

## HERRAMIENTA PARA REUSO DE CÓDIGO JAVASCRIPT ORIENTADO A PATRONES DE INTERACCIÓN

### IMPLEMENTS (TOOLS) FOR THE RE-USE OF JAVASCRIPT CODE ORIENTED TO INTERACCION PATTERNS

GLADYS BENIGNI, OCTAVIO ANTONELLI, YOVANNY VÁSQUEZ

*Coordinación Programa Licenciatura en Informática - Universidad de Oriente - Núcleo Nueva Esparta, Venezuela.  
E-mail: gbenigni@ne.udo.edu.ve, octavioantonelli@gmail.com, yovannyjvv@gmail.com*

#### RESUMEN

Las interfaces de usuario (IU) deben diseñarse cuidadosamente con el fin de obtener una aplicación útil y utilizable, pero los diseñadores Web no son expertos en estos temas y necesitan de una herramienta que solucione sus problemas. Esta investigación obedece a la necesidad de ofrecerle la ayuda necesaria que está buscando el usuario, por esta razón nace ReusMe, que implementada en ambiente Web, permite: personalizar, imprimir y/o copiar código reusable en el lenguaje Javascript, con el fin de generar patrones de interacción o componentes que ayuden al usuario a desarrollar interfaces usables; cada uno de estos patrones posee asociado un conjunto de características que le van a permitir obtener el máximo aprovechamiento de un patrón. ReusMe, se convertirá en el futuro en una herramienta útil y valiosa. *Útil* porque con el uso de patrones se facilitan las tareas de diseño de interfaces de usuarios; y *valiosa*, porque el usuario podrá conseguir ahorrar tiempo y no tener que comenzar su trabajo desde cero, sino, que puede reusar soluciones exitosas descritas en los patrones y así dedicarse a otros aspectos de la aplicación. Se utilizó el método conocido como el Lenguaje Unificado de Modelado (UML), la metodología que se empleó fue la del Proceso Unificado de Desarrollo de Software (PUD), propuesta por Rumbaugh, Jacobson, y Booch (1999), y las clases estereotipadas Conallen (1999) para modelar la arquitectura de la Web.

**PALABRAS CLAVE:** ReusMe, patrones de interacción, interfaces de usuario, usabilidad, reusabilidad.

#### ABSTRACT

User interfaces must be carefully designed in order to achieve a useful and usable application, but Web designers are not experts in this area and need specific methodologies to help solve this problem. This investigation was undertaken in order to provide the user with adequate support. The ReusMe program, when implemented in the web-medium, allows the personalization, printing and /or copying of re-usable codes in Javascript language, in order to obtain interaction patterns or components which help web designers develop usable interfaces. Each one of these patterns has an associated group of characteristics that permit the user to obtain maximum advantage from a pattern. ReusMe will become a useful and valuable tool: useful because the patterns will ease the task of designers of user interfaces; and valuable because the user will not have to start from scratch each time, but will be able to re-use successful solutions described in the patterns, thus saving time and permitting him or her to dedicate more time to other aspects of the application. The methods used were the Unified Modelling Language (UML) and the Unified Software Development Process (UDP), proposed by Rumbaugh, Jacobson.

**KEY WORDS:** ReusMe, Interaccion patterns, user interfaces, usability, re-usability.

#### INTRODUCCIÓN

El proceso de desarrollo de sistemas informáticos es una tarea bastante compleja, debido a la gran magnitud de problemas o situaciones que se pueden presentar al momento de su desarrollo; problemas que empiezan desde el proceso de análisis del mismo hasta

su implementación. Desde 1945 han existido variados lenguajes de programación con diversas ventajas tales como: mayor legibilidad del programa, menor tiempo de programación, mayor facilidad para el aprendizaje por parte de personas sin grandes conocimientos de informática, y sobre todo transportabilidad (Arahal 2003).

Actualmente, existen diversas herramientas de diseño rápido de aplicaciones (RAD) tales como: CBuilder, Visual Basic, Visual Basic.NET, Delphi, entre otras, que incorporan un ambiente visual fácil de usar permitiéndole al usuario desarrollar o diseñar interfaces de usuarios.

La aceptación final de una aplicación (software) por parte del usuario, depende principalmente de la percepción que éste tenga del sistema y ésta se logra mediante la interfaz de la aplicación. El diseño de interfaces habla mucho de la importancia de diseñar interfaces usables, fáciles, robustas, flexibles, entre otras características. Estas interfaces se obtienen a través de componentes tales como: botones, iconos, cajas de texto, menús desplegables, panel, entre otros.

Es importante destacar el papel que juega la usabilidad para diseñar interfaces manejables ya que ella es “la medida en que el producto se puede usar por determinados usuarios para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso particular” (Dix *et al.* 1997). Una interfaz se convierte en usable cuando los usuarios interactúan con la aplicación de tal manera que exista un intercambio de información entre el sistema y el usuario.

En la actualidad, se encuentra una gran cantidad de guías de diseños, pautas, estándares y/o reglas, que ayudan al diseñador o programador a crear interfaces posiblemente usables; estas guías contienen un modelo o muestra de cómo crear una interfaz, cómo usar el tipo de letra, tamaño de la fuente, color, entre otros, garantizándole al usuario final que tendrá una interfaz aceptable.

Hoy en día existe una herramienta informática (no considerado en este artículo como herramienta software) más útil y práctica que las guías de diseño, llamada *patrones de interacción* (Hassan 2004). Un patrón, partiendo de la definición de Christopher (1964), es una solución a un problema que se usa repetidamente en contextos similares con algunas variantes en la implementación.

Acosta y Zambrano (1999), indican que un patrón es un modelo a seguir que describe una solución exitosa de un problema particular en un contexto dado. Por ejemplo, un patrón de costura es un modelo a seguir por una costurera para la elaboración de un vestido, para una talla particular. En informática, un patrón es una descripción de una solución computacional exitosa a un problema recurrente en un contexto particular; esta solución expresa la experiencia y el conocimiento de expertos y, en este

sentido, sirve como medio de comunicación para difundir este conocimiento.

Los patrones de interacción nombran, abstraen e identifican todos los aspectos claves que intervengan en una estructura de diseño común (interfaces, botones, menús desplegables, entre otros), que sean útiles para la creación de un diseño reutilizable, definiendo en su cuerpo o esquema variadas soluciones a diferentes problemas. Cada patrón es adecuado para ser adaptado a un cierto tipo de problema que es planteado por el usuario final, describiendo cuándo aplicarlo y teniendo en cuenta otras restricciones de diseño.

Para esta investigación, se tomó como caso de estudio los patrones de interacción orientado al ambiente Web, en donde cada patrón representa un componente de la Web, que debe cumplir los siguientes atributos según la estructura de los patrones de la colección de Welie (2001):

1. Nombre: el título del patrón, el cual debe ser representativo, claro y conciso del concepto a comunicar.
2. Autor: quién propone al patrón.
3. Problema: descripción del problema desde el punto de vista del usuario.
4. Principio de Usabilidad: describe los principios o criterios de usabilidad en los cuales se basa el patrón.
5. Contexto: una descripción de la situación en la cual puede usarse el patrón, cuáles son las características del contexto, en términos de las tareas, del usuario.
6. Fuerza: aspectos del contexto que necesitan ser optimizados.
7. Solución: descripción clara de la solución propuesta (otros patrones pueden ser necesarios para completar la solución completa del problema).
8. Consecuencias: describe los resultados de aplicar el patrón.

La idea de patrones, por su misma naturaleza genérica de Problema-Solución-Reutilización, se ha difundido a nivel mundial. Actualmente, existen diversas investigaciones al respecto, las cuales se orientan a muy

distintos ámbitos de la actividad humana. Asimismo, dentro de la Ingeniería del Software se ha extendido, tan es así, que han creado y se siguen creando catálogos de patrones para problemas en distintos niveles de abstracción (Hurtado 2005).

En estos momentos la idea de reutilización cobra gran relevancia por parte de las personas que utilizan los patrones; una gran parte de los programadores o diseñadores están de acuerdo en que hay que reusar código, ya que los patrones contribuyen a reutilizar diseño, identificando aspectos claves de la estructura que pueden ser aplicados a una gran cantidad de situaciones. Hoy en día son numerosos los beneficios que se obtendría con la reutilización de código, tales como (Aceves 2004):

1. Rapidez de Desarrollo: puede llegar a ser más rápido ensamblar código que empezarlo de cero.
2. Bajo Costo: al ser más rápido y más fácil ensamblar que hacer se invierten menos recursos.
3. Calidad: es más fácil probar, comprender y adaptar una sola parte de código que todo un sistema.
4. Reducir el Mantenimiento: a menor tiempo de desarrollo menor tiempo de mantenimiento.
5. Incremento en la Consistencia: si los componentes de código siguen estándares serán más consistentes y fáciles de ensamblar.
6. Compartir Experiencia: al reutilizar código se comparte indirectamente la experiencia de desarrollo de otras personas.
7. Beneficios para el Negocio: menores costos, rapidez, estándares y documentación.
8. Aplicar Complejidad al Código: si hay partes complicadas de desarrollar pueden conjuntarse numerosos módulos que permitan lograr dicha complejidad.
9. Facilitar el Aprendizaje a través de un Buen Diseño: el código que se reutilice puede tener las mejores prácticas de desarrollo.
10. Desarrollo de “Una Sola Persona”: una sola persona puede encargarse de reutilizar e integrar código en los lugares donde se necesite.
11. Documentación: normalmente los componentes que son reutilizables están bien documentados o auto documentados.
12. Facilidad de Cambio: un código reutilizable está abierto a sufrir ajustes que puedan complementar las necesidades de su uso.
13. Diversas Propiedades: (a) fácil de entender, (b) una base semántica formal, (c) expresiva, (d) fácil de agregar, modificar o quitarle detalles, (e) haber utilizado las mejores técnicas de desarrollo, (f) interfaces simples, claras y precisas, (g) auto organizado, (h) verificable, (i) flexible sin comprometer su eficiencia, (j) independiente del lenguaje o plataforma.

No hay que reinventar lo que ya está inventado, se puede hacer uso de un código que ya esté probado, validado y que esté siendo usado, para crear ambientes Web usables. De allí, surge la importancia de estudiar lo referente a patrones de interacción y cómo ellos pueden facilitarle al diseñador o desarrollador de páginas web la creación de aplicaciones informáticas partiendo del reuso de código para interfaces gráficas de usuario (IGU), seleccionándose para este propósito una gran cantidad de código en lenguaje Javascript.

La importancia del uso de JavaScript para esta investigación radica en su sencillez y en la potencia para la creación de ambientes Web interactivos y usables; es sencillo porque el código es fácil de entenderlo y/o interpretarlo y es potente porque permite añadir a las páginas Web efectos y funciones adicionales a los contemplados en el estándar HTML (Alvarez 2001). Es por esto que hoy en día sigue siendo un lenguaje útil en la elaboración de páginas Web.

Con base en los aspectos anteriormente expuestos nace ReusMe, la cual es una aplicación en ambiente Web que permite proporcionar componentes Web en código reusable de Javascript basado en el principio de los patrones de interacción que faciliten al usuario final desarrollar interfaces gráficas usables. Por ejemplo: el usuario final podrá seleccionar de una lista de patrones un menú desplegable y, si es el caso, personalizarlo con los siguientes atributos: (a) color del menú: Azul, (b) color de la fuente: Blanco, (c) tamaño de la letra: 12, (d) tipo de letra: Arial, (e) tamaño del menú: 12x12, (f) entre otras solicitudes.

Dependiendo de cuál sea la solicitud del usuario final, el código del menú se va estructurando, en el sentido de que éste se adapte a las necesidades propuestas por el usuario. Al final, el usuario final va a obtener el código

con todas las características que solicitó con la aplicación, en formato de texto plano y PDF.

## MATERIALES Y MÉTODOS

Para la elaboración de ReusMe se contó con computador procesador Intel Pentium D 2.6 ghz, 512 Mb DDR RAM, y otros componentes. Las herramientas de software necesarias para la construcción del sistema se categorizaron de la siguiente manera: a) Plataforma Operativa: Windows XP Professional Server Pack 2; b) Herramientas de Diseño: Macromedia Dreamweaver 8., Macromedia Fireworks 8., Macromedia Flash 8 y Photoshop CS 2; c) Lenguaje de Programación: Php y JavaScript; d) Manejador de Base de Datos: MySql; e) Servidor Web: Apache Server; f) Navegador Web: Internet Explorer 6.0.

Se utilizó el método conocido como el Lenguaje Unificado de Modelado (UML). El UML es una metodología de modelado de desarrollo respaldado por el OMG (Object Management Group) que sigue las especificaciones del Proceso Unificado propuesto por Jacobson, Booch y Rumbaugh en 1997 y que prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan, para describir y modelar el desarrollo de la aplicación mediante una serie de diagramas y en conjunto con el *Proceso Unificado de Desarrollo Software (PUD)*. El PUD, o en sus siglas en inglés RUP, por su parte es desarrollado en sus inicios como Objectory Process que comprende desde la versión 1.0 a 3.8 en los años de 1987 a 1995 hasta alcanzar la versión 5.0 desarrollada en 1995; y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos, propuesto por Rumbaugh *et al.* (1999) y la *Extensión de Aplicaciones Web para UML (WAE)*. Además, el WAE, es el conjunto de extensión de UML propuesto por Conallen (1998), que está formado por valores etiquetados, estereotipos y restricciones (Tagged Values, Stereotypes and Constraints). La parte del mecanismo de la extensión de UML es la habilidad de asignar iconos diferentes a las clases estereotipadas. de Conallen (1999) resulta una metodología de fácil manejo y entendimiento, a la vez de ser robusta al desarrollar sistemas de gran amplitud bajo un entorno web.

PUD divide el proceso de desarrollo en ciclos, teniendo un producto al final de cada ciclo. Cada ciclo se divide en cuatro (4) fases, en donde se hace uso del UML.

La metodología PUD (Rumbaugh *et al.* 1999) se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema, cada ciclo consta de cuatro fases: *Inicio*, su función principal es establecer los objetivos para el ciclo de vida del producto y visión del proyecto. *Elaboración*, esta fase plantea la arquitectura óptima para el ciclo de vida del producto y se capturan la mayor parte de los requerimientos funcionales. *Construcción*, durante esta fase se crea el producto mediante sucesivas iteraciones e incrementos. *Transición*, su objetivo es realizar la entrega del producto operando, una vez realizadas las pruebas de aceptación por un grupo especial de usuarios y habiendo efectuado los ajustes y correcciones que sean requeridos. Los flujos de trabajos fundamentales considerados para cada fase son los siguientes: a) para la Fase de inicio, los diagramas de caso de uso; b) Fase de diseño: diagramas de secuencias y diagrama de clases; c) Fase de Construcción: diagrama de implementación y diagrama de despliegue y; d) Fase de Transición: se realizan las pruebas.

## RESULTADOS

### Herramienta para Reuso ReusMe

ReusMe, es una aplicación Web que nace con la finalidad de ser una herramienta para el desarrollo de sitios Web usables, proporcionando una serie de soluciones reutilizables en lenguaje javascript y que son adaptables o personalizables a las necesidades del usuario. Estas soluciones se encuentran orientadas al área de los patrones de interacción permitiéndole así a los usuarios obtener el máximo aprovechamiento de estos.

### Comparando ReusMe

Actualmente existen herramientas que permiten al diseñador reutilizar soluciones codificadas o crear estas de una manera rápida y eficiente, destacándose entre ellas a Velneo (2006), la cual nace con la concepción de ser una herramienta que facilita soluciones empresariales completas, desde la creación de plantillas, manejadores de base de datos, entre otras. Velneo cuenta con una herramienta para la rápida creación de programas, la cual abarca aspectos generales de la estructura de los datos y la interfaz de usuario a implementar, para el desarrollo de la IU. Welie (2001) es un claro ejemplo de lo que es un sitio Web basado en el uso de patrones. La finalidad de Welie es poner a disposición de los desarrolladores, patrones de diseño ubicados en distintas categorías como: tipo de sitio, navegadores, comercio, entre otros. Cada uno de los proyectos mencionados anteriormente se enfocan en brindar a los usuarios programadores

herramientas que faciliten su trabajo, reduciendo costo, tiempo y esfuerzo. Sin embargo, a nivel de flexibilidad y manipulación de las interfaces de usuario generadas, dichas aplicaciones son bastante rígidas, restringiendo al programador al uso de plantillas preestablecidas, que a pesar de ser personalizables, enmarcan al usuario en un único esquema. En contraposición, ReusMe, además de contar con todos los beneficios antes mencionados, pretende brindar al usuario una interfaz usable, a través del uso de patrones ya probados y evaluados por otros programadores, que aportan una solución viable al problema planteado, mediante el manejo independiente de cada componente en la interfaz diseñada, logrando una mayor aproximación con lo que el usuario desea realizar.

### Desarrollo de ReusMe

La metodología PUD (Rumbaugh *et al.* 1999) se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema, tal como se reseñó en el aparte 2. Cada fase de PUD consta de iteraciones, las cuales se guían a través del conjunto completo de los cinco (5) flujos de trabajos: *Requisitos*, el cual busca modelar todos los requerimientos o requisitos según sea la descripción del

sistema utilizando el modelo de caso de uso (compuesto por los actores y los casos de uso) (Figura 1).

**Análisis**, en este flujo se trabaja con un modelo de objetos conceptual llamado Modelo de Análisis. Este modelo permite refinar o demarcar de mejor manera los requerimientos de los usuarios, identificados en la etapa de requisitos, utilizando para este propósito los diagramas de colaboración que permiten modelar las interacciones entre los objetos entidad, interfaz y control (Figura 2). **Diseño**, en este flujo se identifica la interacción detallada entre los objetos de diseño que tiene lugar cuando se lleva a cabo el caso de uso y para modelar esta interacción se utilizan los diagramas de secuencias, siendo estos diagramas los que modelan las interacciones entre los objetos de diseño (Figura 3); y por otra parte debe modelarse el sistema y encontrar su forma, incluyendo en su arquitectura una forma que dé vida a todos los requisitos incorporados en el sistema a través de un diagrama de clases (Figura 4). En vista que PUD no es una metodología orientada al Web, la misma se complementa utilizando las clases estereotipadas de Conallen (1999), una vez modelado el diseño de la aplicación (Figura 5), permitiendo con estas clases representar objetos como las páginas de cliente y de servidor, formularios, enlaces, entre otros.

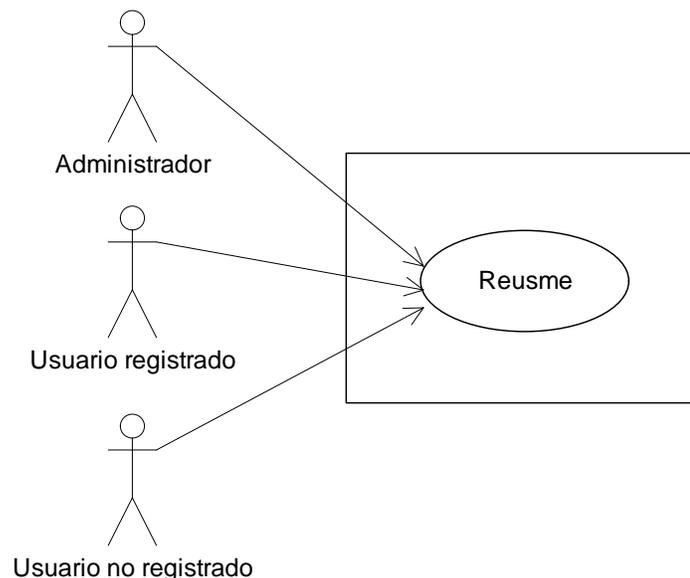


Figura 1. Caso de uso de los actores de ReusMe.

Herramienta para reuso de código...

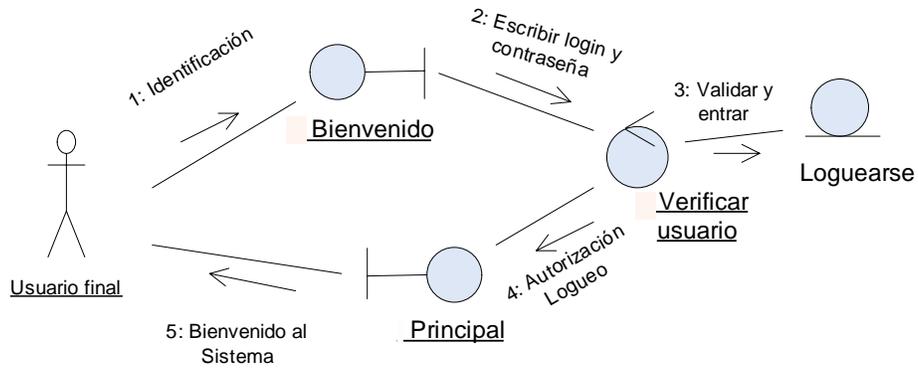


Figura 2. Diagrama de colaboración: Loguearse.

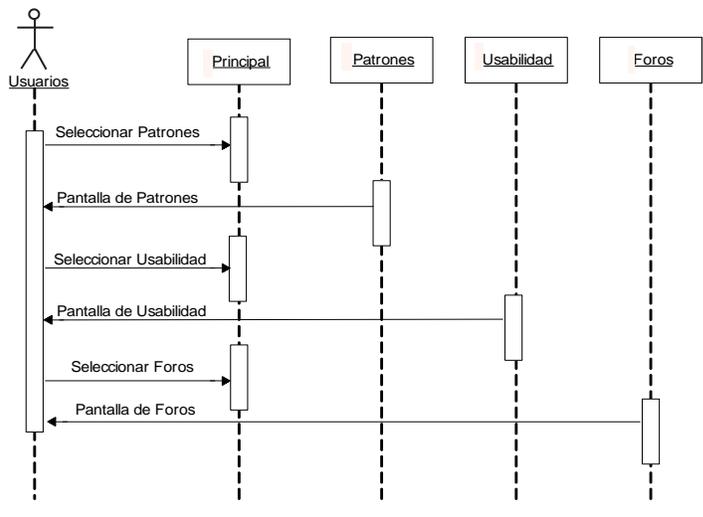


Figura 3. Diagrama de secuencia de la pantalla principal de ReusMe.

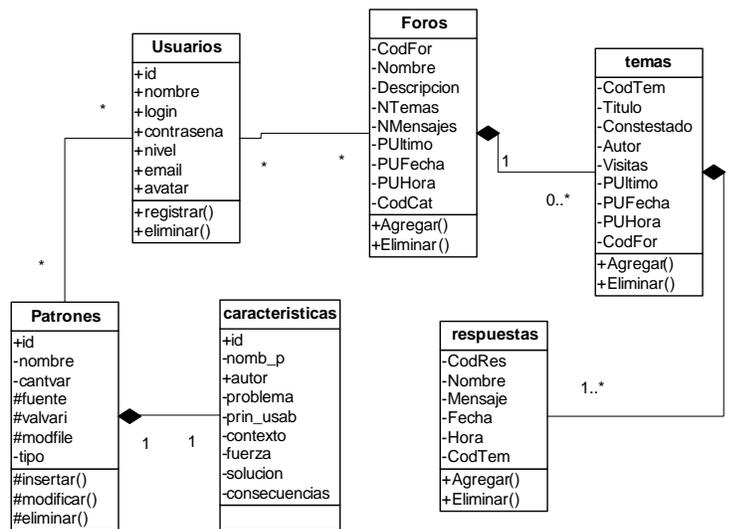


Figura 4. Diagrama de clase de la aplicación ReusMe.

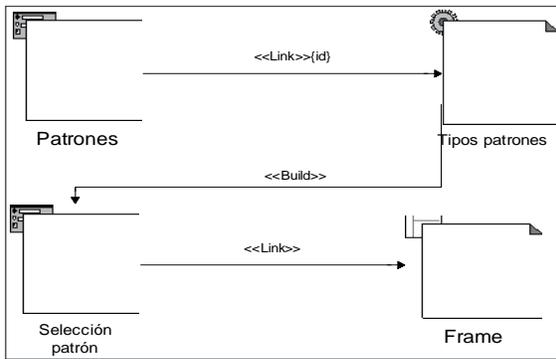


Figura 5. Diagrama de clase estereotipada patrones (ReusMe).

**Implementación,** en este flujo se trabaja con los resultados obtenidos en el diseño para implementar componentes funcionales o útiles sobre el sistema: es decir, ficheros de código fuente, scripts, entre otros (Figura 6). Prueba, en este último flujo se verifican los resultados de la implementación probando cada construcción, incluyendo cada construcción interna como intermedias, así como también las versiones finales.

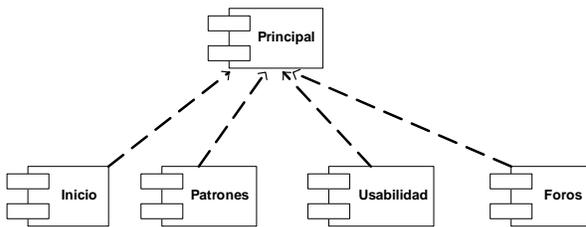


Figura 6. Diagrama de implementación de los componentes del sistema.

**Ejemplo** Para personalizar un patrón de tipo fecha dentro de ReusMe se debe seguir una secuencia de pasos que se detallan seguidamente:

1. Se debe elegir la opción Fecha ubicada en el *Menú Patrones* o a través de los iconos ubicados en el área central de la ventana (Figura 7).

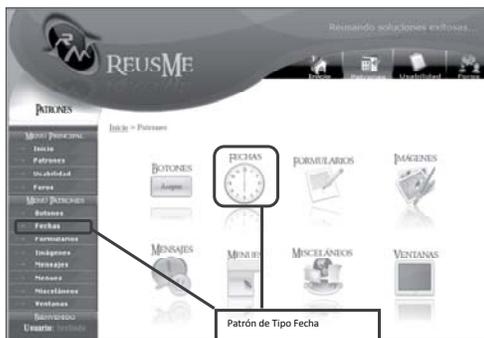


Figura 7. Pantalla de selección del patrón tipo fecha.

2. Una vez elegida la opción *Fecha*, se mostrarán los patrones de tipo fecha, así como su nombre, el autor quien lo propone y un pequeño resumen del mismo (Figura 8).

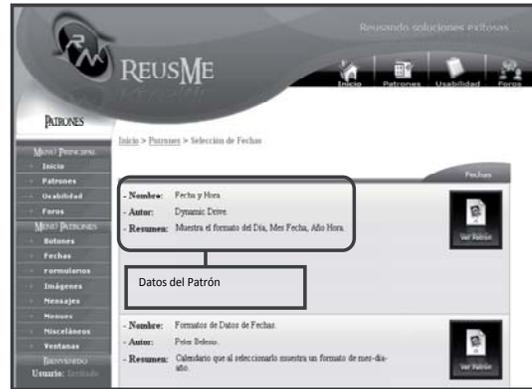


Figura 8. Seleccionar de la lista patrón.

3. Al escoger el tipo de patrón aparecerá una pantalla que contendrá información del patrón seleccionado, así como las propiedades del mismo que pueden ser modificadas a gusto del usuario (Figura 9).

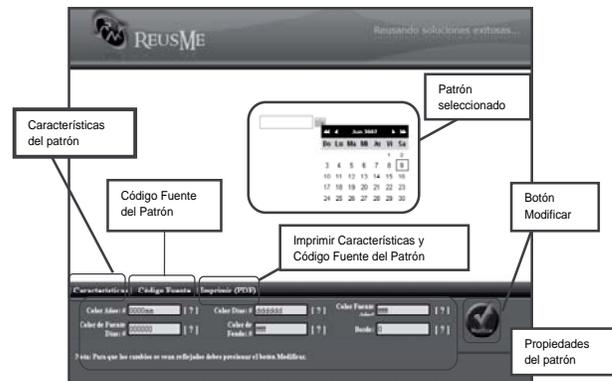


Figura 9. Personalizar patrón.

4. Una vez llenado todos los datos, se debe hacer click en el botón personalizar para de esta manera efectuar todos los cambio solicitados por el usuario.

5. Si el patrón de tipo fecha cumple las expectativas del usuario, éste podrá copiar su código fuente (Figura 10), visualizar sus características (Figura 11) y/o imprimir el código fuente como sus características.

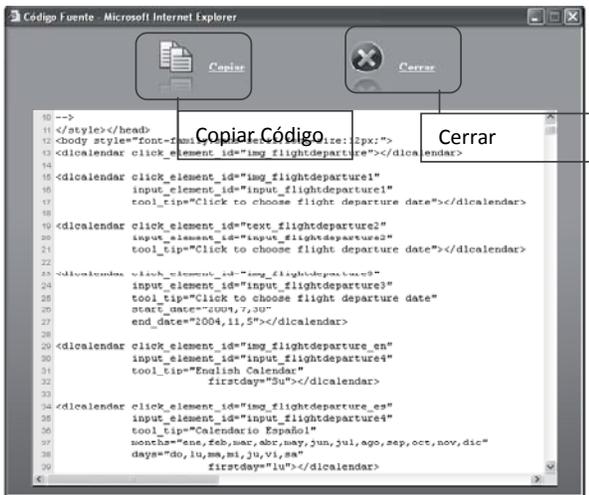


Figura 10. Copiar código fuente.

Características - Microsoft Internet Explorer	
Nombre	Formatos de Datos de Fechas.
Autor	Peter Belesis.
Problema	El usuario desea introducir datos de fechas y no desea preocuparse por la sintaxis del dato.
Principio de Usabilidad	Guiar al usuario y prevenir errores.
Contexto	Todas las páginas Web que requieran que el usuario introduzca fechas.
Fuerza	Los datos de fechas tienen múltiples sintaxis.
Solución	Permitir que el usuario elija la fecha de un calendario como en el mundo real.
Consecuencia	El solo hecho de permitirle al usuario que puede seleccionar una fecha y automáticamente mostrársela en un formato mes-día-año, garantiza prevenir los posibles errores.

Figura 11. Características del patrón.

## DISCUSIÓN

Con el surgimiento de patrones (particularmente los de interacción) y lenguajes como Javascript, es más común ver en Internet páginas Web que rompen con los esquemas de interacción tradicionales y de componentes de interfaces estáticos y casi sin vida, que hacían a los usuarios de estos sitios abandonarlos rápidamente por otros que les ofrecen mayor dinamismo, facilidad y calidad de uso. Estos sitios se enfocan principalmente en las necesidades de los usuarios visitantes a los mismos y que este pueda identificarse mejor con el sitio Web que está visitando, garantizando así que el usuario se desenvuelva

fácilmente y mantenga su permanencia en el mismo.

Con la creciente tecnología y cambios que está sufriendo la Web para ser transformada en la esperada Web 2.0, es cada vez más común ver la migración de herramientas de escritorio al World Wide Web y gracias a esto se puede disponer de las herramientas online en cualquier momento, sin importar en donde nos encontremos.

Las interfaces de usuario deben diseñarse cuidadosamente con el fin de obtener una aplicación útil y utilizable, pero los diseñadores de sitios Web no son expertos en estos temas y necesitan de una herramienta que solucione su problema.

Estos usuarios, se enfrentan a diferentes situaciones al momento de crear su aplicación tales como: ¿cuántos colores debería llevar la interfaz?, ¿qué objetos incluiría en cada una de ellas?, ¿debería utilizar iconos?, ¿qué color debería usar?, ¿cómo ayudar al usuario a encontrar lo que está buscando?, ¿cómo presentárselo?. Darle una respuesta a estas preguntas no es tarea fácil.

Acosta y Zambrano (1999) exponen que en la actualidad, se encuentra una variada disponibilidad de Guías de Diseño que son documentos que recogen normativas y patrones básicos relacionados con el “aspecto” de una interfaz para su aplicación en el desarrollo de nuevas “pantallas” dentro de un entorno concreto. (sitio Web de contenidos, nuevas secciones, entorno de aplicaciones de negocio) (Villa 2004). Estas poseen distintos niveles de abstracción tales como: Guías de Estilos (*Apple, IBM, Sun, W3C, Yale Center for Advanced Instructional Media, National Cancer Institute (NCI)*), Principios (*Simpson, Preece, Mandel y Dix*), Lineamientos (*Microsoft, Apple, Sun, IBM*) y Estándares (*XHTML 1.0 Transicional, CSS nivel 2, WAI-AAA*), las cuales ayudan al usuario final a diseñar interfaces *posiblemente* usables.

Un *principio* es un concepto de alto nivel, que puede ser muy abstracto (aplicables a casi cualquier dominio) o muy específico (aplicable a un dominio en particular). Debido a que los principios son generalizaciones que omiten los detalles, éstos son especializados por lineamientos y estándares de diseño. Un *lineamiento* es también una guía de diseño pero es más específica que un principio. Un *estándar* es aún más específico que un lineamiento, lo que implica que se refiere a un aspecto concreto, en general, asociado a una organización y aplicado a diversos aspectos de la empresa. Los estándares se pueden considerar como la guía de diseño más específica

y concreta para el desarrollo de interfaces de usuario.

En fin, éstas son variadas las herramientas de diseño de interfaces que se dedican a brindar una gran gama de utilidades y funciones que necesite el usuario para desarrollar su aplicación. Cabe destacar que ninguno de los tipos de guías de diseño antes mencionados trabajan de forma detallada, siguen siendo muy generales en relación al tipo de contexto que se esté trabajando. Por otra parte, las guías de diseño no garantizan la usabilidad de un producto, es decir, son elementos muy útiles para la creación de interfaces de usuario, pero deben ser usados adecuadamente (Acosta y Zambrano 1999).

Hoy en día, con el uso de las herramientas de creación de interfaces antes mencionadas, se presentan una serie de problemas debido a que las guías de diseños son: muy generales, difíciles de aplicar, su interpretación puede ser ambigua, abstracta, no ayudan con la especificación del problema y poseen una extensas listas de lineamientos que no dejan claro cuáles aplicar.

Por todo esto mencionado, surgió la necesidad de acudir a la ayuda de los expertos que diseñan páginas Web, de tal manera que se pueda dar reuso a los códigos que nos ofrecen para generar los componentes de la interfaz, y no inventar lo que ya está diseñado, y poder agrupar estos códigos en patrones de interacción, ya que al igual que las guías de estilos son útiles y utilizables.

Existe una gran similitud entre los estándares y los patrones de interacción ya que ambos están orientados a facilitar la tarea del diseñador de interfaces de usuarios; más aún, es factible utilizar patrones para describir estándares y se obtendría una estructura legible, fácil de entender y que proporcione una mejor comunicación.

La diferencia entre patrones de interacción y guías de diseño es que los patrones son centrados en un contexto bien detallado mostrando cuándo, cómo y por qué la solución es aplicable, las guías de diseño no proporcionan una solución exacta por ser tan generales. Otra diferencia es que las guías de diseño sólo orientan a cómo crear la interfaz, y los patrones de interacción, proporcionan los componentes o códigos que el usuario esté necesitando para crear su interfaz (Acosta y Zambrano 1999).

Esta investigación obedeció a la necesidad de ofrecer la ayuda necesaria al usuario que diseña aplicaciones Web, por lo que se desarrolló un software que implementado en ambiente Web, permite: personalizar, imprimir y/o copiar código reusable en el lenguaje Javascript, con el

fin de generar patrones de interacción o componentes que ayuden al diseñador Web a desarrollar interfaces usables.

## CONCLUSIONES

Actualmente son cada vez más las páginas Web que son desarrolladas utilizando los patrones de interacción, pues los mismos, le proporcionan al diseñador una herramienta fiable para el desarrollo de sitios Web usables y en menor tiempo. El uso de patrones permite reutilizar código usable que ya ha sido probado anteriormente, coincidiendo diferentes investigadores en el slogan de “¿Porqué reinventar la rueda?”, ya que el tiempo en reinventar soluciones que ya existen pueden ser aplicadas en el contexto o problemas que se presenten. Los diseñadores de páginas o aplicaciones Web buscan continuamente soluciones exitosas a problemas que son recurrentes en su día a día. ReusMe concentra un conjunto de soluciones que ya han sido probadas y que pueden adaptarse fácilmente al contexto donde desee aplicarse. Al trabajar con patrones obtenemos todos los beneficios que estos ofrecen, como son la reusabilidad, soluciones probadas y sobre todo usables.

## REFERENCIAS BIBLIOGRÁFICAS

- ACEVES L. 2004. *Reutilización de Código*. [Documento en Línea]. Disponible: <http://www.udem.edu.mx/udem/profesores/laceves/rad/m12arad.pdf>. [Consulta: 2006, Noviembre 21].
- ACOSTA E., ZAMBRANO N. 1999. *Interacción Humano Computador: Fundamentos de Diseño*. ND 99-02. Lecturas en Ciencias de la Computación, ISS I316-6239. Escuela de Computación, U.C.V. Caracas, 1999.
- ALVAREZ M. 2001. *¿Qué es Javascript?*. [Documento en Línea]. Disponible: <http://www.desarrolloWeb.com/articulos/25.php>. [Consulta: 2006, Noviembre 21].
- ARAHAL M. 2003. *Fundamentos de Informática para Ingeniería Aeronáutica*. [Documento en Línea]. Disponible: <http://www.esi2.us.es/~jaar/Datos/FIA/fi1aer04.pdf>. [Consulta: 2006, Noviembre 21].
- CHRISTOPHER A. 1964. *An Introduction for Object-Oriented Designers*. [Documento en Línea]. Disponible: <http://g.oswego.edu/dl/ca/ca/ca.html>. [Consulta: 2006, Septiembre 22].
- CONALLEN C. 1999. *Modeling Web Application*

- Architectures with UML*". [Documento en Línea]. Disponible: <http://www.itmWeb.com/essay546.htm>. [Consulta: 2006, Septiembre 22].
- DIX A., FINLAY J., ABOWD G., BEALE R. 1997. *Patrones de Interacción para el Diseño de Interfaces Web Usable*. [Documento en Línea]. Disponible: <http://ccc.inaoep.mx/~grodrig/Descargas/com10017.pdf> [Consulta: 2006, Septiembre 27].
- HASSAN Y. 2004. *Patrones de Diseño de Interacción*. [Documento en Línea]. Disponible: <http://www.bitacoras.sidar.org/g4/index.php?2005/08/16/8-patrones-de-disentildeo-de-interaccioacuten-i-introduccioacuten>. [Consulta: 2006, Septiembre 23].
- HURTADO O. 2005. *Evolución y Orientaciones de Patrones*. [Documento en Línea]. Disponible: <http://www.monografias.com/trabajos27/evolucion-patrones/evolucion-patrones.shtml>. [Consulta: 2006, Noviembre 21].
- RUMBAUGH J., JACOBSON I., BOOCH G. 1999. *El Lenguaje Unificado de Modelado*. Madrid: Addison-Wesley.
- VELNEO S.A. 2006. *Inicio > Velneo.com*. [Documento en Línea]. Disponible: <http://www.velneo.com/>. [Consulta: 2006, Septiembre 26].
- VILLA L. 2004. *Guías de diseño: normalización y usabilidad*. [Documento en Línea]. Disponible: [http://www.alzado.org/articulo.php?id\\_art=327](http://www.alzado.org/articulo.php?id_art=327). [Consulta: 2006, Septiembre 1].
- WELIE M. 2001. *Pattern Languages in Interaction Design*. [Documento en Línea]. Disponible: <http://www.welie.com>. [Consulta: 2006, Septiembre 27].