

Refactorización del Modelo de Presentación de Sitios de Comercio Electrónico

Carlos Albaca Paraván ^{#1}

[#] *Facultad de Ciencias Exactas y Tecnología, Universidad Nacional de Tucumán
Tucumán, Argentina*

¹ calbaca@herrera.unt.edu.ar

Abstract— E-commerce is now a growing reality in the Web, as recent field studies conducted in them confirm the growing use of e-commerce sites to make purchases. In this context, it is essential to analyze the different models that define a web application in order to improve the usability of each model and provide a better user experience to customers sites. The objectives of this paper are to present concepts related to a web application and the three models that compose it, define what a refactoring, then characterize the presentation model refactoring, and finally show a catalog of e-commerce sites presentation model refactoring based on the case study of two e-commerce websites nationwide.

Resumen— El comercio electrónico es hoy una realidad creciente en la Web, ya que estudios de campo recientes realizados en éstos, confirman el creciente uso de los mismos para hacer compras. En este contexto, es imprescindible analizar los los diferentes modelos que definen una aplicación web de manera de mejorar la usabilidad de cada uno para brindar una mejor experiencia de usuario a los clientes de los sitios. Los objetivos de este trabajo son presentar los conceptos relacionados a una aplicación web y los tres modelos que la componen, definir qué es un refactoring, luego caracterizar la refactorización del modelo de presentación, y finalmente presentar un catálogo de refactoring específico del modelo de presentación de sitios de comercio basado en el caso de estudio de dos sitios web de comercio electrónico de alcance nacional.

I. INTRODUCCIÓN

Actualmente el uso que se le da a la Web es bastante diferente del propósito para el cual fue originalmente creada: un medio para compartir información entre algunos científicos. El alcance y la complejidad de los sistemas web actuales varían notablemente: desde servicios efímeros a pequeña escala hasta aplicaciones de empresa a gran escala distribuidas a través de la Internet y de las Intranets/Extranets corporativas [1].

Una aplicación web se basa y extiende un sistema web para agregar la funcionalidad de negocio. En términos más simples, una aplicación web es un sistema web que permite a los usuarios ejecutar lógica de negocio con un navegador web. Las aplicaciones web utilizan tecnologías que permiten hacer su contenido dinámico y permiten a los usuarios del sistema que afecten a la lógica de negocio en el servidor. La distinción entre los sitios web y aplicaciones web es sutil y se basa en la capacidad de un usuario de afectar el estado de la lógica de negocio en el servidor [2].

Muchas veces, la evolución de las aplicaciones es solamente impulsada por la reflexión de los desarrolladores en la estructura de la aplicación, el comportamiento y / o código; en estos casos, la aplicación es modificada no para añadir nuevas funcionalidades, sino para mejorar su

capacidad de mantenimiento y extensibilidad para futuras y eventuales adiciones de funcionalidad [3].

Es en este punto donde la refactorización o *refactoring* entra en juego. Un *refactoring* se define en el contexto de la orientación a objetos como una transformación sintáctica de código fuente, que mejora su estructura interna, preservando el comportamiento externo, es decir, la asignación de entrada a valores de salida [4]. Pero el *refactoring* no sólo puede ayudar a los desarrolladores en la mejora de la calidad del código, sino también puede apoyar el proceso de mejoramiento continuo de diseño de una aplicación web [5].

En este trabajo se caracterizará la refactorización del modelo de presentación de una aplicación web y se presentará un conjunto de *refactoring* que forman parte de un catálogo más extenso, resultado de un trabajo de investigación llevado a cabo mediante el estudio y aplicación de pautas de diseño web y la utilización de patrones de diseño web para mejorar la usabilidad de las aplicaciones web de comercio electrónico.

II. MODELOS DE UNA APLICACIÓN WEB

La complejidad del desarrollo de las aplicaciones web ocurre a diferentes niveles: dominios de aplicación sofisticados (financieros, médicos, geográficos, etc.); la necesidad de proveer acceso de navegación simple a grandes cantidades de datos multimedia, y por último la aparición de nuevos dispositivos para los cuales se deben construir interfaces web fáciles de usar. Esta complejidad en los desarrollos de software sólo puede ser alcanzada mediante la separación de los asuntos de modelización en forma clara y modular. La mayoría de las metodologías de diseño de aplicaciones web formalizan el diseño de una aplicación web a través de tres modelos: el de la aplicación (o contenido), el modelo de navegación, y el modelo de presentación.

A. Modelo de aplicación

El Modelo de Información (de Aplicación o Contenido) detalla los conceptos para especificar el contenido disponible para el usuario y cómo éstos pueden ser accedidos [6].

B. Modelo de navegación

El Modelo de Navegación especifica los conceptos que permiten al diseñador reorganizar la información para fines de navegación. Él debe volver a utilizar los elementos del modelo anterior para especificar los trozos de información actuales, junto con las relaciones entre ellos [6].

C. Modelo de presentación

El Modelo de Presentación define los conceptos necesarios para que el diseñador especifique cómo es publicado el contenido en las páginas y cómo se supone que los usuarios deben llegar a los datos dentro de la misma página o en páginas diferentes [6].

III. REFACTORING DEL MODELO DE PRESENTACIÓN

A. Caracterización de los refactoring del modelo de presentación de una aplicación web

El modelo de presentación de una aplicación web está compuesto por páginas y *widgets* que describen como luce la apariencia de éstas. Específicamente, esto incluye los siguientes elementos que pueden ser objeto de cambio [6]:

- La disposición general de la página.
- La forma gráfica de los *widget* que componen la página, su tipo y posición.
- Los nodos presentes en una página.
- La transformación ocurrida como resultado de una interacción de usuario.

Se define *refactoring* del modelo de presentación como un cambio en el modelo de presentación de la aplicación que preserva [7]:

1. El conjunto de las operaciones puestas a disposición por todas las páginas del modelo, considerado como un todo, y su semántica.
2. La disponibilidad de una interfaz abstracta para los elementos del modelo de navegación.

Bajo esta definición, los *refactoring* válidos del modelo de presentación pueden:

- Dividir o combinar páginas.
- Cambiar el tipo de un *widget* abstracto, si el nuevo tipo preserva la funcionalidad subyacente.
- Reorganizar la disposición de los *widgets* en una página.
- Agregar o cambiar la disponibilidad de efectos de interfaz.

B. Refactoring hacia patrones

Con la salida de metodologías ágiles, los desarrolladores se alejaron de la sobre-ingeniería de una aplicación y de la aplicación de patrones de diseño, siempre que sea posible. En su lugar, comenzaron con diseños simples y la aplicación de patrones solo cuando la solución agrega flexibilidad y no complejidad. Esto no quiere decir que haya una tendencia a la sub-ingeniería, que también es muy peligroso, pero se aplica para evitar la creación de diseños excesivamente complejos que son demasiado difíciles de mantener, donde el *refactoring* trata de ayudar a mantener el equilibrio entre la sobre y sub-ingeniería [8]. El proceso de desarrollo comienza con un diseño simple y, cuando se requiere mayor flexibilidad, este diseño se modifica para incorporar los patrones que resuelven el problema específico. Con el mismo espíritu de [8], en [9] se proponen *refactoring* para introducir patrones web en un modelo existente sólo cuando se necesitan, por ejemplo, porque se ha encontrado un problema en el uso de la aplicación la Web. Los patrones Web son similares a los patrones de diseño porque se ocupan de un problema de diseño recurrente (web) con una solución general que puede ser

instanciada de acuerdo a la aplicación específica que se esté resolviendo.

IV. HACIA UN CATÁLOGO DE REFACTORINGS DEL MODELO DE PRESENTACIÓN

Los *refactoring* presentados en este trabajo son parte de un caso de estudio donde se trabajó en dos sitios de comercio electrónico de tiendas de electrodomésticos de alcance nacional (Frávega y Garbarino), siguen la misma línea que los presentados en [3], [5], [7] y [9] entre otros, y son descriptos de igual manera que en [8], donde cada uno será mostrado usando una plantilla común que contenga el nombre, la motivación, la mecánica y un ejemplo de uso del mismo.

A. Reagrupar información

Motivación: Una página puede tener muchas clases diferentes de objetos de control que proveen diferente funcionalidad asociada con la posibilidad de realizar tareas no relacionadas entre sí, por lo que es conveniente agrupar los controles de acuerdo a la funcionalidad global, contextual y estructural, haciendo destacar cada grupo en un área diferente de la pantalla, proveyendo apariencia similar dentro de cada uno para mejorar la comprensión, tal como lo sugiere el patrón “*Behavioral Grouping*” [10].

Mecánica: Se deben reorganizar los elementos de la página de manera que estén agrupados de acuerdo a un tipo de funcionalidad y destacando de alguna manera cada grupo, manteniendo la homogeneidad de apariencia dentro de cada grupo de elementos.

Ejemplo: En la página de inicio del sitio de Frávega, al final de la misma se encuentra ubicada de manera desorganizada la información tanto de contacto, adhesión al envío de información por mail, lo relativo a las redes sociales, las certificaciones del sitio y un pie de página con información de la empresa. Al aplicar este *refactoring*, se organiza de manera ordenada toda esta información la cual tiene como denominador común que son formas de mantener un vínculo con la empresa y se genera una separación del resto del contenido por medio de la inserción de una línea recta horizontal. Las modificaciones aplicadas se pueden observar en la Figura 1.

B. Reajustar densidad de objetos por página

Motivación: Una página puede contener pocos o muchos objetos, pero estos siempre deben estar correctamente distribuidos en toda la extensión de la página para tener una correcta densidad de elementos. Para lograr esto, es necesario usar correctamente los espacios en blanco entre objetos, de manera de permitir una correcta separación entre ellos sin que esto afecte de manera negativa la navegación del sitio al introducir la utilización de *scrolling* vertical u horizontal quebrantando lo comentado por los patrones “*Homepage Portal*” y “*Above the Fold*” [10], o no teniendo en cuenta lo comentado por las pautas de diseño “*Use Moderate White Space*” y “*Optimize Display Density*” [11].

Mecánica: Se debe reorganizar los elementos de la página de manera que estén correctamente distribuidos a lo largo y ancho de la página web sin que esto genere espacios muy densamente poblado de elementos ni excesos en los espacios en blanco entre ellos.

Ejemplo: En página de inicio del sitio de Frávega, al final de la misma se encuentran ubicados de manera poco homogénea los elementos relacionados a la información de contacto, adhesión al envío de información por mail, lo relativo a las redes sociales, las certificaciones del sitio y un pie de página con listas de enlaces con información de la empresa. Para aplicar el *refactoring* (en este caso en particular primero se aplicó el *refactoring* “Reagrupar Información” para reorganizar los elementos de las secciones) se redujeron extensos espacios en blanco entre las secciones y se movieron elementos de las listas de forma ocupar más eficientemente el espacio libre y reducir la longitud de la página. Las modificaciones aplicadas se pueden observar en Fig. 1.

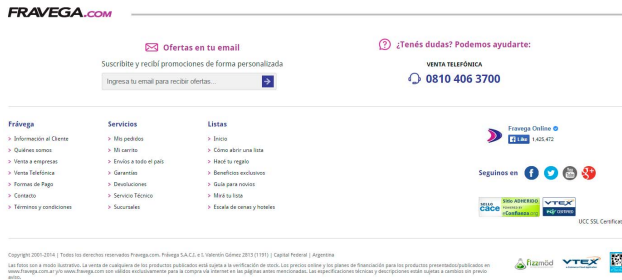


Fig. 1 (a). Sección inferior de la página de inicio de Frávega antes de la aplicación de los *refactoring* “Reagrupar Información” y “Reajustar Densidad de Objetos por Página”.



Fig. 1 (b). Sección inferior de la página de inicio de Frávega después de la aplicación de los *refactoring* “Reagrupar Información” y “Reajustar Densidad de Objetos por Página”.

C. Agregar el número de elementos que se obtuvieron como resultado de una búsqueda

Motivación: Los usuarios del sitio quieren saber el número de elementos que una búsqueda arrojó como resultado, ya que de esta manera tienen una noción rápida de la variedad de productos que posee el sitio y el tiempo que les tomará completar la lectura del listado generado como resultado. Este *refactoring* es una particularización del *refactoring* “Add Widget” [3].

Mecánica: En la página en la que se va a mostrar el resultado de una búsqueda generada por un cliente se debe agregar un widget que posea una etiqueta que indique que el número que se muestra a continuación es el total de elementos que la búsqueda arrojó como resultado. El número de elementos se puede haber obtenido por medio de la capa de aplicación, si es que el sitio ya proveía esta funcionalidad, o simplemente realizando una cuenta a nivel de capa de presentación.

Ejemplo: En el sitio de Frávega, el listado generado a partir de una búsqueda, no posee ninguna referencia al número de productos que arrojó como resultado la misma. Para la aplicación del *refactoring* se agregó en la misma línea, y antes que las opciones de ordenamiento, la etiqueta “X productos encontrados”, donde ‘X’ representa el número de productos que coinciden con los criterios de búsqueda

introducidos previamente por el cliente. Las modificaciones aplicadas se pueden observar en Fig. 2.

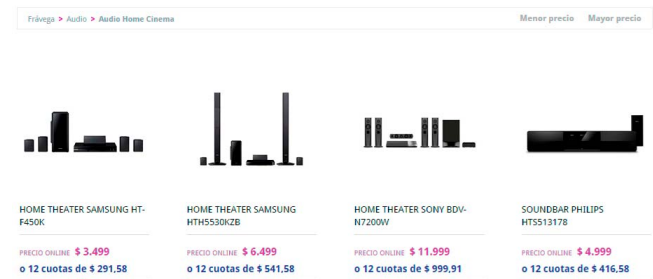


Fig. 2 (a). Parte superior del listado de productos resultado de una búsqueda realizada por el usuario antes de la aplicación del *refactoring* “Agregar el número de elementos que se obtuvieron como resultado de una búsqueda”.

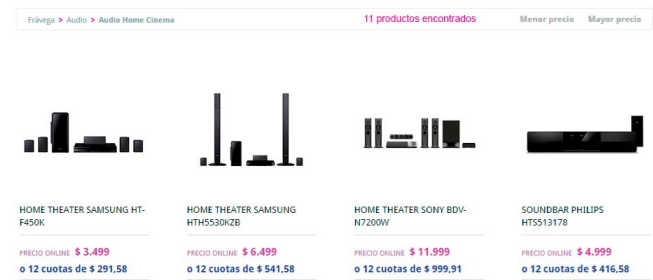


Fig. 2 (b). Parte superior del listado de productos resultado de una búsqueda realizada por el usuario después de la aplicación del *refactoring* “Agregar el número de elementos que se obtuvieron como resultado de una búsqueda”.

D. Organizar información en pestañas

Motivación: La organización de la información en una página es una tarea compleja que tiene como consecuencia la posibilidad de que el usuario permanezca en el sitio o se vaya, dependiendo si encuentra de forma rápida y agradable la información que está buscando. Para lograr una buena organización de la información y reducir la longitud de la página, es conveniente modificar la estructura en la que se muestra la información, pasando de una estructura lineal a una estructura de pestañas. Realizando este cambio en la presentación de la información, se posee la ventaja de estar acorde a los patrones “Above de Fold”, “Clean Product Detail” y “Tab Rows” [10], y se agrega la ventaja de reducir el desplazamiento vertical para la búsqueda de la información en la página lo que mejora su navegabilidad. Por último, la organización de la información en pestañas, permite agrupar la misma en categorías de información de manera que su búsqueda sea más intuitiva para el usuario.

Mecánica: La aplicación de este *refactoring* consiste en crear en la página, una sección que contenga la “Tab Rows” con la cantidad de pestañas necesarias para agrupar cada sección de información que se quiera reorganizar. Una vez creada la estructura, se debe desplazar cada sección de información de la página a la pestaña correspondiente. Es importante destacar que a cada pestaña se le debe poner una etiqueta que sea representativa de la información que contiene, ya que en caso contrario, podría ser contraproducente generando un impacto negativo en el efecto que se quiere producir en el usuario.

Ejemplo: En la página de producto del sitio de Frávega, las secciones de Descripción completa del producto y sus Datos técnicos se encuentran organizadas en 2 secciones que se encuentran una a continuación de la otra, cada una con una etiqueta que titula la sección de información. Para la aplicación de este *refactoring*, se generó una fila de 2 pestañas cada una etiquetada con el título que poseía la sección original y luego se desplazó cada sección a la pestaña correspondiente. Las modificaciones aplicadas se pueden observar en Fig. 3.



Figura 3 (a). Sección de la página de producto antes del *refactoring* “Organizar información en pestañas”.

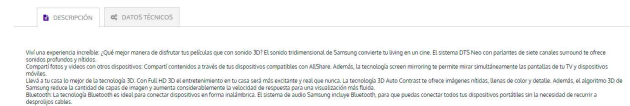


Fig. 3 (b). Sección de la página de producto después del *refactoring* “Organizar información en pestañas”.

E. Eliminar visualmente un widget

Motivación: Un *widget* es un elemento que es usado para interactuar con el usuario del sitio como por ejemplo *input text*, *drop-down list* o un *radio button*. Hay situaciones en las que no es necesaria la utilización de un *widget* ya que no hay interacción entre el sitio y el cliente. Por ejemplo cuando solo hay una opción para elegir, usar un *radio button* que tenga una sola opción y siempre esté marcada es ineficiente y hasta puede confundir al usuario, por lo que en estos casos es mejor reemplazar ese *widget* por una etiqueta que muestre esa información.

Mecánica: Para llevar a cabo este *refactoring* es necesario eliminar visualmente el *widget*, ocultándolo en la página, pero manteniéndolo activo para que el sitio pueda seguir funcionando correctamente si no se realizan cambios a nivel del modelo de aplicación. En caso de ser necesario, se puede agregar una etiqueta que avise acerca de la información que brindaba el respectivo *widget*.

Ejemplo: En la página del proceso de *check out* del sitio de Garbarino, cuando se llenan los datos de entrega se solicita que se elija de un *drop-down list* la fecha y hora en la que se quiere recibir el pedido. Cuando se desea elegir la franja horaria, el *drop-down list* solo posee una opción (de 8:00 a 20:00), por lo que se lo ocultó y se agregó una etiqueta que indique que la franja horaria en la que puede llegar el pedido es esa. Las modificaciones aplicadas se pueden observar en Fig. 4.



Fig. 4 (a). Sección de la página de *check out* antes del *refactoring* “Eliminar visualmente un widget”.



Fig. 4 (a). Sección de la página de *check out* después del *refactoring* “Eliminar visualmente un widget”.

F. Renombrar Elemento

Motivación: El nombre o etiqueta de un elemento (enlace, *widget*, función, etc.) es fundamental a la hora de que el usuario deduzca su forma de uso y el resultado a obtener por medio de la utilización del mismo. Por ello, tal como lo comenta la pauta de diseño “*Label Data Entry Fields Clearly*” [11], un nombre o etiqueta debe ser lo más entendible e intuitivo para el usuario.

Mecánica: La aplicación de este *refactoring* consiste en modificar el texto de un enlace, un elemento de una lista o la etiqueta de un campo de texto, etc. de manera que la misma identifique claramente que es lo que se le pide al usuario, que operación realiza una función o que resultado obtendrá el mismo interactuando con un *widget*.

Ejemplo: La página de listado de productos del sitio de Garbarino posee la función “Ordenar Por” con la posibilidad de ordenar los productos por precio (ascendente y descendente), marca y relevancia. La opción “Relevancia” es poco intuitiva en el criterio de ordenamiento que se usa (Luego de usarla, el usuario se puede dar cuenta que ordena los productos de los que tienen mayor descuento a los de menor descuento). Por esto, es conveniente renombrar esta opción - elemento del *widget drop-down list* - para permitir que la misma sea más entendible para el usuario. De la misma manera hay que renombrar la opción “Marca”, ya que no explica si se ordenan los productos de forma ascendente o descendente. La opción “Relevancia” podría modificarse “Mayores descuentos” de manera de orientar mejor al usuario sobre la funcionalidad de la opción, y en el caso de la opción “Marca”, se la puede modificar a “Marca Ascendente” o “Marca A-Z”. Las modificaciones aplicadas se pueden observar en Fig. 5.



Fig. 5 (a). Sección de la página de listado de productos donde se puede observar el *widget drop-down list* con sus respectivas opciones de ordenamiento antes del *refactoring* “Renombrar Elemento”.



Fig. 5 (b). Sección de la página de listado de productos donde se puede observar el *widget drop-down list* con sus respectivas opciones de ordenamiento después del *refactoring* “Renombrar Elemento”.

V. CONCLUSIONES

Usando como punto de partida pautas de diseño [11] y patrones de diseño web [10] se analizaron los modelos de presentación de dos sitios de comercio electrónico de alcance nacional. Como resultado de este análisis, se pudo inferir que dichos modelos no poseían un buen diseño, y en función de esto, se generó un catálogo de *refactoring* del modelo de presentación que contemple el uso de patrones de diseño web y cumpla con las recomendaciones de diferentes pautas de diseño. Como trabajo futuro, resta verificar mediante pruebas de usabilidad el impacto real de los *refactoring* creados y de esta manera extender su uso en otros sitios de comercio electrónico con falencias similares y obtener como resultado final una mejora en la usabilidad del sitio y en la experiencia del usuario.

REFERENCIAS

- [1] A. Ginige and S. Murugesan, “Web Engineering: An Introduction”, *IEEE Multimedia, Special issues on Web Engineering*, vol. 8, núm. 1, pp. 14-18, 2001.
- [2] J. Conallen, *Building Web Applications with UML*, 2nd ed., Addison-Wesley, 2002.
- [3] A. Garrido, G. Rossi and D. Distanto, “Model refactoring in web applications”, in *Proc. 9th IEEE Int. Symposium on Web Site Evolution*, paper 10.1109, pp. 89-96, 2007.
- [4] W. Opydyke, “Refactoring Object-Oriented Frameworks”, Ph.D. Thesis, Univ. of Illinois, Illinois, U.S.A., 1992.
- [5] L. Olsina, G. Rossi, A. Garrido, D. Distanto and G. Canfora, “Incremental Quality Improvement in Web Applications Using Web Model Refactoring”, *1st Int. Workshop on Web Usability and Accessibility*, pp. 411-422, 2007.
- [6] D. Silva and B. Mercerat, “Construyendo aplicaciones web con una metodología de diseño orientada a objetos”, *Revista Colombiana de Computación – RCC*, vol. 2 núm. 2, 2001.
- [7] A. Garrido, G. Rossi and D. Distanto, “Refactoring for Usability in Web Applications”, *IEEE Software*, vol. 28, núm. 3, pp. 60-67, 2011.
- [8] J. Kerievsky, *Refactoring to Patterns*. Addison-Wesley, 2004.
- [9] A. Garrido, G. Rossi and D. Distanto, “Systematic Improvement of web Application Design”, *Journal of Web Engineering*, vol. 8, núm. 4, pp. 371-404, 2009.
- [10] D. K. van Duyne, J. A. Landay and J. I. Hong, “The Design of Sites: Principles, Processes, and Patterns for Crafting a Customer-Centered Web Experience”, Addison-Wesley, 2003.
- [11] M. O. Leavitt, and B. Shneiderman, “Research-based web design & usability guidelines”, US Department of Health and Human Services, 2006.