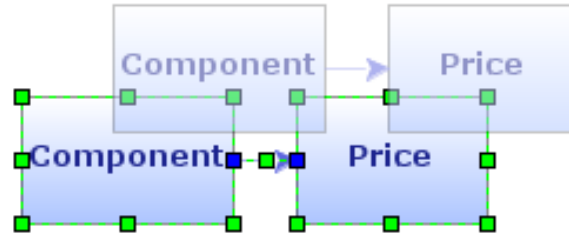


Figura 11: Manipulación en bloque de los elementos en el diagrama

- Los elementos tienen propiedades de herencia dentro del entorno de trabajo. Esto quiere decir que, para cualquier elemento de contenido, es posible generar nuevos elementos de contenido. Esto permite crear nuevos elementos de contenido con las características del elemento de contenido de origen. La tarea se lleva a cabo, posicionándose sobre el elemento de contenido y al cambiar el cursor a un formato de *mano*, se debe seleccionar el elemento y arrastrarlo hacia la posición que se desee crearlo.

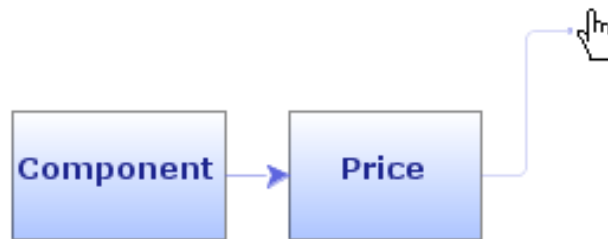


Figura 12: Creación de elementos visuales heredados

En la Figura 12, se muestra un ejemplo de manipulación para crear elementos visuales heredados. Como se puede observar, los elementos de contenido *Component* y *Price* están vinculados a través de un elemento de enlace, en donde se intenta crear elementos visuales heredados desde el elemento de contenido *Price*.

c) Formato y Edición de Componentes

En la parte A de la Figura 7, se muestran las opciones de formato y edición habituales de una aplicación de estas características para manipular elementos dentro del entorno de trabajo: archivos, edición, formato y estilos, vistas del entorno de trabajo, entre otras. Las opciones de archivo, permiten controlar el tipo almacenamiento, impresión y formato de la página del diagrama. Las opciones de edición, proveen opciones para la manipulación de edición de textos y recuperación de acciones en el entorno de trabajo visual de InterArch. Las opciones de vistas, generan distintas alternativas visuales del entorno de trabajo. Las opciones de formato, permiten configurar la forma, tamaño, estilo y comportamiento de los elementos visuales. En resumen, estas opciones de formato y edición de los elementos visuales, permiten configurar cada elemento visual tanto de forma individual como grupal, así como también, definir y manipular los

diagramas. A continuación se describen algunos ejemplos de este tipo de funcionalidades de la herramienta InterArch:

- Una de las opciones de formato corresponde a la configuración del color que puede ser establecido para cualquier elemento visual de la herramienta InterArch. La Figura 13 corresponde a un ejemplo de las opciones de color del formato de línea de los elementos visuales con que cuenta la herramienta InterArch.

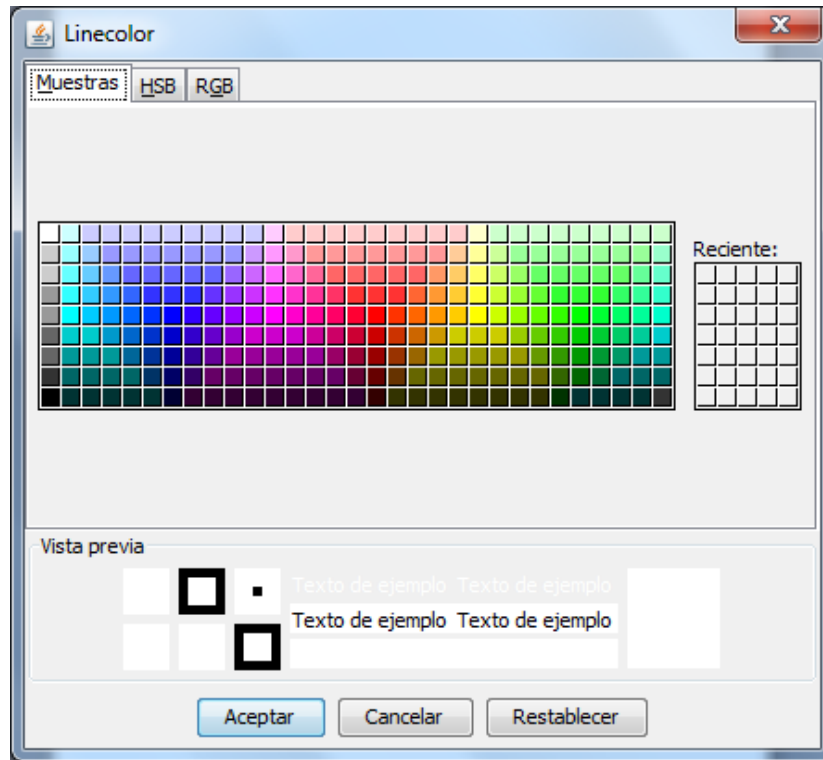


Figura 13: Paleta de colores de formato de línea

- Por otro lado, existen distintos formatos disponibles para almacenar los diagramas elaborados, las cuales se encuentran dentro de las opciones para almacenar el archivo. En la Figura 14, se muestra el cuadro de dialogo para almacenar los diagramas, en el cual se pueden apreciar los distintos formatos disponibles (.xmi, .txt, .svg, .html, .png, .jpg, .bmp, .jpeg, .gif), entre ellos, está la opción de *UML diagram File (.xmi)*, el cual permite transformar el modelo de contenido, elaborado por el Arquitecto de la Información, en diagramas de clases UML en formato XMI. Esta opción, implica procesar el modelo de contenido mediante la aplicación del conjunto de reglas de transformación ideado (ir a la sección 3.4 para mayor información) y que genera automáticamente como resultado un archivo con extensión .xmi. Los otros formatos disponibles permiten almacenar los diagramas en formatos de imagen (.png, .jpg, .bmp, .svg, .jpeg, .gif), paginas HTML (.html) y archivo de texto (.txt).

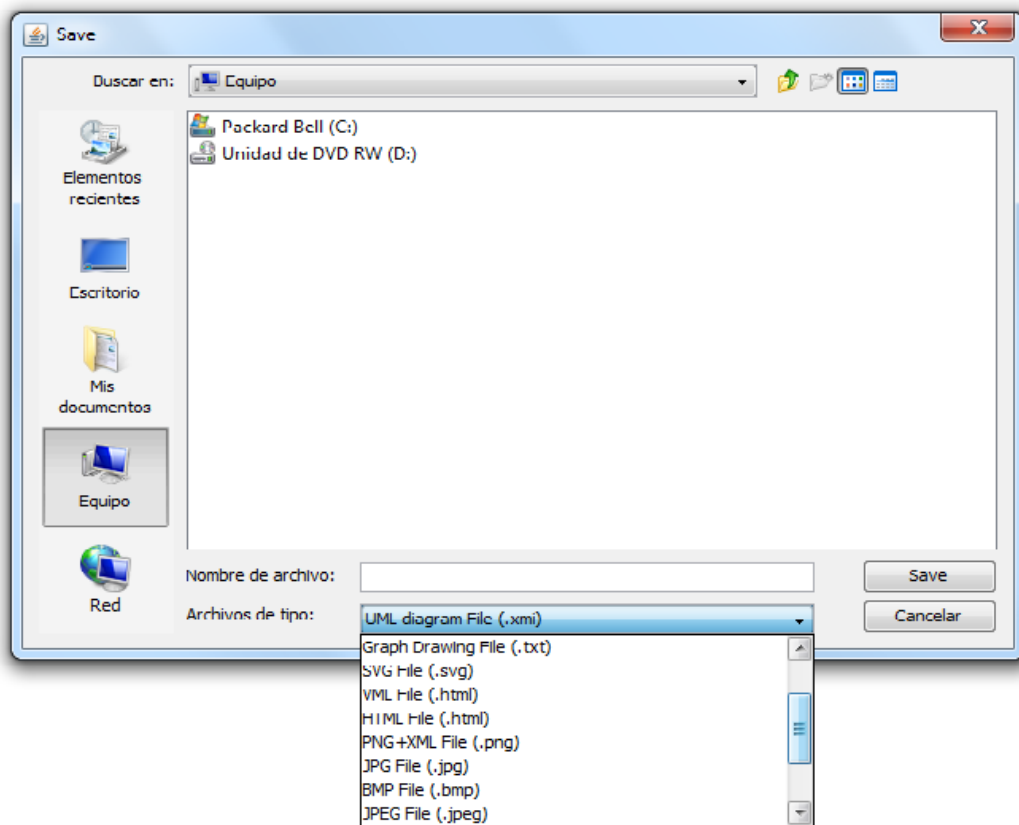


Figura 14: Formatos para almacenar el diagrama

3.3.1. Descripción de las Tecnologías Utilizadas

En la Figura 15, se resume el conjunto de tecnologías empleadas en la programación de la herramienta InterArch. La base de todas las tecnologías utilizadas es el entorno *cliente* de Java, el cual se puede ejecutar en cualquier sistema operativo que soporte el uso de una Máquina Virtual de Java (JVM) (Lindholm & Yellin, 1999). Para diseñar la parte gráfica se ha utilizado la librería JGraph que se apoya en Java como tecnología de soporte. En la parte superior de la pirámide tecnológica se sitúa la librería GraphEditor, que es una implementación de la librería JGraph con la que se ha desarrollado la herramienta InterArch. Finalmente, para la generación de diagramas de clases UML, se ha utilizado el formato textual y transportable XMI. A continuación se describen más en detalle cada una de estas tecnologías:

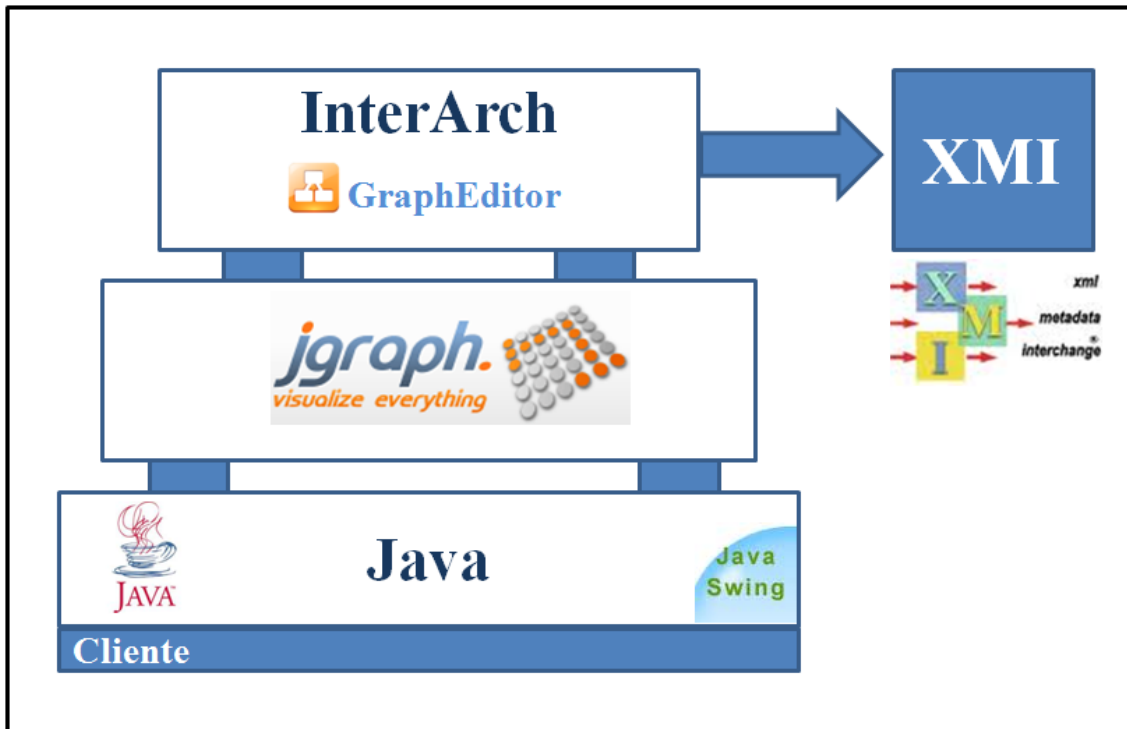


Figura 15: Tecnologías utilizadas en la implementación de InterArch

a) JGraph

Es un componente software de código abierto para diseñar gráficos escrito en el lenguaje de programación Java. JGraph (David & Gaudenz, 2010) está basado en la teoría matemática de grafos y permite crear aplicaciones Java Swing que presentan funcionalidad interactiva de diagramas. InterArch se apoya en la versión 1.5.1.7 de esta librería para crear diagramas interactivos, basados principalmente en la manipulación de Vertex (es la unidad fundamental de la que están formados los grafos) y Edge (corresponde a una relación entre dos vértices de un grafo). Más específicamente, InterArch está construido en base a la versión 1.8 de GraphEditor que permite la edición de gráficos, el cual es una implementación de la librería JGraph. InterArch ha incorporado en GraphEditor funcionalidades para las reglas de transformación (Jerarquía y Configuración) y modificado aspectos relacionados con el formato de la herramienta.

b) Java

Java corresponde a un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90. Java se desarrollo con el propósito de crear un lenguaje que pudiera funcionar en redes computacionales heterogéneas y que fuera independiente de la plataforma de ejecución. Un programa de Java puede ejecutarse en cualquier máquina o plataforma (Gosling et al., 2005). Este lenguaje de programación es el que se ha utilizado como base para desarrollar de forma completa la herramienta InterArch.

c) XMI

XMI (XML de Intercambio de Metadatos) es una especificación para el Intercambio de Diagramas. Es un estándar que permite expresar objetos utilizando XML (lenguaje de marcas extensible). XMI es mucho más que un conjunto de reglas de serialización, está estrechamente relacionado con los estándares de modelado, permitiendo emplear modelos de manera efectiva. Los principales beneficios de XMI son los siguientes (Grose et al., 2002):

- XMI proporciona una representación estándar de objetos en XML, permitiendo el intercambio efectivo de los objetos que utilizan XML.
- XMI especifica cómo crear esquemas XML a partir de modelos.

InterArch utiliza XMI como formato textual transportable de los diagramas de clases UML generados. El tratamiento de este código se ha realizado en base a un desarrollo propio, y no mediante el uso de librerías ad-hoc de procesamiento XMI (DOM, SAX, JOB, XMI Framework, entre otras). El archivo XMI generado por InterArch puede ser procesable por herramientas CASE de modelado que soporten la importación de este formato.

d) Swing GUI Components

Java Swing es una librería de componentes visuales que proporciona Java (Walrath & Campione, 1999). Incluye todo tipo de componentes, desde botones hasta paneles para dividir a las tablas. Mucho de los componentes poseen características para clasificar, imprimir, arrastrar y solar, entre otras. Esta librería es utilizada en el desarrollo de la interfaz de usuario de InterArch, la cual permite adaptar las interfaces a cada sistema operativo sin tener que cambiar el código fuente.

3.4. *Reglas de Transformación*

InterArch tiene como objetivo permitir al Arquitecto de la Información concentrarse en sus tareas de análisis conceptual, de forma que pueda elaborar sus productos de forma habitual para, posteriormente, en base al análisis inicial realizado, generar diagramas de clases UML para analistas e ingenieros software de forma automática. Para generar los diagramas UML de forma automática, InterArch incorpora una capa de interpretación compuesta por un conjunto de reglas de transformación referente a la asociación y jerarquía de los modelos de contenido elaborado por los Arquitectos de la Información, capaz de generar información de análisis y diseño. Como ya se comentó anteriormente, estas reglas de transformación se clasifican en Reglas de Jerarquía y Reglas de Configuración:

a) Reglas de Jerarquía

Este tipo de reglas de transformación corresponden a un conjunto de criterios referentes a la estructura y jerarquía de los diagramas de modelado de contenido elaborados por el Arquitecto de la Información. Las Reglas de Jerarquía contienen los criterios que definen, por cada tipo de relación, el tipo de transformación que se genera. Es decir, considera la estructura y jerarquía de los diagramas de modelado de contenido para definir las clases, atributos, métodos y asociaciones de los diagramas de clases UML. Este tipo de reglas han sido ideadas en base a la propuesta presentada por Pressman para el modelado de análisis para aplicaciones Web (Pressman, 2005), en donde se plantea dar una continuidad a la representación de contenido que se lleva a cabo en las etapas iniciales del desarrollo software de un sitio Web. Más específicamente, en este Trabajo Fin de Máster se ha formalizado e implementado la propuesta anterior, en base a un conjunto de reglas que permiten establecer correspondencias entre modelos de contenidos elaborados por Arquitectos de la Información y diagrama de clases UML requeridos por analistas e ingenieros software. Adicionalmente, se han incorporado nuevas características respecto a la configuración de los métodos, relaciones y reglas en la generación de los diagramas de clases UML.

Para lograr una mejor comprensión de este conjunto de Reglas de Jerarquía, la descripción de las mismas estará basada en una aplicación práctica sobre el diagrama conceptual de la Figura 16, el cual incluye cinco elementos de contenido, donde el elemento principal se corresponde con el elemento de contenido A, el cual contiene a su vez los elementos de contenido descendientes B y C. Asimismo, del elemento de contenido C descienden a su vez los elementos D y E. A continuación se detallan estas Reglas de Jerarquía para transformar el diagrama conceptual de la Figura 16 en un diagrama UML, el cual que se muestra en la Figura 17:

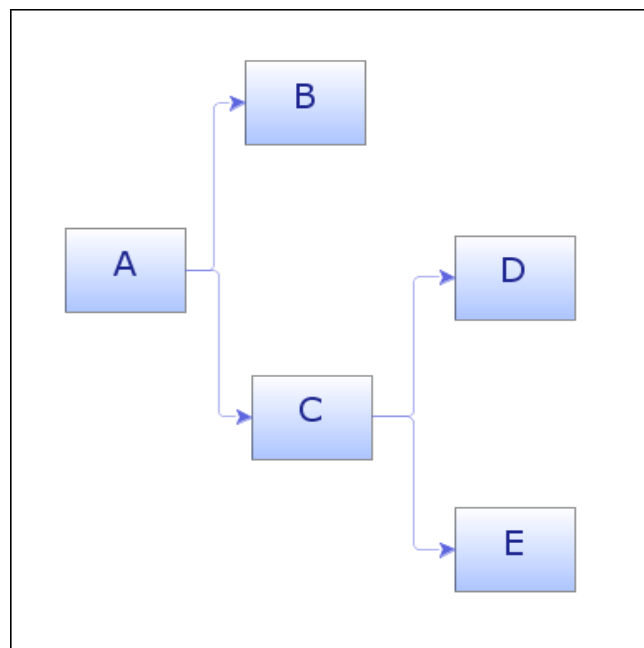


Figura 16: Elementos relacionados de forma jerárquica utilizando InterArch

Solución Propuesta

- **R₁:** *Los elementos que contienen elementos descendientes son considerados directamente como clases en el diagrama de clases UML.*

En la Figura 16, se puede apreciar cómo los elementos A y C cumplen esta condición. En el caso del elemento A, de él descienden los elementos B y C. Y en el caso del elemento C, éste está compuesto por los elementos D y E. En ambos casos, los elementos A y C se transformarían en nuevas clases en el diagrama de clases UML de la Figura 17.

- **R₂:** *Los elementos descendientes, se considerarán atributos en los elementos que descienden.*

En la Figura 16, el elemento B y C son elementos descendientes del elemento A, lo cual generaría los atributos b y c en la clase A (el elemento A cumple con el criterio especificado anteriormente). Del mismo modo, los elementos D y E corresponden a elementos descendientes del elemento C, generando los atributos d y e en la clase C. Los atributos b, c, d y e se pueden visualizar en el diagrama de clases UML de la Figura 17.

- **R₃:** *El elemento principal del diagrama del modelado de contenido, se corresponde con la clase raíz en el diagrama de clases UML.*

En la Figura 16, se puede apreciar que el elemento A cumple esta condición, lo que transformaría al elemento A en la clase raíz del diagrama de clases UML de la Figura 17.

- **R₄:** *Los elementos descendientes correspondientes a nuevas clases, generan una asociación directa con el elemento del que descienden.*

Este criterio se puede apreciar en el elemento C de la Figura 16. El elemento C corresponde a una nueva clase debido a que contiene elementos descendientes (elementos D y E) y al descender de una clase (el elemento A), esto genera una asociación directa entre las clases A y C dentro del diagrama de clases UML de la Figura 17.

- **R₅:** *Por cada asociación generada en el diagrama de clases UML, se crean tres métodos (get, set y new) en la clase donde se inicia la asociación.*

En la Figura 16, los elementos A y C corresponden a clases con una asociación directa en el diagrama de clases UML de la Figura 17. Debido a esta asociación, se crean tres tipos de métodos (get, set y new) en la clase A, los que están compuesto por el tipo de método concatenado con el atributo que hace referencia con la clase que se tiene la asociación, En este caso particular el elemento C, además de ser una clase, corresponde a un atributo de la clase A (atributo c), lo cual genera los métodos getC, setC y newC en la clase A. Estos métodos se generan en las clases donde se inicia la asociación, con el objetivo de crear

operaciones que permitan distintos tipos de comunicación con las clases de destino.

Cabe señalar también, que existe un conjunto de pautas respecto al formato de los nombres que contendrán las clases, atributos y métodos, las cuales son aplicadas conjuntamente con las Reglas de Jerarquía al modelo de contenido elaborado por el Arquitecto de la Información en el proceso automático de transformación:

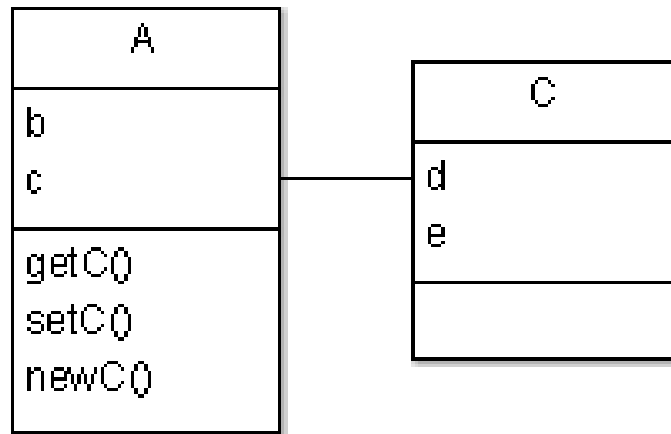


Figura 17: Diagrama de clases UML a partir de la transformación del diagrama conceptual de la Figura 16.

- El nombre de las clases se compone por la primera letra del elemento de contenido en mayúscula junto con el resto de letras de la palabra en minúsculas. Los espacios de separación del nombre del elemento de contenido son eliminados. Por ejemplo, para el caso de un elemento de contenido denominado “component”, al aplicar esta pauta generaría una clase UML de nombre “Component”.
- El nombre de los atributos se compone por la primera palabra del elemento de contenido en minúsculas junto con el resto de las palabras formadas por la primera letra en mayúscula y el resto de las letras de la palabra en minúsculas. Los espacios de separación del nombre del elemento de contenido son eliminados. Por ejemplo, para el caso de un elemento de contenido denominado “descripción of marketing”, al aplicar esta pauta generaría un atributo de nombre “descriptionOfMarketing” en el diagrama de clases UML.
- El nombre de los métodos se compone por el tipo de método (`get`, `set` y `new`) concatenado al nombre del elemento de contenido correspondiente a la clase de destino que se crear la asociación. El nombre del elemento de contenido se compone por la primera letra en mayúscula junto con el resto de las letras de la palabra en minúsculas. Los espacios de separación del nombre del elemento de contenido son eliminados. Por ejemplo, para el caso de un elemento de contenido denominado “description”, al aplicar esta pauta generaría tres

métodos con los nombres "getDescription", "setDescription" y "newDescription" en el diagrama de clases UML.

b) Reglas de Configuración

Las reglas de transformación corresponden a un conjunto de propiedades específicas referentes al nivel de visibilidad, acceso y navegabilidad de las clases, atributos, métodos y asociaciones de los diagramas de clases UML. A diferencia de las Reglas de Jerarquía, este tipo de reglas no consideran la estructura de los diagramas de contenido elaborados por el Arquitecto de la Información, sino que se concentran en los tipos de elementos que componen los diagramas de clases UML y las diferentes propiedades que estos pueden contener.

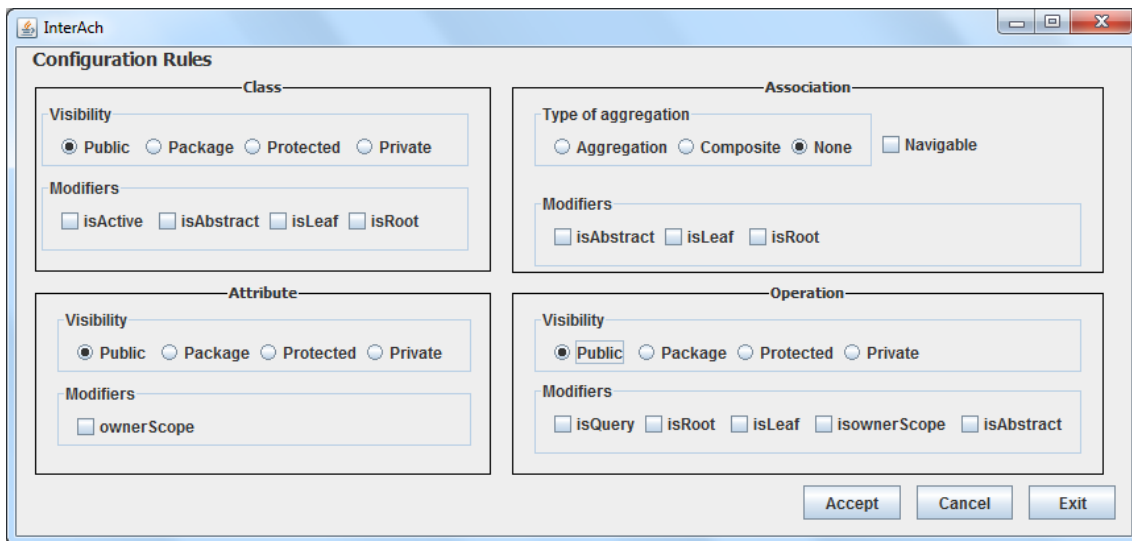


Figura 18: Módulo de Reglas de Configuración de InterArch

La herramienta InterArch provee un conjunto de Reglas de Configuración que son aplicadas al modelo de contenido elaborado inicialmente por el Arquitecto de la Información y que condicionan la generación final del diagrama de clases UML. En la Figura 18, se muestra el módulo de InterArch para las Reglas de Configuración, el cual permite especificar las distintas propiedades de las etiquetas de los elementos de los diagramas de clases UML. En la figura se puede apreciar el conjunto de reglas que están seleccionadas de forma predeterminada y que serán aplicadas por defecto, las cuales se agrupan en reglas de clase, atributo, método y asociación (*Class*, *Attribute*, *Operation*, *Association*, respectivamente). El módulo de Reglas de Configuración, permite ajustar la generación de diagramas de clases UML de acuerdo a las necesidades de analistas e ingenieros software. Este conjunto de Reglas de Configuración corresponden a los tipos de etiquetas propuestos por la OMG (OMG, 2005) para adaptar los esquemas y documentos que se producen utilizando XMI:

- **Clase:** Como se muestra en la Figura 18, este tipo de propiedades corresponden a las que están agrupadas bajo la categoría de *Class* del módulo de Reglas de Configuración, y contienen diferentes opciones para configurar el tipo de

visibilidad (*Public*, *Package*, *Protected* y *Private*) y modificar el tipo acceso (*Active*, *Abstract*, *Leaf* y *Root*) de las clases. Por defecto se encuentra establecido que las clases tendrán una visibilidad pública y sin ningún tipo de acceso definido; dichas propiedades se reflejarían en la generación de las etiquetas de las clases del código XMI.

- **Atributo:** Como se muestra en la Figura 18, este tipo de propiedades corresponden a las que están agrupadas bajo la categoría de *Attribute* del módulo de Reglas de Configuración, y contienen opciones para configurar el tipo de visibilidad (*Public*, *Package*, *Protected* y *Private*) y el tipo de acceso (*OwnerScope*) de los atributos de las clases. Por defecto se encuentra establecido que los atributos tendrán una visibilidad pública y sin ningún tipo de acceso definido, dichas propiedades se reflejarían en la generación de las etiquetas de los atributos del código XMI.
- **Asociación:** Como se muestra en la Figura 18, este tipo de propiedades corresponden a las que están agrupadas bajo la categoría de *Association* del módulo de Reglas de Configuración. En donde, es posible configurar las propiedades de navegabilidad entre las clases, el tipo de agregación (*Aggregation*, *Composite* y *None*) y modificar el tipo acceso (*Root*, *Leaf* y *Abstract*) de todas las asociaciones generadas en el diagrama de clases UML. Por defecto se encuentra establecido que las asociaciones no tendrán un tipo de agregación, sin ningún tipo de acceso definido y que tampoco habrá navegabilidad entre las asociaciones de las clases, dichas propiedades se reflejarían en la generación de las etiquetas de las asociaciones del código XMI.
- **Método:** Como se muestra en la Figura 18, este tipo de propiedades corresponden a las que están agrupadas bajo la categoría de *Operation* del módulo de Reglas de Configuración. Y contienen opciones para configurar el tipo de visibilidad (*Public*, *Package*, *Protected* y *Private*) y modificar el tipo acceso (*Query*, *Root*, *Leaf*, *OwnerScope* y *Abstract*) de los métodos de las clases. Por defecto se encuentra establecido que los métodos tendrán una visibilidad pública y sin ningún tipo de acceso definido, dichas propiedades se reflejarían en la generación de las etiquetas de los métodos del código XMI.

En general, el potencial de los componentes de InterArch y del formalismo propuesto se aprecia mejor de forma conjunta. Es decir, es la interacción y utilización de los distintos componentes de InterArch, junto con la aplicación del formalismo ideado, lo que hace posible la generación de información de análisis y diseño para analistas e ingenieros software en base al trabajo inicial del Arquitecto de la Información. La Figura 19 intenta poner de manifiesto esta idea, en la cual los diagramas de la Arquitectura de la Información son elaborados mediante el uso de los diferentes componentes de la herramienta (*Entorno de Trabajo Visual*, *Generación de Código XMI*, *Formato y Edición de Componentes*, y *Módulo de Reglas de Configuración*) y donde, al aplicar el conjunto de reglas de transformación, se posibilita la generación automática de

diagramas de clases UML en formato XMI, ajustados a las necesidades de analistas e ingenieros software. En ese sentido, se permite dar continuidad al resto de fases del ciclo de vida del proyecto y aumentar la automatización e interoperabilidad en el desarrollo de aplicaciones Web interactivas.

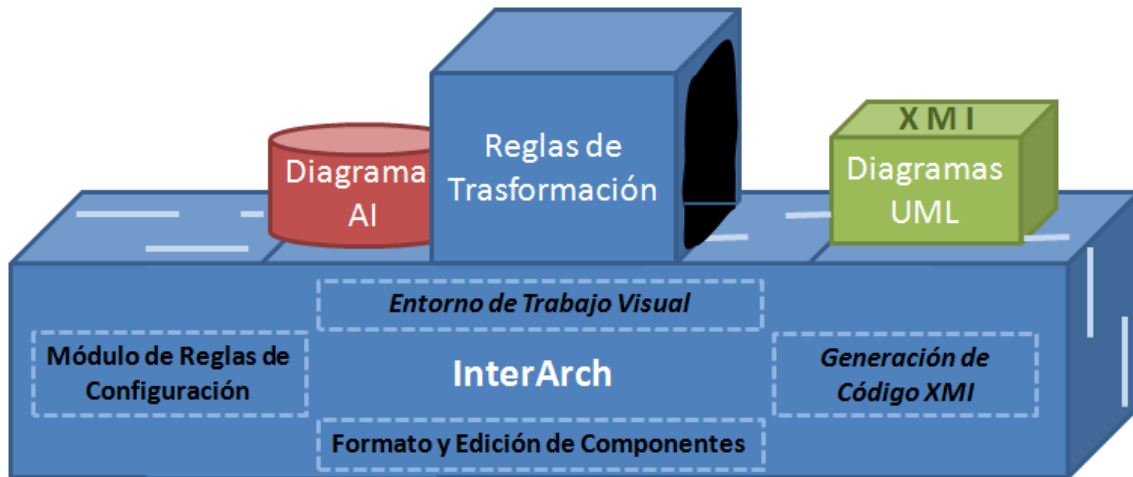


Figura 19: Proceso de transformación de diagramas AI a diagramas UML.

3.5. Caso de Estudio

Con objeto de describir más en profundidad el funcionamiento de InterArch, se concreta un caso de estudio específico bajo la herramienta.

a) Descripción de la tarea para el Caso de Estudio

Supongamos que el Arquitecto de la Información pretende trabajar sobre un modelo de contenido de la información que describe la composición, descripción y precios de los componentes de cada producto en una tienda on-line. Este ejemplo se puede apreciar en la Figura 20 a través de la creación de esta maqueta de fidelidad media, la cual se obtuvo mediante una fase de educación de requisitos con los encargados de ventas. Como se puede observar, esta maqueta contiene información referente al nombre, tipo y número del producto. También incluye información sobre la descripción técnica y de marketing, e información respecto a las características de la imagen y el video que describen visualmente el producto. A partir de la maqueta proporcionada, el Arquitecto de la Información trabajará en la parte de modelado conceptual, llevando a cabo las siguientes tareas:

- *Segmentación del contenido* de los componentes de los productos de la tienda on-line.
- *Creación de un modelo de contenido*, utilizando la herramienta InterArch.

La tarea inicial corresponde a la *segmentación del contenido* a partir de la maqueta proporcionada en la Figura 20. Esta tarea consiste en identificar los distintos componentes y estructuras de la información, de manera que se pueda establecer de forma clara y precisa la composición y jerarquía de los diferentes elementos de

información que contiene la maqueta. El resultado de la segmentación del contenido, permite al Arquitecto de la Información contar con información necesaria para la elaboración del modelo de contenido.

Component	
Part Number	857789
Part Name	LED Screen
Part Type	TV
Description	
Schema	Model UN40D5800VGXZS
Description of Marketing	Full HD. Certified Product
Technical Description	Resolution On Line 1920 X 1080.
Video	
Url	\products\TV\FullHD\LS857789.mpeg
Tag	TV, Full HD, LED Screen.
Photography	
Size	411 x 334
Title	Full HD LED Screen
Price	
Wholesale Price	US 315
Retail Price	US 380

Figura 20: Maqueta de productos en una tienda on-line

La siguiente tarea corresponde a la *creación de un modelo de contenido*, utilizando la herramienta InterArch (la Figura 4 muestra un ejemplo de un modelo de contenido). Esta tarea se realiza en base a la información de la composición de los contenidos obtenida en la tarea anterior de segmentación del contenido, lo cual genera como resultado un diagrama jerárquico de información conforme a la maqueta proporcionada en la Figura 20. En este caso, el elemento de contenido Component, más general, se describe a su vez por medio de cinco elementos de contenido (Part Number, Part Name, Part Type, Description y Price), donde Description y Price presentan a su vez datos compuestos que se definen en base a los elementos de contenido que descienden de ellos de forma jerárquica. El elemento Description se compone de elementos que representan distintos tipos de descripciones del producto (Video, Schema, Technical Description, Photography y Description of Marketing), de los cuales Video (con Url y Tag) y Photography (con Size y Title) presentan también datos compuestos. El elemento Price, indica los precios al por mayor y al por menor que puede tener el componente en cuestión: Wholesale Price y Retail Price. El resultado del proceso se corresponde con el diagrama que aparece en la parte B de la Figura 7 presentada con anterioridad.

Una vez diseñado el modelo de contenido por parte del Arquitecto de la Información, InterArch incorpora por debajo una capa de interpretación capaz de reconocer las

distintas correspondencias entre estos diagramas de análisis de la información y las clases UML requeridas por los Ingenieros del Software. La cual permite generar finalmente información de análisis y diseño para analistas e ingenieros software. En el siguiente apartado, se describe en detalle el resultado de aplicar el conjunto de reglas de transformación al modelo de contenido inicialmente que aparece en la Figura 7.

b) Reglas de Asociación y Relación para los Elementos Visuales

Una vez que el Arquitecto de la Información ha creado el diagrama de contenidos, lo siguiente sería generar las clases de contenido, aprovechables por el Ingeniero del Software, en formato UML. Este paso es completamente transparente para el Arquitecto de la Información y se logra al almacenar el modelo de contenido con la opción de “UML diagram File (.xmi)” de los formatos disponibles con que cuenta la herramienta InterArch.

Para la generación de diagramas UML a partir de los diagramas AI, se aplican las reglas de transformación referidas en la sección 3.4. En este caso concreto, a partir del elemento principal, `Component`, se generaría una clase raíz y sus elementos descendientes se evaluarían de la siguiente forma: los elementos terminales, y descendientes directamente del elemento principal, pasan a ser atributos de la clase `Component`. Si estos elementos tienen a su vez descendientes, estos pasan a ser nuevas clases relacionadas directamente con el elemento `Component`. Si los elementos descendientes generan atributos y si son elementos compuestos, de forma recursiva generarían nuevas clases relacionadas con el elemento del que descienden. La aplicación de esta regla genera una clase `Component` con cinco atributos: `partNumber`, `partName`, `partType`, `description` y `price`. Siguiendo con la aplicación de estas reglas, el elemento `Price`, al descender directamente del elemento principal y contener a su vez elementos descendientes `Wholesale Price` y `Retail Price` se transforma en una nueva clase, y sus elementos `Wholesale Price` y `Retail Price` en atributos para la clase `Price`. El elemento `Description` presenta una situación similar al elemento `Price`, es decir, al ser elemento compuesto por los elementos descendientes `Video`, `Schema`, `Technical Description`, `Photography` y `Description of Marketing`, estos se transforman en atributos de la nueva clase `Description`. En ambos casos, las nuevas clases generadas (`Price` y `Descripción`), continúan siendo atributos de la clase principal `Component`. Las clases `Price` y `Descripción` pasan a tener una relación directa con la clase principal `Component` (el tipo de asociación dependerá de la configuración de las reglas de transformación descritas anteriormente). Finalmente, los elementos `Video` y `Photography` al ser también elementos compuestos por `Url`, `Tag`, y por `Size` y `Title`, respectivamente, se convierten en nuevas clases relacionadas directamente con la clase descendiente (`Descripción`).

En cuanto a los métodos de clase, se generan por defecto tres métodos por cada atributo que representa una clase agregada (`get`, `set` y `new`). Cada elemento compuesto

corresponde a una nueva clase relacionada con el elemento del cual desciende, por cada relación de este tipo, se generan tres métodos en la clase donde se encuentra el elemento compuesto. Por ejemplo, en la clase `Component` que contiene una relación con la clase `Descripción`, se generan los métodos `getDescripción`, `setDescripción` y `newDescripción` (La composición de los nombres de los métodos se describe en la sección 3.4).

c) Generación de Código XMI

La aplicación de las reglas comentada en el apartado anterior, genera como salida un diagrama de clases UML almacenado en formato textual XMI. A modo de ejemplo, en el siguiente fragmento de código se puede apreciar la representación XMI de las clases `Component`, `Price`, `Description`, `Video` y `Photography` a través de la etiqueta `<UML:Class>`, la cual contiene el elemento `xmi.id` para indicar el identificador único de la clase y el elemento `name` para el nombre. Así también, por ejemplo se pueden visualizar los atributos `partNumber`, `partName` y `partType` de la clase `Component`, los cuales se representan a través de la etiqueta `<UML:Attribute>`, la cual contiene el elemento `xmi.id` que permite indicar el identificador único del atributo y el elemento `name` para el nombre del atributo. Respecto a los métodos de las clases, en el código se pueden apreciar, a modo de ejemplo, los métodos `getDescription`, `setDescription` y `newDescription` de la clase `Component`, que se representan a través de la etiqueta `<UML:Operation>`, la cual contiene el elemento `xmi.id` para indicar el identificador único del método y el elemento `name` para el nombre.

En cuanto a las asociaciones entre las clases del diagrama UML, estas se representan a través de la etiqueta `<UML:Association>`, junto con las etiquetas `<UML:AssociationEnd>` y `<UML:AssociationEnd.participant>` que permiten indicar el tipo de asociación (`Aggregation`, `Composite` y `None`) e identificar las clases que participan en la asociación, respectivamente. Finalmente, todas las etiquetas que permiten definir las clases, atributos, métodos y asociaciones cuentan con propiedades obtenidas del conjunto de Reglas de Configuración, las que se obtienen de forma predeterminada a través del módulo de Reglas de Configuración de la herramienta `InterArch` (ir al apartado b de la sección 3.4 para mayor información).

```
<UML:Class xmi.id = 'x232' name = 'Component'>
<UML:Classifier.feature>
<UML:Attribute xmi.id = 'x232:81B' name = 'partNumber' visibility =
'public'>
<UML:Attribute xmi.id = 'x232:83B' name = 'partName' visibility =
'public'>
<UML:Attribute xmi.id = 'x232:85B' name = 'partType' visibility =
'public'>
<UML:Attribute xmi.id = 'x232:87B' name = 'description' visibility =
'public'>
<UML:Attribute xmi.id = 'x232:89B' name = 'price' visibility =
'public'>
```

Solución Propuesta

```
</UML:Attribute>...
<UML:Operation xmi.id = 'x232:01C' name = 'getDescription' visibility
= 'public'> <UML:Operation xmi.id = 'x232:02C' name = 'setDescription'
visibility = 'public'> <UML:Operation xmi.id = 'x232:05C' name =
'newDescription' visibility = 'public'> <UML:Operation xmi.id =
'x232:06C' name = 'getPrice' visibility = 'public'> <UML:Operation
xmi.id = 'x232:07C' name = 'setPrice' visibility = 'public'>
<UML:Operation xmi.id = 'x232:09C' name = 'newPrice' visibility =
'public'> </UML:Operation>...
<UML:Class xmi.id = 'x235' name = 'Price'><UML:Classifier.feature>
<UML:Attribute xmi.id = 'x235:81H' name = 'wholesalePrice' visibility
= 'public'>
<UML:Attribute xmi.id = 'x235:83H' name = 'retailPrice' visibility =
'public'>
</UML:Attribute>...
<UML:Class xmi.id = 'x237' name = 'Description'>
<UML:Classifier.feature>
<UML:Attribute xmi.id = 'x237:81B' name = 'descriptionOfMarketing'
visibility = 'public'>
<UML:Attribute xmi.id = 'x237:83B' name = 'photography' visibility =
'public'>
<UML:Attribute xmi.id = 'x237:85B' name = 'technicalDescription'
visibility = 'public'>
<UML:Attribute xmi.id = 'x237:87B' name = 'scheme' visibility =
'public'>
<UML:Attribute xmi.id = 'x237:89B' name = 'video' visibility =
'public'>
</UML:Attribute>...
<UML:Operation xmi.id = 'x237:01C' name = 'getPhotography' visibility
= 'public'> <UML:Operation xmi.id = 'x237:02C' name = 'setPhotography'
visibility = 'public'> <UML:Operation xmi.id = 'x237:05C' name =
'newPhotography' visibility = 'public'> <UML:Operation xmi.id =
'x237:06C' name = 'getVideo' visibility = 'public'> <UML:Operation
xmi.id = 'x237:07C' name = 'setVideo' visibility = 'public'>
<UML:Operation xmi.id = 'x237:09C' name = 'newVideo' visibility =
'public'> </UML:Operation>...
<UML:Class xmi.id = 'x239' name = 'Video'><UML:Classifier.feature>
<UML:Attribute xmi.id = 'x239:81H' name = 'url' visibility = 'public'>
<UML:Attribute xmi.id = 'x239:83H' name = 'tag' visibility = 'public'>
</UML:Attribute>...
<UML:Class xmi.id = 'x241' name =
'Photography'><UML:Classifier.feature>
<UML:Attribute xmi.id = 'x241:81H' name = 'size' visibility =
'public'>
<UML:Attribute xmi.id = 'x241:83H' name = 'title' visibility =
'public'>
</UML:Attribute>...
<UML:Association xmi.id='871'> <UML:Association.connection>
<UML:AssociationEnd xmi.id='891' aggregation='none'>...
<UML:AssociationEnd.participant>
<UML:Class xmi.idref='x232' />
</UML:AssociationEnd.participant></UML:AssociationEnd>
```

```
<UML:AssociationEnd xmi.id = '874'  
aggregation='none'><UML:AssociationEnd.participant>  
<UML:Class xmi.idref='x235'>...  
<UML:Association xmi.id='868'> <UML:Association.connection>  
<UML:AssociationEnd xmi.id='889' aggregation='none'>...  
<UML:AssociationEnd.participant>  
<UML:Class xmi.idref='x232' />  
</UML:AssociationEnd.participant></UML:AssociationEnd>  
<UML:AssociationEnd xmi.id = '877'  
aggregation='none'><UML:AssociationEnd.participant>  
<UML:Class xmi.idref='x237'>...  
<UML:Association xmi.id='872'> <UML:Association.connection>  
<UML:AssociationEnd xmi.id='891' aggregation='none'>...  
<UML:AssociationEnd.participant>  
<UML:Class xmi.idref='x237' />  
</UML:AssociationEnd.participant></UML:AssociationEnd>  
<UML:AssociationEnd xmi.id = '879'  
aggregation='none'><UML:AssociationEnd.participant>  
<UML:Class xmi.idref='x239'>...  
<UML:Association xmi.id='874'> <UML:Association.connection>  
<UML:AssociationEnd xmi.id='893' aggregation='none'>...  
<UML:AssociationEnd.participant>  
<UML:Class xmi.idref='x237' />  
</UML:AssociationEnd.participant></UML:AssociationEnd>  
<UML:AssociationEnd xmi.id = '882'  
aggregation='none'><UML:AssociationEnd.participant>  
<UML:Class xmi.idref='x241'>...
```

El archivo generado en formato XMI es transportable, y puede ser utilizado por herramientas de diagramación UML que permitan la importación de código XMI, como es el caso, por ejemplo, de las herramientas ArgoUML, StarUML, BOUML, Visual Paradigm, Circa y Mia-Generation, entre otras, ampliamente utilizadas en la actualidad.

d) Diagramas de Clases UML

En la Figura 21, se puede apreciar la visualización final de las clases UML conceptuales que se generan. El diagrama de clases está compuesto por una clase raíz Component que contiene los atributos *partNumber*, *partName*, *partType*, *description* y *price*. Los métodos creados para esta clase corresponden a la relación directa con las clases *Description* y *Price*. A su vez, la clase *Description* tienen relación directa con las clases *Video* y *Photography*. Esta generación se hace en base a las reglas de transformación que facilitan la automatización del proceso.

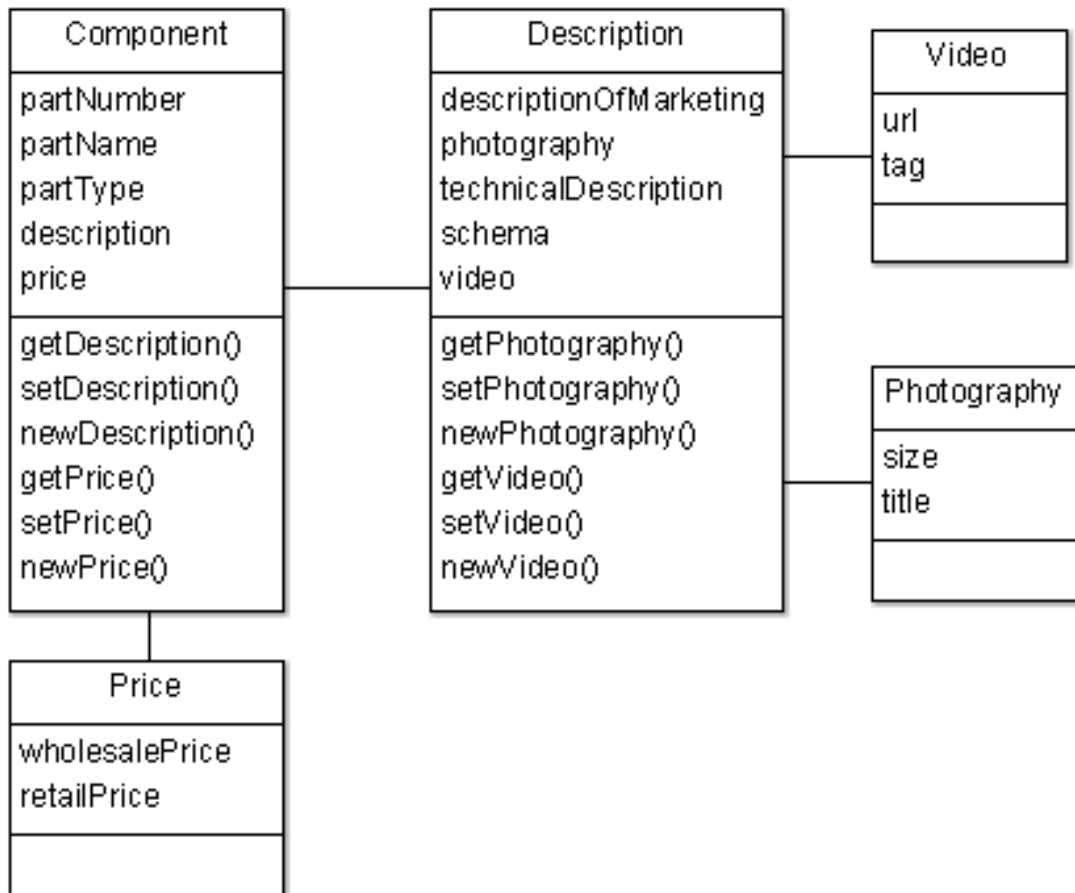


Figura 21: Diagrama de clases UML resultante del análisis de la AI del ejemplo propuesto

Sin embargo, podría ser necesario, para una solución de diseño más concreta, establecer el tipo de visibilidad, o incluso establecer relaciones de composición (agregación fuerte) entre ellas. Esto puede suceder en relaciones entre clases como `Price` y `Description` con `Component`, ya que probablemente no tenga sentido que exista el precio sin el componente (es decir, no tiene sentido que exista la parte sin el todo). El tipo y nivel de dependencia entre relaciones, además de la visibilidad y navegabilidad, pueden ser modificados por el analista o ingeniero software mediante las reglas configurables de InterArch. La Figura 22 corresponde a una adaptación del diagrama de clases UML mediante el uso del módulo de Reglas de Configuración de InterArch, en la cual se ha establecido:

- Una agregación de tipo *compuesto* (*Composite*) para las asociaciones entre las clases `Component-Description`, `Component-Price`, `Description-Video` y `Description-Photography`.
- La visibilidad de las clases del tipo paquete (*Package*).
- La visibilidad de los atributos del tipo protegida (*Protected*).
- La visibilidad de los métodos del tipo privada (*Private*).

- Las asociaciones entre las clases son de tipo *navegable*.

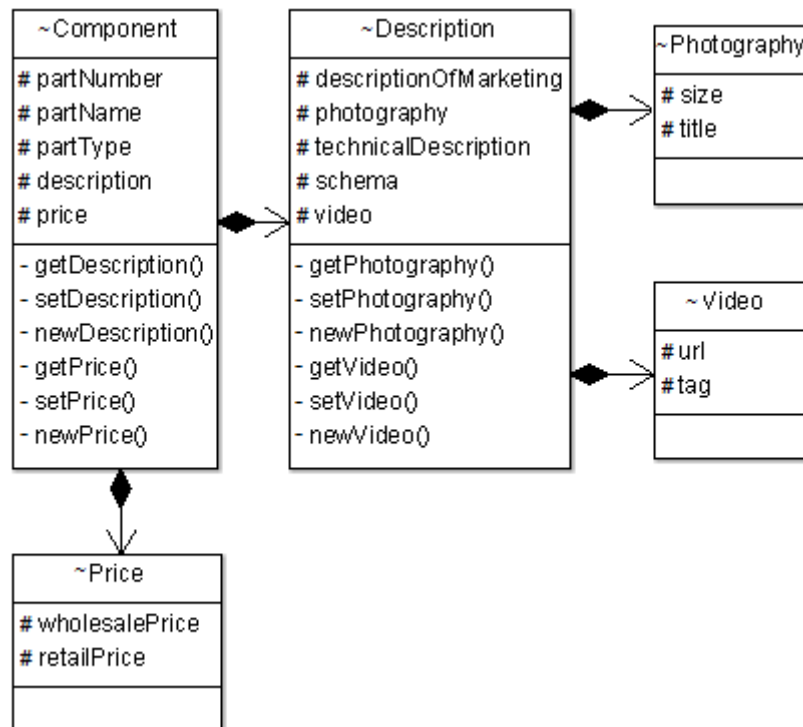


Figura 22: Adaptación del diagrama de clases UML con el módulo de Reglas de Configuración de InterArch

Como se puede observar en la Figura 22, las adaptaciones realizadas mediante el módulo de Reglas de Configuración de InterArch son aplicadas a todos los objetos (clases, atributos, métodos y asociaciones) del diagrama de clases UML. Por ejemplo, la configuración de una agregación de tipo compuesta (*Composite*) y la navegabilidad para las asociaciones entre las clases, es aplicada a las cuatro asociaciones que existen en el diagrama de clases UML. Del mismo modo, la visibilidad de clases, atributos y métodos también se aplican a todos estos elementos.

Por otro lado, para llegar a una solución de diseño, más cercana al dominio de la solución, de la que se beneficien los ingenieros del proyecto, es posible especificar otro tipo de propiedades tales como la cardinalidad de las asociaciones o el tipo de datos de los atributos, entre otras, a través de la edición del archivo XMI generado por InterArch en una herramienta de diagramación UML. En el siguiente apartado se describe un caso de adaptación del diagrama de clases UML mediante el uso de la herramienta de diagramación UML ArgoUML.

e) Adaptación de Diagramas de Clases UML

La herramienta InterArch genera diagramas de clases UML en formato XMI, estableciendo por defecto características específicas respecto al tipo y el nivel de dependencia entre las relaciones, la visibilidad de las clases, atributos y métodos. Estas propiedades pueden ser modificadas a través del módulo de Reglas de Configuración y

Solución Propuesta

así poder cambiar esta generación por defecto que realiza la herramienta InterArch. También, estos diagramas de clases UML en formato XMI pueden ser importados por herramientas CASE de modelado para continuar con las fases y actividades del proyecto. Una de estas actividades corresponde a adaptar los diagramas de clases UML a requerimientos específicos de analistas e ingenieros software para una solución de diseño más detallada en base a las especificaciones del proyecto.

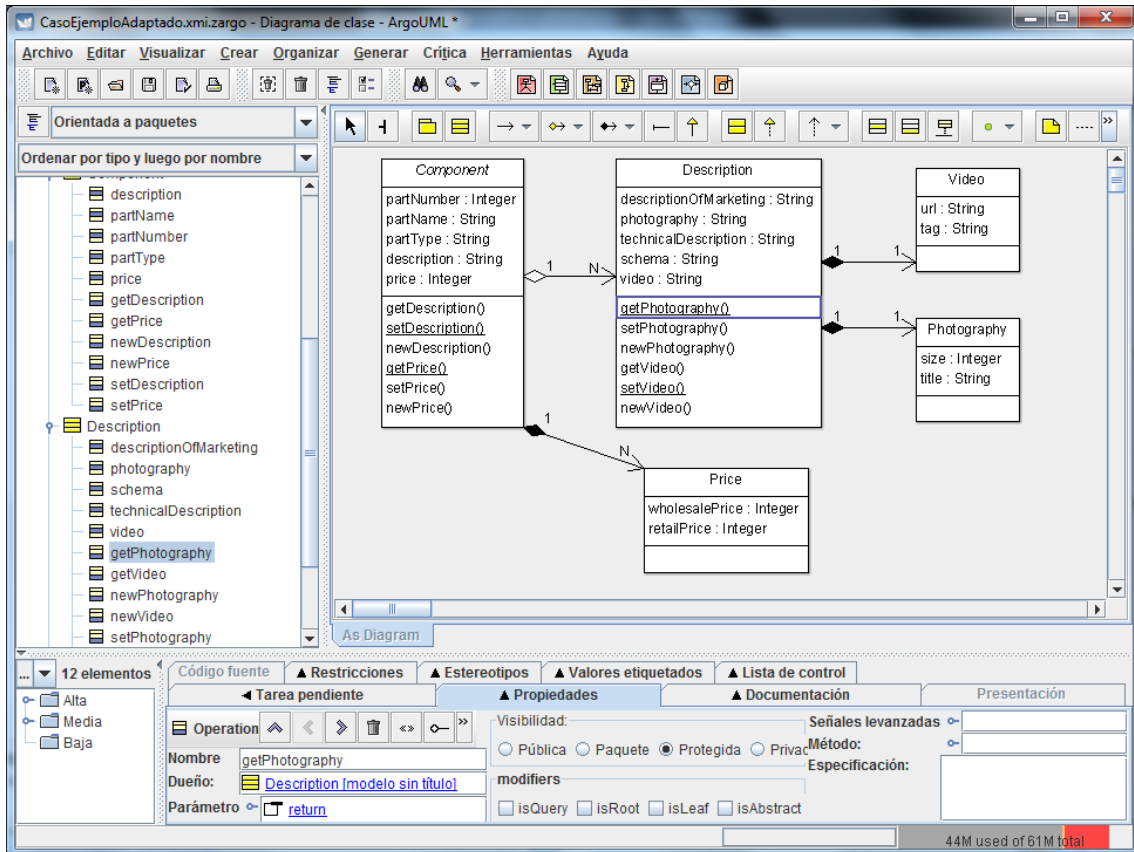


Figura 23: Adaptación del diagrama de clases UML con ArgoUML

La Figura 23 muestra una adaptación del diagrama de clases UML inicialmente generado por la herramienta InterArch. Esta adaptación fue realizada a través de la importación del archivo XMI, que genera InterArch, en la herramienta ArgoUML. ArgoUML es una herramienta CASE de modelado, con un entorno gráfico que soporta el diseño, desarrollo y documentación de las aplicaciones de software (Tigris.org, 2009). En este caso particular, a través del uso de la herramienta ArgoUML, se han configurado las siguientes propiedades:

- Se ha definido el tipo de dato para todos los atributos del diagrama de clases UML. Por ejemplo: en la clase *Component* se ha definido como tipo de dato entero a los atributos *partNumber* y *price*, y como tipo de dato de texto a los atributos *partName*, *partType* y *description*.
- La asociación entre las clases *Component* y *Description* se ha cambiado al tipo agregación (Aggregation).

- Se ha modifica el tipo de visibilidad de los métodos `setDescription`, `getPrice`, `getPhotography` y `setVideo`.
- Se ha establecido la cardinalidad de las asociaciones entre las clases `Component-Description`, `Component-Price`, `Description-Video` y `Description-Photography`.

Recapitulación

A lo largo de este capítulo, se ha presentado un formalismo basado en la representación y Arquitectura de la Información y su posterior transformación en información de análisis y diseño. Para poner en práctica este formalismo y validarlo, se ha desarrollado la herramienta InterArch que permite generar de forma automática información de análisis y diseño para analistas e ingenieros software a partir de un modelado de contenido elaborado por el Arquitecto de la Información.

Adicionalmente, la información generada por la herramienta puede ser adaptada en términos generales a través de las opciones de configuración que se incluyen en InterArch. También, es posible ajustar la información de análisis y diseño a partir de requerimientos más específicos de analistas e ingenieros software mediante la manipulación del archivo XMI en otras herramientas CASE de modelado. Esto permite además aprovechar otras ventajas con las que cuentan estas herramientas, como son la ingeniería inversa de código fuente, la incorporación de base de datos y las restricciones OCL para clases, entre otras, aportando un valor añadido. En cualquier caso, InterArch hace de puente entre estas especificaciones, aportando un entorno para el Arquitecto de la Información y, a nivel de customización, para el Ingeniero del Software.

Esta implementación de la metodología de gestión de conocimiento conceptual del dominio del problema, y su posterior transformación en un modelo del dominio cercano a la solución, permite validar los planteamientos iniciales y corroborar la hipótesis de partida H_1 . No obstante, se hace también necesario proporcionar características de facilidad de uso y aprendizaje en la manipulación de la herramienta InterArch. Es por esto que, en el siguiente capítulo, se realiza una evaluación con usuarios reales para determinar el grado de usabilidad de la herramienta CASE InterArch que permitan corroborar la última hipótesis de partida H_2 en la que se basa este trabajo.

Capítulo 4: Experimentación

En este capítulo se describe la experimentación llevada a cabo para medir el grado de usabilidad alcanzado en la herramienta InterArch. En la sección 4.1, se describen las técnicas de evaluación y tareas para evaluar la usabilidad de la herramienta. En la sección 4.2, se lleva a cabo un análisis de los diferentes resultados obtenidos tanto de la sesión experimental con los 12 usuarios, como del cuestionario de usabilidad. Finalmente, en la sección 4.3 se realiza una discusión respecto a los resultados generales obtenidos de la experimentación.

4.1. *Técnicas de Evaluación Utilizadas*

Desarrollar sistemas informáticos exige generar productos finales que cumplan tanto características de funcionalidad como de usabilidad. Las deficiencias en las características de usabilidad del sistema informático pueden derivar de forma segura en un rechazo por parte de los usuarios y el posterior fracaso del sistema informático en sí.

Para comprobar que un sistema informático contiene características de usabilidad, se utilizan métodos de evaluación. “La evaluación comprende un conjunto de metodologías y técnicas que analizan la usabilidad de un sistema interactivo en diferentes etapas del ciclo de vida” (Abascal et al., 2001). Los métodos habituales de evaluación de la usabilidad se clasifican en métodos de *Inspección*, *Indagación* y *Test*.

Para efectuar la evaluación de la herramienta CASE InterArch, se han seleccionado las técnicas de evaluación *Test Retrospectivo* y *Thinking Aloud*. Estas dos técnicas de evaluación se encuentran clasificadas dentro de los métodos de evaluación de *Test*, en donde los usuarios representativos trabajan en tareas utilizando el sistema, y los evaluadores utilizan los resultados para ver cómo la interfaz de usuario responde a los usuarios en sus tareas.

La primera técnica de evaluación seleccionada corresponde a *Test Retrospectivo*, la cual consiste en revisar los registros de videos almacenados durante la realización del test a los usuarios. Esto permite recopilar información adicional mediante la revisión de la grabación del test de usuario (Nielsen, 1993), permitiendo adquirir más información de cada test realizado. Una de las desventajas es que se requiere tiempo, debido a que se deben revisar por los menos dos veces cada grabación de test.

La segunda técnica de evaluación seleccionada es la denominada *Thinking Aloud*. Esta técnica, introducida por Clayton Lewis (Lewis, 1982), es considerada una de las más valiosa en ingeniería de la usabilidad (Holzinger, 2005). Se trata de tener al usuario final continuamente pensando en voz alta mientras se utiliza el sistema. Al verbalizar los pensamientos de los usuarios, la técnica permite entender cómo ven ellos el sistema computacional, lo que hace que sea más fácil identificar conceptos erróneos (Nielsen, 1993). Para la implementación de esta técnica de evaluación, se solicitó a los usuarios participantes que indicaran en voz alta sus sentimientos y opiniones mientras interactuaban con la herramienta InterArch en la realización de una tarea específica.

Para la consecución de estas técnicas de evaluación, se requería capturar la interacción que ocurriera en la pantalla durante la interacción con la herramienta InterArch, registrando además el audio y vídeo de los usuarios durante toda la sesión experimental. En base a estas exigencias, se ha utilizado la aplicación Camtasia Studio (Camtasia-Studio, 2011), que cumple con estos requisitos específicos.

Para la evaluación de la herramienta se utilizaron 12 personas que trabajan de forma habitual en empresas y consultoras de proyectos informáticos, específicamente en tareas concernientes a la Arquitectura de la Información. Fueron 9 hombres y 3 mujeres, con edades comprendidas entre 24 y 43 años (Media = 32, Desviación = 8.062¹), la dispersión alta en las edades de los usuarios permite indicar que la muestra de la población no estuvo sesgada a un estrato específico de edades, esta diversidad en las edades de los usuarios puede considerarse como un aspecto positivo para la realización de la evaluación de la herramienta. En general, los usuarios contaban con conocimiento de partida sobre análisis y documentación en la estructuración y categorización de contenidos de sitios Web.

Para llevar a cabo la experimentación con los usuarios, se llevaron a cabo los siguientes pasos:

1. En primer lugar, para cada usuario individualmente, se realizó una demostración-tutorial de las distintas funcionalidades y características de la herramienta InterArch que tuvo una duración promedio de 7 minutos y 48 segundos, con una desviación de 2 minutos y 22 segundos.
2. A continuación, ya como parte de la tarea a realizar en el experimento, se solicitó a los usuarios ponerse en el papel del Arquitecto de la Información y realizar un modelo de contenido relacionado con el proceso actual de creación de registros de anuncios de una empresa de anuncios de productos de segunda mano. El proceso actual de registros de anuncios, se efectúa en base a un registro por formulario impreso, el cual permite obtener información sobre el detalle del anuncio y los datos particulares del anunciante. En concreto, se dispuso de una “maqueta” de fidelidad media sacada de uno de los anuncios de la versión impresa de la revista Segunda Mano y se solicitó crear el modelo de contenido utilizando la herramienta InterArch (para mayor información sobre esta tarea, ir a la sección 4.1.1 del presente capítulo). Los usuarios fueron grabados en video y audio en cada sesión. Esto permitió contar con un registro tanto de las interacciones con la herramienta InterArch como de las expresiones y comentarios efectuados durante la realización de la tarea.
3. Finalmente, al terminar cada sesión con los usuarios, se les solicitó completar un cuestionario para medir la usabilidad de la herramienta InterArch. El cuestionario usado está basado principalmente en el cuestionario USE (Lund, 2001), con algunas variaciones proporcionadas por el cuestionario de Percepción de Utilidad y Facilidad de Uso de Davis (Davis, 1989), así como el Cuestionario

¹ Se utiliza el punto (.) como separador de decimales

de Usabilidad de Purdue (Lin et al., 1997). El cuestionario completado por los usuarios, fue dividido en 6 partes (ver Anexo A):

- Una primera parte que contiene preguntas para obtener información sobre el perfil del usuario.
- Una segunda, tercera, cuarta y quinta parte, correspondientes a preguntas cerradas para medir las variables de *Utilidad*, *Facilidad de Uso*, *Facilidad de Aprendizaje* y *Satisfacción* de la herramienta InterArch, respectivamente, lo que se corresponde con las 4 dimensiones medidas para la estimación de la usabilidad.
- Una última parte con preguntas abiertas enfocadas a obtener información sobre aspectos tanto positivos como negativos de la herramienta InterArch, y sobre posibles tareas de labor cotidiana que podrían ser realizadas por los usuarios con el uso de la herramienta InterArch.

4.1.1. Descripción de la Tarea Propuesta para Evaluar la Herramienta InterArch

A cada usuario del experimento, se le proporcionó el siguiente enunciado ficticio para que se metiera en el papel del Arquitecto de la Información y pudiera realizar la educación de requisitos no funcionales correspondientes sobre la información:

Una revista de anuncios de productos de segunda mano, que en la actualidad se distribuye a partir de su versión impresa, quiere cambiar a una versión digital. Inicialmente, la empresa pretende modernizar su proceso actual de creación de registros de anuncios. El registro de anuncios se efectúa en base a un registro por formulario impreso, que permite obtener información del detalle del anuncio y de los datos particulares del anunciante. Más específicamente, la empresa requiere que el actual registro de anuncios sea a través de una página Web que incluya información sobre la provincia, categoría y tipo de anuncio, junto con el nombre, email y teléfono del anunciante, así como datos específicos del anuncio referentes al título, descripción, precio (fijo, negociable o gratis), video e imagen del producto.

El trabajo se ha encargado a la empresa informática donde usted trabaja como Arquitecto de la Información. A partir de la maqueta proporcionada, que está sacada de uno de los anuncios de la versión impresa de la revista de venta Segunda Mano (ver Figura 24), deberá trabajar en la parte de modelado conceptual, llevando a cabo las siguientes tareas:

- *Segmentación del contenido del registro de anuncios.*
- *Creación de un modelo de contenido, utilizando la herramienta InterArch.*

Ad Registration Form	
City	Pulaski
Category	Sport
Ad Type	For Sale
Ad Interested	Particular
Advertiser Details	
Name	John Perez
E-mail	John-perez@email.com
Phone	85558548
Ad Details	
Title	Bicycle for children
Description	Perfect conditions: Damping front and center.
Price	
Fixed Price	90 US
Negotiable	No
Gift/Free	No
Image	
Video	

Figura 24: Maqueta de registro de anuncios para productos de segunda mano.

A modo ilustrativo, se ha elaborado con la herramienta InterArch, el posible modelo de contenidos esperado que resultaría de la tarea propuesta a los usuarios. En la Figura 25, se muestra la composición de este modelo de contenido, el cual está compuesto principalmente por el elemento Ad Registration Form, que a la vez está compuesto por los elementos City, Category, Advertiser Details, Ad Details, Ad Type y Ad Interested. El elemento Advertiser Details está compuesto por los elementos Name, E-mail y Phone. El elemento Ad Details está compuesto por los elementos Title, Description, Price, Image y Video. Finalmente, el elemento Price está compuesto por los elementos Fixed Price, Negotiable y Gift/Free.

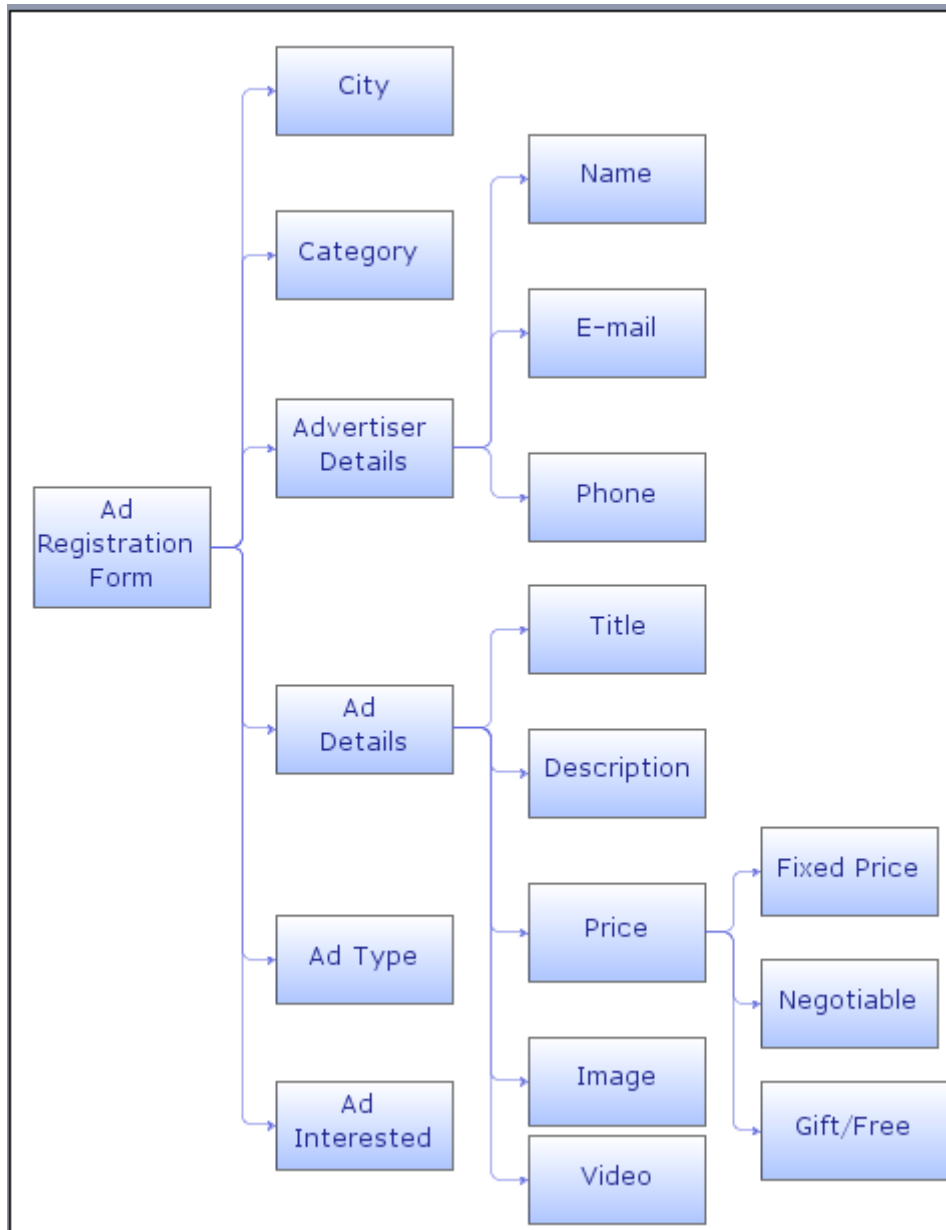


Figura 25: Modelo de contenido de registro de anuncios para productos de segunda mano

Del mismo modo, en la Figura 26 se muestra el diagrama de clases UML que resultaría al aplicar el conjunto de reglas de transformación de la herramienta InterArch al modelado de contenido de la Figura 25. Este diagrama de clases UML está compuesto por las clases `AdRegistrationForm`, `AdDetails`, `Price` y `AdvertiserDetails`, en donde cada clase está compuesta por atributos, operaciones y asociaciones de acuerdo al conjunto de reglas de transformación descritas en la sección 3.4 del capítulo 3. En cuanto a las Reglas de Configuración que actúan por defecto, éstas generan clases, atributos y métodos con una visibilidad de tipo *public*. Además, las asociaciones entre clases se generan sin navegabilidad ni agregación entre ellas.

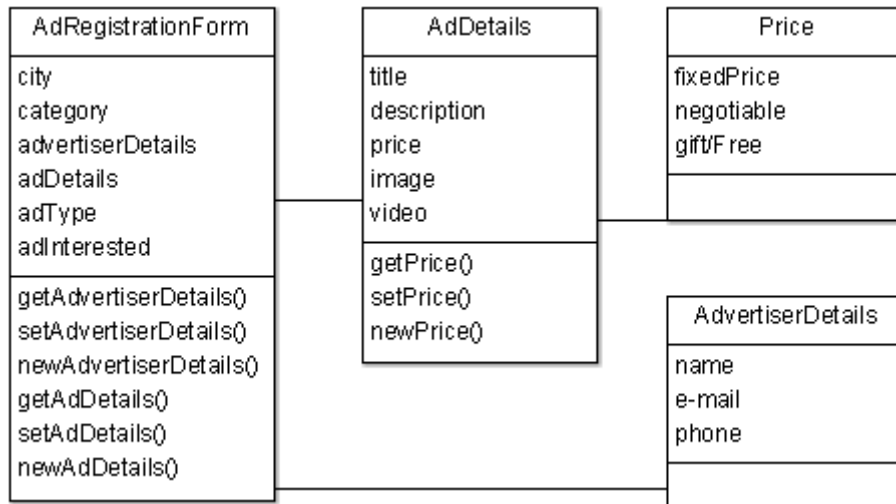


Figura 26: Diagrama de clases UML de registro de anuncios para productos de segunda mano

4.2. Resultados Obtenidos

En esta sección se describen y analizan los diferentes resultados obtenidos de la sesión experimental con los 12 usuarios y del cuestionario de usabilidad.

4.2.1. Resultados de la Sesión Experimental

En esta sección se describen los distintos resultados de la sesión experimental, los cuales se presentan en base al orden de actividades establecidas para el proceso de evaluación de la herramienta InterArch.

a) Demostración-tutorial de las funcionalidades y características de la herramienta InterArch

La evaluación de la herramienta InterArch se inició con una introducción, a modo de tutorial, dirigida al usuario. Dicha introducción abarcaba las principales funcionalidades y características de la herramienta InterArch. Esta demostración tuvo una duración promedio de 7 minutos y 48 segundos, con una desviación de 2 minutos y 22 segundos. El usuario que más tiempo tardó lo hizo en 14 minutos y 3 segundos, mientras que el menos tardó lo hizo en 5 minutos y 23 segundos. Esta etapa tuvo una duración total, con los 12 usuarios, de 1 hora, 33 minutos y 41 segundos. Las diferencias entre los tiempos máximos y mínimos, así como el resultado de la dispersión de los tiempos empleados en esta tarea, son reflejo del nivel de interacción que se tuvo con los usuarios, donde en algunos casos se produjeron preguntas respecto a ciertas características de la herramienta. En otros casos, y en función del grado de asimilación personal, otros usuarios apenas hicieron preguntas.

El objetivo principal de esta introducción, fue familiarizar a los usuarios con las funcionalidades básicas con que cuenta la herramienta InterArch, de manera que adquirieran un conocimiento en términos generales de las capacidades que

Experimentación

ofrece la herramienta. Los usuarios no manifestaron mayores dudas respecto a las funcionalidades y características de la herramienta InterArch, debido principalmente a que InterArch replicaba muchas de las características de herramientas similares empleadas por Arquitectos de la Información. Algunas de estas características corresponden a:

- La estructura de diseño de la herramienta InterArch cuenta con una paleta de elementos visuales, un entorno de trabajo y opciones de formato.
- La manipulación de los elementos se inicia al arrastrar los elementos al entorno de trabajo desde la paleta de elementos visuales.
- Los elementos de contenido se relacionan mediante el uso de los elementos de enlace o de forma directa en base a seleccionar y arrastrar desde el elemento de contenido de origen hasta el elemento de contenido de destino.
- Los elementos del diagrama pueden ser agrupados y manipulados en bloque.
- Los elementos pueden ser creados mediante mecanismos de herencia, seleccionando y arrastrando los elementos creados en el entorno de trabajo.
- El control de errores se consigue a base a deshacer y rehacer acciones. Es decir, los usuarios pueden volver a estados anteriores y rehacer acciones a través de estas funcionalidades. Esto se puede conseguir a través de los iconos de la barra de herramientas, o directamente con la combinación de las teclas Ctrl+Z (deshacer) y Ctrl+Y (rehacer).
- La introducción de información en los elementos de contenido, enlace y etiquetado, se realiza al hacer doble clic sobre dichos elementos.

b) Segmentación de contenido

Una vez finalizado el adiestramiento sobre las funcionalidades y características de la herramienta InterArch, la segunda etapa de la sesión experimental con usuarios consistió en entregarles un enunciado que contenía la descripción de la tarea. El enunciado completo se encuentra en la sección 4.1.1, y la primera tarea consistió en realizar una segmentación del contenido en base a la información proporcionada.

La segmentación del contenido tuvo una duración promedio de 7 minutos y 5 segundos, con una desviación de 3 minutos y 12 segundos. El usuario que más tiempo tardó lo hizo en 12 minutos y 24 segundos, y el que menos tardó lo hizo en 2 minutos y 55 segundos. La notable diferencia entre el tiempo máximo y mínimo empleados para llevar a cabo la tarea de segmentación de contenido, se debe principalmente a que 2 usuarios decidieron volcar su esfuerzo en elaborar directamente el modelado de contenido utilizando la herramienta InterArch, dedicando por tanto menos tiempo al preámbulo (2 minutos y 55 segundos, y 3 minutos y 56 segundos), mientras que 3 usuarios le dedicaron una cantidad de tiempo mayor a 10 minutos para llevar a cabo la tarea de segmentación del

contenido (12 minutos y 24 segundos, 11 minutos y 50 segundos, y 10 minutos y 9 segundos). Sin embargo, esta diferencia entre el tiempo máximo y mínimo no muestra una alta dispersión en los tiempos empleado por los usuarios con respecto al tiempo promedio (7 minutos y 5 segundos). Lo anterior, se puede apreciar en el valor de la desviación estándar de 3 minutos y 12 segundos. Esta etapa tuvo una duración total, con los 12 usuarios, de 1 hora, 25 minutos y 2 segundos.

En general, la tarea de segmentación de contenido no exigió un mayor esfuerzo por parte de los usuarios, la cual se efectuó sin mayores inconvenientes y de forma exitosa. Esta tarea, permite a los usuarios identificar oportunidades para reutilizar el contenido y controlar la granularidad de la información, ambos aspectos son fundamentales de cara a abordar la siguiente tarea de crear un modelo de contenido en base a la segmentación realizada.

c) Modelado de contenido utilizando la herramienta InterArch

Como última tarea de la sesión experimental, se solicitó a los usuarios que, a partir de la segmentación de contenido efectuada anteriormente, elaboraran un modelo de contenido utilizando la herramienta InterArch. Los usuarios manipularon sin mayor dificultad, ni esfuerzo, las distintas funcionalidades y características de la herramienta InterArch. Aún cuando el objetivo no era medir la validez de los modelos de contenido creados por los usuarios, sí es interesante indicar que la gran mayoría de estos modelos de contenido corresponden al que se esperaba en base a la tarea planteada y el cual se puede observar en la Figura 25. Lo anterior, permite validar el nivel de conocimiento y grado de experiencia de los usuarios en el área de análisis y categorización de contenidos de sitios Web.

La experiencia de elaboración del modelo de contenido utilizando la herramienta InterArch tuvo una duración promedio de 11 minutos y 54 segundos, con una desviación estándar de 3 minutos y 14 segundos. El usuario que más tiempo tardó lo hizo en 18 minutos y 45 segundos, y el que menos tardó lo hizo en 8 minutos y 40 segundos. Aun existiendo una alta diferencia entre el tiempo máximo y mínimo en la elaboración del modelo de contenido utilizando la herramienta InterArch, la realización de esta tarea no presentó una dispersión significativa en los tiempos empleados con respecto al tiempo promedio (11 minutos y 54 segundos), así lo demuestra la desviación estándar de 3 minutos y 14 segundos. La alta diferencia entre el tiempo máximo y mínimo en la elaboración del modelo de contenido se debió a que algunos usuarios manifestaron mayor habilidad en el manejo de la herramienta InterArch, lo que los condujo a terminar el modelo de contenido en un periodo muy corto de tiempo (tiempo mínimo: 8 minutos y 40 segundos), mientras que los dos usuarios que exigieron mayor tiempo en la consecución de la tarea (18 minutos y 45 segundos, y 17 minutos y 13 segundos) coincidieron casualmente con los 2

Experimentación

usuarios que menos tiempo dedicaron en la tarea de segmentación de contenido (3 minutos y 56 segundos, y 2 minutos y 55 segundos), lo que confirma que la segmentación es un paso previo importante para agilizar el proceso de Arquitectura de la Información. Sin embargo, por lo general, la gran mayoría de los usuarios terminaron la elaboración del modelo de contenido en un tiempo promedio similar.

La tarea de elaborar el modelo de contenido utilizando la herramienta InterArch, tuvo una duración total, con los 12 usuarios, de 2 horas, 22 minutos y 50 segundos.

En cuanto al estilo de manipulación de los elementos visuales en la elaboración del modelo de contenido, los usuarios optaron mayoritariamente (8 usuarios) por generar nuevos elementos de contenido en base a la funcionalidad de seleccionar y arrastrar sobre los elementos de contenido creados (herencia). El resto de los usuarios, para crear nuevos elementos de contenido, emplearon la funcionalidad de arrastrarlos desde la paleta de elementos de la herramienta InterArch, no manifestando mayores dificultades. De hecho, la funcionalidad para crear nuevos elementos visuales que fue empleada por la mayoría de los usuarios (8) presentaba mayor dificultad que la que utilizó la minoría (4). Lo anterior confirma el éxito de la funcionalidad de creación de elementos por herencia que permite crear elementos de contenido enlazados y con características similares a elementos de contenido origen, facilitando y agilizando la forma de crear y enlazar los nuevos elementos.

Otro aspecto a destacar, es que los usuarios manifestaron la necesidad de que la herramienta incluyera una funcionalidad que permita que los elementos visuales del diagrama se organicen de forma automática, en base a su jerarquía y dependencia entre ellos. Cabe señalar que, actualmente, la herramienta cuenta con la funcionalidad de organizar los elementos de enlace de forma automática, lo cual facilita en parte lo requerido por los usuarios. No obstante, la opción de organizar de forma automática los elementos visuales del diagrama, es un aspecto importante a considerar en futuras mejoras de la herramienta InterArch.

En cuanto al control de errores en la manipulación e interacción con la herramienta, se destaca que la gran mayoría de los usuarios hizo uso de la funcionalidad de deshacer acciones mediante la combinación de las teclas Ctrl+Z cada vez que consideraban necesario volver a un estado anterior. Aun cuando era posible también alcanzar esta funcionalidad dentro de las opciones de edición de la herramienta, los usuarios optaron por conseguirla a través de la combinación de las teclas. Estos resultados, permiten reafirmar la utilidad de esta funcionalidad para los usuarios y los habituados que están en utilizarla independiente del entorno de trabajo en que se encuentren interactuando.

En términos generales, se puede indicar como dato, que la sesión completa de interacción con todos los usuarios (demostración de la herramienta, segmentación del

contenido y modelado del contenido utilizando la herramienta InterArch) tuvo una duración total de 5 horas, 21 minutos y 33 segundos. El usuario que más tiempo empleó en la sesión experimental completa lo hizo en 30 minutos y 51 segundos, y el que empleó menos tiempo lo hizo en 19 minutos y 44 segundos, con una desviación estándar de 3 minutos y 55 segundos.

La etapa de la sesión experimental que menos tiempo requirió para llevarse a cabo correspondió a la de **segmentación del contenido**, la cual tuvo una duración total de 1 hora, 25 minutos y 2 segundos, seguida por la de **demostración de la herramienta InterArch**, con una duración total de 1 hora, 33 minutos y 41 segundos. Finalmente, la tarea que más tiempo exigió para fue la de **crear el modelado de contenido utilizando la herramienta InterArch**, con una duración total de 2 horas, 22 minutos y 50 segundos. Lo anterior era de esperar, ya que las fases de demostración de la herramienta y de segmentación del contenido correspondían a etapas que no requerían de un esfuerzo especial, mientras que la etapa de creación del modelo contenido demandaba mayor dedicación de tiempo, dado que exigía manipular e interactuar con la herramienta InterArch en base al conocimiento adquirido en las etapas anteriores. A modo de resumen, se presenta la Figura 27 con los tiempos promedio empleados en las diferentes tareas, como así una barra correspondiente (+/- σ) a la dispersión de cada tarea (desviación estándar):

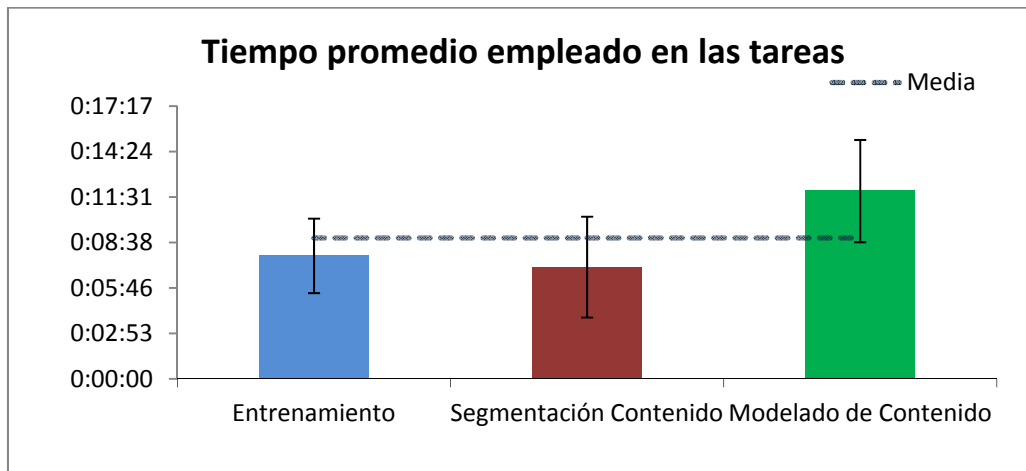


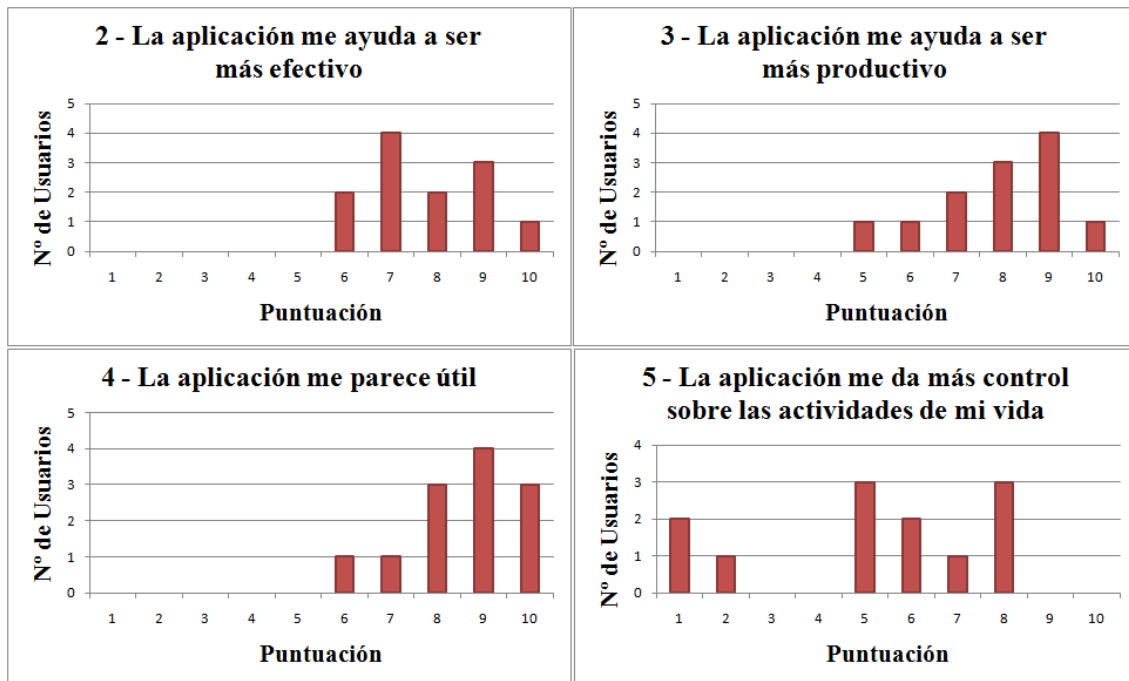
Figura 27: Tiempo promedio empleado por los usuario en las tareas

El uso de la técnica Thinking Aloud en la evaluación de la herramienta InterArch, ha permitido obtener información valiosa respecto a cómo los usuarios utilizan la herramienta, identificando conceptos erróneos a través de los usuarios. De igual forma, la técnica de evaluación de Test Retrospectivo facilitó un análisis más exhaustivo de las interacciones con la herramienta InterArch mediante la revisión de las grabaciones de audio y video de los usuarios. Los resultados obtenidos a través de estas técnicas de evaluación, se utilizarán para mejorar la herramienta, de modo que esta responda de forma más eficiente y en concordancia a las necesidades de los usuarios.

4.2.2. Resultados y Análisis de la Usabilidad

La sesión experimental con los usuarios estaba compuesta por tres etapas: una demostración de la herramienta, una segmentación de contenido y la creación de un modelado del contenido utilizando la herramienta InterArch. Las dos últimas etapas fueron desarrolladas en base a una tarea específica descrita en la sección 4.1.1. Una vez finalizada la sesión experimental, se les solicitó a los usuarios completar un cuestionario para medir la usabilidad de la herramienta InterArch. Este cuestionario se diseñó con 31 preguntas cerradas (ir al Anexo A para mayor información) para medir las variables de: *Utilidad*, *Facilidad de Uso*, *Facilidad de Aprendizaje* y *Satisfacción*. Además, se incluyeron 7 preguntas abiertas con el objetivo de obtener información respecto al perfil de los usuarios, aspectos positivos y negativos de la herramienta InterArch, y sobre posibles tareas de labor cotidiana que podrían ser realizadas con el uso de la herramienta InterArch.

En esta sección, se presentan los resultados del cuestionario agrupados de acuerdo a las diferentes variables medidas (*Utilidad*, *Facilidad de Uso*, *Facilidad de Aprendizaje* y *Satisfacción*). Estas corresponden a preguntas cerradas, puntuadas por los usuarios entre 1 y 10, donde los usuarios distinguen la puntuación 10 como máximo, 1 como mínimo y 5 como valor intermedio. Se ha utilizado este tipo de escala con el objetivo de proveer a los usuarios un mayor grado de libertad en cuanto a la amplitud de las opciones de las evaluaciones, relacionándolo con puntuaciones porcentuales más fáciles de vislumbrar por los usuarios a la hora de evaluar cuantitativamente cada aspecto.



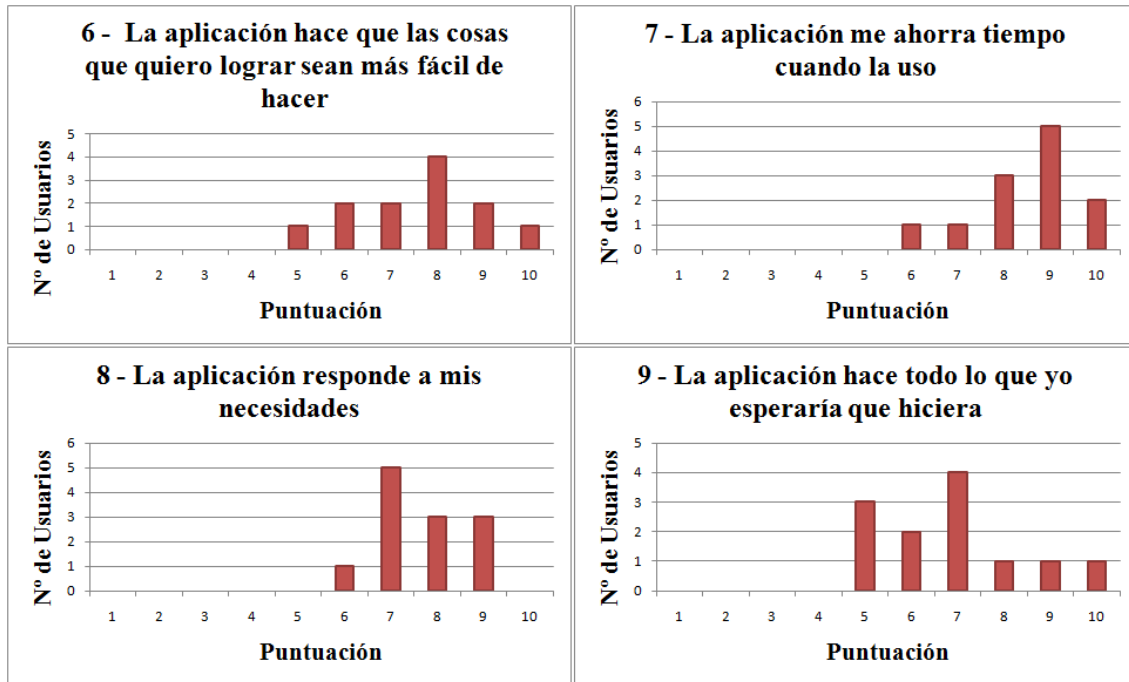


Figura 28: Gráficos con las frecuencias para la categoría *Utilidad*

a) Utilidad

En este apartado se presentan los resultados obtenidos en la medición de la variable *Utilidad* para la herramienta InterArch. La Figura 28, muestra 8 gráficas correspondientes a las 8 preguntas de esta categoría (números 2 a 9 del cuestionario). En cada gráfica, aparece el número de usuarios (frecuencia) que contestaron a cada pregunta en el eje Y, frente a la puntuación (de 1 a 10) en el eje X que los usuarios otorgaron en cada caso.

En términos generales, se puede indicar que la herramienta obtiene puntuaciones altas con respecto a que los usuarios están de acuerdo en que la herramienta InterArch presenta características de utilidad, con una puntuación media de 7.5 y con una desviación de 1.089. De hecho, sólo la pregunta número 5 (“La aplicación me da más control sobre las actividades de mi vida”) presenta puntuaciones dispersas e inferiores a 5 (obtiene una puntuación media de 5.16), lo cual se podría deber a que los usuarios pudieran considerar que las actividades de su vida son mucho más amplias y complejas como para que la aplicación les entregue un mayor control sobre éstas. No obstante, sí se puede concluir que la utilidad alcanzada es positiva en la totalidad de los casos.

Destaca la alta puntuación de las preguntas número 4 (La aplicación me parece útil) y 7 (La aplicación me ahorra tiempo cuando la uso), ambas puntuadas con 8 en adelante por más del 80% de los usuarios (obtienen una puntuación media de 8.58 y 8.5, respectivamente). Esto podría deberse que la herramienta cuenta con las características necesarias que requiere el Arquitecto de la Información para elaborar, en este caso, el modelo de contenido, y que además emplea un conjunto de funcionalidades similares a

Experimentación

otras herramientas, lo que al final se traduce en una mayor utilidad y ahorro en el tiempo que exige el utilizar la herramienta.

La pregunta número 9 (La aplicación hace todo lo que yo esperaría que hiciera) destaca por su concentración del 80% de los usuarios con puntuaciones entre 5 y 7, aún cuando se consideran como puntuaciones positivas (obtiene una puntuación media de 6.83), esto se podría explicar debido a que algunos usuarios manifestaron en la sesión experimental, la necesidad de que la herramienta InterArch incluyera una funcionalidad para organizar de forma automática los elementos visuales del diagrama, lo cual llevo al usuario a considerar que la herramienta no realiza completamente lo que él esperaría. Sin embargo, se destaca que la herramienta no obtiene puntuaciones inferiores a 5, considerándose esto como un aspecto muy positivo, respecto a que la herramienta satisface lo que los usuarios esperan que hiciera.

b) Facilidad de Uso

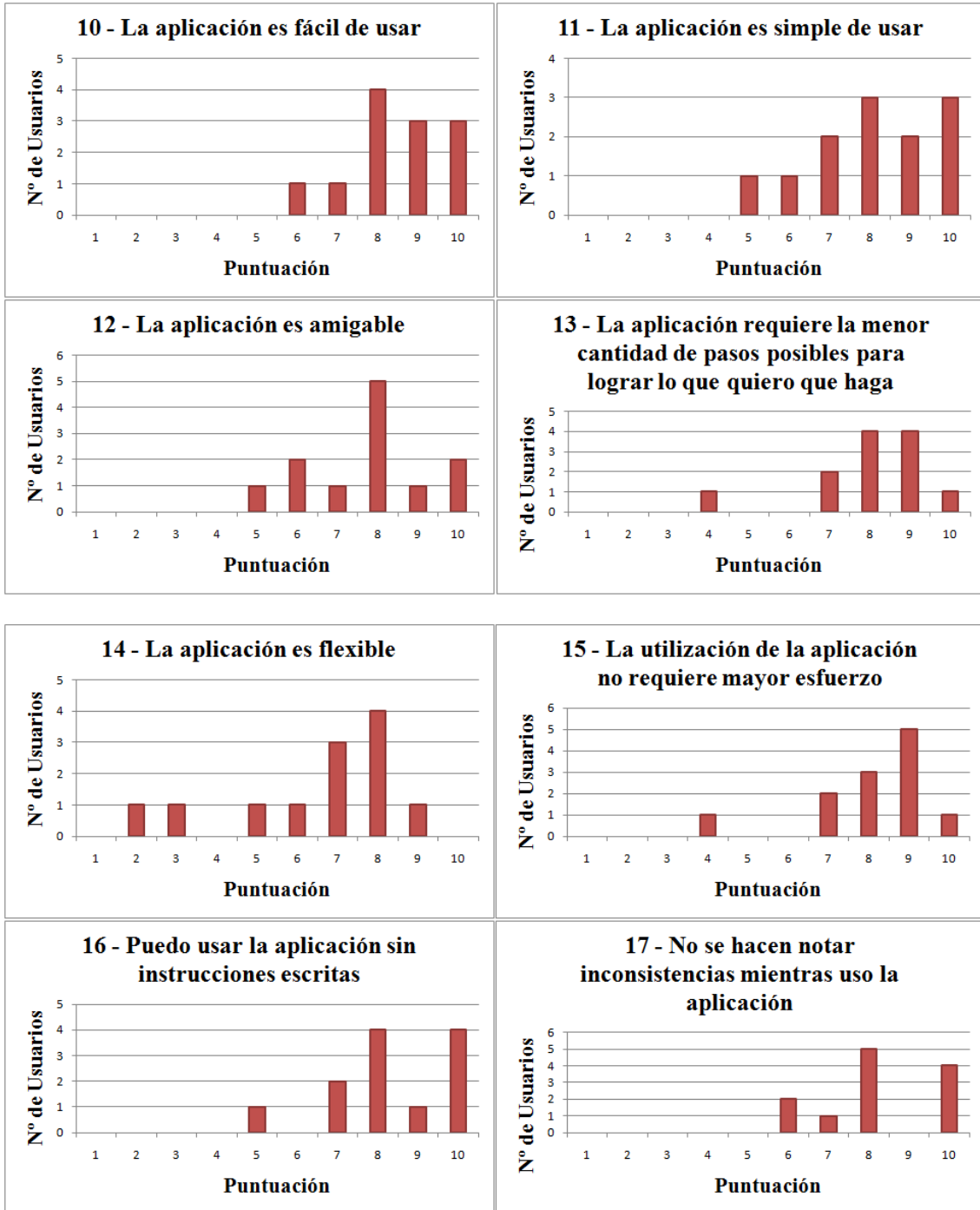
En este apartado se presentan los resultados obtenidos en la medición de la variable *Facilidad de Uso* para la herramienta InterArch. La Figura 29, muestra 10 gráficas correspondientes a las 10 preguntas de esta categoría (números 10 a 19 del cuestionario). En cada gráfica, aparece el número de usuarios (frecuencia) que contestaron a cada pregunta en el eje Y, frente a la puntuación (de 1 a 10) en el eje X que los usuarios otorgaron en cada caso.

En términos generales, se puede indicar que la herramienta obtiene mayoritariamente puntuaciones satisfactorias en todas las preguntas, con una puntuación media de 7.95 y con una desviación de 1.083. Esta vez, en tan sólo 3 preguntas (13, 14 y 15) obtienen puntuaciones inferiores a 5. Sin embargo, en estas 3 preguntas (13, 14 y 15), las puntuaciones sobre 5 corresponden a la mayoría (obtienen puntuaciones medias de 8, 6.5 y 8.08, respectivamente). Lo anterior, permite concluir que el grado de facilidad de uso alcanzado es positivo en la mayoría de los casos.

Las preguntas número 10 (La aplicación es fácil de usar) y 11 (La aplicación es simple de usar) obtienen un 100% de puntuaciones positivas (≥ 5), con puntuaciones de 8 en adelante, en más del 75% de los usuarios (obtienen puntuaciones medias de 8.5 y 8.08, respectivamente). Lo anterior, se podría explicar debido a que los usuarios perciben que la herramienta tiene niveles bajos de dificultad.

La pregunta número 14 (La aplicación es flexible) obtiene 3 puntuaciones menores o igual a 5. Esto se debe a que algunos usuarios en la sesión de interacción manifestaron dificultad en el uso de la funcionalidad para crear elementos de contenido y plantearon la necesidad de contar con otras funcionalidades para esta acción. Lo que al final, se tradujo en que estos usuarios tuvieron la impresión de que la herramienta no contaba con una flexibilidad suficiente. Aún así, esta pregunta obtiene una puntuación de 7 en adelante, por parte del 67% de los usuarios. Esto permite afirmar, en términos generales, que la herramienta cuenta con un nivel alto de aprobación en cuanto a las características de flexibilidad.

Destaca la pregunta número 19 (Puedo usar la aplicación siempre con éxito) con una puntuación de 9 en adelante, por parte del 66% de los usuarios. Estos niveles altos de aceptación en cuanto al uso exitoso de la herramienta, se pueden deber a que no se registraron mayores inconvenientes en la sesión de interacción con los usuarios, logrando que la totalidad de ellos pudiera realizar y terminar la tarea sin percepción de errores por parte de la herramienta. Esto también explica que las puntuaciones fueran de 6 en adelante, por parte del 100% de usuarios.



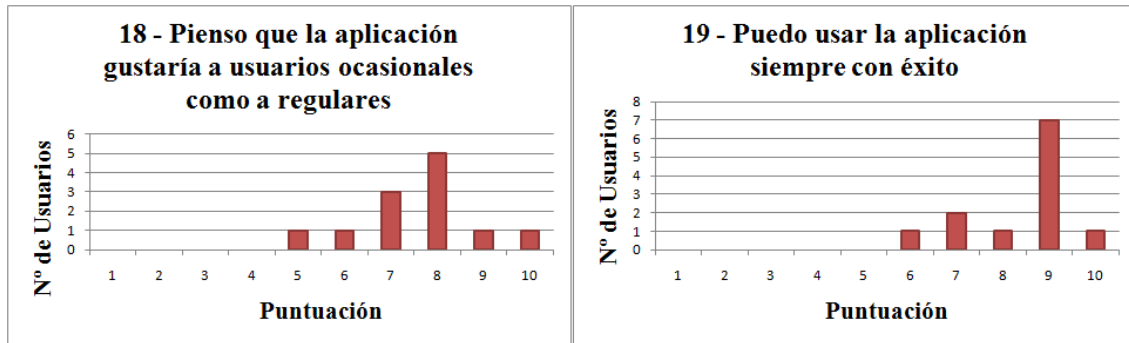


Figura 29: Gráficos con las frecuencias para la categoría *Facilidad de Uso*

c) Facilidad de Aprendizaje

En este apartado se presentan los resultados obtenidos en la medición de la variable *Facilidad de Aprendizaje* para la herramienta InterArch. La Figura 30, muestra 6 gráficas correspondientes a las 6 preguntas de esta categoría (números 20 a 25 del cuestionario). En cada gráfica, aparece el número de usuarios (frecuencia) que contestaron a cada pregunta en el eje Y, frente a la puntuación (de 1 a 10) en el eje X que los usuarios otorgaron en cada caso.

Nuevamente destaca, al igual que en las variables descritas anteriormente, la alta puntuación obtenida en todas las preguntas (la mayoría obtiene una puntuación sobre 5), con una puntuación media de 8.3 y una desviación de 1.23. Lo que permite concluir que el grado de facilidad de aprendizaje alcanzado es positivo en todos los casos.

Las preguntas número 20 (Aprendí a usar la aplicación rápidamente), 21 (Recuerdo fácilmente cómo usar la aplicación) y 22 (Es fácil de aprender a usar la aplicación) obtienen puntuaciones de 8 en adelante (obtienen puntuaciones medias de 9, 8.8 y 8.9, respectivamente), por parte de más del 90% de los usuarios. Esto se explica en parte por el bajo tiempo promedio (7 minutos y 48 segundos) que demandaron los usuarios en el entrenamiento de la herramienta. Y que además, durante la fase de entrenamiento, los usuarios no requirieron de repeticiones en las explicaciones, ni aclaraciones respecto a las funcionalidades de la herramienta, lo que finalmente se tradujo en que los usuarios aprendieran y recordarán de forma fácil y rápida la herramienta.

La pregunta número 24 (No es necesario explorar nuevas características por ensayo y error) obtiene el 58% de sus puntuaciones entre los valores 4 y 7. Esto podría tener una explicación similar a lo que sucede con la pregunta número 9 (La aplicación hace todo lo que yo esperaría que hiciera) de la variable *utilidad*, debido a que se observó que los usuarios que manifestaron la necesidad de contar con la funcionalidad de organizar de forma automática los elementos visuales del diagrama, exploraron por ensayo y error en busca de esta funcionalidad.

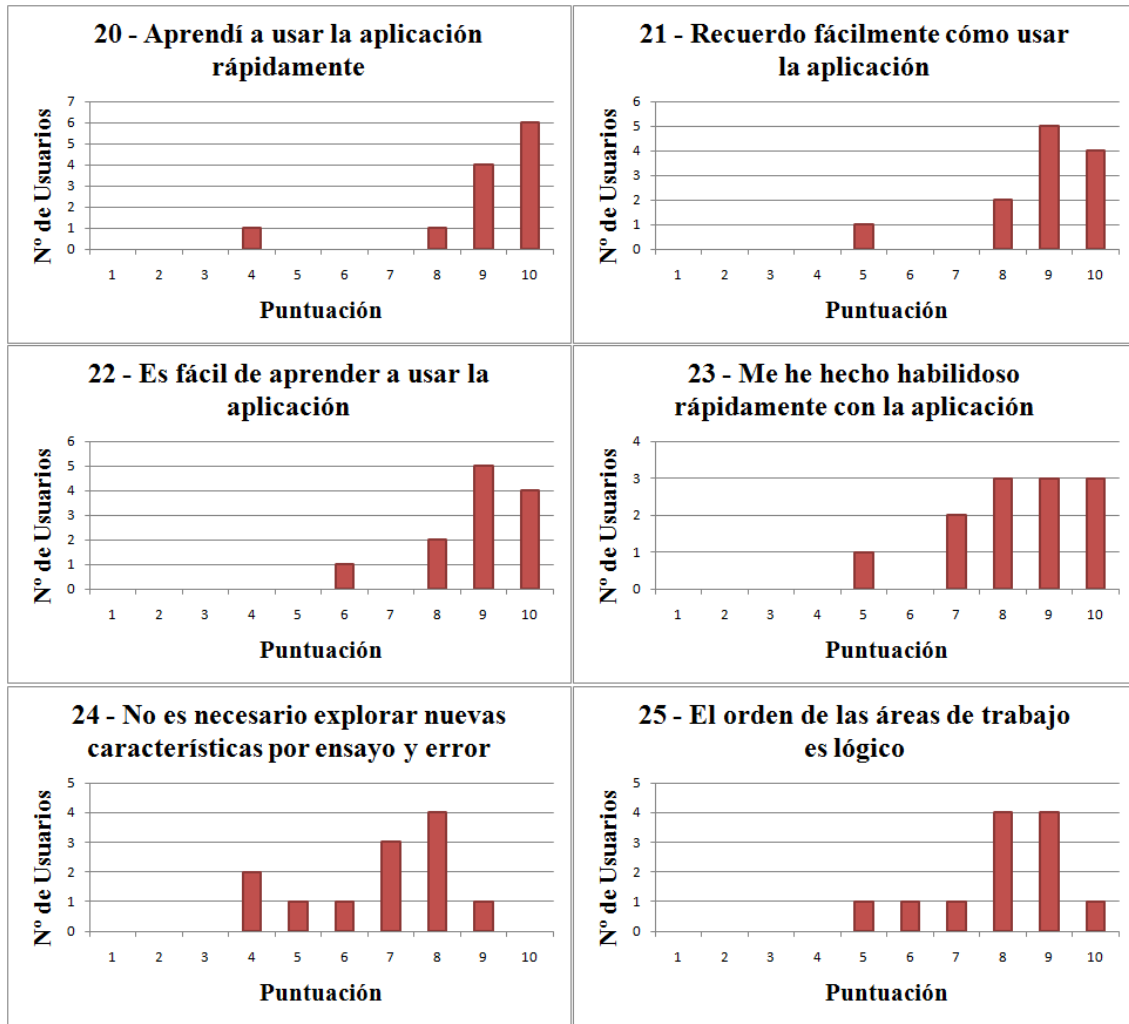


Figura 30: Gráficos con las frecuencias para la categoría *Facilidad de Aprendizaje*

d) Satisfacción

En este apartado se presentan los resultados obtenidos en la medición de la variable *Satisfacción* para la herramienta InterArch. La Figura 31, muestra 7 gráficas correspondientes a las 7 preguntas de esta categoría (números 26 a 32 del cuestionario). En cada gráfica, aparece el número de usuarios (frecuencia) que contestaron a cada pregunta en el eje Y, frente a la puntuación (de 1 a 10) en el eje X que los usuarios otorgaron en cada caso.

Experimentación

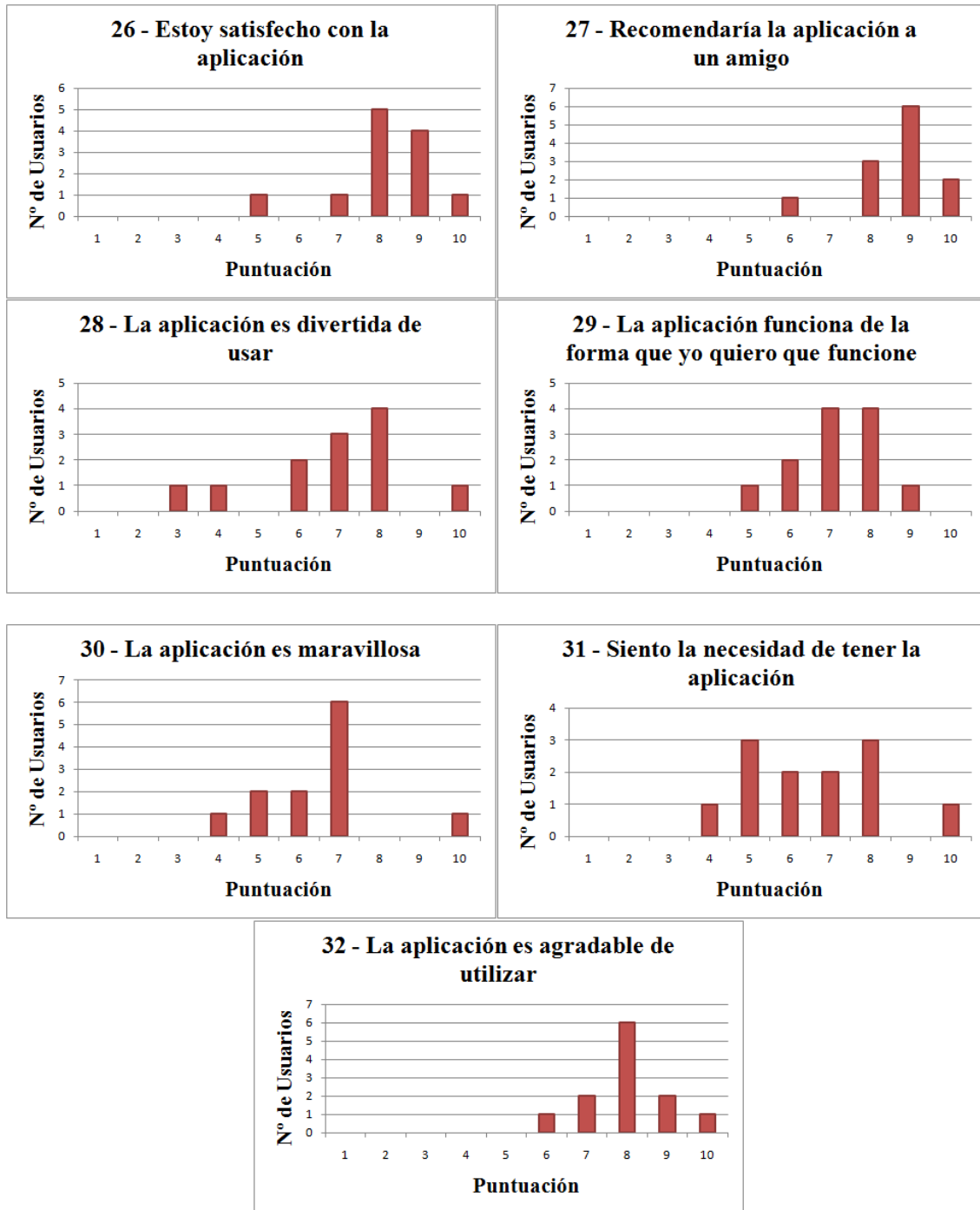


Figura 31: Gráficos con las frecuencias para la categoría *Satisfacción*

A diferencia de las otras variables analizadas, la variable de satisfacción muestra más valores dispersos en las puntuaciones de las distintas preguntas, principalmente en las preguntas números 28, 30 y 31. Aún así, las preguntas obtuvieron puntuaciones altas (la mayoría obtiene una puntuación sobre 5), con una puntuación media de 7.4 y una desviación de 1.05, lo que permite concluir que el grado de satisfacción alcanzado es positivo en la mayoría de los casos.

Las preguntas número 27 (Recomendaría la aplicación a un amigo) y 32 (La aplicación es agradable de utilizar) coinciden en la obtención de puntuaciones de 6 en adelante, en

el 100% de los usuarios (obtienen puntuaciones de 8.6 y 8, respectivamente). Lo cual refleja un alto nivel de satisfacción de los usuarios al manifestar su deseo de recomendar la aplicación y la percepción agradable en el uso de la herramienta. La pregunta número 28 (La aplicación es divertida de usar) obtiene puntuaciones dispersas, es posible que la inclusión del concepto *diversión*, en la pregunta ocasione en el usuario cierta incertidumbre respecto al contexto en que se encuentra. Una situación similar, se corresponde con las preguntas 30 (La aplicación es maravillosa) y 31 (Siento la necesidad de tener la aplicación), las cuales también obtienen puntuaciones dispersas, lo que lleva al usuario a responder de forma neutral en la mayoría de los casos.

4.2.3. Fiabilidad de la Evaluación

La fiabilidad de la evaluación permite validar el nivel de consistencia de una determinada medida, la cual debería mostrar resultados similares, libres de error, en repetidas mediciones. Para medir la fiabilidad de la evaluación realizada a los 12 usuarios, se ha utilizado el indicador Alfa de Cronbach. Este indicador determina la consistencia interna de una escala y toma valores entre 0 y 1; en ocasiones toma valores negativos en los casos donde hay ítems que miden lo opuesto al resto. Cuanto más se acerque el valor calculado a 1, más alta será la fiabilidad de la evaluación, considerándose como un nivel de fiabilidad aceptable entre 0.7 y 0.8, un nivel bueno entre los valores 0.8 y 0.9 y en valores mayores a 0.9 es considerado como excelente.

El indicador de Alfa de Cronbach, se ha calculado para las 31 preguntas cerradas del cuestionario. El resultado muestra un valor de fiabilidad del 95.18% ($\alpha = 0.9518$), lo cual permite indicar que el cuestionario presenta un nivel de fiabilidad excelente, pues supera en un 25.18% el umbral de aceptación. El recíproco del Alfa se utilizará como Nivel de Confianza para efectuar el contraste de hipótesis en el análisis de la varianza (1-Alfa de Cronbach; $\alpha = 0.0482$).

4.2.4. Análisis Multivariable

Hasta al momento, sólo se han presentado los resultados del cuestionario agrupados de acuerdo a las diferentes variables medidas (sección 4.2.2), lo cual no incluye una evaluación orientada a establecer comparaciones entre las variables de forma conjunta. Es por esto que, en esta sección, se presenta un análisis multivariable, el cual tiene como objetivo analizar de forma simultánea las cuatro variables medidas.

Para llevar a cabo este análisis, primero se efectuaron una serie de cálculos con los datos correspondientes a cada una de las variables medidas (*Utilidad, Facilidad de Uso, Facilidad de Aprendizaje y Satisfacción*), tales como la puntuación promedio y la desviación estándar de cada una de las variables y del total de la muestra. La Figura 32 presenta el resultado de los cálculos de las cuatro variables mencionadas anteriormente. Principalmente, la grafica muestra los distintos valores medios de las variables, como así una barra correspondiente (+/- σ) a la dispersión de cada variable (desviación estándar).

Experimentación

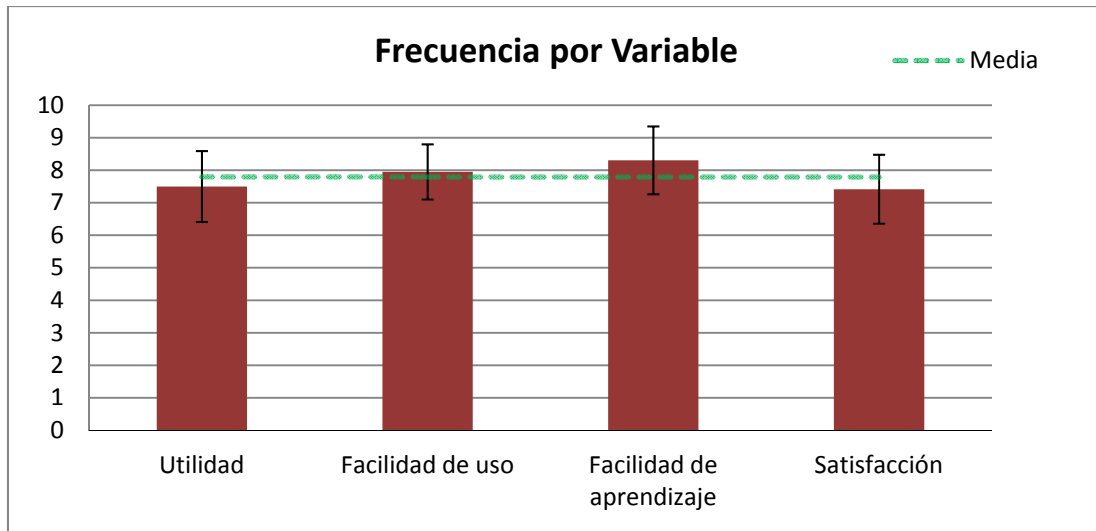


Figura 32: Grafico de frecuencia por variable

En la Figura 32, se puede observar que la variable de *Facilidad de aprendizaje* es la que obtiene la puntuación promedio más alta de toda la muestra, con una media de 8.3 y con una desviación estándar de 1.23, la cual corresponde también a la dispersión más alta entre las variables medidas. La variable de *Facilidad de uso* obtiene el segundo promedio más alto de las puntuaciones medidas, con un promedio de 7.95 y con una desviación estándar de 1.083. Le sigue la variable de *Utilidad*, que obtiene una puntuación promedio de 7.5 y una desviación estándar de 1.089. Finalmente, la variable *Satisfacción* fue la que obtuvo el promedio de puntuaciones más bajo, con un promedio de 7.41 y con una desviación estándar de 1.059, la cual corresponde también a la dispersión más baja entre las variables medidas.

En términos generales, se puede señalar que el promedio de las puntuaciones de toda la muestra es de 7.79, con una desviación estándar de 1.14. Destaca el hecho de que todas las variables obtienen puntuaciones superiores a 7, con lo cual se esta medida se puede considerar como un buen indicador de la usabilidad general de la herramienta InterArch, en base a la percepción de los usuarios.

Debido a que los resultados de las medias de las 4 variables están próximos entre sí, se estudio la posible variabilidad de los valores medios y su posible correlación. Para estudiar la variabilidad, se llevo a cabo una ANOVA de un factor sobre las 4 variables estudiadas. Antes de realizar este cálculo, se analizó si las cuatro variables pueden ser adecuadamente modeladas por una distribución normal. Para realizar este análisis, se utilizo test de Shapiro–Wilk, obteniendo valores de significación de 0.850, 0.241, 0.147 y 0.228 para las variables de *Utilidad*, *Facilidad de Uso*, *Facilidad de Aprendizaje* y *Satisfacción*, respectivamente. En base a estos resultados, se puede concluir que los datos experimentales siguen una distribución normal, debido a que cada variable tiene un valor de significación mayor que 0.0482, correspondiente al nivel aceptado en base al cálculo del recíproco del Alfa de Cronbach, como se explicó anteriormente. Para realizar el estudio de la ANOVA, se planteo como hipótesis inicial que las medias de las 4 variables son similares. Este análisis consiste en poner a prueba la hipótesis inicial

mediante el cálculo del estadístico F_{test} , que indica la proporción entre los estimadores de la varianza poblacional sobre las medias que se están comparando. El resultado del cálculo de la ANOVA obtiene un valor para el estadístico F_{test} de 2.163 y un nivel crítico F_{crit} de 2.848 ($\alpha = 0.0482$; 3; 44; 1). Debido a que F_{crit} es mayor a F_{test} , no es posible rechazar la hipótesis inicial, lo que permite indicar que no hay diferencias entre los 4 variables analizadas, es decir, éstas tienen niveles similares de aceptación entre los usuarios. Lo anterior permite confirmar que, a vista de los resultados del cuestionario, la herramienta presenta cualidades aceptables de usabilidad.

Respecto a la posible correlación de las variables, se realizó un cálculo para medir la correlación (coeficiente de correlación) entre las variables medidas. Este análisis tiene como objetivo medir el grado de intensidad lineal de vinculación de las variables. Para llevar a cabo este cálculo, primero se elaboró una matriz de correlación para estudiar la correlación de todas las variables y así poder seleccionar las variables que estaban más correlacionadas. A continuación, se efectuaron gráficas de dispersión de las puntuaciones promedio de cada una de las variables seleccionadas, confrontando las puntuaciones unas con otras, y calculando el coeficiente de correlación lineal (R^2). Este coeficiente se sitúa entre 0 y 1, y cuanto más cerca esté de 1 mayor relación lineal hay entre las variables estudiadas.

La correlación más alta encontrada fue entre las variables de *Facilidad de uso* y *Facilidad de aprendizaje*, con un coeficiente de correlación lineal en sentido positivo entre las variables de $R^2 = 0.796$. Como se puede observar en la Figura 33, tan sólo un usuario ha entregado una puntuación neutra respecto a la conformidad de estas dos variables, esto se corresponde con el mismo usuario que en promedio entregó la puntuación más baja en todas las variables, con una puntuación promedio de 5.39. Sin embargo, como se muestra en la gráfica, la mayor parte de los usuarios manifiesta niveles altos de conformidad en cuanto a las cualidades de facilidad de uso de la herramienta, la cual se ve incrementada en la medida en que la herramienta ofrece mayores facilidades en el aprendizaje de su utilización.

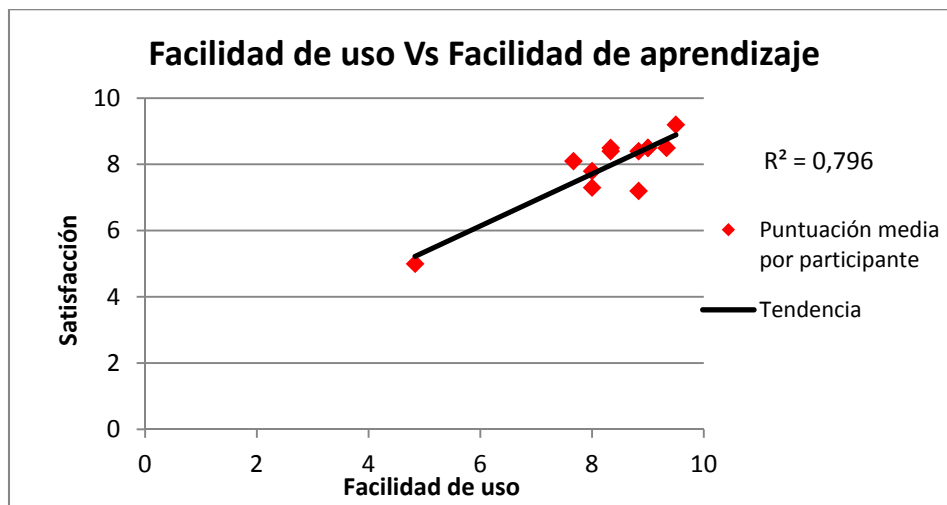


Figura 33: Correlación entre las variables *Facilidad de uso* y *Facilidad de aprendizaje*

Experimentación

La Figura 34, muestra la segunda correlación más alta y que fue encontrada entre las variables *Satisfacción* y *Utilidad*. La cual obtiene un coeficiente de correlación lineal de $R^2 = 0.7626$, lo que demuestra que existe una correlación en sentido positiva entre estas variables. Esta correlación permite indicar que los usuarios manifestarán mayor satisfacción en la utilización de la herramienta, en la medida que esta cumpla con las expectativas de utilidad.

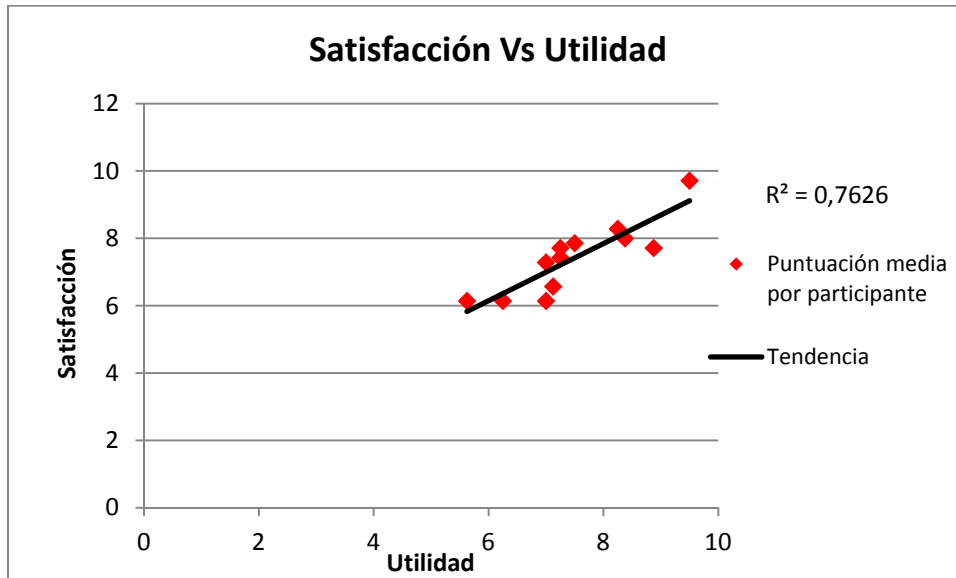


Figura 34: Correlación entre las variables *Satisfacción* y *Utilidad*

4.3. *Discusión de los Resultados*

En esta sección se realiza una discusión en conjunto de las dos partes de la experimentación llevada a cabo, es decir, la sesión experimental con los 12 usuarios y la relacionada con el cuestionario de usabilidad.

a) **La sesión experimental**

La sesión experimental con los 12 usuarios ha permitido obtener información valiosa respecto a la interacción de los usuarios con la herramienta y los comentarios efectuados mientras la utilizaban. En esta primera parte de la experimentación se consideraron las siguientes medidas:

- El tiempo empleado en las distintas etapas que requería la evaluación de la herramienta InterArch.
- Las tomas de audio y video de las sesiones experimentales.

Estas medidas fueron utilizadas en la sesión completa con los usuarios (demostración de la herramienta, segmentación de contenido y de creación de un modelo de contenido utilizando la herramienta InterArch). En cuanto a los tiempos medios empleados por los usuarios, los resultados fueron bastante variados. El resultado de los valores en los tiempos empleados en las distintas etapas, permite indicar que los usuarios que menos tiempo emplearon en la tarea de segmentación de contenido fueron los que más tiempo

dedicaron en la tarea de crear el modelo de contenido utilizando la herramienta. Esto sólo muestra un determinado estilo de trabajo para llevar a cabo las tareas, pues no significó que fueran ni los que más ni menos tiempo emplearon en toda la sesión de interacción. En cuanto a los tiempos empleados en la etapa de entrenamiento de los usuarios con la herramienta, se puede señalar que estos tiempos tienen una correlación baja con respecto al tiempo total empleado en cada una de las etapas ($R^2 = 0.3081$), especialmente con los tiempos empleados en la etapa de utilización de la herramienta ($R^2 = 0.1088$), lo cual demuestra que no se relaciona la pericia alcanzada por los usuarios en la manipulación de la herramienta en función al tiempo de entrenamiento recibido, es decir, el tiempo empleado en el entrenamiento de la herramienta no incidió en forma directa en la habilidad que finalmente alcanzaban los usuarios en la manipulación e interacción con la herramienta. Sí se puede indicar, que las cualidades que provee la herramienta respecto a la facilidad de aprendizaje permiten a los usuarios alcanzar una pericia más que aceptable, independiente del tiempo empleado en su entrenamiento. Lo anterior se reafirma con los resultados sobre el grado de facilidad de aprendizaje, el cual fue positivo en todos los casos, y con un resultado del 100% de puntuaciones positivas en cuanto a las características de facilidad de uso que manifestaron los usuarios.

Con respecto a las tomas de audio y video de las sesiones experimentales, estas han permitido obtener información valiosa respecto a cómo los usuarios utilizan la herramienta InterArch, y analizarlas de forma más exhaustiva. Por ejemplo, aunque los usuarios optaron mayoritariamente por utilizar la funcionalidad de seleccionar y arrastrar sobre los elementos de contenido creados (herencia), se detectó que esta funcionalidad presenta cierta dificultad en su manipulación, y que los usuarios necesitaban efectuar varios intentos antes de conseguir de forma exitosa lo que querían. También, se observó que los usuarios realizan varios pasos en la búsqueda de las distintas opciones de la herramienta. Esto permitió detectar la necesidad de contar con más opciones de acceso directo y definir una organización que permita conseguir una consecución más ágil, en función del número de pasos necesarios para alcanzar las opciones que requieren los usuarios.

b) Cuestionario de usabilidad

La segunda parte del experimento, corresponde a la relacionada con el cuestionario de usabilidad, la cual muestra diversos resultados acerca de la percepción de los usuarios con respecto a la herramienta InterArch. En la sección 4.2.2, se han efectuado medidas de forma satisfactoria, obteniendo los siguientes valores:

- La usabilidad de la herramienta mediante la medición de las variables *Utilidad*, *Facilidad de Uso*, *Facilidad de Aprendizaje* y *Satisfacción* (sección 4.2.2).
- El nivel de fiabilidad del cuestionario (sección 4.2.3).
- Las puntuaciones medias y dispersión de las variables de usabilidad (sección 4.2.4).

Experimentación

- La existencia de posibles diferencias en las variables de usabilidad estudiadas en grupo (sección 4.2.4).
- Las correlaciones existentes entre pares de variables (sección 4.2.4).

La evaluación presenta un nivel de fiabilidad más que aceptable (fiabilidad del 95.18%), lo que permite efectuar diversos análisis con un alto nivel de confianza. Los resultados en las cuatro variables de usabilidad arrojan valoraciones más que aceptables, pues en todas ellas se obtienen puntuaciones sobre 7 (media de 7.79, con una desviación estándar de 1.14), y además el resultado del estudio de la ANOVA indican que estas cuatro variables tienen niveles similares de aceptación entre los usuarios, lo que es considerado muy positivo en cuanto a la percepción que tienen los usuarios respecto a la usabilidad de la herramienta, en donde los usuarios principalmente destacan que la herramienta InterArch es fácil de aprender y de usar. Respecto a las correlaciones de las variables, se encontró cierto grado de relación entre la facilidad de uso y aprendizaje. La segunda correlación encontrada indica que el nivel de satisfacción de los usuarios en la utilización de la herramienta está muy relacionado con las expectativas de utilidad.

En cuanto a las preguntas abiertas del cuestionario, éstas reportaron información valiosa respecto a los aspectos positivos y de mejora para la herramienta. Si bien estas preguntas abiertas fueron presentadas a los usuarios de forma opcional, destaca que todos los usuarios se interesaron en realizar observaciones y entregar comentarios respecto a su experiencia. Sobre los comentarios de los aspectos positivos observados, cabe comentar que los usuarios resaltaron las siguientes propiedades: sencillez, facilidad de uso, rapidez, utilidad, intuición, amigabilidad, sentido práctico y parecido con otros entornos, entre otras. Lo que permite corroborar las evaluaciones positivas obtenidas en las preguntas cerradas, y que además están en concordancia con los aspectos que se plantearon al desarrollar la herramienta. Respecto a los aspectos susceptibles de mejora que requeriría la herramienta, los usuarios aportaron información valiosa a través de los siguientes comentarios: *“faltan tipos de elementos para utilizar”*, *“es muy sensible al mouse”*, *“las cajas se deberían representar en distintos tamaños según el nivel de jerarquía en el modelo de contenido”*, *“el texto debería adaptarse al tamaño de la caja”*, y *“falta una organización automática del diagrama completo”*. Todos estos comentarios reportaron una valiosa información de cara a la mejora continua, y serán considerados en futuras mejoras de la herramienta InterArch.

Finalmente, los resultados experimentales obtenidos permiten corroborar la última hipótesis de partida H_2 en la que se basa esta investigación, es decir, que InterArch es fácil de usar y de aprender por parte del Arquitecto de la Información, y que además permite generar información de análisis y diseño para analistas e ingenieros software de forma automática a partir de modelos de contenido que el Arquitecto de la Información crea, uniendo puentes entre las fases más iniciales del proyecto, dentro del dominio del problema, y las fases previas al desarrollo, más cercanas al dominio de la solución, como se quería demostrar. Finalmente, estos resultados permiten también afirmar, como se indicó al inicio del trabajo, que tanto el formalismo como la herramienta son una aportación al paradigma del Desarrollo por el Usuario Final, y más en concreto a

paradigmas relacionados como el EUSE (End-User Software Engineering) cuyo objetivo es involucrar a usuarios no expertos en computación en el ciclo de vida del desarrollo de aplicaciones software.

Capítulo 5: Conclusiones y Trabajo Futuro

En el presente capítulo, se presentan las conclusiones más importantes del trabajo y las actividades que se deben considerar como trabajo futuro o líneas futuras para continuar la investigación. En la sección 5.1, se describen las conclusiones principales del trabajo, y en la sección 5.2, se presentan líneas de continuidad del trabajo.

5.1. Conclusiones

En este trabajo, se ha presentado una solución automatizada de soporte válida para el análisis y diseño de aplicaciones Web interactivas consistente en un formalismo que permite representar el conocimiento proveniente de las definiciones conceptuales del Arquitecto de la Información, así como su posterior transformación en información de análisis y diseño para el Ingeniero del Software, el cual se ha materializado en una herramienta CASE llamada InterArch, que es fácil de usar y de aprender por parte del Arquitecto de la Información.

Respecto al formalismo elaborado, éste ha sido ideado en base a la propuesta presentada por Pressman para el modelado de análisis para aplicaciones Web (Pressman, 2005), en donde se plantea dar una continuidad a la representación de contenido que se lleva a cabo en las etapas iniciales del desarrollo software de un sitio Web. Más específicamente, en este Trabajo Fin de Máster se ha formalizado e implementado la propuesta anterior, en base a un conjunto de reglas que permiten establecer correspondencias entre modelos de contenidos elaborados por Arquitectos de la Información y diagrama de clases UML requeridos por analistas e ingenieros software. Adicionalmente, se han incorporado nuevas características respecto a la configuración de los métodos, relaciones y reglas en la generación de los diagramas de clases UML.

La herramienta CASE InterArch se ha desarrollado con el objetivo de unir puentes entre las representaciones conceptuales de alto nivel de la Arquitectura de la Información y la representación no funcional e ingenieril del contenido. Esto permite obtener finalmente clases de análisis y diseño necesarias para la implementación del software dentro del dominio de la solución. Para ello, la herramienta permite generar automáticamente diagramas de clases UML a partir de la definición de modelos de contenidos de la Arquitectura de la Información de un sitio Web interactivo, utilizando XMI como lenguaje intermedio de representación que además puede ser procesado por un gran número de herramientas de modelado CASE, lo que permite una mayor interoperabilidad para integrar clases UML funcionales y no funcionales durante el ciclo de vida de una aplicación Web interactiva.

Aun cuando la presencia de herramientas para el análisis de la Arquitectura de la Información en entornos de desarrollo Web no es algo novedoso, se han encontrado ciertas deficiencias y carencias en las herramientas existentes, las cuales han sido detectadas a través de un análisis específico que ha tenido como objetivo averiguar cómo estas herramientas enfrentan la necesidad de permitir una continuidad entre la información de análisis conceptual de la AI y la información de análisis y diseño que

necesitan los analistas informáticos. En cuanto a estos inconvenientes, el presente trabajo provee una solución probada, y proporciona ventajas competitivas con respecto a otras herramientas, entre ellas:

- Proveer un formalismo para representar información no funcional y transformarla, mediante reglas específicas, en diagramas de estructura customizables.
- Minimizar en términos de coste, tiempo y esfuerzo, el desarrollo de aplicaciones Web interactivas. La herramienta permite generar información necesaria para implementar las clases funcionales del dominio de la solución, sin que esto se signifique un mayor coste, tiempo o esfuerzo por parte de Arquitectos de la Información y analistas e ingenieros software.
- Incorporar una generación automática interoperable de diagrama de clases UML en formato transportable XMI, a partir de definiciones de los modelos de contenidos elaborados por los Arquitectos de la Información. La información generada es útil para el desarrollo de un proyecto informático, relacionado con el análisis y desarrollo de Software.

En cuanto a la experimentación llevada a cabo para medir la usabilidad de la herramienta, ésta permitió obtener información valiosa respecto al estilo de interacción de los usuarios con la herramienta, así como información sobre su usabilidad. Para ello, se han usado las técnicas de evaluación de Test Retrospectivo y Thinking Aloud, que permitieron obtener mayor detalle respecto a la interacción de los usuarios al utilizar la herramienta. Por ejemplo, se ha descubierto que las funcionalidades más utilizadas por los usuarios corresponden a aquellas que reportaban un mayor nivel de utilidad aún cuando estas exigen una mayor destreza para su utilización.

Los resultados de las evaluaciones con los 12 usuarios sobre los aspectos de usabilidad de la herramienta, muestran valoraciones positivas y aceptables en cuanto a la percepción que tienen los usuarios con respecto a las variables de *Utilidad*, *Facilidad de Uso*, *Facilidad de Aprendizaje* y *Satisfacción* de la herramienta InterArch. La mayor parte de los usuarios manifiesta niveles altos de conformidad en cuanto a las cualidades de facilidad de uso de la herramienta, la cual se ve incrementada en la medida en que la herramienta ofrece mayores facilidades en el aprendizaje de su utilización. Del mismo modo, los usuarios manifiestan mayor satisfacción en la utilización de la herramienta, en la medida que esta cumple con las expectativas de utilidad.

El trabajo se ha desarrollado en base a una serie de hipótesis de partida, que se han corroborado a lo largo del trabajo. Principalmente, que es posible generar información de análisis y diseño procesable por analistas e ingenieros software a partir de una representación inicial conceptual de los contenidos por parte del Arquitecto de la Información, correspondiente a la hipótesis de partida H_1 , la cual ha sido validada mediante la implementación de la metodología de gestión de conocimiento conceptual del dominio del problema, y su posterior transformación en un modelo del dominio cercano a la solución. Y que en base a lo anterior, es viable construir una herramienta

CASE fácil de usar y de aprender por el Arquitecto de la Información (hipótesis de partida H_2), la que también se ha corroborado en base a la evaluación con los usuarios reales, donde se obtuvieron resultados satisfactorios respecto al grado de usabilidad de la herramienta.

Finalmente, se puede indicar que la solución presentada es ideal para el análisis de la Arquitectura de la Información en entornos de desarrollo Web, ya que permite representar modelos de contenidos, así como su transformación en diagrama de clases UML, aportando nuevas funcionalidades para aumentar la automatización e interoperabilidad entre las definiciones conceptuales de los contenidos de una aplicación Web y los elementos de análisis y diseño procesables por analistas e ingenieros software, y disminuyendo en términos de coste, tiempo y esfuerzo, el desarrollo de aplicaciones Web interactivas.

5.2. Trabajo Futuro

En esta sección, se presentan las actividades que se proponen como trabajo futuro, así como mejoras propuestas, las cuales se han redactado en base a los resultados obtenidos de la evaluación de la herramienta y de las observaciones realizadas por los usuarios en el cuestionario de usabilidad, junto con otras ideas de continuidad de la investigación:

1. *Dotar a la herramienta con características de aporte semánticos.* Por ejemplo, la inclusión de comentarios en los elementos de contenido a la hora de ser creados por el Arquitecto de la información, permite dotar de información semántica, no sólo sintáctica, para su uso por analistas, de forma que se puedan crear restricciones en el diseño más avanzadas en base a una educación de información sobre los requisitos no funcionales planteados por los propios Arquitectos de la Información explícitamente. Estos comentarios serán procesados posteriormente por los analistas para concretar soluciones de diseño específicas que puedan satisfacer dichos requisitos, dotando así a la herramienta de un mecanismo semántico para la comunicación con los ingenieros del software que acabarán diseñando la aplicación final. Dichos comentarios pueden aparecer como tales en el diagrama UML generado, ya que el estándar así lo contempla. Por otro lado, la posibilidad de incluir mayor nivel semántico en el análisis de contenidos abre las puertas a otro tipo de actividades útiles en la educación de requisitos no funcionales, como pueden ser la incorporación automática de *propiedades de accesibilidad*, que podrían incluirse en el análisis de contenidos o bien de forma automática o concretando más explícitamente dichos elementos de contenido, anotando mediante comentarios el objetivo o propiedades de dichos elementos. Por ejemplo, cuando un Arquitecto de la Información está diseñando contenidos para especificar un catálogo de productos para una tienda on-line, además del texto descriptivo del producto, una descripción en forma de audio podría resultar útil para usuarios invidentes. Este tipo de campos, basados en *propiedades de accesibilidad*, podrían considerarse por defecto mediante reglas específicas o, como se ha argumentado

anteriormente, de forma explícita, con la versión actual de la herramienta, por parte del Arquitecto de la Información.

2. *Aumentar la expresividad en la parte de las reglas de transformación del modelo visual.* La idea sería permitir a analistas e ingenieros software adaptar aún más la generación de información para el proyecto de acuerdo a sus necesidades específicas. Actualmente, se provee un conjunto de reglas de transformación muy susceptibles de ampliación. Por ejemplo, las Reglas de Jerarquía pueden ser creadas y agrupadas de acuerdo a diferentes tipos de transformaciones que se requieran, y en función de los tipos de productos que se deseen conseguir. Del mismo modo, las Reglas de Configuración pueden ser enriquecidas tanto en la ampliación del número de propiedades que pueden ser configuradas en las etiquetas de las clases, atributos, métodos y asociaciones que se generan mediante el código XMI, tales como el tipo de dato y valor inicial de los atributos, la cardinalidad de las asociaciones y las restricciones de las clases, entre otras. También se podrían concretar los niveles en que estas propiedades pueden ser especificadas y aplicadas. Por ejemplo, las Reglas de Configuración podrían ser aplicadas a elementos o asociaciones específicas del diagrama.
3. *Encapsular el formalismo propuesto, así como su transformación en diagramas de clases UML, a través de un Plug-in.* Se trataría de generar un *Plug-in* con las funcionalidades de InterArch, el cual sea capaz de integrarse con los entornos de trabajos más usados comúnmente por los Arquitectos de la Información y los analistas e ingenieros software, esto permitiría prescindir de la exigencia de tener que emplear la herramienta InterArch para beneficiarse de sus ventajas y funcionalidades, las cuales se integrarían en los entornos de trabajo en que se encuentren más familiarizado, mediante una extensión de las funcionalidades que tienen estos entornos de trabajo.
4. *Estudiar la utilidad y manipulación de los diagramas de clases UML.* Se trataría de llevar a cabo un estudio respecto de la utilidad y manipulación de los diagramas de clases UML generados para analistas e ingenieros software, de modo que se pueda obtener información sobre el uso e impacto de los diagramas de clases UML generados dentro del ciclo de vida del desarrollo de aplicaciones Web interactivas, con el objetivo de perfeccionar y adecuar el proceso actual de generación de diagramas.
5. *Estudiar correspondencias de otros productos entregables de la AI.* Estudiar las posibles correspondencias que puedan existir entre los productos entregables de la Arquitectura de la Información y la información de análisis y diseño que necesitan analistas e ingenieros software u otros profesionales implicados en el proceso de desarrollo de software. Este estudio, permitiría establecer un mecanismo para generar esta información de forma automática, por ejemplo, los blueprints. Al describir la composición y relación de las diferentes páginas de las que se compone un sitio Web, podrían ser útiles para generar de forma

automática el mapa del sitio Web o código fuente respecto a la composición existen entre las páginas del sitio Web.

6. *Proveer mecanismos de ingeniería inversa para la generación de modelos de contenido.* Se basaría en crear los modelos de contenido a partir de la lectura y análisis de los diagramas de clases UML (ingeniería inversa), con el objetivo de organizar la forma de desplegar la información en las situaciones en que se requiera rediseñar o analizar un sitio Web, aportando información conceptual a los Arquitectos de la Información para la concreción de los aspectos no funcionales de un sitio Web, y facilitando así el rediseño de aplicaciones en general.
7. *Mejorar la herramienta con nuevas funcionalidades para la gestión y manipulación de los elementos visuales.* La idea es incorporar funcionalidades que permitan a los usuarios obtener una mayor eficiencia en la interacción con la herramienta, tales como: organización automática de acuerdo a la jerarquía y asociación de contenidos, visualización de diagramas de clases UML, dimensiones ajustables al tamaño del texto y nivel de jerarquía, etc.
8. *Utilizar repositorios de información para construir modelos de contenido semi-automáticamente.* Se trataría de poder acceder, desde la herramienta, a otros repositorios de información (bases de datos, repositorios de ontologías, etc.) para obtener los datos que permitan la obtención automática, y posterior manipulación por parte del Arquitecto de la Información, de modelos de contenidos. Sería útil, por ejemplo, incorporar una opción para importar o leer elementos de información enriquecidos, como es el caso de ontologías RDF que ya incluyan relaciones específicas entre los elementos de información. Esto además ayudaría a obtener el contenido de fuentes de datos externas cuando el volumen de información es grande.

Bibliografía

- Abascal, J., Aedo, I., Cañas, J. J., Gea, M., Gil, A. B., Lorés, J., et al. (2001). *La interacción persona-ordenador*. (AIPO, Ed.) Barcelona, España: Jesús Lorés.
- Alfa de Cronbach*. (2011). Obtenido de http://es.wikipedia.org/wiki/Alfa_de_Cronbach
- Camtasia-Studio*. (2011). <http://www.camtasiaoftware.com>.
- David, B., & Gaudenz, A. (2010). *JGraph*. Reuperado desde www.jgraph.org
- Davis, F. D. (1989). *Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology*. *MIS Quarterly*, 13, 319-340.
- Elaine, G. T. (2002). *Information interaction: Providing a framework for information architecture*. *Journal of the American Society for Information Science and Technology*, 53, 855-862.
- Erlin, Yunura, Y. A., & Azizah, A. R. (2008). *The evolution of Information Architecture*. *Information Technology, ITSIM*, 4, 1-6.
- Garrett, J. J. (2002). *A visual vocabulary for describing information architecture and interaction design*. Recuperado en Febrero del 2011, de www.jjg.net/ia/visvocab
- Gosling, J., Joy, B., Steele, G., & Bracha, G. (2005). *The Java language specification* (tercera edición ed.). Addison-Wesley.
- Grose, T., Doney, G., & Stephen. (2002). *Mastering XMI: Java Programming with XMI, XML, and UML*. John Wiley & Sons.
- Gutierrez, M. P.-M. (2010). *Arquitectura de la informacion en entornos Web*. TREA.
- Hassan, Y., & Núñez, A. (2005). *Diseño de Arquitecturas de Información: Descripción y Clasificación*. No Solo Usabilidad, 4.
- Holzinger, A. (2005). *Usability engineering methods for software developers*. *Communications ACM* 48, 71-74.
- Katsanos, C. T. (2008). *AutoCardSorter: designing the information architecture of a Web site using latent semantic analysis*. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems (CHI '08)*, 875-878.
- Klann, M. F. (2003). *End-User Development*. D1.1 Roadmap. *Proceedings of the End User Development Workshop at CHI'2003 Conference*, 5-10.
- Leise, F., Fast, K., & Stecke, M. (2002). *What Is A Controlled Vocabulary?* Obtenido de boxesandarrows.com

- Lewis, C. (1982). *Using the "thinking aloud" method in cognitive interface design*. IBM Res. Rep. RC-9265.
- Lieberman, H., Paternò, F., & Wulf, V. (2006). *End-User Development*. Human Computer Interaction Series .
- Lin, H. Y. (1997). *A Proposed Index of Usability: A Method for Comparing the Relative Usability of Different Software Systems*. Behaviour and Information Technology, 16, 267-278.
- Lindholm, T., & Yellin, F. (1999). *Java Virtual Machine Specification*. Inc. Boston, MA, USA: Addison-Wesley Longman Publishing Co.
- Lund, A. (2001). *Measuring Usability with the USE Questionnaire*. Usability and User Experience Special Interest Group, 8.
- Macías, J. A., & Castells, P. (2007). *Providing end-user facilities to simplify ontology-driven Web application authoring*. Interacting with Computers, 19, 563-585.
- Morville, P., & Rosenfeld, L. (2006). *Information Architecture for the World Wide Web*. O'Reilly Media Inc.
- Morville, P., Rosenfeld, L., & Koloski, B. (2009). *The Information Architecture Institute*. Obtenido de ia.institute.org
- NetBeans. (2010). Recuperado en Junio de 2011, de <http://netbeans.org/>
- Nielsen, J. (1993). *Usability Engineering*. Morgan Kaufmann Publishers.
- Nørgaard, M., & Hornbæk, K. (2006). *What do usability evaluators do in practice?: an explorative study of think-aloud testing*. Proceedings of the 6th conference on Designing Interactive systems, 209-218.
- Nørgaard, M., & Hornbæk, K. (2006). *What do usability evaluators do in practice?: an explorative study of think-aloud testing*. Proceedings of the 6th conference on Designing Interactive systems, 209-218.
- OMG. (2005). *MOF 2.0/XMI Mapping Specification*. Obtenido de <http://www.omg.org/spec/XMI/2.1/PDF>
- Pérez, M., & Codina, L. (2010). *Software de prototipado para la arquitectura de la información: funcionalidad y evaluación*. El profesional de la información, 417-424.
- Pressman, R. S. (2005). *Ingeniería del software : un enfoque práctico*. McGraw-Hill.
- Ramey, J., Cuddihy, E., Dumas, J., Guan, Z., Van, M., & Jong, M. (2006). *Does think aloud work?: how do we know?* CHI '06 extended abstracts on Human factors in computing systems, 45-48.

Bibliografía

- Rice, Sarah A. (2005). *Seneb Consulting*. Obtenido de http://www.seneb.com/example_content_inventory.xls
- Rojas, L. A., & Macías, J. A. (2011a). *End-User Support for Information Architecture Analysis in Interactive Web Applications*. 13th IFIP TC 13 International Conference (Interact 2011). Campos, Nunes, Graham, Jorge and Palanque (editors). Lisbon, Portugal, September 5-9, 2011. Lecture Notes in Computer Science, Volume 6949, Springer NY, 2011, pp. 515-518. ISSN: 0302-9743. ISBN: 978-3-642-23767-6. [CORE A]
- Rojas, L. A., & Macías, J. A. (2011b). *Soporte para el Análisis de la Arquitectura de la Información en Aplicaciones Web*. Actas del XII Congreso Internacional de Interacción Persona Ordenador (Interacción 2011). Nestor Garay Vitoria y Julio Abascal González (Editores). Ibergarceta Publicaciones. Lisboa, Portugal, del 02 al 05 de Septiembre de 2011, pp. 89-98. ISBN: 978-84-9281-234-9.
- Ronda, R. (2007). *Revisión de técnicas de arquitectura de información*. No Solo Usabilidad (6).
- Sachs, H. (2008). *Content Modelling*. Recuperado en Febrero de 2011, de http://www.openpublish.com.au/pdf/Howard_Sachs.pdf
- Schwabe, D. R. (1995). *The object-oriented hypermedia design model*. ACM 38 , 45-46.
- Shiple, J. (1998). *Information Architecture Tutorial*. Recuperado desde <http://hotwired.lycos.com/webmonkey/98/28/index0a.html>
- TerpSys. (s.f.). *User-Centered Design Methodology*. Recuperado en Junio de 2011, de <http://www.terpsys.com/ia.html>
- Tigris.org. (2009). Recuperado en Junio de 2011, desde <http://argouml.tigris.org/>
- Walrath, K., & Campione, M. (1999). *The JFC Swing Tutorial: A Guide to Constructing GUIs*. Addison-Wesley.
- XMI. (1999). *XML Metadata Interchange*. Obtenido de www.omg.org.
- Yusef, H., & Francisco, M. (2003). *Web Semántica: El papel del Arquitecto de la Información*. No Solo Usabilidad, 2.

,

Anexo A: Cuestionario de Usabilidad

Este anexo contiene el cuestionario completo utilizado para medir la usabilidad de la herramienta InterArch. Inicialmente, contiene preguntas referentes al perfil del usuario. Además, contiene preguntas cerradas para medir las variables de *Utilidad*, *Facilidad de Uso*, *Facilidad de Aprendizaje* y *Satisfacción* de la herramienta InterArch. Finalmente, incluye preguntas abiertas para obtener información respecto a los aspectos tanto positivos como negativos de la herramienta InterArch y sobre las tareas específicas que podrían ser apoyadas con el uso de la herramienta InterArch.

I. PERFIL DEL USUARIO

Nombre y Apellido

Sexo

Edad

II. UTILIDAD

(1 es completamente en desacuerdo, 10 es totalmente de acuerdo y 5 neutral)

(*) Indica que la pregunta es obligatoria

2. La aplicación me ayuda a ser más efectivo. (*)

1 2 3 4 5 6 7 8 9 10

3. La aplicación me ayuda a ser más productivo. (*)

1 2 3 4 5 6 7 8 9 10

4. La aplicación me parece útil. (*)

1 2 3 4 5 6 7 8 9 10

5. La aplicación me da más control sobre las actividades de mi vida. (*)

1 2 3 4 5 6 7 8 9 10

6. La aplicación hace que las cosas que quiero lograr sean más fácil de hacer. (*)

1 2 3 4 5 6 7 8 9 10

7. La aplicación me ahorra tiempo cuando la uso. (*)

1 2 3 4 5 6 7 8 9 10

8. La aplicación responde a mis necesidades. (*)

1 2 3 4 5 6 7 8 9 10

9. La aplicación hace todo lo que yo esperaría que hiciera. (*)

1 2 3 4 5 6 7 8 9 10

III. FACILIDAD DE USO

(1 es completamente en desacuerdo y 10 es totalmente de acuerdo)

10. La aplicación es fácil de usar. (*)

1 2 3 4 5 6 7 8 9 10

11. La aplicación es simple de usar. (*)

1 2 3 4 5 6 7 8 9 10

12. La aplicación es amigable. (*)

1 2 3 4 5 6 7 8 9 10

13. La aplicación requiere la menor cantidad de pasos posibles para lograr lo que quiero que haga. (*)

1 2 3 4 5 6 7 8 9 10

14. La aplicación es flexible. (*)

1 2 3 4 5 6 7 8 9 10

15. La utilización de la aplicación no requiere mayor esfuerzo. (*)

1 2 3 4 5 6 7 8 9 10

16. Puedo usar la aplicación sin instrucciones escritas. (*)

1 2 3 4 5 6 7 8 9 10

17. No se hacen notar inconsistencias mientras uso la aplicación. (*)

1 2 3 4 5 6 7 8 9 10

18. Pienso que la aplicación gustaría a usuarios ocasionales como a regulares. (*)

1 2 3 4 5 6 7 8 9 10

19. Puedo usar la aplicación siempre con éxito. (*)

1 2 3 4 5 6 7 8 9 10

IV. FACILIDAD DE APRENDIZAJE

(1 es completamente en desacuerdo y 10 es totalmente de acuerdo)

Cuestionario de Usabilidad

20. Aprendí a usar la aplicación rápidamente. (*)

1 2 3 4 5 6 7 8 9 10

21. Recuerdo fácilmente cómo usar la aplicación. (*)

1 2 3 4 5 6 7 8 9 10

22. Es fácil de aprender a usar la aplicación. (*)

1 2 3 4 5 6 7 8 9 10

23. Me he hecho habilidoso rápidamente con la aplicación. (*)

1 2 3 4 5 6 7 8 9 10

24. No es necesario explorar nuevas características por ensayo y error. (*)

1 2 3 4 5 6 7 8 9 10

25. El orden de las áreas de trabajo es lógico. (*)

1 2 3 4 5 6 7 8 9 10

V. SATISFACCIÓN

(1 es completamente en desacuerdo y 10 es totalmente de acuerdo)

26. Estoy satisfecho con la aplicación. (*)

1 2 3 4 5 6 7 8 9 10

27. Recomendaría la aplicación a un amigo. (*)

1 2 3 4 5 6 7 8 9 10

28. La aplicación es divertida de usar. (*)

1 2 3 4 5 6 7 8 9 10

29. La aplicación funciona de la forma que yo quiero que funcione. (*)

1 2 3 4 5 6 7 8 9 10

30. La aplicación es maravillosa. (*)

1 2 3 4 5 6 7 8 9 10

31. Siento la necesidad de tener la aplicación. (*)

1 2 3 4 5 6 7 8 9 10

32. La aplicación es agradable de utilizar. (*)

1 2 3 4 5 6 7 8 9 10

VI. PREGUNTAS ABIERTAS

33. ¿Qué aspectos consideras negativos de la aplicación?

34. ¿Qué aspectos consideras positivos de la aplicación?

35. ¿Para qué tareas de tu labor cotidiana emplearías o ves que puede ser útil la aplicación?

36. Observaciones