

Desarrollo de Software EUD Centrado en el Usuario Final

José Antonio Macías Iglesias
Universidad Autónoma de Madrid
Tomás y Valiente 11
28049 Madrid
+34 914976241
j.macias@uam.es

RESUMEN

Recientemente, el auge de las TIC ha motivado que usuarios no programadores hayan tomado un papel activo en el entorno de la computación construyendo y personalizando artefactos software, lo que ha provocado una tendencia que aboga por transformar al usuario final en programador o diseñador de proyectos software. Este hecho puede desequilibrar las competencias profesionales, introducir un factor de riesgo implícito y producir un impacto socio-económico importante. En este artículo se analiza esta problemática y se proponen directrices que permitan potenciar los principios de diseño y asistir al usuario final en las tareas más complejas de personalización de soluciones software relacionadas con su dominio, en vez de que éste acabe programando y/o gestionando proyectos software. Para ello, se introducirán tres herramientas de soporte, y se evaluarán con respecto a su usabilidad alcanzada para demostrar el alcance y adecuación de dichas herramientas para el usuario final experto en su dominio pero no en computación.

Palabras Clave

Desarrollo por el Usuario Final, Diseño Centrado en el Usuario, Usabilidad.

1. INTRODUCTION

En los últimos años se han producido cambios notables en el mundo TIC motivados, principalmente, por el abaratamiento del hardware, la implantación de internet y por el hecho de que gran parte de la población mundial haya acabado manejando software de ordenador. Esto ha motivado pasar de tener personas que utilizaban el ordenador de forma esporádica a tener usuarios finales expertos en su dominio que manipulan software para fines muy concretos como la simulación, experimentación, desarrollo de pequeñas aplicaciones o módulos con entornos de alto nivel, macros, aplicaciones web, etc. Este tipo de usuarios se diferencian de los usuarios poco expertos, con un perfil más ofimático, pero también de los muy expertos entre los que se encuentran programadores y expertos en computación (ver Figura 1).

Este cambio de mentalidad provocó el surgimiento de nuevas disciplinas, como el EUD (*End-User Development*), basadas en el concepto del usuario final, es decir, usuarios no informáticos con diferentes conocimientos y habilidades cognitivas que acaban tomando un papel activo en la creación y personalización de software. El EUD, o Desarrollo por el Usuario Final, se define como un conjunto de actividades o técnicas que permiten a personas, que no son desarrolladores profesionales, crear o modificar software [8][10]. Posteriormente, a raíz del EUD, surgen nuevos y recientes sub-paradigmas relacionados que abogan por implicar aún más al usuario final en las tareas de

desarrollo y construcción de software. Se trata del EUP (*End-User Programming*), donde el objetivo es que el usuario pueda crear programas software [6], y el EUSE (*End-User Software Engineering*), donde el objetivo va más allá de que el usuario programe, involucrándole en aspectos concretos del desarrollo de un proyecto informático relacionados con la reusabilidad, seguridad y verificabilidad del software a desarrollar [22].

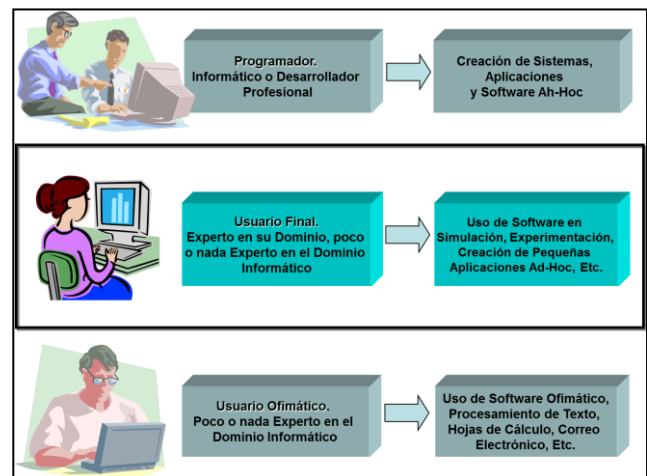


Figura 1. Categorización del usuario final entre los niveles de usuario ofimático y experto programador.

1.1 Estado del Arte

El Desarrollo por el Usuario Final surge como iniciativa a partir de consorcios como EUSES [7] en Estados Unidos o EUD-Net [17] en Europa, cuyos objetivos se basan en hacer posible, de una forma rápida y barata, la co-evolución de sistemas y usuarios finales a partir de la adaptación explícita de las aplicaciones informáticas a las actividades de los propios usuarios. No obstante, y desde el punto de vista de la industria, no existe una adopción sistematizada del EUD, sino que se manifiesta parcialmente en algunas técnicas pensadas para ayudar o asistir al usuario y evitarle adquirir conocimientos irrelevantes no relacionados directamente con su actividad, como la creación de menús adaptativos, la generación de macros y pequeños programas mediante interfaces de alto nivel conceptual, la realización de cálculos o procesos de inferencia automáticos en celdas de hojas de cálculo, etc.

En el plano académico, podemos encontrar algunas propuestas más concretas a partir de herramientas como TERESA [15], enfocada al diseño y desarrollo de aplicaciones *nómadicas*, que ofrece un soporte integrado para representaciones formales e informales, y permitiendo además la modificación de aplicaciones

existentes o la creación de nuevas aplicaciones desde cero. También, WebRevenge [19] lleva a cabo un proceso de ingeniería inversa, es decir, genera un modelo de tareas a partir del análisis de la interacción y los componentes de una interfaz Web final. iCAP [6] es también otro trabajo de investigación cercano al EUD que permite el desarrollo de aplicaciones con sensibilidad al contexto, definiendo salidas o respuestas en función de las entradas contextuales. Por otro lado, ICO [1], compuesto por un lenguaje visual distribuido y concurrente de objetos para la modelización compartida, y PetShop [16], un entorno de edición y modificación de programas para prototipado rápido que permite la modelización interactiva de una aplicación, intentan combinar lenguaje de modelado y herramienta de edición, dando un soporte único para la creación de aplicaciones por parte de usuarios cualesquiera sin necesidad de conocer para ello ningún lenguaje de programación concreto. Otra interesante aportación a este campo es el del concepto de Web Pragmática (The Pragmatic Web) [20]. La Web Pragmática no trata sobre forma o significado, como es el caso de la Web normal, sino sobre cómo la información va a ser usada finalmente. Para ello utiliza agentes que transmiten y captan información relevante que será usada en aplicaciones prácticas, en función de las necesidades propias de cada usuario, y teniendo en cuenta en todo momento el contexto de uso. Bajo el paradigma de las interfaces inteligentes, Personal Wizards [2] captura los eventos generados por un usuario experto bajo una aplicación, y seguidamente el sistema crea un asistente, a partir de los eventos generados, para guiar a usuarios menos expertos paso a paso a través de las tareas propuestas.

No obstante, y en la actualidad, no existe un estudio concienzudo o guías de diseño explícitas que permita elaborar tecnología EUD de manera sistematizada. En ese sentido, urge la creación de guías o directrices para el diseño de sistemas realmente centrados en el usuario, sin exigir de éste una intervención explícita en el desarrollo, pero orientando las soluciones a los usuarios finales del sistema, de forma que el software esté más acorde con sus necesidades reales.

1.2 El Usuario Final como Desarrollador

En cierta medida, pudiera parecer razonable que la sociedad de la información avance hasta tal punto que permita garantizar plenamente el hecho de que el usuario final acabe desarrollando sus propios artefactos software. En general, la mejora de la experiencia del usuario, a partir de la mejora de atributos de calidad como la usabilidad, ha hecho progresar las técnicas IPO (Interacción Persona-Ordenador) notablemente. No obstante, surgen aspectos, que no han sido completamente tratados hasta el momento, que dejan entrever cierta incongruencia sobre esta idea:

- ¿Debería un usuario no experto en computación acabar programando software? ¿Es conveniente que un usuario final intervenga en los aspectos técnicos del desarrollo de un proyecto software?
- ¿Sería asumible el coste en aprendizaje de estos usuarios? ¿Este coste podría ser asumido si, por otro lado, se prescindiera de personal experto en programación y/o gestión de proyectos?
- ¿Podrían considerarse estas iniciativas una intromisión en las competencias de informáticos, analistas y programadores?

En cierta forma, existen factores de riesgo que pueden hacer poco recomendable que personas no expertas en software acaben programando. Algunos de los accidentes ocurridos en los últimos

años, como los producidos en el metro de distintas ciudades, o el accidente aéreo de la Tam, ponen de manifiesto problemas de usabilidad inherentes en las aplicaciones diseñadas, que podrían ser mucho más graves si son los usuarios finales, sin ciertas competencias relacionadas con la creación de software, los que crean o manipulan este tipo de aplicaciones informáticas.

En general, para garantizar el EUD en toda su extensión sería necesario un cambio sistematizado en las técnicas actuales de construcción de software, ya que se requieren métodos de desarrollo más orientados a las personas que a los procesos, enfatizando en la flexibilidad, la adaptabilidad y la asistencia al usuario final. No sería adecuado, por tanto, que el usuario final acabe desarrollando mediante herramientas o entornos especializados, ya que conllevan un coste de aprendizaje elevado. Por otro lado, está presente el ya mencionado factor de riesgo, pues muchos usuarios no tienen el desarrollo como tarea principal. Debe haber, por tanto, un equilibrio entre la motivación del usuario y el coste que conlleva poner en práctica el EUD.

La hipótesis de partida de este trabajo se basa en que es posible potenciar los principios de diseño, llegando a un compromiso aceptable entre utilidad y usabilidad, potenciando técnicas relacionadas con la interacción que permitan asistir al usuario en las tareas más complejas, en vez de que el usuario acabe programando y/o gestionando proyectos software. Se trata, por tanto, de que los usuarios puedan diseñar o personalizar fácilmente nuevas aplicaciones, a la vez que los expertos realizan las partes más complejas (componentes, extensiones, herramientas de desarrollo, etc.), buscando un equilibrio que permita optimizar las competencias de ambos colectivos en beneficio del usuario final, lo que supondría un compromiso aceptable para ambas partes, una disminución del riesgo y una solución a ciertos problemas de usabilidad actuales que tienen un impacto socio-económico importante.

1.3 Aportación del Trabajo

Se propone un planteamiento basado en el paradigma EUD que permita formalizar y definir los rasgos principales, procesos, actividades y herramientas que satisfagan la hipótesis de partida anteriormente enunciada. Para ello, se proponen tres herramientas de soporte, DESK [13], VISQUE [3] e InterArch [21], que permiten al usuario final, no desarrollador y experto en su dominio, concentrarse en sus tareas de creación y personalización de artefactos software, mientras los expertos en informática se concentran en tareas de soporte o desarrollo avanzado de las propias herramientas, como es el caso, o continuar su labor a partir de las salidas que éstas generan. Las principales aportaciones de este trabajo, y de las herramientas presentadas, son las siguientes:

- a) Concreción de la participación del usuario en el proceso de diseño de soluciones programáticas mediante herramientas de autor de distintos dominios de aplicación. Se trata de mejorar los métodos tradicionales y tener en cuenta la opinión del usuario en todo momento, haciendo un diseño más centrado en el usuario final y adaptado a sus propias necesidades.
- b) Explotación de lenguajes visuales de modelado. Es primordial conseguir lenguajes fáciles de usar para el usuario final, más intuitivos, así como lenguajes de dominio específico. Esto permitirá hacer frente a la diferencia de

abstracción existente entre los profesionales del software y el modelo mental de los usuarios finales.

- c) Establecimiento de un compromiso aceptable entre expresividad y facilidad de uso. Lo deseable es conseguir entornos de creación de artefactos software fáciles de utilizar, intuitivos y visuales, disminuyendo, si fuera necesario, la capacidad expresiva de los mismos. En definitiva, se trata de conseguir un compromiso aceptable entre la expresividad y la facilidad de uso a base de reducir la complejidad conceptual de las aplicaciones.

Las herramientas presentadas reúnen características no presentes en otras soluciones existentes, incorporando aspectos tales como permitir al usuario seleccionar y parametrizar objetos fácilmente, integrar componentes existentes en el sistema y extenderlos construyendo nuevos componentes visualmente. Por otro lado, el lenguaje visual empleado en todas ellas permite al usuario concentrarse en sus tareas de diseño, definiendo a su vez un método de trabajo basado en su dominio que permite crear y customizar soluciones software específicas. Se trata de evitar limitar la capacidad creativa de estos usuarios. Estos objetivos se garantizan a partir de una medición específica de la usabilidad realizada, y mediante las técnicas centradas en el usuario empleadas para la creación de estas herramientas, enfocadas al colectivo de usuarios objetivo.

2. HERRAMIENTAS DESARROLLADAS

Con objeto de satisfacer los objetivos anteriormente expuestos, se presentan tres herramientas diseñadas en el ámbito del Desarrollo por el Usuario Final que pretenden satisfacer los planteamientos enunciados anteriormente.

Estas herramientas han sido construidas en base a las siguientes hipótesis de partida:

- 1) Es posible idear un formalismo visual, manipulable por los usuarios finales (lenguaje visual centrado en el usuario) que permita representar ideas o conocimiento conceptual procesable por este tipo de usuarios.
- 2) Es posible sistematizar el proceso descrito en la primera hipótesis mediante la construcción de una herramienta fácil de usar y de aprender por el usuario final, que permita personalizar o crear elementos programáticos u artefactos software con el menor esfuerzo posible a partir de las facilidades visuales y de manipulación concebidas.

2.1 DESK

DESK es una herramienta web, similar a un editor web, que permite al usuario, no experto en lenguajes de edición para la web, modificar una página web generada dinámicamente, siendo la herramienta capaz de seguir el camino inverso en la generación de dicha página para poder llegar a identificar los objetos del dominio, presentes en el modelo de la presentación, que generaron dicha página. De esta forma, DESK cambia información procedural sobre cómo deberá generarse la página modificada en el futuro, o incluso cómo deberán generarse páginas del mismo tipo. La herramienta se apoya en un sistema de generación de páginas web a través de ontologías, denominado PEGASUS [4], donde gracias a la separación explícita que hace de presentación y contenidos, permite la edición de estos dos conceptos por separado, permitiendo a DESK llevar a cabo el proceso de manera automática.

PEGASUS es un sistema de representación del conocimiento y generación automática de páginas (interfaces) web dinámicas, que utiliza ontologías del dominio a la medida para la descripción y estructuración conceptual de la información. PEGASUS proporciona un lenguaje de representación de ontologías del dominio y bases de conocimiento. Una ontología describe conceptos de un dominio determinado, y consiste básicamente en una jerarquía de categorías, con propiedades y relaciones entre ellas. El vocabulario definido por la ontología se utiliza para especificar entidades del dominio (instancias de las categorías) y relacionarlas unas con otras, estructurando el conocimiento en forma de red semántica. PEGASUS además aporta un lenguaje de representación de un modelo abstracto explícito de la presentación, que determina la composición y apariencia de las páginas a generar. Estas presentaciones no se asocian a unidades de conocimiento concretas, sino a clases y relaciones de la ontología, es decir, a categorías de conocimiento. Las plantillas de presentación de PEGASUS permiten incluir referencias y condiciones, además de reglas, sobre el modelo del usuario y de la plataforma, asimismo la utilización de un modelo abstracto de la presentación, que supone la separación entre presentación y contenidos, favorece la reutilización y consistencia en las interfaces web diseñadas.

La herramienta de autor DESK utiliza técnicas basadas en el EUD, como es el caso de la *Programación por Demostración* [9], y monitoriza al usuario durante toda la interacción, construyendo un modelo estructurado de la interacción. Posteriormente procesa este modelo para extraer información sobre los cambios realizados por el usuario. De esta forma es posible inferir automáticamente las modificaciones llevadas a cabo, evitando así que el usuario tenga que manipular los lenguajes de modelado interno de PEGASUS.

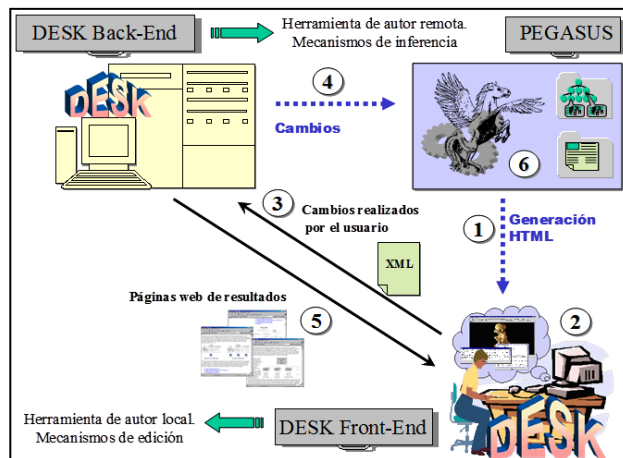


Figura 2. La herramienta de autor DESK.

Para llevar a cabo su cometido, DESK distribuye su funcionalidad en dos partes construidas a partir de una arquitectura cliente-servidor, tal y como se muestra en la Figura 2. La parte del cliente (*front-end*) es la herramienta de autor WYSIWYG encargada de soportar la mayor parte de la interacción directa con el usuario. La parte del servidor (*back-end*), es donde se efectúan los mecanismos de inferencia que permiten llevar a cabo los cambios realizados por el usuario sobre la interfaz web. Atendiendo a la información mostrada en la Figura 2, los principales pasos en el proceso de interacción con DESK son los siguientes:

- 1) En primer lugar el usuario edita una página concreta, generada por PEGASUS, mediante la herramienta DESK del *front-end*.
- 2) DESK recoge información sobre la interacción del usuario y la representa en un modelo de monitorización, que es un modelo estructurado codificado en XML, el cual refleja información de alto nivel sobre las acciones o cambios llevados a cabo por el usuario durante la edición de un documento web.
- 3) Los cambios realizados por el usuario se envían al servidor. La parte de DESK que se encuentra en el *back-end* procesa los cambios a partir del modelo de monitorización, empleando para ello los correspondientes mecanismos de inferencia mediante el uso de conocimiento de alto nivel del dominio.
- 4) Como resultado final del proceso anterior de inferencia, se modifican los modelos subyacentes de PEGASUS: presentación y dominio.
- 5) Finalmente se envía al usuario información relativa a los cambios efectuados y a lo acontecido durante todo el proceso de inferencia.
- 6) La próxima vez que se genere una página del mismo tipo, los cambios surtirán efecto, habiendo modificado así el mecanismo por el cual se generan las páginas web de forma persistente

El proceso de inferencia llevado a cabo por DESK consta de heurísticas que buscan en la ontología correspondencias entre los objetos del dominio y los cambios que el usuario ha llevado a cabo en la interfaz. De esta forma se localizan relaciones entre objetos concretos, del dominio y de la presentación, y se construyen caracterizaciones (rutas de acceso) que determinan exactamente qué es lo que el usuario ha tratado de cambiar en cada caso, distinguiendo el sistema qué modelo modificar (dominio, presentación) y, a menor nivel de granularidad, qué información debe ser actualizada dentro de los modelos. De forma adicional, y aplicando heurísticas apropiadas que trabajan con información del dominio, DESK trata de solventar las posibles ambigüedades que se producen en ese camino inverso a partir del proceso de ingeniería inversa llevado a cabo. Esto permite reconocer y caracterizar los cambios de una forma más precisa.

Por otro lado, DESK aporta, a partir del proceso de ingeniería inversa explicado, una visión enfocada a alejar al usuario del conocimiento de la especificación de la interfaz web modelizada a partir de PEGASUS. Es decir, se trata de aplicar técnicas de Desarrollo por el Usuario Final, como es el caso de la *Programación por Demostración*, para reducir la complejidad del sistema, persiguiendo conseguir un compromiso aceptable entre expresividad y complejidad de uso, siendo un aporte importante al diseño de la interacción en entornos web semánticos [13].

2.2 VISQUE

VISQUE es una herramienta de autor que tiene como propósito la creación de consultas *por demostración* [9] enfocadas a almacenes de datos, permitiendo a usuarios de bases de datos, no expertos en SQL, construir consultas de una forma visual y con menos limitaciones de expresividad. La herramienta utiliza distintas técnicas de visualización, y proporciona expresividad para construir consultas complejas que incluyen operaciones de conjuntos, consultas anidadas, y operaciones de agregación.

La herramienta está basada en la web, y aporta un conjunto de técnicas novedosas de visualización de información para la consulta de esquemas de almacenes de datos. VISQUE toma la información de conexión de una base de datos relacional e infiere el esquema de datos correspondiente mediante un algoritmo que analiza la estructura de las tablas y determina cuál es la tabla de *hechos* y cuáles son las tablas de *dimensiones*. Esta información se almacena en un fichero XML, que sirve como descriptor del esquema y proporciona información a otros componentes de la herramienta.

A partir del componente visual para la interacción con el esquema, el usuario puede construir una consulta visualmente. Posteriormente, el sistema utiliza un intérprete de consultas y, apoyándose en el descriptor del esquema, genera una sentencia SQL a ser ejecutada en el Sistema Gestor de Bases de Datos para visualizar los resultados. Cuando la consulta en cuestión implica una operación sobre conjuntos o existen consultas anidadas, el sistema es capaz de generar visualizaciones para mostrar dichas operaciones. Para ello, se genera automáticamente otro fichero XML (Descriptor de Visualización de operaciones) a partir de los datos que posee la aplicación acerca de la consulta, que posteriormente se representará en el navegador web del usuario.

La interfaz Web de VISQUE incluye varias secciones que comprenden distintas funcionalidades del sistema. En la Figura 3 se puede observar la pantalla principal de la herramienta, en la cual se muestran 6 regiones (cada una de ellas señalada con una flecha numerada). La primera (1) muestra el esquema de interacción y visualización, anteriormente comentado, sobre el esquema de almacenes de datos, y la manera en la cual el usuario puede manipular los campos, seleccionándolos desde dicha estructura jerárquica que muestra una clara diferencia entre las *medidas* (rectángulos verdes) y las *dimensiones* (rectángulos rojos), así como las operaciones que se pueden aplicar a cada uno de los elementos del esquema (expandiendo cada uno de los nodos). Cuando se expande un elemento hasta el último nivel, se muestran todas las operaciones posibles sobre un determinado campo, bien sea una *medida* o un *atributo dimensional*.

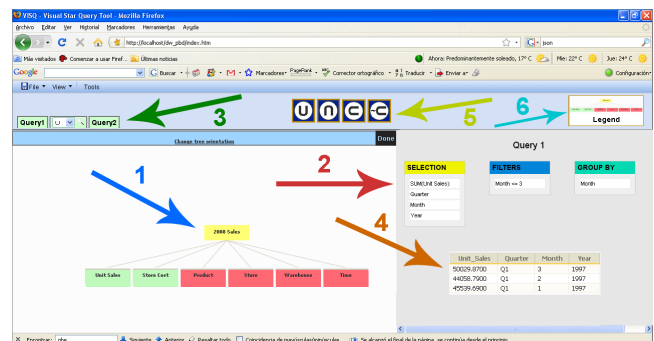


Figura 3. Interfaz de usuario para la manipulación de consultas visuales en VISQUE.

Entre las operaciones que permite la herramienta, y que el usuario puede realizar de forma visual, cabe mencionar la selección, el filtrado, los criterios de ordenamiento y el agrupamiento. Siguiendo con la Figura 3, la segunda región (2), es una representación de los elementos para la consulta que han sido seleccionados desde la estructura jerárquica mencionada de la región 1. Cada uno de dichos elementos se muestra en cajas con distintos colores, y especifican distintas partes de la cláusula SQL inferida. Una de las principales características de la herramienta es

la habilidad de representar visualmente operaciones tales como UNION, INTERSECTION, IN y NOT IN, que no son muy comunes en otras herramientas de consultas visuales. La tercera región (3), comprende un área tabular que representa independientemente cada una de las subconsultas que pueden generarse dentro de la consulta global, indicando las operaciones correspondientes entre las distintas consultas, y permitiendo al usuario observar los elementos de la operación resultante de forma visual. La cuarta región (4) se actualiza automáticamente para cada subconsulta independiente, y muestra los resultados (datos) en tiempo real. La quinta región (5) muestra un conjunto de botones que se utilizan para incluir nuevas subconsultas en función de una operación determinada. Por último, la sexta región (6) muestra una imagen que indica la leyenda del esquema de datos; al hacer clic sobre ésta, se proporciona una ayuda al usuario para entender la correspondencia entre colores, metáforas y elementos del esquema en caso de que el usuario lo necesite en cualquier momento.

Una vez que el usuario ha diseñado la consulta visualmente, el sistema la convierte automáticamente a código SQL, la envía al servidor de base de datos seleccionado a priori y VISQUE visualiza la consulta final de manera visual, así como los resultados obtenidos de su ejecución. El sistema permite una visualización de datos y de la consulta en tiempo real, por tanto al usuario se le muestra la información resultante de la consulta según éste la va construyendo paso a paso, lo que permite obtener una retroalimentación inmediata sobre la información que desea consultar en cada caso. Por otro lado, la representación de operaciones sobre conjuntos de forma visual ayuda al usuario a entender las relaciones entre los datos. Este aspecto no ha sido tratado en otras herramientas incluso del ámbito comercial, ya que la mayoría de estas herramientas se basan en modelos de tabla para realizar las consultas de manera visual. VIQUE cambia el modelo visual clásico para realizar consultas y permite su representación, y de los datos obtenidos, de manera más usable.

2.3 InterArch

InterArch es una herramienta CASE que permite la creación de la Arquitectura de la Información (AI) para el diseño de sitios web interactivos. 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 estético y organización de la información, InterArch le permite concentrarse en sus tareas de análisis conceptual dentro del dominio del problema. Esto facilita 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 genera diagramas UML para analistas e ingenieros software de forma automática, concretando los elementos que tienen su correspondencia con los diagramas de clase y objetos de contenido que utilizan los profesionales del software. Para ello, dicha información se generará en un formato textual y transportable en XMI, de forma que sea procesable por cualquier herramienta CASE existente con objeto de dar continuidad al resto de fases y actividades del proyecto software.

La herramienta de autor InterArch está compuesta principalmente por una serie de procesos que se encargan de la gestión y transformación de modelos en un entorno visual orientado al Arquitecto de la Información. 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 analistas e ingenieros del software.

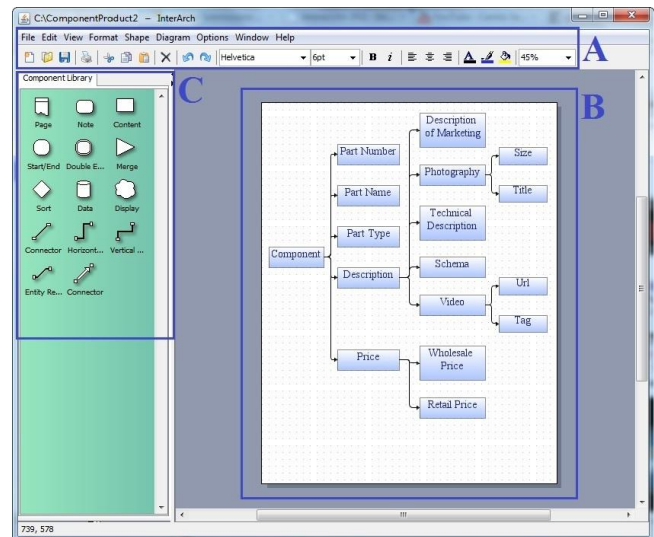


Figura 4. Interfaz de Usuario de InterArch dividida en tres partes: A, B y C.

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 para manipular e interactuar con los elementos visuales a través de distintas opciones de formato y edición. En la Figura 4 se muestra esta interfaz de usuario, cuyas partes principales están marcadas con letras mayúsculas (A, B y C).

En la parte A de la Figura 4, 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, etc. En la parte B, se detalla el entorno de trabajo principal que permite manipular y relacionar los distintos elementos visuales de que dispone la herramienta. En el ejemplo que aparece, se pueden apreciar las relaciones entre los elementos de contenido representados, las cuales describen la estructuración y precios de los componentes de cada producto en una tienda on-line. En la parte C de la Figura 4, 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. Existen tres tipos de elementos principales para el modelado visual: los elementos para la interpretación de contenido del modelado visual-conceptual de la AI, los elementos de enlace que permiten definir los tipos de asociaciones y relaciones entre los elementos de contenido, y los elementos de etiquetado que permiten incorporar información adicional (semántica) al modelado.

La herramienta InterArch tiene la ventaja de incorporar un mecanismo automático de transformación de modelos, no presente en las herramientas habituales de la AI, lo que permite aislar al experto de la información de aspectos técnicos relacionados con la programación del sistema interactivo, pero involucrándolo a su

vez en el propio proyecto software, principalmente en las fases tempranas de análisis de contenidos. De esta forma, la herramienta permite al experto en información concentrarse en el dominio del problema, y al profesional del software en el dominio de la solución a través de los diagramas UML que la propia herramienta genera, y que pueden ser integrados con el resto de elementos funcionales y no funcionales en el proyecto software para el diseño de la aplicación web interactiva que se está desarrollando.

3. EXPERIMENTACIÓN CON USUARIOS

Para corroborar las hipótesis de partida anteriormente descritas sobre el compromiso aceptable entre expresividad y facilidad de uso de las herramientas EUD presentadas, se idearon experimentos con usuarios utilizando las tres herramientas descritas. Esta experimentación así como los resultados más importantes alcanzados, se describe en los siguientes apartados.

3.1 Experimentos Realizados

En los tres casos, y con el objetivo de obtener una estimación de la usabilidad de las herramientas, se utilizaron las técnicas de evaluación *Test Retrospectivo* y *Thinking Aloud* [18], las cuales permiten obtener una grabación de la interacción del usuario con las herramientas y revisar posteriormente los registros de videos almacenados durante la realización del test, recompilando la información adicional necesaria e identificando, más fácilmente, tiempos medidos y conceptos erróneos al verbalizar los usuarios su experiencia con las herramientas.

De esta forma, en el experimento participaron 12 personas con perfil empresarial no expertos en informática, pero con conocimientos del dominio de las herramientas, es decir, con conocimientos en edición web, creación de consultas y contenidos web. La experimentación con usuarios se llevó a cabo siguiendo los pasos que se muestran a continuación para cada herramienta:

- 1) En primer lugar, se realizó una demostración de las distintas funcionalidades y características de las herramientas a evaluar, que tuvo una duración promedio de 10 minutos.
- 2) Seguidamente, se les explicó a los usuarios las tareas que tenían que realizar según la herramienta a evaluar en cada caso:
 - a. Para la herramienta DESK, los usuarios debían modificar una página web, relacionada con la temática del submarinismo, generada inicialmente por PEGASUS. En concreto, se le pidió al usuario realizar cambios que afectaban a algunos de los componentes que se generaron dinámicamente a través de relaciones específicas de los elementos de contenido a través de las ontologías, con el objetivo de sacar el máximo partido al proceso de ingeniería inversa seguido por la herramienta, y evaluar así la eficacia de la identificación y modificación de los elementos del dominio que dieron lugar a la generación de la página.
 - b. Para el caso de VISQUE, se solicitó a los usuarios realizar una serie de consultas sobre una base de datos de películas, de forma que pudieran explotar las capacidades visuales de la herramienta para llevar a cabo la extracción de la información de la base de datos, en concreto fueron 5 consultas las que se les pidió realizar a los usuarios.

- c. Para el caso de InteraArch, se solicitó a los usuarios elaborar un modelo de contenidos relacionado con el proceso de creación de registros de anuncios de una empresa de productos de segunda mano. En concreto, se les entregó a los usuarios de una maqueta de fidelidad media sacada de uno de los anuncios de la versión impresa de la revista Segunda Mano, y se les solicitó crear el modelo de contenidos utilizando la herramienta InteraArch.

- 3) Finalmente, y después de haber realizado las tareas correspondientes con cada una de ellas, se pidió a los usuarios que completaran un cuestionario para medir la usabilidad de las tres herramientas por separado. Para ello, se utilizó el cuestionario USE [12], con algunas variaciones proporcionadas por el cuestionario de Percepción de Utilidad y Facilidad de Uso de Davis [5], así como el Cuestionario de Usabilidad de Purdue [11]. Cada cuestionario contenía 31 preguntas cerradas para medir las variables relativas a *utilidad* (8), *facilidad de uso* (10), *facilidad de aprendizaje* (6) y *satisfacción* (7) de las herramientas, lo que se corresponde con las 4 dimensiones medidas para la estimación de la usabilidad. Cada pregunta se evaluó en una escala entre 1 (mínimo) y 10 (máximo). Además, se incluyeron preguntas abiertas enfocadas a obtener información sobre aspectos tanto positivos como negativos de las tres herramientas evaluadas.

3.2 Resultados Obtenidos

En la Figura 5 se muestra un gráfico con los resultados de la percepción de usabilidad de los usuarios en base a la experiencia realizada a partir de los resultados obtenidos del cuestionario de usabilidad para cada herramienta. El gráfico de barras representa los valores medios obtenidos para cada dimensión, además se incluye la barra de error ($\pm \sigma$) para cada dimensión analizada. Se muestra, de forma comparativa, los valores para las tres herramientas en distintos colores.

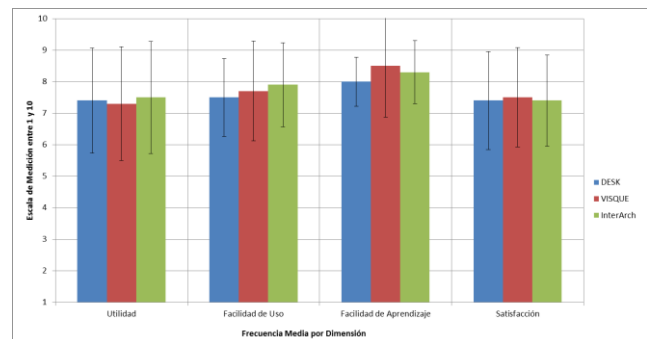


Figura 5. Usabilidad de las Herramientas DESK, VISQUE e InteraArch. Valores medios y errores ($\pm \sigma$) por dimensión en una escala entre 1 (mínimo) y 10 (máximo).

Como se puede observar en la Figura 5, la variable *facilidad de aprendizaje* es la que obtiene la puntuación promedio más alta de toda la muestra para las tres herramientas analizadas, con una media superior a 8 en ambos casos. La variable de *facilidad de uso* obtiene el segundo promedio más alto de las puntuaciones medidas, superior en todos los casos a 7.6. Le sigue la variable *utilidad*, que obtiene una puntuación promedio superior a 7.2 en todos los casos. Finalmente, la variable *satisfacción* obtuvo un promedio superior a 7.3 para las tres herramientas analizadas. En

general, las tres herramientas obtuvieron medidas muy similares en cuanto a las dimensiones de usabilidad analizadas.

En términos generales, se puede decir que el promedio de las puntuaciones de toda la muestra es de 7.7, para las tres herramientas, con una desviación estándar promedio de 1.4. Destaca el hecho de que todas las variables obtienen puntuaciones superiores a 7, con lo cual esta medida se puede considerar como un buen indicador de la usabilidad general de las herramientas, en base a la percepción de los usuarios.

Por otro lado, se estudió el grado de correlación entre las distintas dimensiones de usabilidad de las herramientas. Para ello, se calculó el coeficiente de correlación entre las variables medidas. Este análisis tiene como objetivo medir el grado de intensidad lineal de vinculación entre 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 crearon graficas 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). A ese respecto, se encontraron algunas correlaciones interesantes en las herramientas VISQUE e InterArch que se comentan a continuación.

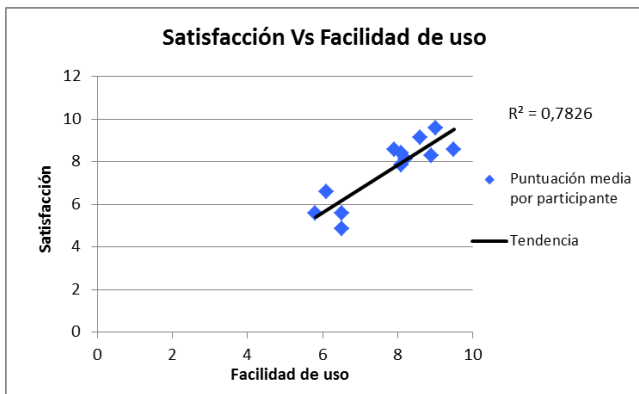


Figura 6. Correlación entre las variables *satisfacción* y *facilidad de uso* en la herramienta VISQUE.

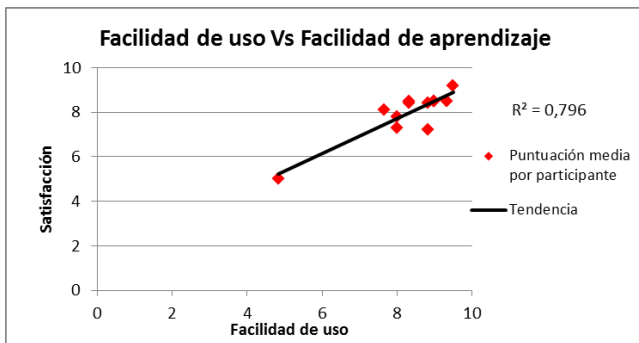


Figura 7. Correlación entre las variables *facilidad de uso* y *facilidad de aprendizaje* en la herramienta InterArch.

La Figura 6 muestra la correlación más alta encontrada para la herramienta VISQUE, que fue entre las variables *satisfacción* y *facilidad de uso*. Como se puede observar en la gráfica, el ajuste lineal reporta un coeficiente de correlación entre las dos variables de $R^2 = 0,7826$, indicando que existe una correlación positiva entre la percepción que tienen los usuarios de la facilidad de uso y

su satisfacción en el uso de la herramienta que aumenta según la herramienta es más sencilla de utilizar.

En el caso de la herramienta InterArch (Figura 7), la correlación más alta encontrada fue entre las variables de *facilidad de uso* y *facilidad de aprendizaje*, con un coeficiente de correlación lineal positivo entre las variables de $R^2 = 0.796$. Como se puede observar, la mayor parte de los usuarios manifiesta niveles altos de conformidad en cuanto a las cualidades de facilidad de uso de la herramienta. Esto se ve incrementado en la medida en que la herramienta ofrece mayores facilidades de aprendizaje.

Con respecto a las tomas de audio y video de las sesiones experimentales, los resultados han permitido analizar las herramientas de forma más exhaustiva, obteniendo información valiosa respecto a cómo los usuarios las utilizan. Por ejemplo, en el caso de DESK, los usuarios realizaron sin dificultades las tareas de edición propuestas, demostrando la herramienta un proceso correcto de inferencia en todos los casos. Los usuarios utilizaron la herramienta de forma similar a cómo lo hacen habitualmente en otros editores web, pero sacando partido al hecho de modificar enlaces entre contenidos para variar las relaciones entre los elementos de contenido en las ontologías del dominio. En el caso de VISQUE, en general la mayoría de los usuarios realizaron correctamente y de forma intuitiva las consultas, encontrando la herramienta, así como la metáfora visual y el método de interacción, útiles para la construcción de las consultas, y aprovechando las facilidades para crear consultas más complejas mediante operaciones visuales sobre conjuntos. Finalmente, en el caso de InterArch, los usuarios optaron mayoritariamente por utilizar funcionalidades visuales como seleccionar y arrastrar sobre los elementos de contenido creados (aprovechando mecanismos de herencia). No obstante, 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. Esto permitió detectar la necesidad de contar con más opciones de acceso directo y definir una organización que permita conseguir una consecución más ágil para alcanzar las opciones que requieren los usuarios.

4. CONCLUSIONES

Hoy en día, muchos usuarios no programadores han tomado un papel activo en la computación tradicional, construyendo y personalizando artefactos software para resolver problemas cotidianos. Esto corrobora que la computación actual está centrada en lo que los usuarios pueden hacer con los ordenadores, a diferencia de la computación tradicional donde lo importante era lo que los ordenadores podían hacer por sí mismos [23].

Este hecho ha provocado que surjan tendencias orientadas a transformar al usuario final en programador o diseñador de proyectos software, lo que puede desequilibrar las competencias profesionales, introducir cierto factor de riesgo y producir un impacto socio-económico importante. El trabajo que se presenta apuesta por las técnicas y herramientas basadas en el Desarrollo por el Usuario Final, lo que permite potenciar los principios de diseño, llegando a un compromiso aceptable entre utilidad y usabilidad, potenciando técnicas relacionadas con la interacción que permitan asistir al usuario final en las tareas más complejas, en vez de que éste acabe programando y/o gestionando sus propios proyectos software. Se trata, por tanto, de que los usuarios puedan diseñar o personalizar fácilmente nuevas aplicaciones, a la vez que los expertos realizan las partes más complejas (componentes, extensiones, herramientas de desarrollo, etc.),

buscando un equilibrio que permita optimizar las competencias de ambos colectivos. Este aspecto provoca un cambio de tendencia que desplaza a los actuales diseños desde el concepto de «facilidad de uso» (*easy-to-use*) hasta el concepto de «facilidad para el desarrollo de aplicaciones software interactivas» (*easy-to-develop interactive software systems*) [14].

Para llevar a cabo este objetivo, en este artículo se proponen tres herramientas basadas en este paradigma y orientadas al usuario final, experto en su dominio, que permiten diseñar y customizar artefactos software fácilmente por parte de usuarios no expertos en programación. Estas herramientas permiten la creación de contenidos y elementos que ayudan en la construcción de software, como es la autoría de páginas web dinámicas, la gestión de la arquitectura de la información, y la construcción de consultas complejas sobre bases de datos sin la necesidad de conocer SQL. Las tres herramientas aportan un lenguaje visual de manipulación fácil y directo en el dominio del análisis o del problema enfocado al usuario final, generando automáticamente la herramienta los elementos programáticos necesarios – modificaciones sobre ontologías del dominio, código SQL y diagramas UML, en el dominio de la solución.

Las tres herramientas han sido evaluadas, obteniendo resultados aceptables de usabilidad, con la idea de garantizar un compromiso aceptable entre usuarios y diseñadores, apostando por una disminución del riesgo y una solución a ciertos problemas de usabilidad actuales que tienen un impacto socio-económico importante.

5. AGRADECIMIENTOS

Esta investigación ha sido subvencionada por el Ministerio de Educación, proyectos TIN2011-24139 y TIN2011-15009-E.

6. BIBLIOGRAFÍA

- [1] Bastide R., Palanque, P.: A visual and Formal Glue between Application and Interaction. *Internacional Journal of Visual Language and Computing*. Academia Press Vol. 10, No. 5, pp. 481-507, 1999.
- [2] Bergman, L., Lau, T., Castelli, V., Oblinger, D.: Personal Wizards: collaborative end-user programming. *Proceedings of the End User Development Workshop at CHI'2003 Conference*. Ft. Lauderdale, Florida, USA. April 5-10, 2003.
- [3] Borges, C.R. and Macías, J.A. Feasible Database Querying Using a Visual End-User Approach. *Proceedings of the 2010 ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS 2010)*, Berlin, Germany, 19-23 June 2010, pp. 187-192.
- [4] Castells, P. Macías, J.A. An Adaptive Hypermedia Presentation Modeling System for Custom Knowledge Representations. *Proceedings of WebNet 2001 - World Conference on the WWW and Internet*. Orlando, Florida; October 23-27, pp. 148-153, 2001.
- [5] Davis, F.D.: Perceived Usefulness, Perceived Ease of Use and User Acceptance of Information Technology, *MIS Quarterly*, 1989, no. 13, pp. 319-340.
- [6] Dey, A.K., Shon, T.: Supporting End User Programming of Context-Aware Applications. *Proceedings of the End User Development Workshop at CHI'2003 Conference*. Ft. Lauderdale, Florida, USA. April 5-10, 2003.
- [7] EUSES Consortium: <http://eusesconsortium.org>
- [8] Klann, M.; Fit, F.: End-User Development. D1.1 Roadmap. *Proceedings of the End User Development Workshop at CHI Conference*. Ft. Lauderdale, Florida, USA. April 5-10, 2003.
- [9] Lieberman, H. (ed): *Your Wish is my Command. Programming By Example*. Morgan Kaufmann Publishers. Academic Press, USA. 2001.
- [10] Lieberman, H., Paternò, F., and Wulf, V. (eds). *End-User Development*. HCI Series. Springer Verlag, 2006.
- [11] Lin, H.Y.: A Proposed Index of Usability: A Method for Comparing the Relative Usability of Different Software Systems. *Behaviour and Information Technology*, 1997, no. 16, pp. 267-278.
- [12] Lund, A.: Measuring Usability with the USE Questionnaire, Usability and User Experience. *Special Interest Group 8*, 2001.
- [13] Macías, J.A. Intelligent Assistance in Authoring Dynamically-Generated Web Interfaces. *World Wide Web Journal*. Springer. Volume 11, Number 2, pp. 253-286, 2008.
- [14] Macías, J.A. and Paternò, F. Customization of Web Applications through an Intelligent Environment Exploiting Logical Interface Descriptions. *Interacting with Computers – The Interdisciplinary Journal of Human-Computer Interaction*. Elsevier. Volume 20, Issue 1, pp. 29-47, 2008.
- [15] Mori, G., Paternò, F., Santoro, C.: CTTE: Support for Developing and Analysing Task Models for Interactive System Design. *IEEE Transactions on Software Engineering*. IEEE Press. Vol. 28, No. 8, pp. 797-813, 2002.
- [16] Navarre, D., Palanque, P., Bastide, R. Sy, O.: Structuring interactive systems specifications for executability and prototypability. *7th Eurographics workshop on Design, Specification and Verification of Interactive Systems, DSV-IS'2000*; Springer Verlag LNCS. N° 1946, 2000.
- [17] Network of Excellence on End-User Development (EUD-Net). <http://giove.cnuce.cnr.it/eud-net.htm>.
- [18] Nielsen, J.: *Usability Engineering*, Morgan Kaufmann Publishers, 1993.
- [19] Paganelli, L., Paternò, F.: Automatic Reconstruction of the Underlying Interaction Design of Web Applications, *Proceedings of SEKE Conference*, pp. 439-445, 2002.
- [20] Repenning, A.: The Pragmatic Web: Customizable Web Applications. *Proceedings of the End User Development Workshop at CHI'2003 Conference*. Ft. Lauderdale, Florida, USA. April 5-10, 2003.
- [21] Rojas, L.A. y Macías, J.A. Soporte para el Análisis de la Arquitectura de la Información en Aplicaciones Web. *Revista Faz*, no. 5, enero de 2012, pp.10-27.
- [22] Scaffidi, C., Shaw, M. and Myers, B.: Estimating the Numbers of End Users and End User Programmers. *Proceedings of the 2005 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'05)*.
- [23] Shneiderman, B.: *Leonardo's Laptop*. The MIT Press, 2003.