

Hacia Aplicaciones Web con Términos Difusos

José Ángel Labbad¹, Rosseline Rodríguez², Leonid Tineo²
joseangellabbad@gmail.com, crodrig@usb.ve, leonid@usb.ve

¹ Universidad Metropolitana, Caracas, Venezuela

² Departamento de Computación, Universidad Simón Bolívar, Caracas, Venezuela

Resumen: En la actualidad, casi cualquier ámbito de la vida cuenta con una aplicación Web relacionada que dispone de formularios de búsqueda. Lamentablemente, estas aplicaciones tienen problemas de rigidez propios del uso subyacente de la lógica booleana, los cuales se pueden superar gracias a la existencia de lenguajes de consulta a bases de datos, extendidos con lógica difusa. Así, los formularios de búsqueda pueden enriquecerse usando términos difusos en los controles Web. En este trabajo se hace una propuesta metodológica para abordar la migración de una aplicación Web existente hacia una nueva que haga uso de formularios Web con términos difusos, superando los problemas de rigidez de las aplicaciones tradicionales. Para ello, se realizó un estudio de opinión a programadores y usuarios Web para conocer si los programadores desean utilizar términos difusos en sus desarrollos, y si los usuarios se inclinan a realizar búsquedas en formularios Web que proveyeran de estos términos para expresar sus preferencias. Posteriormente, se diseñó un mecanismo sistemático que permitiera analizar los controles Web existentes en los formularios de búsqueda de una aplicación para incorporarles términos difusos en sus opciones de entrada de datos. Se evaluaron las diferentes extensiones con lógica difusa de los SGBD que existen actualmente, en base a criterios como completitud, disponibilidad, visibilidad, calidad, soporte, portabilidad y documentación. Se seleccionó SQLf como la que satisface la mayoría de estos criterios. Finalmente, se plantea una propuesta metodológica para la migración de un sitio Web con el fin de que sus formularios de búsqueda usen términos difusos presentes en el lenguaje natural.

Palabras Clave: Formularios Web; Términos Difusos; Controles Web; Lenguaje Natural; Migración.

Abstract: Currently, almost any area of life has a related Web application that contains search forms. Unfortunately, these applications have rigidity problems inherent to the underlying use of Boolean logic, which can be overcome thanks to the existence of database query languages extended with fuzzy logic. Thus, search forms can be enriched by using fuzzy terms in Web controls. In this work a methodological proposal is made to approach the migration of an existing Web application to a new one that makes use of Web forms with fuzzy terms, overcoming the rigidity problems of traditional applications. For this, an opinion study was made to programmers and Web users to know if programmers want to use fuzzy terms in their developments, and if users are inclined to make searches in Web forms that provide these terms to express their preferences. Subsequently, a systematic mechanism is designed to analyze the existing Web controls in the search forms of an application to incorporate fuzzy terms in their data input options. The different extensions DBMS with fuzzy logic that currently exist were evaluated, using criteria such as completeness, availability, visibility, quality, support, portability and documentation. SQLf was selected as that one that satisfies most of these criteria. Finally, a methodological proposal for the migration of a Web site is suggested to use fuzzy terms present in the natural language in search forms.

Keywords: Web Forms; Fuzzy Terms; Web Controls; Natural Language; Migration.

I. INTRODUCCIÓN

En la actualidad los usuarios de aplicaciones Web demandan mayor flexibilidad y adaptación a sus requisitos o preferencias. La mayoría de estas aplicaciones cuentan con formularios de búsqueda para dar acceso a los datos que ellas gestionan, los cuales en muchas oportunidades presentan rigidez en la adaptación a las demandas de los usuarios.

Para flexibilizar la gestión de datos y búsquedas a través de formularios Web [1], se ha propuesto la lógica difusa [2] como

herramienta para la especificación de términos del lenguaje natural. Esta lógica [2] fue definida en base a la teoría de conjuntos difusos [3], propuesta por Zadeh, como mecanismo para dar tratamiento matemático y computacional a los términos vagos del lenguaje natural. Es por ello que se han hecho estudios dirigidos a definir lenguajes que permitan realizar consultas difusas a bases de datos relacionales [4]. Estos lenguajes son extensiones al lenguaje estándar para consultas a bases de datos, llamado SQL (*Structured Query Language*) [5], los cuales se han incorporado en los Sistemas Gestores de Bases de Datos (SGBD) más usados [6][7][8][9].

Algunas de estos SGBD extendidos están disponibles para ser utilizados libremente, sin embargo, no se observa que se haya masificado su uso a nivel de aplicaciones Web. Por otro lado, algunas de las extensiones de SQL con lógica difusa cuentan con implementaciones en distintos SGBD, sea en el código fuente o mediante una capa lógica. Sin embargo, aún no existe un número significativo de aplicaciones que aprovechen las ventajas de trabajar con lógica difusa usando un SGBD.

En este trabajo, se muestra el resultado de un estudio que valida la factibilidad de usar términos difusos en formularios Web, por parte de programadores y usuarios. Además, la principal contribución de este trabajo es una propuesta sistemática y metodológica que permite transformar los formularios de búsqueda existentes dentro de una aplicación Web operativa a una versión modificada que permita la gestión de términos lingüísticos vagos en dicho formularios usando lógica difusa.

En tal sentido el trabajo se estructura como sigue. En la Sección II se plantean los antecedentes de la investigación. En la Sección III, se exponen los conceptos que fundamentan la investigación. En la Sección IV, se muestra un estudio de opinión realizado a programadores y usuarios de aplicaciones Web sobre la posibilidad de realizar búsquedas online usando términos difusos. En la Sección V, se explica el proceso metodológico a seguir para incorporar términos difusos a formularios Web. En la Sección VI, se propone la metodología de migración de formularios Web tradicionales para que incluyan términos difusos. Finalmente, se presentan las conclusiones y trabajos futuros.

II. ANTECEDENTES

A pesar del reciente surgimiento de tendencias NO-SQL [10], hoy en día, el lenguaje de consultas a bases de datos más utilizado a nivel mundial sigue siendo SQL [11]. Gracias a su uso masivo en diversas áreas de aplicación, la perspectiva es que SQL siga siendo ampliamente usado en el futuro.

La incorporación de lógica difusa como mecanismo para flexibilizar los lenguajes de consultas a base de datos se ha venido estudiando por varios años. Las propuestas más completas son SQLf [7] y FSQ [8]. Usando SQLf como lenguaje de consulta, en [12] se desarrollaron aplicaciones basadas en lógica difusa para expresar las preferencias de usuario: un sistema de evaluación de cursos docentes, un sistema de promoción de personal, y un sistema de compra y venta de vehículos. Estas aplicaciones fueron implementadas sobre una extensión en capa lógica del SGBD Oracle denominada SQLfi [12].

En base a esa experiencia y otras previas, en [13] se definieron elementos gramaticales y características que determinan cuando una aplicación posee requisitos difusos, es decir, aquellos que poseen términos lingüísticos vagos. En este trabajo se definen parámetros que ayudan a establecer cuándo una aplicación podría aprovechar el uso de la lógica difusa dentro de un SGBD. Por otro lado, se dan lineamientos para que los desarrolladores puedan identificar, en los requisitos escritos en lenguaje natural, los términos de la lógica difusa que luego serán usados en la definición de consultas a la base de datos.

Posteriormente, en [14] se estudió la incorporación de los beneficios de la lógica difusa en las metodologías de desarrollo de software a fin de expresar requisitos que involucran preferencias de usuario. En este trabajo, se añade al ciclo de vida de desarrollo de software, en las fases de análisis, diseño e implementación, los aspectos necesarios para que las aplicaciones usen consultas difusas sobre bases de datos.

En [15] se propuso un método formal para implementar requisitos difusos dentro de sistemas de software que soporten consultas difusas. De tal forma que, utiliza el cálculo relacional de tuplas para especificar formalmente las consultas difusas, además, provee reglas que permiten traducir una especificación formal a una consulta escrita en el lenguaje SQLf.

Por otro lado, en [16] se propone un perfil UML para representar requisitos difusos a base de datos relacionales. Este perfil hace uso de OCL (*Object Constraint Language*) extendido con condiciones difusas, a fin de completar los diferentes artefactos de la notación UML con requisitos difusos expresados formalmente, lo cual facilita el proceso de traducción a un lenguaje de consultas como SQLf. En [17] se define un algoritmo de traducción de requisitos difusos en OCL a consultas difusas en SQLf. Este algoritmo es de fácil implementación sobre un SGBD difuso, como SQLfi [12] o PostgreSQL [6].

Los trabajos anteriores [16][17] fueron considerados en la construcción de un *framework* de persistencia [18] que permite el desarrollo de aplicaciones con consultas difusas, el cual es una extensión de Spring, *framework* muy usado actualmente en la comunidad de desarrolladores. Esta extensión puede ser incorporada dentro de ambientes de trabajo, así como, ser integrada con librerías, para el desarrollo de aplicaciones Java.

A pesar que los trabajos aquí mencionados muestran el creciente interés del uso de la lógica difusa como mecanismo para incorporar flexibilidad en diversos sistemas de software sobre bases de datos, ninguno de esos trabajos ha sido enfocado directamente a los formularios de búsqueda presentes en las aplicaciones Web, las cuales tienen mucho impacto en la actualidad. Además, la incorporación de información imprecisa e incierta en el modelo de base de datos sigue siendo un tema importante de investigación debido a que existen datos imprecisos e inciertos en muchas de las aplicaciones del mundo real [19]. Por tal razón, es razonable pensar que incorporar términos difusos en los formularios de búsqueda como funcionalidad agregada dentro de las aplicaciones Web puede resultar de impacto en el presente.

III. MARCO TEÓRICO

En esta sección se describe la teoría de los Conjuntos Difusos que es la base matemática sobre la cual se define la lógica difusa, de manera que permita especificar términos del lenguaje natural. Además, se da un panorama general de los elementos presentes en los formularios de búsqueda en la Web, conocidos como controles Web.

A. Conjuntos Difusos

El lenguaje natural que usamos a diario está lleno de imprecisión. Muy rara vez se habla de temperaturas exactas como “38 grados”, más bien se emplean frases como “está caliente”, “hace frío” o “tiene fiebre alta”. Si se opine sobre la edad de una persona es común decir que es “joven”, “adulta” o

“anciana” en lugar de un valor preciso. Al comentar sobre distancias recorridas se indica si algún lugar está “lejos” o “cerca”. Para enfatizar algo se usan adverbios dentro de frases imprecisas como “es muy joven” o “está extremadamente lejos”. En cuanto a cantidades, es común usar términos como “pocos”, “la mayoría” o “muchos”. Estos términos del lenguaje natural son mayormente adjetivos o adverbios que se caracterizan por ser impresos, vagos o difusos [13].

Una herramienta matemática que permite modelar estos términos es la Teoría de Conjuntos Difusos [3]. Un **conjunto difuso** permite membresía gradual. En los conjuntos clásicos, se dice si un elemento pertenece o no al conjunto. En los conjuntos difusos, la pertenencia de un elemento x viene dada por un grado de membresía $\mu(x)$ cuyo valor está en el intervalo $[0,1]$. Entonces, si el grado es cero el elemento está “completamente excluido”, y si el grado es 1, el elemento está “completamente incluido”. En todos los demás casos la pertenencia es gradual. Al subconjunto de los elementos que están completamente incluidos se les conoce como **núcleo**, a los que no están completamente excluidos (grado de pertenencia mayor que 0), pero tampoco son completamente incluidos se les conoce como **borde**. El **soporte** es la unión del núcleo y el borde.

Para especificar un conjunto difuso, se puede usar una notación trapezoidal de la forma (x_1, x_2, x_3, x_4) , donde $x_1 \leq x_2 \leq x_3 \leq x_4$ señalan los puntos de inflexión del trapecio. También, se puede especificar un conjunto difuso con una notación por extensión $\{\mu(x_1)/x_1, \dots, \mu(x_n)/x_n\}$, donde cada elemento x_i del dominio se acompaña de su grado de membresía $\mu(x_i)$. Estos grados son colocados según la preferencia del usuario.

B. Términos Difusos

Fundamentada en los conjuntos difusos, la lógica difusa permite expresar condiciones con términos difusos, que pueden usarse al realizar consultas o búsquedas sobre un grupo de datos. En esta lógica los valores de verdad son representados por números reales comprendidos en el intervalo $[0,1]$. Así el cero (0) significa completamente falso y el uno (1) significa completamente cierto, y los demás valores del intervalo representan el grado de verdad obtenido para una proposición que se está evaluando.

El operador lógico negación se interpreta como el complemento de conjuntos, la conjunción como la intersección y la disyunción como la unión. El grado de membresía de un elemento x en el complemento del conjunto difuso F se calcula como $1-\mu_F(x)$. El grado de membresía para el conjunto intersección se calcula mediante una norma triangular, que es operador binario, cerrado en $[0,1]$, conmutativo, con elemento neutro uno (1). A cada norma triangular le corresponde una co-norma, que es el concepto dual, la cual se usa para el cálculo de la unión. Usualmente se adopta la norma triangular mínimo (*min*) y su co-norma máximo (*max*).

Las expresiones en lógica difusa se construyen utilizando términos lingüísticos, tales como predicados, modificadores, comparadores, conectores y cuantificadores [20].

Los **predicados** [13] representan adjetivos del lenguaje natural, que corresponden a los componentes atómicos de la lógica difusa, los cuales pueden ser definidos con un conjunto difuso en tres formas diferentes: una función trapezoidal, una

expresión con rango en el intervalo $[0,1]$ y por extensión indicando para cada valor del dominio del predicado su respectivo grado de verdad. Por ejemplo, los adjetivos “joven”, “adulto” y “anciano” para la edad, pueden ser definidos de forma trapezoidal como se observa en la Figura 1. En este caso, el usuario especifica el predicado “joven” con el trapecio $(0, 0, 20, 40)$, “adulto” con el trapecio $(20, 40, 60, 80)$, y “anciano” con el trapecio $(40, 60, 100, 100)$.

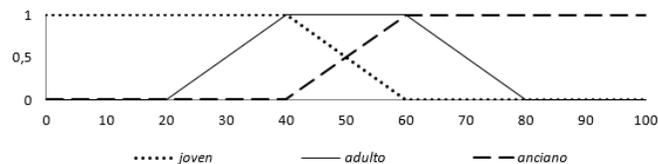


Figura 1: Definición Trapezoidal de los Términos Difusos Joven, Adulto y Anciano

Otro ejemplo, donde se usa una definición por extensión, sería el predicado “agradable”, relacionado al universo de los colores = {amarillo, verde, rojo, azul, negro}, puede representarse con el conjunto difuso $agradable = \{0.4/amarillo, 1/verde, 0.1/rojo, 0.7/azul, 0.3/negro\}$. Un ejemplo, que usa una expresión con rango en el intervalo $[0,1]$ es el predicado “alto” definido con la expresión $x/250$, siendo x un valor en el rango $0 \dots 250$.

Los **modificadores** [13] son adverbios que sirven para expresar una propiedad adicional sobre un predicado, las cuales intensifican, relajan, desplazan o invierten el concepto que representa el predicado. Se pueden definir de tres formas: a través de una potencia a la cual será elevado el valor resultante de un predicado difuso, a través de un valor de traslación que indica lo que le será sumado o restado (según sea el caso) al valor de entrada de un predicado y con una función de norma. Por ejemplo, el modificador “extremadamente” pudiera representarse con la potencia al cubo del valor de la función de membresía, y el modificador “muy” como una traslación negativa.

Los **comparadores** [13] permiten definir una comparación para dos argumentos recibidos. Los términos lingüísticos que los representan son adjetivos de grado comparativo o adverbios. Se pueden definir de dos formas: trapezoidal, si se especifica una expresión con dos variables, cuyo resultado luego de ser evaluada cae en el conjunto difuso definido por un trapecio; o por medio de una relación difusa, donde se especifica para cada par de valores su valor de verdad. Por ejemplo, se podría definir el comparador “cercano”, para los valores en el intervalo $[-30,30]$, con la expresión $x-y$; donde el resultado obtenido se evalúa en la función de membresía representada por el trapecio $(-20, -1, 1, 20)$. Un segundo ejemplo es definir el comparador “similar” para colores, usando la relación difusa $\{(negro,gris)/0.5, (azul,gris)/0.7, (blanco,gris)/0.5, (azul,negro)/0.8, (azul,blanco)/0.2\}$. Aquí se indica que el negro es similar a gris en un grado de 0.5, mientras que el azul es similar a gris en un grado de 0.7.

Los **conectores** [13] son operadores lógicos que puede definir el usuario, para combinar dos condiciones difusas. Por ejemplo, la negación, la conjunción y la disyunción clásicas se pueden extender de forma difusa, preservando su correspondencia con los operadores de conjunto: complemento, intersección y unión. Así, el $x \Rightarrow y$ puede definirse con la expresión $max(1 - x, y)$.

Los **cuantificadores** [13] permiten describir cantidades difusas. Son representados por adverbios superlativos o cuantitativos, o también, con frases imprecisas que expresan cantidad. Pueden definirse de forma absoluta o proporcional dependiendo de los valores facilitados, usándose en ambos casos la forma trapezoidal. Los cuantificadores clásicos, existencial y universal, se mantienen como cuantificadores difusos. Por ejemplo, un cuantificador absoluto podría ser la frase “porlomenos3” representado por la función trapezoidal $(3,8,\infty,\infty)$. Un cuantificador relativo podría ser “muchos” representado por el trapecio $(0.75, 0.90, 1, 1)$.

C. Formularios Web

Para interactuar con un sitio o página Web están las interfaces conocidas como formularios Web. Tienen muchos usos, desde funciones tan simples como casillas de búsqueda, suscripciones en listas de correos, libros de visitas o encuestas, así como sistemas de comercio en línea. Los formularios reúnen información del usuario a través de controles como botones, campos de texto o menús deslizantes, pero no procesan los datos. Los controles Web más usados son (ver Figura 2):

- **Campos de entrada de texto:** sirven para introducir una sola palabra, una línea de texto, una contraseña con caracteres ocultos, direcciones de email, enlaces a recursos o sitios Web, y áreas de texto multilínea, con control deslizante.

- **Casillas de verificación:** interruptores que pueden ser activados y desactivados por el usuario, para preguntas multiselección en el caso de aceptar más de una respuesta.
- **Botón de opción:** interruptor que sólo puede activarse o desactivarse un control al mismo tiempo mientras todos los demás quedan desactivados.
- **Número:** Campos de entrada que contienen un valor numérico, pueden aplicar restricciones.
- **Rango:** para registrar valores que deben estar delimitados en un rango definido.
- **Menús desplegables:** sólo puede seleccionarse un elemento cada vez.
- **Menús deslizantes:** permite que los usuarios seleccionen más de una opción de la lista.
- **Color:** Permite al usuario seleccionar un color de la escala RGB.
- **Tiempo:** Permite indicar valores del tiempo como fecha, mes, año, semana, hora específica del día.

IV. CONTROLES WEB CON TÉRMINOS DIFUSOS

Muchas aplicaciones utilizan controles Web en las interfaces con los usuarios, los cuales no fueron concebidos para usar lógica difusa desde su diseño. Por tal motivo se realizó un estudio de las posibilidades de conversión de un sistema que ya funciona con lógica precisa a uno que funcione con lógica

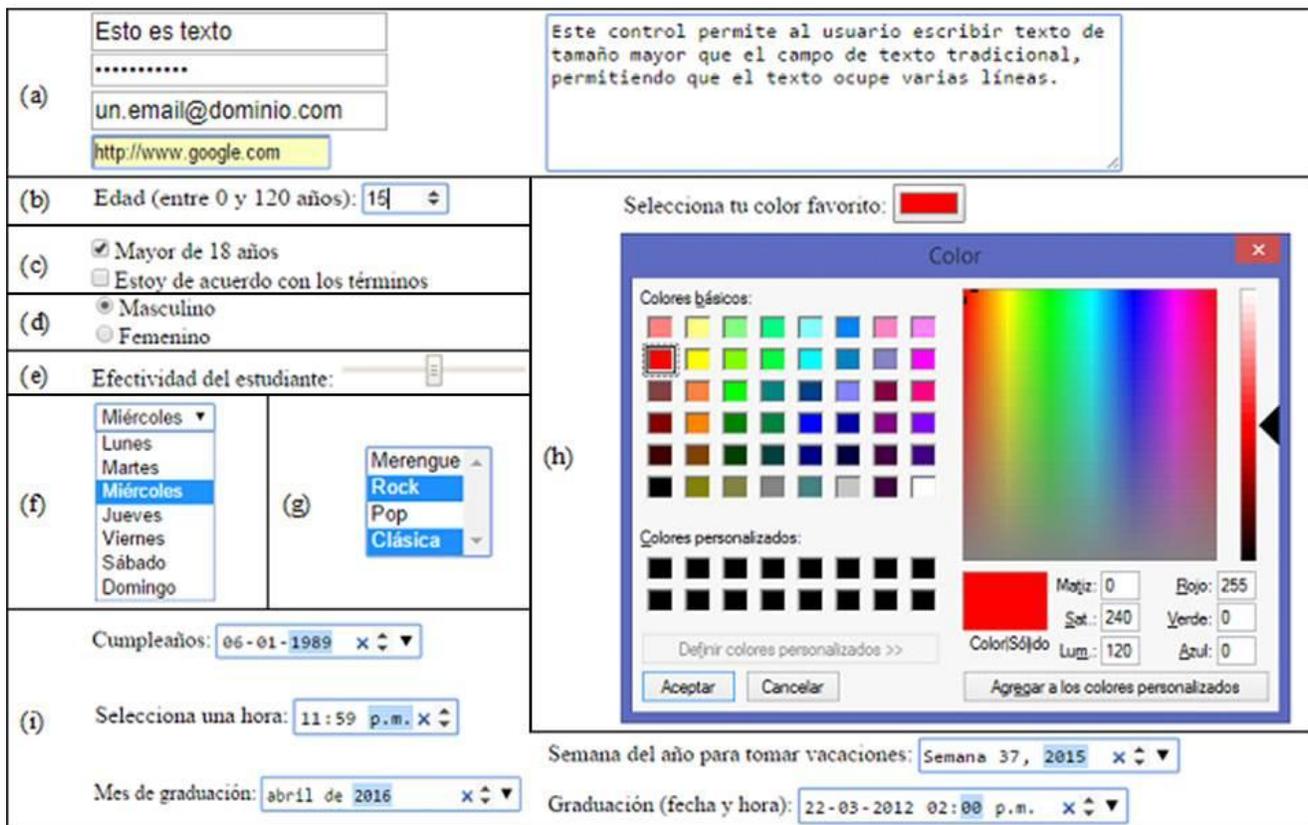


Figura 2: Tipos de Controles Web más Usados: (a) Campos de Entrada de Texto, (b) Número, (c) Casillas de Verificación, (d) Botón de Opción, (e) Rango, (f) Menú Desplegable, (g) Menú Deslizante, (h) Color, (i) Tiempo

difusa. Los detalles de este estudio y sus resultados se describen en [1].

A manera de resumen se mencionarán algunos aspectos. Primero, existen controles Web con entradas precisas que no pueden ser transformados a difusos: campos de entrada de textos con nombres, contraseñas, dirección email o url y texto multilíneas. Además de los cuales no suelen usarse en formularios de búsqueda (i.e, contraseñas o texto multilínea).

Campos de entrada numéricos, casillas de verificación, botones de opción, menús desplegables, menús deslizantes en la mayoría de los casos pueden asociarse a predicados difusos, representables con un menú desplegable o con casillas de verificación para múltiples selecciones. Las casillas de verificación con múltiples opciones, en algunos casos, también pueden transformarse a un cuantificador difuso.

Los controles relacionados con fechas y tiempo también pueden representarse con predicados difusos (como cercano y lejano) para búsquedas difusas. La transformación de números también puede ser realizada con un comparador difuso.

Los controles tipo rango, donde se especifica un valor dentro de un dominio definido, pueden representarse con un comparador difuso especificado sobre el mismo dominio. Para los colores se puede agregar la funcionalidad de búsquedas con colores parecidos definiendo un predicado difuso que permita evaluar lo cercano que es un color de otro.

A todos los controles convertidos, se les puede aumentar la intensidad a través de modificadores difusos “muy” o “extremadamente”, si aporta algo al usuario. Estos se pueden agregar con casillas de verificación o botones de opción. Además, se puede especificar en las búsquedas el grado mínimo de verdad que deben satisfacer todos los resultados presentes en la misma. A esto se le conoce como calibración. Hay algunos controles equivalentes, en esos casos las decisiones se toman por aspectos relacionados con la interfaz.

V. ESTUDIO DE OPINIÓN

En el caso de los comercios Web, el impacto de una función de búsqueda efectiva puede ser muy importante. Los clientes que visitan estos sitios, ya saben lo que necesitan y por lo tanto van directamente a la sección de búsqueda. Esta sección muy probablemente es su primer contacto con el sitio Web, por lo que se debe asegurar que los usuarios no reciban resultados equivocados o mal clasificados, sino que realmente reciban lo que están buscando. De lo contrario, la frustración puede llevarlos a un sitio Web diferente.

En muchos casos esto ocurre cuando se hacen búsquedas precisas. Los formularios al tener controles Web que exigen entradas precisas (el valor exacto de una edad, precio, día, distancia, temperatura, etc.) puede ocasionar que las respuestas de las búsquedas sean incorrectas, incompletas o insatisfactorias. De allí surge la propuesta agregar términos difusos a los formularios Web.

Para analizar el impacto y el deseo que tendrían los programadores y usuarios Web de incluir lógica difusa en los formularios de búsqueda, se hizo una encuesta a 249 personas, de 24 países diferentes siendo el grupo más representativo el de Venezuela. Por ello, se analizaron estas 182 respuestas correspondientes a un 43% de usuarios y un 57% de

programadores Web, a fin de ver si el proyecto era interesante. La información en común recabada fue edad, sexo, nivel educativo y si es programador o usuario Web. A los programadores se les consultó los SGBD que usaban y si estaban dispuestos a usar lógica difusa en sus desarrollos. A los usuarios se les consultó si al realizar búsquedas en formularios Web habían tenido respuestas negativas dándose cuenta luego que lo buscado si estaba. Además, se le consultó si estarían dispuestos a realizar búsquedas usando términos del lenguaje natural.

En los resultados recabados, un 95% de los encuestados estaban dispuestos a usar términos difusos y un 90% de los programadores estaban dispuestos a incluirlos en sus desarrollos. Ante la pregunta de si les ha ocurrido que utilizando formularios de búsqueda no encuentran lo que necesitan y posteriormente se dieron cuenta de que sí estaba, 94% respondió afirmativamente, concluyendo que los formularios presentan problemas para reflejar las preferencias de los usuarios.

En cuanto a los factores edad, sexo y nivel de educación alcanzado, se observó que no marcan una tendencia claramente diferenciada en ninguno de los casos. Por lo cual las descripciones y tendencias facilitadas en el análisis descriptivo de los datos se pueden considerar generalizadas, de forma independiente a estos factores mencionados.

Sobre la pregunta realizada a los programadores Web, referente a los SGBD preferidos para desarrollar, se observó que, aunque hay una tendencia actual de usar NO-SQL, las respuestas permiten concluir que el mercado sigue dominado el por el lenguaje SQL, con un 90% de preferencia.

VI. SELECCIÓN DE LA EXTENSIÓN CON LÓGICA DIFUSA

Algunos esfuerzos se han realizado para dar mayor flexibilidad al lenguaje estándar de bases de datos SQL, incorporando elementos de datos y condiciones de consultas basados en los conjuntos difusos. FSQl y SQLf son las extensiones más completas existentes para la incorporación de conjuntos difusos en SQL. Estas dos propuestas tienen enfoques complementarios. SQLf fue propuesta por Bosc y Pivert [7], se enfoca en la extensión de las expresiones de consulta. FSQl, propuesta por Galindo [8] y otros colaboradores, se centra en la extensión de los datos. Muchos trabajos de investigación y desarrollo se han realizado a partir de estas dos propuestas. SQLf es la extensión que permite expresar la mayor cantidad de términos difusos. Varias implementaciones de estas extensiones de SQL han sido realizadas.

SQLfi [12] es un sistema de consultas difusas que recibe instrucciones en el lenguaje SQLf, las cuales pueden ser: definiciones de términos difusos (predicados, modificadores, comparadores, conectores o cuantificadores), definiciones de objetos de datos (tipos, tablas, vistas, aserciones, procedimientos almacenados o funciones), manipulación de datos (inserción, eliminación, actualización), instrucciones transaccionales (*commit* o *rollback*) o consultas difusas. Si bien SQLfi fue desarrollado sobre Oracle 9i, su diseño se hizo pensando en la portabilidad del sistema, por lo que se implementó en Java, usando el estándar de conexión JDBC, como una capa lógica que se encarga del procesamiento de las nuevas funcionalidades.

PostgreSQLf [21] es una extensión de PostgreSQL para el procesamiento de consultas difusas que usa la estrategia de acoplamiento fuerte en su implementación. El *parser* del SGBD fue extendido para el reconocimiento de términos difusos y condiciones difusas. Asimismo, el catálogo y las estructuras abstractas de representación, de manera que son reconocidos por el optimizador, el ejecutor y las funciones de comandos. El uso de acoplamiento fuerte provee mayor escalabilidad y mejor desempeño. Por otro lado, esta complejidad lo hace no portable y con un desarrollo de gran esfuerzo, lo cual ha dificultado su actualización a las últimas versiones del SGBD y completar la implementación de todos los términos difusos.

FSQL [8] posee un servidor desarrollado en Oracle para una base de datos difusa, siguiendo el modelo teórico GEFRED que permite atributos difusos, los cuales almacenan etiquetas lingüísticas, en las tablas de una base de datos. Además, este servidor provee la facilidad de realizar consultas flexibles sobre atributos precisos o difusos. También, el uso y definición de constantes, operadores, comparadores y cuantificadores difusos.

SQLf_j [22] es un sistema que permite consultas difusas a los SGBD relacionales: MySQL y PostgreSQL. Las consultas se especifican en SQLf. SQLf_j funciona como un traductor, de consultas difusas a consultas SQL estándar, las cuales se ejecutan en el SGBD relacional. SQLf_j fue escrito en Java con el *parser* de SQLf creado con yacc y flex, el cual fue reemplazado por un analizador orientado a objetos programado en SableCC.

La metodología desarrollada debía apoyarse en una de las extensiones existentes de SQL que maneje lógica difusa, se decidió establecer algunas características que permitieran evaluar tales extensiones después de realizar un proceso exhaustivo de investigación. Las características se describen a continuación y la presencia en las extensiones con lógica difusa de estas características se muestra, como un cuadro comparativo, en la Tabla I.

Tabla I: Presencia de Características Deseables en las Extensiones con Lógica Difusa de SGBD Existentes

Característica	SGBD con lógica difusa			
	SQLfi	PostgreSQLf	FSQL	SQLf_j
Complejidad	Sí	Sí	No	Desconocido
Disponibilidad	Sí	Sí	Sí	Sí
Visibilidad	Baja	No visible	Alta	Alta
Calidad	Sí	No	No	No
Soporte	No	No	No	No
Portabilidad	6 SGBD	No	2 SGBD	2 SGBD
Documentación	Instalación y usuarios	Instalación y usuarios	Instalación y usuarios	Instalación
TOTAL	5	3	3	3

- **Complejidad:** se refiere a si la extensión maneja todos los términos difusos definidos por Zadeh [22], así como, posee suficiente capacidad expresiva para que sea provechoso y aumente las posibilidades de realizar una migración exitosa.
- **Disponibilidad:** si la extensión está disponible en la Web para su descarga.

- **Visibilidad:** si la extensión es fácilmente ubicable en la Web para su descarga. Relacionada con la anterior, ya que el producto podría estar disponible, pero con muchas dificultades para ubicarlo.
- **Calidad:** si la extensión cuenta con documentación sobre pruebas de calidad, en términos de rendimiento y correcto funcionamiento de los términos difusos implementados en la misma. Todas las extensiones son desarrollos experimentales, por lo cual es importante saber si tienen un control de calidad que permita su utilización en un ambiente de producción.
- **Soporte:** si existen personas u organizaciones con disposición de responder correos electrónicos aclarando dudas sobre la extensión en cuestión.
- **Portabilidad:** si la extensión puede funcionar con múltiples SGBD, y no está atada a uno específico.
- **Documentación:** si la extensión cuenta con documentación para los desarrolladores de software y a nivel de infraestructura para su instalación.

La Tabla I muestra que la extensión SQL que cumple con la mayor cantidad de criterios deseados es SQLf, por tal razón se selecciona como la extensión a ser utilizada en la metodología.

VII. PROYECTO DE MIGRACIÓN

En esta sección se describe en detalle la propuesta metodológica denominada Proyecto de Migración, cuyo objetivo es transformar un sitio Web con formularios de búsqueda que utilicen lógica clásica para que gestione términos difusos. Se explican los roles a considerar durante el proceso de migración, las diferentes etapas, la documentación y algunos aspectos relacionados a la calidad y viabilidad de la migración. La Figura 3 resume los diferentes componentes del proyecto de migración que se describen a continuación.

A. Roles

En los proyectos de tecnologías de información, particularmente en proyectos de desarrollo o actualización de software, existen diferentes roles a lo largo de la planificación y ejecución de los mismos. En ocasiones, estos roles son cubiertos por una misma persona o grupo de trabajo.

Según Pressman [23], todo proyecto de software requiere de varios participantes que deben organizarse por roles. Tomando como base la propuesta de Pressman [23] se definen los roles para la planificación y ejecución del proyecto de migración de formularios Web.

- **Gerente del proyecto:** encargado de gestionar la planificación del proyecto en términos de tiempo, ejecución, costo y resultados esperados.
- **Especialista en infraestructura TI:** experto en instalaciones de SGBD, configuraciones de sistemas operativos y conocimientos en redes de computadoras.
- **Programador SQL:** persona conocedora de las consultas utilizadas en los formularios de búsqueda a migrar, capacitada en programación en el lenguaje SQL, dispuesta a entrenarse en la extensión SQLfi.
- **Programador web:** personal capacitado en programación, en el lenguaje utilizado en el desarrollo funcional del sitio Web a migrar.

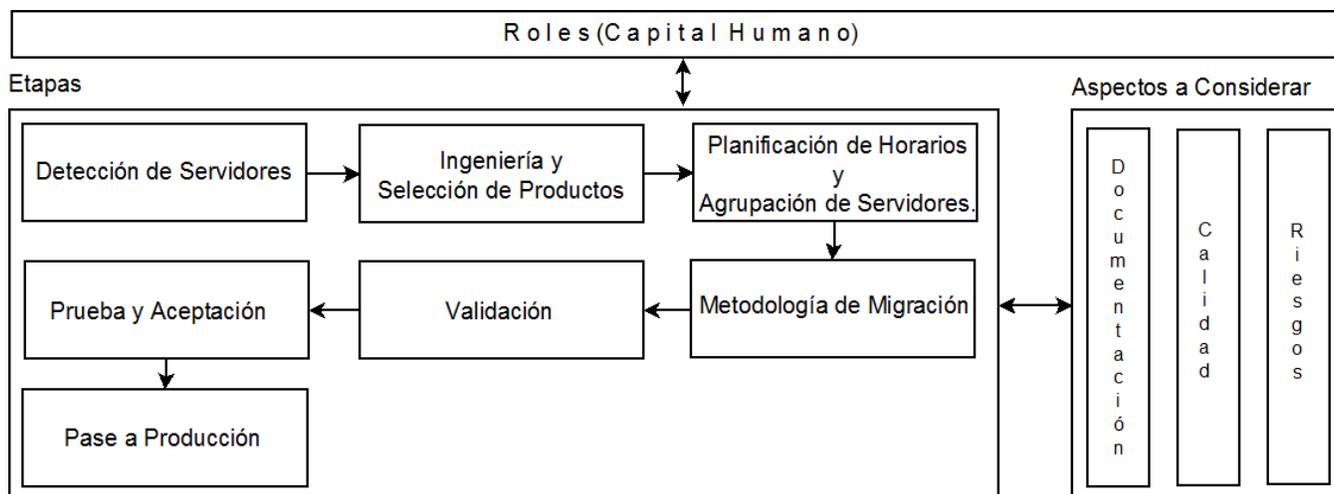


Figura 3: Esquema Resumen del Proyecto de Migración

- **Probador de software/encargado de la calidad:** persona con habilidades para la realización de pruebas y detección de posibles fallas o bugs en los desarrollos de software, específicamente en sitios Web.
- **Diseñador web:** encargado(s) del aspecto visual del sitio Web, preferiblemente quien participó en el diseño inicial.
- **Clientes:** conocedores expertos del modelo de negocio y la funcionalidad del sitio Web. Son los que dan las indicaciones de los cambios a realizar en el sitio Web para incorporarle lógica difusa.
- **Usuarios finales:** grupo de personas que utilizarán el sitio Web una vez migrado, quienes participan en las pruebas de migración. Sólo deben saber navegar en internet y utilizar un formulario de búsqueda. Se sugiere que este grupo sea lo más numeroso posible, y con diversidad en términos de localización, edad, sexo, nivel de conocimiento, entre otros factores.

Adicionalmente, todos los roles deben contar con habilidades de comunicación efectiva y asertiva, así como de trabajo en equipo. A lo largo de todo el proceso será necesario el intercambio de ideas, así como la generación de propuestas.

B. Etapas

En base a la propuesta de Martínez [24], la cual afirma que un proceso de migración de sistemas incluye seis componentes principales, se describen a continuación cada una de etapas con las adaptaciones correspondientes a la propuesta metodológica presentada en este trabajo.

1) *Detección de Servidores:* Corresponde a determinar los requisitos de diseño relacionados a los productos o las soluciones que se utilizarán en el proyecto de migración. Para el caso de migración a formularios Web con lógica difusa, el requisito a considerar es que el servidor de SQLfi utilizado sea compatible con la arquitectura actual del sitio Web.

2) *Ingeniería y Selección de Productos:* Se identifica y establece toda la información básica para determinar los sistemas que se deben migrar. Para el caso de migración a formularios Web con lógica difusa se proponen dos actividades concretas:

- **Entrenamiento:** Los roles profesionales, el gerente de proyecto y los clientes deben comprender el funcionamiento y las capacidades de los términos difusos provistos por SQLfi. En el caso de los clientes, deben recibir capacitación en lo que son términos difusos y lo que son capaces de hacer, para ayudar a proponer los cambios a realizar. El programador SQL y programador Web, requieren el mismo conocimiento que el experto del negocio y adicionalmente requieren entrenamiento en el lenguaje SQLf.
- **Selección de controles a migrar:** Los clientes deben analizar uno a uno los controles existentes en los formularios de búsqueda, considerando todas las propuestas y tomando la decisión de migrar o no de forma individual. En algunos casos puede resultar que existe más de una migración viable para un determinado control. Es en este momento cuando el diseñador Web evalúa ambas propuestas, y opta por la que sea más amigable para los usuarios finales y más acorde con el diseño del sitio Web en cuestión.

Una vez que se han diseñado los controles Web difusos, es necesario organizar o coordinar un proyecto de migración de formularios con términos precisos a formularios con términos difusos.

3) *Planificación de Horarios y Agrupación de Servidores:* Se determina el estado futuro de cada servidor según su configuración actual y se agrupan por grupo de coherencia. Algunos servidores tienen interdependencias que requieren que se los migre de forma conjunta para eliminar la interrupción de servicios esenciales para los usuarios finales. Una vez que se hayan establecido los grupos de coherencia se hacen los cambios por olas de migraciones. Estos grupos sientan las bases para un plan de proyectos real, conformar el cronograma y los hitos de la migración.

4) *Metodología de Migración:* Corresponde a los procesos que conforman la propuesta de este trabajo de investigación. Las actividades propuestas son:

- **Instalación e integración:** El especialista de infraestructura TI instala y deja operativa la extensión SQLfi sobre el SGBD utilizado por el sitio Web a migrar.

Esto se realiza en un ambiente de pruebas lo más similar posible al ambiente de producción. Luego, el programador de la aplicación Web redirecciona todas las consultas de los formularios de búsqueda hacia SQLfi. Posteriormente, el probador de software verifica que los formularios sigan operando de la forma habitual y sin ninguna incidencia.

- **Definición de términos difusos:** El cliente, junto al programador SQL y programador Web definen los nuevos términos difusos usando SQLfi. Se hacen consultas de prueba sobre este SGBD para verificar que todos los términos están funcionando según lo esperado y acorde a su definición.
- **Definición de consultas SQL:** El programador SQL precisa las consultas apropiadas para las nuevas búsquedas a realizar, en todas sus combinaciones, a fin que el sitio Web sea modificado por el programador Web. La meta es que los formularios de búsqueda envíen las consultas para ser ejecutadas por SQLfi.

5) *Validación:* Conjunto de tareas que aseguran que el software que se construye sigue los requerimientos del cliente. En esta etapa se realizan pruebas con el objeto de descubrir y prevenir problemas futuros. También, se establecen planes sobre mitigación de riesgos, que consideren los siguientes aspectos:

- Cuando las consultas no funcionen como se espera, no se hacen suficientes pruebas o el ambiente de producción y de prueba no son completamente equivalentes. Se recomienda estar preparado para regresar a la versión anterior del sitio Web.
- Cuando los usuarios no comprendan el funcionamiento del nuevo sistema de búsqueda, debido a un mal diseño de los términos difusos, o de los nuevos controles Web, o no se hicieron suficientes pruebas con múltiples usuarios externos al proyecto. Se sugiere regresar a la versión anterior del sitio Web.

6) *Prueba y Aceptación:* Se asegura que todos los sistemas funcionen en su totalidad antes de entrar en producción. Para ello, se realizan pruebas de aceptación y un plan de recuperación. En esta etapa el probador de software se encarga de revisar exhaustivamente que el sitio Web no presenta errores en los formularios de búsqueda. También se valida el nivel semántico, es decir, si los resultados son los esperados y que se excluye cualquier otro. Luego de estas verificaciones, se integra a los usuarios finales en el proceso de prueba, así como, en conocer su opinión sobre los cambios realizados. Si éstos no resultan adecuados y agradables, se regresa a la etapa de selección de controles a migrar a fin de tomar nuevas decisiones en base a la experiencia reportada por los usuarios.

7) *Pase a Producción:* Una vez culminadas todas las etapas anteriores se deben realizar los cambios en el ambiente de producción, y validar su funcionamiento como se hizo en el ambiente de pruebas. Algunas recomendaciones a considerar durante este procedimiento:

- Realizar los cambios en un horario donde el sitio Web tenga el menor tráfico posible.

- Informar a los usuarios del cambio a realizar mediante avisos en el sitio Web.
- Durante el proceso debe estar disponible todo el equipo de trabajo del proyecto para canalizar cualquier inconveniente y solventarlo en la forma más rápida posible.

Desde el punto de vista técnico, se sugiere realizar los cambios en el siguiente orden:

- a. Instalar la extensión SQLfi del SGBD usado por el sitio Web en el ambiente de producción.
- b. Probar el sitio Web original sobre la extensión, a fin de verificar que los formularios de búsqueda continúan funcionando. Esto es posible pues SQLfi tiene la capacidad de ejecutar consultas clásicas, por lo cual el sitio Web debería operar con total normalidad. Si hay algún problema, revisar el paso anterior.
- c. Si todo funciona apropiadamente, se procede a migrar los formularios de búsqueda con los nuevos términos difusos, y se indica a los usuarios el cambio realizado.

En la Tabla II, se resumen las etapas del proyecto de migración y los actores que participan en cada una ellas.

C. Documentación

Durante el proyecto de migración se deben elaborar los siguientes informes a fin de documentar todo el proceso realizado:

- **Cambios en infraestructura de TI:** el especialista en infraestructura de TI reporta todos los cambios a nivel de puertos, las rutas utilizadas para nuevas instalaciones, y cualquier otro cambio pertinente que haya aplicado.
- **Cambios en la arquitectura de la aplicación:** se detalla la arquitectura anterior, los cambios a realizar, y la nueva arquitectura del sitio Web producto de la incorporación de consultas sobre la extensión SQLfi.
- **Términos difusos:** se debe reportar todos los términos difusos definidos, su significado, sus valores, su concepción original y motivación. Además, a nivel técnico, se debe especificar los nombres y dominios que les fueron asignados dentro de SQLfi.
- **Pruebas de calidad:** se deben describir en detalle todas las pruebas realizadas durante la etapa seis, Pruebas y Validación, así como los resultados obtenidos.

Se recomienda que cada documento sea elaborado una vez terminada la etapa correspondiente al mismo. Luego que el sitio Web esté en el ambiente de producción, con resultados satisfactorios, se debe revisar toda la documentación nuevamente para verificar que cualquier cambio ocurrido ha sido reportado apropiadamente. Finalmente, se debe realizar cualquier documentación adicional que aporte valor al proyecto.

D. Calidad

Se propone como enfoque para comprobar la calidad del proyecto comisionar a un grupo de personas que se encarguen de revisar el software, su documentación y la supervisión de los procesos utilizados en el desarrollo. Este equipo verifica que se sigan los estándares seleccionados para el proyecto en

Tabla II: Etapas del Proyecto de Migración y los Actores que Participan en cada Una

Rol \ Actividad	Detección de Servidores	Ingeniería y Selección de Productos	Planificación de Horarios	Metodología de Migración	Validación	Pruebas y Aceptación	Pase a Producción
Gerente del Proyecto	✓	✓	✓	✓	✓	✓	✓
Especialista en Infraestructura TI	✓		✓	✓			
Programador SQL	✓	✓		✓			✓
Programador Web	✓	✓		✓			✓
Probador de Software					✓	✓	✓
Diseñador Web		✓				✓	
Clientes		✓		✓		✓	
Usuarios Finales						✓	

los tres aspectos mencionados. Cualquier estándar que no haya sido cumplido debe ser atendido por el gerente del proyecto.

Además, se sugiere realizar las siguientes verificaciones de calidad:

- A nivel de infraestructura de TI, todos los cambios indicados en la documentación coinciden con el ambiente final de ejecución del sitio Web.
- Los términos difusos escogidos tienen una sola interpretación en lenguaje natural.
- Los usuarios finales seleccionados para las pruebas semánticas y de usabilidad representan apropiadamente todo el universo de usuarios del sitio Web.
- Se realizaron todas las pruebas propuestas con rigurosidad, sin omitir ningún caso.
- La documentación de los términos difusos coincide con lo que está operando en el ambiente de producción.

Estas verificaciones deben realizarse durante las etapas del proyecto que lo ameriten, buscando siempre que el resultado sea lo más satisfactorio posible. Por otro lado, no se excluye cualquier validación adicional que se desee realizar.

E. Viabilidad de la Migración

Para determinar si un sitio Web es factible de ser migrado a trabajar con lógica difusa en sus formularios de búsqueda, se debe contar con algunos requisitos:

- Disponer de capital humano suficiente para cubrir todos los roles descritos.
- Llevar a cabo las dos primeras etapas del proyecto de migración a fin de determinar si al menos un control Web es seleccionado para ser migrado.
- El equipo de trabajo está dispuesto a asumir los riesgos asociados al proyecto.

Se deben considerar todos los riesgos posibles, y planificar para cada uno de ellos una estrategia de mitigación y una de contingencia.

VIII. CONCLUSIONES Y TRABAJOS FUTUROS

En casi cualquier ámbito de la vida se pueden conseguir aplicaciones Web, las cuales usualmente están dotadas de formularios que permiten la realización de búsquedas para satisfacer requerimientos de los usuarios. De manera que resulta de gran interés el tratar con este tipo de formularios, considerando su extensión usando términos lingüísticos vagos.

En este artículo se han relacionado algunos de los varios trabajos previos que se han hecho en cuanto a la incorporación de lógica difusa en SQL y la consecuente extensión de los SGBD. Como resultado de esos trabajos se cuenta con SGBD que proveen mayor expresividad para búsquedas basadas en preferencias del usuario, los cuales están disponibles para ser utilizadas libremente, pero no se ha masificado su uso en aplicaciones Web. También se han mostrado los pocos esfuerzos previos en el uso metodológico de la incorporación de SQL extendido difuso. Estos trabajos relacionados no habían tenido en cuenta la extensión de un formulario de búsqueda Web existente para agregarle términos difusos.

Se expuso en este artículo un marco conceptual resumido de la investigación realizada que involucra la teoría de conjuntos difusos, los términos lingüísticos y los controles web. La teoría de conjuntos difusos es una herramienta para el tratamiento matemático y computacional de términos vagos del lenguaje natural, la cual da soporte a una lógica gradual, conocida como lógica difusa. Los términos en esa lógica son palabras del lenguaje natural, las cuales se clasifican en: predicados, modificadores, comparadores, conectores y cuantificadores. Por otro lado, los formularios Web permiten el uso de diferentes tipos de controles: campos de texto, casillas de verificación, botón de opción, números, rangos, ingreso de fechas, menú desplegable, menú deslizante y selector de colores.

En un trabajo previo se ha reportado la incorporación de términos difusos en estos controles, a fin de dar mayor flexibilidad a los formularios Web de búsqueda. De tal forma que se provee de un mecanismo sistemático, que permite extender formularios de búsquedas existentes, que no fueron

concebidos originalmente con lógica difusa, para que se facilite la expresión de preferencias de usuarios al involucrar términos difusos. La mayoría de los controles Web existentes puede incorporar términos difusos, por lo que se dispone de una guía que puede ser usada por cualquier programador Web que quiera flexibilizar las búsquedas en una aplicación Web existente, a fin de lograr mayor satisfacción del usuario.

A los fines de validar la aceptación que tendría el desarrollo y uso de aplicaciones Web incluyan controles Web con términos difusos en sus formularios de búsqueda, se realizó un estudio de opinión. Con base a este estudio, se puede concluir que los programadores Web tienen el deseo de utilizar herramientas de lógica difusa. Por otro lado, la encuesta deja ver que actualmente los usuarios de aplicaciones Web demandan mayor flexibilidad y adaptación a sus requisitos o preferencias. Más aún, en muchas oportunidades los usuarios quedan insatisfechos con los formularios de búsqueda presentes en aplicaciones Web debido a su rigidez. De manera que el estudio muestra tanto que los usuarios están deseando este tipo de términos en sus formularios de búsqueda Web, así como los programadores están dispuestos a incorporarlos en sus desarrollos. El soporte para estos desarrollos serían los SGBD basados en SQL extendido mediante la aplicación de la teoría de conjuntos difusos. La encuesta muestra que, a pesar de la nueva tecnología NO-SQL, los programadores prefieren actualmente el uso de SGBD basados en SQL.

Antes de proponer el proceso de migración, se analizaron las diferentes extensiones del lenguaje SQL que se implementan sobre los diferentes SGBD existentes, las cuales usan conjuntos difusos en los elementos de datos y las condiciones de consulta. De esta forma, se incorpora la lógica difusa en los SGBD como mecanismo para flexibilizar las consultas y expresar preferencias de usuario. Se estudiaron cuatro extensiones distintas: SQLfi, PostgreSQL, FSQL y SQLfj. Además, se propusieron ciertas características que permitían evaluar la más conveniente para el proyecto de migración. Como resultado de esta investigación, se concluyó que SQLfi es la extensión con lógica difusa que posee mayor cantidad de características deseables: completitud, disponibilidad, visibilidad, calidad, soporte, portabilidad y documentación. Por tal motivo, se seleccionó SQLfi para ser utilizada en la propuesta aquí presentada.

El principal aporte de este trabajo es una propuesta metodológica que permite realizar la migración de sitios Web existentes para que sus formularios de búsqueda incorporen términos difusos. Se definen los roles de los participantes en el Proyecto de Migración, estos son: Gerente del proyecto, Especialista en infraestructura TI, Programador SQL, Programador Web, Probador de software o encargado de la calidad, Diseñador Web, Clientes, Usuarios Finales. Además, se describen las etapas por las que debe pasar este proyecto: Detección de servidores, Ingeniería y selección de productos, Planificación de horarios y agrupación de servidores, Metodología de migración, Validación, Prueba y aceptación, Pase a producción. Estas etapas son similares a otros proyectos de migración, aunque requieren aspectos distintivos al objetivo del trabajo aquí presentado. En particular, en la etapa de la Metodología de migración es necesario instalar e integrar la extensión SQLfi al SGBD usado por el sitio Web a migrar, establecer las definiciones de los términos difusos y especificar

las nuevas consultas que gestionarán estos términos. Finalmente, se dan sugerencias para garantizar la calidad y operatividad del sitio Web migrado, así como, recomendaciones para que el proyecto sea viable.

Como trabajos futuros se espera realizar experiencias prácticas de migración de aplicaciones Web existentes usando la propuesta metodológica aquí presentada. Así mismo, se procederá con el análisis estadístico de estas experiencias a fin de observar los beneficios obtenidos de los sitios Web migrados. Sería interesante evaluar el esfuerzo que debe realizar un programador al tratar de incorporar términos difusos durante la migración de un sitio Web. La expectativa de los autores es que este esfuerzo no sea significativo.

Por otro lado, se espera hacer nuevas versiones de la metodología que abarquen aplicaciones de escritorio y aplicaciones móviles.

AGRADECIMIENTOS

Agradecemos Aquél que nos da fe y valor para emprender proyectos hacia lo desconocido: “Por la fe Abraham, siendo llamado, obedeció para salir al lugar que había de recibir como herencia; y salió sin saber a dónde iba” (Hebreos 11:8).

REFERENCIAS

- [1] J. A. Labbad, R. Rodríguez, y L. Tineo, *Formularios Web con Lógica Difusa*, unpublished.
- [2] L. A. Zadeh, *Fuzzy Logic - A Personal Perspective*, Fuzzy Sets and Systems, vol. 281, no. C, pp. 4-20, December 2015.
- [3] L. A. Zadeh, *Fuzzy Sets*. Information Control, vol. 8, no. 3, pp. 338-353, June 1965.
- [4] O. Pivert and P. Bosc, *Fuzzy Preference Queries to Relational Databases*, Imperial College Press, February 2012.
- [5] ISO/IEC. *Information Technology — Database Languages — SQL — Part 2: Foundation (SQL/Foundation)*, ISO/IEC 9075-2:2011 (en), 2011.
- [6] A. Aguilera, L. Borjas, R. Rodríguez, and L. Tineo, *Experiences on Fuzzy DBMS: Implementation and Use*, in proceedings of the XXXIX Latin American Computing Conference (CLEI 2013), Naguayá, Venezuela, October 2013, pp. 478-485.
- [7] P. Bosc and O. Pivert, *SQLf: A Relational Database Language for Fuzzy Querying*, IEEE Transactions on Fuzzy Systems, vol. 3, no. 1, pp. 1-17, February 1995.
- [8] J. Galindo, *FSQL (Fuzzy SQL) A Fuzzy Query Language*. Universidad de Málaga, Málaga, España, 2008, <http://www.lcc.uma.es/~ppgg/FSQL>.
- [9] S. Carrasquel, A. Gyomrey, S. Moreau, R. Rodríguez, B. Stornelli, C. Timaury y L. Tineo, *Extensión de MariaDB para Ordenamiento y Agrupamiento Difuso*, Novática. Revista de la Asociación de Técnicos de Informática, no. 229, pp. 92-97, julio-septiembre 2014.
- [10] E. Lai, *No To SQL? Anti-database Movement Gains Steam*. Computerworld, 2009, http://www.computerworld.com/s/article/9135086/No_to_SQL_Anti_database_movement_gains_steam.html.
- [11] MetalByte, *¿Cuáles son los Lenguajes de Programación más Demandados en la Empresa?*, muylinux, 2013. <http://www.muylinux.com/2013/04/18/cuales-son-los-lenguajes-de-programacion-mas-demandados-en-la-empresa>.
- [12] M. Goncalves y L. Tineo, *SQLfi y Sus Aplicaciones*. Revista Avances en Sistemas e Informática, vol. 5, no. 2, pp. 33-40, Mayo 2008.
- [13] R. Rodríguez y L. Tineo, *Elementos Gramaticales y Características que Determinan Aplicaciones con Requerimientos Difusos*, Revista Tekhné, vol. 12, pp. 50-64, Enero 2009.
- [14] M. Goncalves, R. Rodríguez y L. Tineo, *Incorporando Consultas Difusas en el Desarrollo de Software*, Revista Avances en Sistemas e Informática, vol. 6, no. 2, pp. 87-101, Noviembre 2009.

- [15] M. Goncalves, R. Rodríguez y L. Tineo, *Formal Method to Implement Fuzzy Requirements*, DYNA, Revista de la Facultad de Minas, vol. 173, no. 2, pp. 15-24, Enero 2012.
- [16] R. Rodríguez y M. Goncalves, *Perfil UML para el Modelado Visual de Requisitos Difusos*, Enl@ce: Revista Venezolana de Información, Tecnología y Conocimiento, vol. 6, no. 3, pp. 29-46, septiembre-diciembre 2009.
- [17] R. Rodríguez y M. Goncalves, *Implementación de Requisitos Difusos en Sistemas Orientados a Datos Utilizando el Lenguaje OCL y Lógica Difusa*, Enl@ce: Revista Venezolana de Información, Tecnología y Conocimiento, vol. 8, no. 1, pp. 31-54, Enero-Abril 2011.
- [18] A. Aguilera, M. Goncalves, and R. Rodríguez, *Framework for Fuzzy Application Development*, in proceedings of the XXXVIII Latin American Computing Conference (CLEI 2012), Medellín, Colombia, pp. 478-485, October 2012.
- [19] L. Yan and Z.M. Ma, *Modeling Fuzzy Information in Fuzzy Extended Entity-Relationship Model and Fuzzy Relational Databases*, Journal of Intelligent & Fuzzy Systems, vol. 27, no. 4, pp. 1881-1896, July 2014.
- [20] L. A. Zadeh, *PRUF – A Meaning Representation Language for Natural Languages*. International Journal of Man-Machine Studies, vol. 10, no. 4, pp. 395-460, January 1978.
- [21] A. Aguilera, J. Cadenas, and L. Tineo, *Fuzzy Querying Capability at Core of a RDBMS*, Advanced Database Query Systems: Techniques, Applications and Technologies, IGI Global. New York, USA, pp. 160-184, March 2011.
- [22] P. Kalinowski, *SQLF_J*, Poznan University of Technology, Polonia. 2006. http://calypso.cs.put.poznan.pl/~sqlf_j/en/index.php?
- [23] R. Pressman, *Ingeniería del Software: Un Enfoque Práctico*. México D.F., McGraw Hill, 2010.
- [24] H. Martínez, *Ocho Pasos para el Éxito en Migraciones*. The GMB Journal. 2013. <https://es.scribd.com/document/246408796/Pasos-Para-El-Exito-en-Migraciones-GBM-Journal>.