

Advances in Intelligent Information Technologies

Research in Computing Science

Series Editorial Board

Editors-in-Chief:

Grigori Sidorov (Mexico)
Gerhard Ritter (USA)
Jean Serra (France)
Ulises Cortés (Spain)

Associate Editors:

Jesús Angulo (France)
Jihad El-Sana (Israel)
Jesús Figueroa (Mexico)
Alexander Gelbukh (Russia)
Ioannis Kakadiaris (USA)
Serguei Levachkine (Russia)
Petros Maragos (Greece)
Julian Padget (UK)
Mateo Valero (Spain)

Editorial Coordination:

María Fernanda Rios Zacarias

Research in Computing Science es una publicación trimestral, de circulación internacional, editada por el Centro de Investigación en Computación del IPN, para dar a conocer los avances de investigación científica y desarrollo tecnológico de la comunidad científica internacional. **Volumen 79**, octubre 2014. Tiraje: 500 ejemplares. *Certificado de Reserva de Derechos al Uso Exclusivo del Título* No. : 04-2005-121611550100-102, expedido por el Instituto Nacional de Derecho de Autor. *Certificado de Licitud de Título* No. 12897, *Certificado de licitud de Contenido* No. 10470, expedidos por la Comisión Calificadora de Publicaciones y Revistas Ilustradas. El contenido de los artículos es responsabilidad exclusiva de sus respectivos autores. Queda prohibida la reproducción total o parcial, por cualquier medio, sin el permiso expreso del editor, excepto para uso personal o de estudio haciendo cita explícita en la primera página de cada documento. Impreso en la Ciudad de México, en los Talleres Gráficos del IPN – Dirección de Publicaciones, Tres Guerras 27, Centro Histórico, México, D.F. Distribuida por el Centro de Investigación en Computación, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, México, D.F. Tel. 57 29 60 00, ext. 56571.

Editor responsable: *Grigori Sidorov, RFC SIGR651028L69*

Research in Computing Science is published by the Center for Computing Research of IPN. **Volume 79**, October 2014. Printing 500. The authors are responsible for the contents of their articles. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior permission of Centre for Computing Research. Printed in Mexico City, in the IPN Graphic Workshop – Publication Office.

Volume 79

Advances in Intelligent Information Technologies

Edmundo Bonilla Huerta
Federico Ramírez Cruz
Eduardo Sanchez Lucero (eds.)



Instituto Politécnico Nacional
"La Técnica al Servicio de la Patria"



Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2014

ISSN: 1870-4069

Copyright © Instituto Politécnico Nacional 2014

Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal
Unidad Profesional “Adolfo López Mateos”, Zacatenco
07738, México D.F., México

<http://www.rcs.cic.ipn.mx>

<http://www.ipn.mx>

<http://www.cic.ipn.mx>

The editors and the publisher of this journal have made their best effort in preparing this special issue, but make no warranty of any kind, expressed or implied, with regard to the information contained in this volume.

All rights reserved. No part of this publication may be reproduced, stored on a retrieval system or transmitted, in any form or by any means, including electronic, mechanical, photocopying, recording, or otherwise, without prior permission of the Instituto Politécnico Nacional, except for personal or classroom use provided that copies bear the full citation notice provided on the first page of each paper.

Indexed in LATINDEX and Periodica / Indexada en LATINDEX y Periódica

Printing: 500 / Tiraje: 500

Printed in Mexico / Impreso en México

Preface

This volume of the journal “Research in Computing Science” contains selected papers on intelligent and information technologies, which is an interdisciplinary area where several fields converge, such as machine learning, data mining, human computer interaction, computational intelligence, artificial vision, virtual learning environments, data bases, software engineering, Web based systems, mobil computing, systems distributed and parallel, natural language and text processing. The combination and cooperation between these fields has been the key of advance of Information Technologies.

The papers in this volume have been carefully chosen by the Editorial Board based on evaluation by at least two members of the reviewing committee. The main criteria for the selection were originality and technical quality of the papers.

Submission, reviewing, and selection process was supported free of charge by the EasyChair system, www.EasyChair.org.

This volume contains 13 papers related to various aspects of the development and applications of intelligent and information technologies, organized in the following sections:

- Computational intelligence
- Software engineering
- Web Based Systems
- Systems distributed and parallel

This volume is the result of hard work and collaboration of many people. First of all, we thank the authors of the papers included in this volume for their technical excellence, which made possible the high quality of this volume. We also thank the members of the Editorial Board of the volume and the reviewing committee for their hard work on selection of the best papers out of the fifty five submissions we received.

Perfecto Malaquias Quintero Flores
Crispin Hernández Hernández
Edmundo Bonilla Huerta
Federico Ramirez Cruz
October 2014

Table of Contents

Page

Un Modelo Híbrido de Inteligencia Computacional para Resolver el Problema de Job Shop Scheduling.....	9
<i>Jacob Meneses Angel, Marcela Rivera Martínez, Luis René Marcial Castillo, Lourdes Sandoval Solis</i>	
Técnica para la recomendación de productos turísticos en Cuba usando GIS.....	21
<i>Jesse Daniel Cano Otero, Marlon Jorge Remedios Gonzalez, Maria Teresa Morell Franco</i>	
Programación generativa en Java y herramientas de meta-programación	37
<i>Ismael Beristain, Ulises Juárez</i>	
SISFIUX: Adaptación de Feature-driven Development para el desarrollo de un sistema financiero para una universidad.....	49
<i>César Ricardo Alducin-Ruiz, Jorge Octavio Ocharán-Hernández, Lizbeth A. Hernández-González</i>	
Desarrollo de la versión móvil para la red social FEIBook	63
<i>Luis Abraham Zavaleta Ibarra, Gerardo Contreras Vega, Karen Cortés Verdín, Juan Carlos Pérez Arriaga</i>	
Desarrollo de una arquitectura orientada a servicios para un prototipo de una línea de productos de Software	75
<i>Ramón Gómez-Romero, Karen Cortés, Juan Carlos Pérez Arriaga, Ángeles Arenas Valdés</i>	
Software para evaluación de construcciones sustentables: caso práctico	87
<i>Eunice Villicaña Ortiz, Ubaldo R. Montes Juárez, Olga E. Serena Ibáñez, Ana G. Pérez Aguilar, Gabriela Martínez Rosiles, María R. Zarate Nava, Jesús L. López Hernández</i>	
Metodología ágil Scrumban en el proceso de desarrollo y mantenimiento de software de la norma MoProSoft	97
<i>David Ibarra Guzmán, Ulises Castañeda Islas, Carlos Pérez Corona, Blanca Estela Pedroza Méndez</i>	
Especificación de requerimientos con Áncora y el estándar 830.....	109
<i>Yesenia Nohemí Gonzales Meneses, Nayely Yaline León Padilla, José Juan Hernández Mora, María Guadalupe Medina Barrera</i>	

Análisis del estado actual de certificaciones CMMI-DEV ver. 1.3 año 2013 y 2014, a nivel mundial y en México	121
<i>Yesenia Nohemí Gonzales Meneses, Nayely Yaline León Padilla, José Juan Hernández Mora, María Guadalupe Medina Barrera</i>	
Propuesta de sistema Web personalizable para el control de reservaciones en Hoteles	135
<i>Betzabet García Mendoza, Carlos R. Jaimez-González</i>	
Análisis y evaluación de marcos de trabajo para el desarrollo de aplicaciones y servicios en la nube PAAS.....	147
<i>Rafael Antonio Fernández Domínguez, Giner Alor Hernández, Leticia Dávila Nicano</i>	
SBDA-GRID: Implementación de un servicio de base de datos en una arquitectura Grid	159
<i>Nancy Ocotitla, Axel E. Moreno, Rogelio Castro</i>	

Un modelo híbrido de inteligencia computacional para resolver el problema de Job Shop Scheduling

Jacob Meneses Angel, Marcela Rivera Martínez, Luis René Marcial Castillo, Lourdes Sandoval Solís

Benemérita Universidad Autónoma de Puebla, Puebla, Pue., México
{snake_n313, mrvimar}@hotmail.com
{lmc, sandoval}@cs.buap.mx

Resumen. En este trabajo se presenta un modelo híbrido de inteligencia computacional para resolver el problema de job shop scheduling, problema de secuenciación, clasificado como NP Completo. Se propone resolverlo usando la técnica de colonia de hormigas asistida con recocido simulado. Como estrategia de búsqueda global, se usó la técnica de colonia de hormigas y como estrategia de búsqueda local, la de recocido simulado. Dicha propuesta se validó experimentalmente con problemas prueba reportados en la literatura.

Palabras clave: Inteligencia Computacional, Job Shop Scheduling, Colonia de Hormigas, Recocido Simulado, Secuenciación, Makespan.

1 Introducción

El problema de *job shop scheduling* (JSS), es un problema de optimización combinatoria muy conocido en el área de investigación de operaciones e inteligencia artificial. En la manufactura de un producto se requiere de una serie de procesos, cada uno de éstos se efectúa en una máquina determinada. Esto origina diferentes problemas entre los que se encuentra secuenciar n trabajos en m máquinas. La secuenciación tiene como objetivo encontrar el orden de ejecución de n trabajos, que requieren una serie de procesos en m máquinas, de manera que se optimice el tiempo [14]. Acorde a la teoría de complejidad computacional, es un problema NP Completo, por lo tanto no puede resolverse usando métodos exactos [7]. En este trabajo se propone el uso de la metaheurística colonia de hormigas (Ant Colony Optimization, ACO) asistido con recocido simulado (Simulated Annealing, SA).

En los últimos años, de acuerdo con la literatura publicada, se han propuesto varios algoritmos híbridos para solucionar este problema en [9, 11, 15], pero no hay antecedentes de propuestas híbridas de colonia de hormigas con recocido simulado.

En la sección 2, se presentan los conceptos básicos del problema de *job shop scheduling*, la sección 3 describe el algoritmo de recocido simulado; el algoritmo de colonia de hormigas se describe en la sección 4, en la sección 5 se presenta el algoritmo propuesto en el presente trabajo, en la sección 6 se muestran las pruebas del algoritmo propuesto a problemas reportados en la literatura, en la sección 7 se mencionan las conclusiones del trabajo.

2 Conceptos básicos

En el esquema de *job shop scheduling* cada trabajo tiene su propia ruta predeterminada a seguir. Las prioridades dependen de cada trabajo debido a diferentes factores como: mínima manipulación del material, fácil transporte, no existe regreso, manufacturas modernas. En el *job shop scheduling* las operaciones consecutivas de un trabajo nunca se llevan a cabo en la misma máquina, cuando esto ocurre, las dos operaciones se consideran como una sola cuyo tiempo de procesamiento es la suma de los tiempos de procesamiento de las operaciones individuales. Cada trabajo consta de un conjunto de operaciones con una secuencia específica, las cuales requieren de una máquina para su procesamiento y cada máquina puede realizar sólo una tarea a la vez.

Formalmente, un problema de *job shop scheduling* se define como un conjunto finito J de n trabajos $J = \{J_1, \dots, J_n\}$ que tienen que ser programados en un conjunto finito M de m máquinas, $M = \{M_1, \dots, M_m\}$. Cada trabajo J_i está compuesto de una serie de k operaciones O_{ik} , donde el subíndice k indica la máquina M_k en que las operaciones se deben procesar. El orden de las máquinas (secuenciación) para un trabajo J_i está predefinido; a cada operación O_{ik} se le asigna un tiempo de procesamiento entero no negativo P_{ik} . El objetivo es determinar la programación de una secuencia de operaciones en cada máquina M_k tal que las restricciones de precedencia y de capacidad del problema sean satisfechas [6].

Las permutaciones de operaciones representan una solución al problema, aunque no son prácticas con respecto al post procesado. Además de la posición dentro de la secuencia, se está interesado principalmente en que las operaciones de un mismo trabajo no se lleven a cabo al mismo tiempo en una misma máquina. Por esta razón, un candidato solución se describe preferentemente como un conjunto S de tiempos iniciales de operación S_{ik} , $S = \{S_{ik} \mid 1 \leq i \leq n, 1 \leq k \leq m\}$.

Por otra parte, la inteligencia computacional se ocupa de la teoría, diseño, desarrollo y aplicaciones de paradigmas computacionales motivados lingüística y biológicamente. La inteligencia computacional aplicada es un sistema de métodos e infraestructuras que mejora la inteligencia humana aprendiendo y descubriendo nuevos patrones, relaciones y estructuras complejas en ambientes dinámicos para resolver problemas prácticos [13].

3 Recocido simulado

En la metalurgia, el método del *recocido* se utiliza para obtener materiales más resistentes o más cristalinos, en general, para determinar las cualidades de un material. El proceso consiste en calentar el material a temperatura muy alta, en esa situación los átomos adquieren una distribución *azarosa* dentro de la estructura del material y la energía del sistema es máxima.

Después se hace descender la temperatura muy lentamente por etapas, de acuerdo con una calendarización, dejando que en cada una de esas etapas los átomos queden en equilibrio, es decir, que los átomos alcancen una configuración óptima para esa temperatura. Al final del proceso, los átomos forman una estructura cristalina altamente regular, el material alcanza las cualidades buscadas y la energía del sistema es mínima.

Experimentalmente se comprueba que si se hace descender la temperatura bruscamente, o si no se espera suficiente tiempo en cada etapa, al final, la estructura del material no es la óptima. El algoritmo de recocido simulado se desarrolló para simular el proceso de recocido con el fin de encontrar un mínimo global de la función objetivo. En el algoritmo de recocido simulado, la función objetivo se trata como la función de energía de un metal fundido y una calendarización de temperaturas artificiales se establece para enfriar gradualmente el material, análogo a la técnica de recocido. Esta temperatura artificial o conjunto de temperaturas actúa como una fuente de aleatoriedad, que es conveniente para evitar eventualmente un mínimo local [10].

El algoritmo requiere de una configuración inicial dentro del espacio de soluciones o conjunto de configuraciones R , una función de costo $C: R \rightarrow R$, una estructura de vecindad $\hat{V}: R \rightarrow R^2$, y $\hat{V}(r)$ como un subconjunto de \hat{V} , vecinos de la configuración $r, \forall r \in R$.

El algoritmo se define de la siguiente manera: Dada una configuración inicial X , y un número de iteraciones, en cada iteración se selecciona $Y \in \hat{V}(X)$ y se evalúa la expresión $\delta = C(Y) - C(X)$. Se realiza una transición de estado si y solo si $\delta < 0$ o usando una probabilidad de aceptación definida como $\exp^{-\delta/T}$, donde T es la temperatura, la cual sufre un decremento durante la ejecución del algoritmo. Un algoritmo de recocido simulado es mostrado en la figura 1.

En la figura 1, X es la configuración inicial y solución local durante la ejecución, la asignación $X \leftarrow Y$ representa la aceptación de una nueva configuración Y . En las primeras etapas del algoritmo, si la temperatura es lo suficientemente grande, la expresión $\exp^{-\delta/T}$ tiende a 1 haciendo que siempre se acepten nuevas configuraciones por la condicional $u < \exp^{-\delta/T}$ cuando $C(Y) > C(X)$. A medida que se disminuye la temperatura, la probabilidad de que se acepten nuevas configuraciones tiende a 0, haciendo que la configuración X tienda a un mínimo.

```
Algoritmo 1. Recocido Simulado
Sea  $X \in R$  la configuración inicial
Sea  $T > 0$  la temperatura inicial
Sea  $N$  un número máximo de iteraciones.
Repetir
   $n \leftarrow 0$ 
  Repetir
    Generar un vecino  $Y \in V(X)$ 
     $\delta \leftarrow C(Y) - C(X)$ 
     $X \leftarrow Y \leftrightarrow \delta < 0 \quad u < \exp^{-\delta/T}; u \in (0,1)$  aleatorio
     $n \leftarrow n + 1$ 
  Mientras  $n \leq N$ 
Disminuir  $T$ 
Mientras la temperatura  $T > 0$ 
```

Fig. 1. Algoritmo de recocido simulado.

En la figura 1, X es la configuración inicial y solución local durante la ejecución, la asignación $X \leftarrow Y$ representa la aceptación de una nueva configuración Y . En las primeras etapas del algoritmo, si la temperatura es lo suficientemente grande, la expresión $\exp^{-\delta/T}$ tiende a 1 haciendo que siempre se acepten nuevas configuraciones por la condicional $u < \exp^{-\delta/T}$ cuando $C(Y) > C(X)$.

A medida que se disminuye la temperatura, la probabilidad de que se acepten nuevas configuraciones tiende a 0, haciendo que la configuración X tienda a un mínimo.

4 Colonia de hormigas

Los algoritmos de colonia de hormigas son metaheurísticas propuestas por Dorigo et al, para la resolución de problemas de optimización combinatoria tipo NP Completo [4]. ACO tiene su fuente inspiradora en el comportamiento de las hormigas en el rastro de feromona, mecanismo que utilizan para la comunicación entre ellas.

ACO basa su funcionamiento en una colonia de agentes simples, hormigas artificiales, que similar al proceso biológico, se comunicarán indirectamente mediante el depósito, evaporación y seguimiento de feromona artificial.

Los rastros de feromona en ACO sirven de información distribuida entre las hormigas para construir, de forma probabilística, soluciones al problema que se va a resolver [5]. Dorigo propuso el esquema básico para la metaheurística de ACO [3]:

```

Algoritmo 2.Optimización por Colonia de Hormigas (ACO)
Paso1: Inicialización de parámetros y rastros de feromona
Mientras las condiciones de paro no se cumplan hacer
    Paso 2: Las Hormigas construyen soluciones.
    Paso 3: Aplicar búsqueda local (opcional).
    Paso 4: Actualizar feromonas.
Fin Mientras
    
```

Fig. 2. Algoritmo de Optimización por Colonia de Hormigas.

Para poder aplicar el algoritmo ACO, el problema de optimización debe ser traducido a un grafo $G = (V, L)$. V y L representan los vértices y las aristas del grafo respectivamente, y son definidos con las siguientes características y notaciones:

- Un conjunto finito de vértices del problema $V = \{V_1, V_2, \dots, V_n\}$.
- Un conjunto finito E de conexiones entre un subconjunto de elementos de V , tal que $|E| \leq N^2$. (N representa el número de vértices, $N = |V|$).
- Para cada $e_i \in E$ existe un costo de conexión $d_{i,j}$ que representa el costo de transitar del vértice i al vértice j , $\eta_{i,j} = 1/d_{i,j}$ es llamada la distancia heurística.
- Un número finito de restricciones Ω definidos sobre los elementos V y E .
- Dado un conjunto S de todas las posibles secuencias $\langle V_i, V_j, \dots, V_k, \dots \rangle$ sobre los elementos de V , se requiere un subconjunto $\hat{S} \subseteq S$, que represente las secuencias factibles con respecto a Ω .
- Una solución Ψ es subconjunto de \hat{S} , $\Psi \subseteq \hat{S}$.
- Una función de costo $\Phi_\Psi(E, t)$ que represente el costo total de la solución, donde t representa el tiempo.

El algoritmo comienza por asignar un valor real positivo a cada $e_i \in E$ que va a representar el rastro de feromona y se evaporará cada cierto tiempo. Posteriormente, se crearán h hormigas artificiales junto con una lista asociada a ellas denotada como $Tabu_h$. Iniciarán desde una posición inicial arbitraria y la posición inicial será añadida a su lista $Tabu$.

En el paso 2 de la figura 2, las hormigas empiezan a construir soluciones visitando cada uno de los vértices V_i del grafo solo una vez, respetando las restricciones Ω del problema. Para toda hormiga h , se define la probabilidad de transición como la probabilidad de que la hormiga transite de un vértice V_i a un V_j en un tiempo t mostrada en:

$$P_{ij}^h = \begin{cases} \frac{[\tau_{i,j}(t)]^\alpha [\eta_{i,j}]^\beta}{\sum_{x \in Permittedos_h} [\tau_{i,x}(t)]^\alpha [\eta_{i,x}]^\beta} & \text{si } (i, j) \in Tabu_k \\ 0 & \text{en cualquier otro caso} \end{cases} \quad (1)$$

donde $\tau_{i,j}(t)$ es el rastro de feromona en la arista del grafo (i, j) en el instante de tiempo t . $Permittedos_h$ es la lista de aristas por las que la hormiga h puede transitar manteniendo la

factibilidad con respecto de Ω ; α y β son parámetros de control que determinan la importancia de rastro con respecto a la distancia heurística respectivamente.

Una vez que todas las hormigas hayan completado una secuencia Ψ , en el paso 3 de la figura 2 se actualizará el rastro de feromona en el grafo usando la información recolectada en la lista *Tabu* de las hormigas usando la fórmula:

$$\tau_{i,j}(t+n) = (1-\rho) \cdot \tau_{i,j} + \Delta\tau_{i,j} \quad (2)$$

donde $\rho \in (0,1)$ representa un coeficiente real y $1-\rho$ representa el coeficiente de evaporación de la feromona en la arista (i,j) y $\Delta\tau_{i,j}$ representa la cantidad total de feromona depositada por las h hormigas y se calcula con la fórmula:

$$\Delta\tau_{i,j} = \sum_{k=1}^h \Delta\tau_{i,j}^k \quad (3)$$

la cantidad de feromona depositada en la arista (i,j) por una hormiga h es calculada por la fórmula:

$$\Delta\tau_{i,j}^h = \begin{cases} \frac{Q}{L_h} & \text{si la } h\text{-ésima hormiga pasa por la arista } (i,j) \\ 0 & \text{en cualquier otro caso} \end{cases} \quad (4)$$

donde Q es un valor real positivo y $L_h = \Phi_\Psi$ el costo de la solución Ψ o longitud de la ruta encontrada por la hormiga h . Posteriormente se vacían las listas *Tabu* y el algoritmo comienza de nuevo, hasta completar el número máximo de iteraciones.

5 Modelo híbrido

El objetivo de la solución propuesta es resolver el problema de *job shop scheduling*, para ello se debe encontrar una secuencia factible de operaciones con el menor tiempo total de proceso posible, conocido como *makespan*. Para el problema del *job shop scheduling* se tienen las siguientes restricciones:

- Cada trabajo se procesa por una máquina solamente una vez.
- Cada operación de los trabajos deben ser procesados en el orden en que son dados, propiedad conocida como restricciones tecnológicas.
- Cada máquina solo puede procesar un trabajo a la vez.
- Cada operación debe ser procesada sin interrupción.
- Las operaciones de un mismo trabajo no pueden ser procesadas de manera concurrente.

Un problema de *job shop scheduling* se denota formalmente de la forma $n/m/G/C_{max}$ donde n es el número de trabajos, m el número de máquinas, G las restricciones tecnológicas del problema y C_{max} la función objetivo que debe minimizarse, en este caso el *makespan*.

Las restricciones tecnológicas G son representadas por una matriz. Por ejemplo para un problema de 2 trabajos y 3 máquinas, se tendría la matriz:

$$G = \begin{pmatrix} O_{1,3} & O_{1,2} & O_{1,1} \\ O_{2,2} & O_{2,1} & O_{2,3} \end{pmatrix} \quad (5)$$

para la matriz G , la i -ésima fila representa el i -ésimo trabajo y los elementos en la fila representan las operaciones. Por ejemplo la fila 1 corresponde al primer trabajo que se procesa de acuerdo con la siguiente secuencia: primero la máquina 3 después la máquina 2 y finalmente la máquina 1. Similarmente la secuencia del trabajo 2 establecido por la fila 2, se interpreta como máquina 2, máquina 1 y máquina 3. Además, se tiene una matriz P mostrada en (6) que representa los tiempos de procesamiento de las operaciones de los trabajos.

$$P = \begin{pmatrix} t_{1,1} & t_{1,2} & t_{1,3} \\ t_{2,1} & t_{2,2} & t_{2,3} \end{pmatrix} \quad (6)$$

para cada elemento $t_{i,j} \in P$, i corresponde el número de trabajo y j al número de máquina.

Para aplicar el algoritmo propuesto primero se requiere una representación adecuada del problema. La figura 3 muestra un ejemplo de la representación del grafo que será explorado por la colonia de hormigas para un problema de 2 trabajos y 3 máquinas.

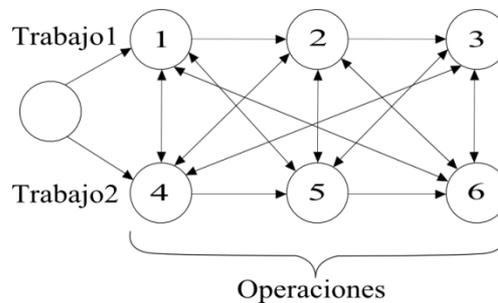


Fig. 3. Representación del problema requerido para aplicar ACO.

Para construir el grafo, cada nodo corresponderá a un elemento en la matriz G de la fórmula (5). Para 2 nodos i, j existirá un arco dirigido (i, j) en el grafo si y solo si i precede a j respecto a G , o si i y j no pertenecen al mismo trabajo.

La solución al problema de *job shop scheduling* propuesta está pensada para utilizar la capacidad de ACO de generar buenas soluciones y aprovechar la propiedad de convergencia de recocido simulado [12], para obtener el mejor local del subconjunto del espacio de soluciones (paso 3 de la figura 2) marcado por el rastro de feromona.

La figura 4 muestra la propuesta para el problema de *job shop scheduling*, usando colonia de hormigas asistida con recocido simulado.

En el paso 2 de la figura 4, cada hormiga tiene un nodo inicial asociado a su lista *Tabu*, y en el paso 3 cada hormiga *h* comienza a recorrer el grafo, transitando de un nodo a otro utilizando la probabilidad de transición de la ecuación (1).

Algoritmo 3. Modelo híbrido ACO-SA

```
Paso 1: Inicializar la matriz de feromonas, representación del grafo,
representación de las hormigas e inicialización de la posición inicial
de estas.
Para e ← 1 hasta NIteraciones hacer
    Paso 2: Agregar el nodo inicial a la lista de cada hormiga.
    Paso 3: Cada hormiga construye una solución factible.
    Paso 4: Calcular el mejor makespan encontrado por las hormigas,
    junto con su secuencia.
    Paso 5: Actualizar el makespan global.
    Paso 6: Determinar qué secuencia se usará como configuración inicial
    por recocido simulado.
    Paso 7: Aplicar recocido simulado a la configuración inicial elegida.
    Paso 8: Actualizar el makespan global por segunda vez.
    Paso 9: Actualizar la matriz de feromonas y vaciar las listas de
    cada hormiga.
Fin_para
```

Fig. 4. Modelo híbrido propuesto ACO-SA.

Para la representación de las soluciones y la configuración inicial de recocido simulado, se utiliza una secuencia factible de nodos recorridos por las hormigas. La figura 5 muestra dos ejemplos de secuencias sobre el grafo de la figura 3. La figura 5a muestra el caso cuando una secuencia de nodos es factible, y la figura 5b muestra el caso cuando una secuencia no es factible, es decir, no se respeta el conjunto de restricciones Ω del problema de *job shop scheduling*.

Para la figura 5b, la secuencia es correcta porque sí existe una arista del nodo 2 al nodo 5 pero no es factible ya que el nodo 5 solo puede ser visitado una vez que se visita el nodo 4. Esta situación es de importancia mientras las hormigas construyen las soluciones (paso 3 de la figura 4).

En el paso 6 de la figura 4, el criterio para elegir la configuración inicial para recocido simulado es el siguiente: Si las hormigas encontraron una nueva mejor solución, entonces esa secuencia es utilizada como configuración inicial, en caso contrario se elige al azar uno de los siguientes criterios:

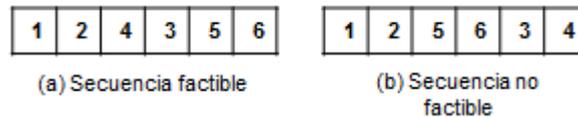


Fig. 5. Ejemplos de secuencias sobre el grafo.

- La mejor secuencia global con probabilidad menor a 0.15.
- La mejor secuencia encontrada por las hormigas en el paso 3 con probabilidad entre 0.15 y 0.50.
- Cualquier otra secuencia encontrada por las hormigas en el paso 3 con probabilidad mayor a 0.50.

El algoritmo de recocido simulado aplicado en el paso 7 de la figura 4, es el mismo descrito en la figura 1. En la solución propuesta, para una secuencia de nodos X , se obtiene un vecino $Y = \hat{V}(X)$ eligiendo una posición en la secuencia e intercambiándola con la siguiente siempre y cuando se mantenga la factibilidad de la solución. El método utilizado para disminuir la temperatura es multiplicando la temperatura T por un valor $\alpha_r \in (0,1)$.

Después de aplicar recocido simulado en el paso 6 de la figura 4, se vuelve a actualizar en el paso 7, la mejor solución encontrada respecto a la solución encontrada por recocido simulado y finalmente en el paso 8, se actualiza la información de la matriz de feromonas igual al paso 4 de la figura 2.

6 Experimentos y resultados

El algoritmo propuesto se validó utilizando problemas prueba reportados en el repositorio de la librería Operation Research [2], en una máquina cuyas características son: procesador Intel Core i3 2.27Ghz, 3GB de memoria RAM, lenguaje de programación OCTAVE [8].

Para cada problema prueba de *job shop scheduling* se procesa la información de la matriz publicada, ya que ésta contiene secuenciación y tiempo de forma alternada.

Los parámetros utilizados en el algoritmo híbrido se calibraron para el *job shop scheduling*. Por ejemplo, para el problema la20 con 50 hormigas y 5 iteraciones, se obtuvo en 10 ejecuciones un promedio de 968 unidades de tiempo, para el mismo problema con 25 iteraciones se llegó a 942.

Se realizaron 10 ejecuciones del algoritmo para cada problema prueba, con los siguientes parámetros calibrados: cantidad de iteraciones 15, cantidad de hormigas 50, feromona inicial 20, $\rho = 0.30$, $\alpha = 1$, $\beta = 10$, $Q = 100$. En el caso de la técnica de recocido simulado los valores de los parámetros calibrados son: $T = 50$, $N = 70$, $\alpha_r = 0.20$.

Los resultados obtenidos se comparan con los resultados reportados hasta ahora como los mejores para cada problema, aunque se hayan obtenido con diferentes técnicas [1, 16, 17].

En la tabla 1 se presenta los resultados de las pruebas, donde la primera columna tiene el nombre del problema, en la segunda columna está el tamaño del problema, cantidad de trabajos por la cantidad de máquinas, en la columna tres el mejor valor óptimo reportado, las columnas de la 4 a la 7 son del modelo híbrido propuesto, la primera de ellas es el mejor valor obtenido, en la siguiente es el peor valor obtenido y la última el promedio de correr 10 veces el problema.

Tabla 1. Resultados de las pruebas, donde la primera columna tiene el nombre del problema, en la segunda está el tamaño del problema, cantidad de trabajos por la cantidad de máquinas, en la columna tres el mejor valor óptimo reportado, las columnas de la 4 a la 7 son del modelo híbrido propuesto, la primera de ellas es el mejor valor obtenido, en la siguiente es el peor valor obtenido y la última el promedio de correr 10 veces el problema.

Problema	Tamaño	Mejor re- portado	ACO-SA		
			Mejor	Peor	Promedio
La01	10x5	666	666	666	666
La02	10x5	655	655	663	656.6
La03	10x5	597	603	626	614.6
La04	10x5	590	590	600	596.2
La05	10x5	593	593	593	593
La06	15x5	926	926	926	926
La07	15x5	890	890	900	892
La08	15x5	863	863	863	863
La09	15x5	951	951	951	951
La10	15x5	958	958	958	958
La11	20x5	1222	1222	1222	1222
La12	20x5	1039	1039	1039	1039
La13	20x5	1150	1150	1150	1150
La14	20x5	1292	1292	1292	1292
La15	20x5	1207	1207	1212	1208
La16	10x10	945	978	988	984.2
La17	10x10	784	983	1016	1002
La18	10x10	848	897	935	921.8
La19	10x10	842	876	907	888.4
La20	10x10	902	914	961	934.6
La30	20x10	1355	1469	1540	1504
La40	15x15	1222	1407	1479	1444.2

Se puede observar en la tabla 1 que en 14 (resaltados en negritas) de 22 problemas prueba, es decir, en el 63.63% se llega a encontrar la mejor solución reportada hasta el

momento y que para los problemas en los cuales no se logra llegar al mejor reportado, se tiene una aproximación promedio del 92%.

7 Conclusiones

Con los resultados obtenidos se concluye que el algoritmo híbrido propuesto, ACO con recocido simulado como búsqueda local, converge en quince iteraciones, es decir, no se requiere de un número grande de iteraciones para encontrar una solución óptima.

En el 63.63% de los problemas prueba, se llega a la mejor solución reportada. En el resto de los problemas se tiene una aproximación promedio del 92%.

Como trabajo futuro se probará en recocido simulado, otras calendarizaciones del enfriamiento así como la elección de los vecinos. En el algoritmo de colonia de hormigas se propondrán formas alternativas para la actualización y evaporación de la feromona.

Referencias

1. Banharnsakun, A., Sirinaovakul, B., Achalakul, T.: Job Shop Scheduling with the Best-so-far ABC. *Engineering Applications of Artificial Intelligence*, Volume 25 Issue 3, 583-593 (2011)
2. Beasley, J.: OR-Library: Distributing test problems by electronic mail. *The Journal of the Operational Research Society*, 41(11), 1069-1072 (1990)
3. Dorigo, M., Birattari, M., Stutzle, T.: Ant Colony Optimization: Artificial Ants as a Computational Intelligence Technique. Technical Report, Iridia - Technical Report Series No.23 (2006)
4. Dorigo, M., Maniezzo, V., Colomi, A.: Ant System: Optimization by a colony of cooperating agents. In: *IEEE Transactions on Systems, Man and Cybernetics, Part B*, pp. 29-41 (1996)
5. Dorigo, M., Stutzle, T.: The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances. In: *International Series in Operations Research & Management Science*. Vol.57, pp. 250-285 (2003)
6. Duan, L., Havens, W. S.: *Applying Systematic Local Search to Job Shop Scheduling Problems: Basic Concepts and Methods*. VDM Verlag (2008)
7. Garey, M. R., Johnson, D. S., Sethi, R.: The Complexity of Flowshop and Jobshop Scheduling. *Mathematics of operations research*, 117-129 (1976)
8. GNU Octave, <http://www.gnu.org/software/octave/>
9. Huang, K. L., Liao, C. J.: Ant colony optimization combined with taboo search for the job shop scheduling problem. *Computers and Operations Research*, Vol. 35, No. 4, pp. 1030-1046 (2008)
10. Kirkpatrick Jr., S., Gelatt, C. D., Vecchi, M.: Optimization by simulated annealing. *Science*, Vol.220, No.4598, pp. 671-680 (1983)
11. Meeran, S., Morshed, M. S.: A hybrid genetic tabu search algorithm for solving job shop scheduling problems: a case study. *J. of Intelligent Manufacturing*, Vol. 23, No.4, pp. 1063-1078 (2012)
12. Mitra, D., Romeo, F., and Vincentelli, A.S.: Convergence and Finite-Time Behavior of Simulated Annealing. *Advances in Applied Probability* Vol. 18, No. 3, 747-771 (1986)

13. Muñoz Pérez, J.: *Inteligencia Computacional Inspirada en la Vida*. Servicio de Publicaciones de la Universidad de Málaga, (2010)
14. Prawda, J.: *Métodos y Modelos de Investigación de Operaciones, Vol. 2 Modelos estocásticos*. Limusa, México (2000)
15. Rojas Santiago, M., Damodaran, P., Muthuswamy, S.: Makespan minimization in a job shop with a BPM using simulated annealing. *International Journal of Advanced Manufacturing*, Vol. 68, No.9, pp. 2283-2391 (2013)
16. Sureshkumar, S., Saravanan, G., Thiruvankadam S.: Optimizing Makespan In JSSP Using Unordered Subsequence Exchange Crossover In GA. *IOSR Journal of Computer Engineering*, Volume 8 Issue 5, 41-46 (2013)
17. Xueni Q., Henry, Y. K. L.: An AIS-based hybrid algorithm for static job shop scheduling problem. *J. Intelligent Manufacturing*, Vol. 25, No.3, pp. 489-503 (2014)

Esbozo de una técnica para la recomendación de productos turísticos en Cuba usando GIS

Jesse Daniel Cano Otero¹, Marlon Jorge Remedios Gonzalez¹, Maria Teresa Morell Franco²

¹Universidad de las Ciencias Informáticas. Facultad I. Departamento de Programación.
{jdcano, mjremedios}@uci.cu

²Grupo Empresarial BIOCUBAFARMA. Departamento de Informática
maria@sd.biocubafarma.cu

Resumen. El turismo cubano se ha desarrollado tanto en los últimos 30 años al punto de convertirse en uno de los motores de la economía cubana. Con este desarrollo también han crecido las empresas cubanas que apuestan por el turismo electrónico como vía para dar a conocer sus productos y captar clientes. A pesar de este hecho, la mayoría de los sitios web cubanos de temática turística se limitan a brindar información de los distintos productos y servicios lo que provoca, en muchos casos, que el usuario se abrume con el cúmulo de información y abandone la búsqueda antes de encontrar un producto que se adecue a sus necesidades. La personalización ha sido reconocida como un factor crítico para las empresas de turismo electrónico exitosas y la utilización de sistemas de recomendación es el mejor enfoque para tratar con el problema de la personalización. El presente trabajo tiene como objetivo realizar un acercamiento preliminar a una técnica que permita obtener predicciones sobre a qué productos un usuario en particular daría una mejor valoración; estos productos serían los que se le mostrarían en primera instancia. Para lograr esto en el trabajo se analizan los elementos teóricos del entorno turístico en Cuba; de los sistemas de recomendación y de los sistemas de información geográfica como herramientas para la representación de la información. Por último se definen, por cada componente estructural identificado, un conjunto de reglas que permitirán obtener un sistema de turismo electrónico que maneje la personalización del servicio que presta de manera efectiva.

Palabras Clave. Productos turísticos; recomendación; sistema de información geográfica; técnica.

1 Introducción

Luego de la caída del campo socialista, el gobierno revolucionario apostó por el desarrollo turístico en Cuba como vía de proveer un alivio a la economía. Este sector se desarrolló al

punto de convertirse en uno de los motores de la economía cubana. Con el auge de internet en los noventa la industria turística mundial fue cambiando dando lugar al turismo electrónico (e-turismo). De hecho, varios estudios recientes indican que la Web se ha convertido en la principal fuente de información en el mundo occidental para las personas que buscan obtener información de los destinos turísticos cuando se encuentran planificando sus viajes [1]. A pesar de este hecho, la mayoría de los sitios web cubanos de temática turística se limitan a brindar información de los distintos destinos y en algunos casos brindan la posibilidad de realizar reservaciones para algún que otro producto turístico. Es obvio que la cantidad de información que recibe el usuario cuando desea encontrar un producto turístico cubano en la web puede convertirse en un problema porque el usuario se vería abrumado por el cúmulo de información disponible y es probable que abandone el sitio web antes de encontrar el producto de su preferencia.

La personalización ha sido reconocida como un factor crítico para las empresas de e-turismo exitosas [2]. El enfoque más exitoso para lidiar con el problema de la personalización es la utilización de los sistemas de recomendación (SR y SRs para el plural) [3,4]. Este enfoque provee a los usuarios técnicas para el filtrado de información usando el conocimiento sobre sus preferencias. Los SRs pueden proveer al visitante del sitio un conjunto de opciones, desde destinos, hoteles, museos y lugares de interés que se ajusten al perfil y presupuesto del visitante. Para la personalización del servicio en el e-turismo, los enfoques más populares son el uso de SRs con filtrado colaborativo y basados en el contenido.

Por otro lado, muchos SRs hacen uso de sistemas de información geográfica (GIS por sus siglas en inglés) como una vía de mejorar la presentación de las recomendaciones y, de esta manera, incrementar la interactividad del sistema con los usuarios. La utilización de GIS para el e-turismo es totalmente razonable dado a que la mayoría de la información turística tiene una estrecha relación con el espacio [5].

Dada la dinámica del sector turístico y el emergente, pero actualmente muy competitivo, mercado electrónico del turismo, casi todas las organizaciones presentes en la industria han implementado una estrategia que explota las tecnologías de la información (TI) e internet [6]. Esto sucede porque la utilización de las TI ha permitido que los turistas tomen un rol más activo en la producción de productos turísticos porque ya no se conforman con productos estandarizados [8].

Dependiendo de la localización de los clientes respecto al índice de inclusión digital [7], más y más clientes adquieren sus productos turísticos a través de sitios web y perciben que la imagen del sitio y su usabilidad afectan directamente sus intenciones adquisitivas [9]. Por lo tanto, entender las percepciones de los clientes y sus comportamientos online son determinantes para el desarrollo de un sitio de e-turismo exitoso [10]. Un SR puede ser de gran utilidad ya que provee asistencia en el proceso de brindar – recibir indicaciones acerca de las opciones que se ajustan mejor en un caso específico para individuos específicos [11]. Ricci [12] dice que un SR puede brindar información valiosa para asistir en el proceso de toma de decisiones del cliente a partir de la identificación de los

requisitos de este y mediante la identificación de su correlación con los requisitos y preferencias de otros clientes. Kaplanidou y Vogt [13] demostraron que la utilidad, para un usuario en particular, de un sitio web constituye un predictor significativo de la intención de viajar al destino.

Con la rapidez de las transmisiones de datos en internet, los tiempos de respuesta esperados de las organizaciones para con los usuarios han sido reducidos drásticamente. La reacción de los sistemas a las consultas online pueden, por lo tanto, influenciar la satisfacción del cliente y su comportamiento adquisitivo. Lograr un entendimiento de los distintos comportamientos online de los clientes puede incrementar la posibilidad de que el cliente complete la transacción [14]. Teniendo en cuenta esto, las características del comportamiento de compras online de los viajeros han sido estudiadas en profundidad por los investigadores. Por ejemplo, los estudios han mostrado que los clientes chinos son menos propensos a confiar en los anuncios publicitarios de productos turísticos y se enfocan comúnmente en el word-of-mouth (comentario, sugerencia, rumor) electrónico y en la información online [14].

A pesar de la gran importancia que, para Cuba, tiene la industria del turismo las empresas que utilizan el e-turismo como vía de captar clientes saturan de información la Web de manera que los potenciales clientes se enfrentan al problema de determinar qué producto turístico consumir a partir de un monto abrumador de información. Esto invariablemente atenta contra la eficiencia con que operan estas empresas y contra el consumo de los productos turísticos que se ofertan. A las organizaciones dedicadas al turismo que han fallado en aplicar correctamente las tecnologías de la información y las comunicaciones (TICs) les es muy difícil gestionar sus negocios sensibles a la información sin afectar su competitividad [15]. Esta situación puede ser mitigada con el uso de un método que permita predecir qué producto turístico puede ser de interés para un grupo determinado de clientes. En este contexto se puede hacer uso de un SR, pero, para esto, se deben caracterizar los distintos elementos que componen el entorno turístico cubano con el objetivo de proveer una recomendación de productos turísticos con exactitud y de manera precisa. En el presente trabajo se esbozará una técnica para obtener recomendaciones efectivas de productos turísticos en Cuba a partir de la definición de un conjunto de reglas que indicarán cómo deben ser implementados cada uno de sus componentes.

2 Sistemas de recomendación

Los SRs son herramientas de software y técnicas que proveen sugerencias o recomendaciones acerca de los ítems que pueden ser útiles para un usuario [4,16]. Desde este punto de vista se puede decir, entonces, que los SRs son la respuesta técnica al hecho de que las personas frecuentemente se basan en la experiencia y recomendaciones de otras personas cuando se enfrentan a un nuevo campo del conocimiento, cuando no conocen con amplitud todos los hechos o cuando el conocimiento excede la cantidad de información que un

humano puede manejar cognoscitivamente [8]. Estas recomendaciones se relacionan a varios procesos de tomas de decisiones, por ejemplo: qué ítem comprar, qué canción escuchar o qué noticia leer.

El término “ítem” se usa, de manera general, para denotar qué es recomendado a los usuarios por el sistema. Los SRs normalmente se centran en un tipo de ítem específico (dígase canciones, noticias o libros) y de acuerdo con esto su diseño, su interfaz gráfica de usuario (GUI por sus siglas en inglés), y la técnica de recomendación se configura para proveer sugerencias útiles y efectivas para ese tipo específico de ítem.

Luego, los SRs se dirigen principalmente a los individuos que carecen de la suficiente experiencia personal o la competencia para evaluar el potencialmente abrumador número de alternativas que un sitio Web, por ejemplo, puede ofrecer [4].

En su forma más simple, las recomendaciones se ofrecen como una lista ordenada de ítems. Al ejecutar este ordenamiento, los SRs tratan de predecir cuáles son los productos o servicios más convenientes para el usuario. Para lograr esto, computacionalmente, los SRs recolectan las preferencias de los usuarios, ya sea explícitamente, por ejemplo: solicitando a los usuarios valoraciones sobre los ítems, o infiriéndolas a partir de las acciones de los usuarios. En la actualidad, los SRs han demostrado ser un medio valioso de lidiar con el problema del exceso de información.

La taxonomía propuesta en el artículo “Hybrid web recommender systems” [17] se ha convertido en un clásico a la hora de distinguir los distintos tipos de SRs, en dicho artículo se proponen 6 enfoques de recomendación diferentes:

- Sistemas de recomendación basados en contenido: Este tipo de SR computan las recomendaciones basándose en las características de los ítems que el usuario ha valorado positivamente en el pasado.
- Sistemas de recomendación colaborativos (SRC): Estos sistemas usan las valoraciones de los usuarios para categorizarlos en grupos de acuerdo con la similitud de sus preferencias. Las recomendaciones se infieren teniendo en cuenta las valoraciones de los usuarios categorizados en un mismo grupo.
- Sistemas de recomendación demográficos: Estos categorizan a los usuarios en grupos demográficos basándose en atributos personales. Los usuarios, entonces, reciben recomendaciones de acuerdo con los grupos en los que se encuentran clasificados.
- Sistemas de recomendación basados en conocimiento: Estos sistemas infieren las recomendaciones basándose en las necesidades del usuario y en el conocimiento de las características de los ítems. Típicamente, computan las recomendaciones usando procesos de razonamiento basado en casos, por ejemplo: el usuario provee un ejemplo de ítem similar al que le interesa y el sistema, a partir de este conocimiento, infiere un perfil de ítem que permita encontrar los mejores ítems en el espacio de búsqueda.
- Sistemas de recomendación basados en comunidades: Este tipo de SR sugiere los ítems basándose en las preferencias de las “amistades” del usuario en cuestión. Esta técnica sigue el epigrama “Dime con quién andas y te diré quién eres” [18,19].

- Sistemas de recomendación híbridos. Este tipo de sistema se basa en la combinación de cualquiera de las técnicas mencionadas anteriormente. Un sistema híbrido que combina las técnicas A y B trata de aprovechar las ventajas de A para solucionar las desventajas de B.

De acuerdo con varias investigaciones [20,21], la aplicación de los sistemas de recomendación colaborativos es el enfoque más exitoso para realizar recomendaciones.

2.1 Sistemas de recomendación colaborativos

Como se había mencionado anteriormente los SRC computan las recomendaciones a partir de la clasificación de los usuarios en grupos con valoraciones similares. Luego, el problema de la recomendación puede ser definido como: la tarea de estimar la respuesta de un usuario para un ítem que es nuevo para él, basándose en la información histórica almacenada en el sistema. El tipo de respuesta usuario – ítem varía de una aplicación a otra y se puede categorizar de 3 maneras: escalar, binaria y unitaria. Las respuestas escalares son numéricas (ejemplo: valoraciones de 1 a 5 estrellas) u ordinales (ejemplo: Muy de acuerdo, De acuerdo, Neutral, En desacuerdo, Muy en desacuerdo) siempre representando los posibles niveles de apreciación de los usuarios respecto a los ítems. Las respuestas binarias solo tienen dos posibles valores donde se capturan dos niveles de apreciación opuestos (ejemplo: Me gusta, No me gusta). Finalmente, las respuestas unitarias capturan la interacción de un usuario con un ítem (ejemplo: compra, acceso online, etc.) sin brindar una información explícita de la apreciación del usuario por el ítem en cuestión.

Las maneras en las que las respuestas del usuario son capturadas pueden ser distintas también. Una manera es solicitar explícitamente una valoración al usuario pero otra manera podría ser el almacenamiento del historial de compras o acceso de un usuario a un grupo de ítems [22].

Se puede decir, entonces, que la idea central de los SRC es que la valoración de un usuario u para un nuevo ítem i será probablemente similar a la valoración de otro usuario v sobre i si u y v han valorado otros ítems de manera similar. El enfoque colaborativo resuelve alguna de las limitaciones del enfoque basado en contenido. Por ejemplo, los ítems para los cuales no se tiene información de sus características, o esta información resulta difícil de obtener, pueden ser recomendados a través de la retroalimentación provista por otros usuarios. Además, las recomendaciones obtenidas por un SRC se basan en la calidad de los ítems según el criterio de valoración de los pares del usuario activo en vez de basarse en el contenido, pues este puede ser un mal indicador de la calidad. Finalmente el enfoque colaborativo puede recomendar ítems con un contenido diametralmente distinto siempre y cuando otros usuarios hayan mostrado interés por estos ítems.

Los SRC automatizan el concepto de “rumor”, donde se confía en la opinión de personas con similitud de pensamiento u otra fuente confiable para evaluar la relevancia de un ítem (película, libro, etc.).

Existen 2 enfoques para realizar recomendaciones a partir de un SRC: “usuario - usuario” e “ítem - ítem”. El enfoque “usuario - usuario” se basa en que para obtener la predicción de la valoración de un usuario activo a cualquier ítem no valorado se haya la vecindad de ese usuario (usuarios que han efectuado valoraciones similares) para luego inferir la valoración que le daría al ítem el usuario activo. Por otro lado, el enfoque “ítem - ítem” se basa en que para obtener la valoración del usuario activo a cualquier ítem no valorado se haya la vecindad del ítem (ítems que han recibido valoraciones similares) y con esa información se predice la valoración que le daría el usuario activo. La selección de uno u otro enfoque depende básicamente en la estabilidad del grupo, por ejemplo, si los usuarios del sistema tienden a variar (se incorporan o se eliminan usuarios frecuentemente) se debe utilizar el enfoque “ítem - ítem”; si por el contrario la comunidad de usuarios del sistema tiene una mayor estabilidad sobre el conjunto de ítems ofertados por el sistema es recomendable utilizar el enfoque “usuario - usuario”.

Para cualquiera de los enfoques a utilizar dentro de los SRCs existen 2 métodos que permitirán obtener la predicción de la valoración del usuario activo sobre cualquier ítem no valorado por este. Estos métodos son el de regresión y el de clasificación. La opción de implementar el método de regresión o clasificación “usuario - usuario” depende, en gran medida, de la escala de valoración. Si el sistema utiliza una escala continua que va, por ejemplo, desde 1 a 10, es recomendable la utilización del método de regresión. Por el contrario si la escala de valoración contiene pocos valores discretos por ejemplo “Me gusta” y “No me gusta” o si los valores de la escala no pueden ser ordenados de una manera trivial, es recomendable la utilización el método de clasificación.

De manera general la predicción r_{ui} “ítem - ítem” del usuario u a ítem i por el método de regresión se calcula como el promedio de las valoraciones normalizadas y ponderadas dadas por los k usuarios más cercanos a u mediante la siguiente fórmula [23] :

$$r'_{ui} = h^{-1} \left(\frac{\sum_{j \in N_u(i)} w_{ij} h(r_{uj})}{\sum_{j \in N_u(i)} |w_{ij}|} \right) \quad (1)$$

donde w_{ij} es el coeficiente de similitud entre el ítem i (ítem del que se quiere predecir la valoración que daría el usuario activo) y el ítem j (uno de los vecinos de i) y $h(r_{uj})$ sería la normalización de la valoración dada por el usuario activo a j .

De manera análoga la predicción “ítem - ítem” por el método de clasificación se obtendría mediante la siguiente fórmula [23]:

$$r'_{ui} = h^{-1} \left(\operatorname{argmax}_{r \in S'} \sum_{v \in N_u(i)} \partial(h(r_{vj}) = r) w_{ij} \right) \quad (2)$$

donde la función $\partial(h(r_{vj}) = r)$ toma valor 1 si $r_{vj} = r$ y 0 en caso contrario.

2.2 Uso de los GIS con los SRs de productos turísticos

Un GIS es un sistema de software diseñado para permitir la entrada, el almacenamiento, la edición, la captura, el análisis y la salida de información geográficamente referenciada [24]. Los GIS son herramientas con un alto valor para los servicios turísticos porque los productos que se ofrecen en este sector normalmente pueden ser ubicados en un mapa.

El diseño de la interacción y la interfaz gráfica de un SR puede afectar profundamente el proceso de toma de decisiones del usuario. La utilización de un GIS como vía de representar las recomendaciones de productos turísticos obtenidas por un SR, brinda la ventaja de permitir al usuario la visualización geográfica de las entidades e ítems recomendados de una manera intuitiva, permitiéndoles, entre otras cosas, evaluar la distancia que existe entre ítems de los que puede convertirse en un consumidor. Por lo que al pensar en la implementación de un SR para productos turísticos, la utilización de un GIS para la presentación de los resultados se torna prácticamente obvia.

3 Técnica para la recomendación de productos turísticos en Cuba

Para obtener un sistema que recomiende ítems de manera efectiva se deben tener en cuenta cada una de las partes de este sistema, no solo el motor de recomendación. El objetivo de este trabajo es esbozar una técnica que permita la obtención de este sistema. Para lograr esto se parte de una primera regla que define una estructura general para el sistema y luego se definen un conjunto de reglas para cada uno de los componentes de esta estructura teniendo en cuenta las características contextuales del entorno turístico en Cuba. En la figura 1 se define la estructura general que tiene como componentes principales la interfaz gráfica de usuario, el selector de recomendación y el motor de recomendación.

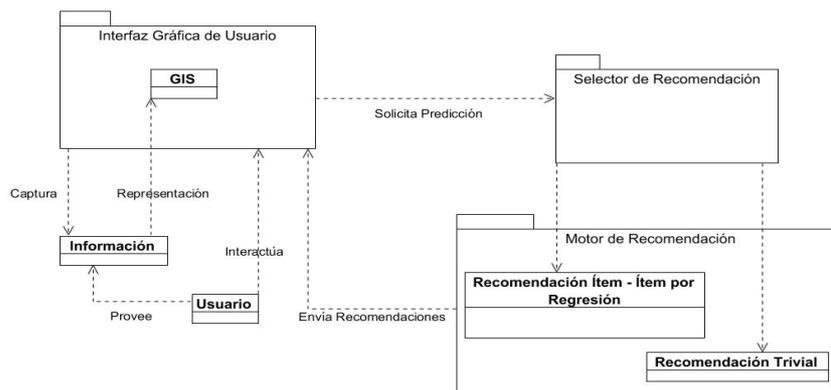


Fig. 1. Esquema del funcionamiento de la técnica para la recomendación de productos turísticos en Cuba.

3.1 Interfaz gráfica de usuario

La responsabilidad de este componente es la de garantizar la captura de la información que provee el usuario a partir de sus entradas de datos o de sus acciones sobre el sistema; además debe brindar una representación correcta de la información de manera que el usuario se sienta cómodo en su interacción con el sistema.

La información que debe ser representada es, básicamente, la referente a los productos turísticos que la organización que utilice el e-turismo desea ofertar a los usuarios. La manera en que esta información es presentada incide directamente en cómo los usuarios interactúan con el sistema así como el nivel de aceptación que estos le confieren. Para la representación de la información, luego, se propone utilizar un GIS debido a la característica inherente de los productos turísticos de poder ser ubicados geográficamente. El GIS debe mostrar un mapa de la región geográfica en cuestión donde se ubiquen los productos turísticos a ofertar y debe contar con funcionalidades básicas de zoom y paneo para permitir al usuario moverse libremente por el mapa.

Además, este componente debe representar la información referente a las recomendaciones que se le van a realizar al usuario, esta información constituye una entrada de este componente y se obtienen a partir del componente Motor de Recomendación. Se propone separar la visualización de las recomendaciones del mapa, es importante no efectuar recomendaciones sobre el mapa pues esto limitaría la variedad de productos con una buena predicción de valoración que pudieran ser recomendados cuando el usuario esté visualizando un fragmento de toda la región geográfica representada en el mapa.

La información que debe ser capturada, dando cumplimiento a la segunda responsabilidad de este componente, es aquella referente a la opinión del usuario sobre algún ítem así como aquella asociada a la descripción de la situación del usuario en el sistema. La valoración de los usuarios sobre los ítems que le son mostrados es la base del sistema de recomendación, estas valoraciones constituyen la información sobre la cual serán calculadas todas las predicciones que utilizará el sistema para decidir que ítems debe mostrar prominentemente a un usuario u otro.

Para el contexto de las organizaciones dedicadas al turismo en Cuba se propone la utilización de un método de captura de las valoraciones explícito y con una escala continua; esta manera debe implementarse de forma tal que no se interrumpa la navegación del usuario. Por la relación que guarda con la actividad turística y la manera en que son categorizados la mayoría de las instalaciones de alojamiento en Cuba se propone usar una escala de valoración de 1 a 5 esta escala podría ser presentada en forma de estrellas cuando el usuario esté visualizando la información detallada de algún ítem así éste podría valorar la calidad de un servicio o producto de 1 a 5 estrellas sin ver afectada, además, su interacción con el sitio como lo sería si se le mostrara un formulario adicional solo para que la valoración sea provista.

Por otro lado, la información situacional o contextual del usuario en un instante de tiempo determinado es la base que utilizará el componente Selector de Recomendación

para decidir qué tipo de predicción debe utilizar para obtener el listado de ítems que debe ser recomendado. En este componente debe implementarse la captura de la información relevante a ser usada por el selector de recomendación, esta información sería la referencia geográfica del centro del fragmento de la región geográfica que está visualizando el usuario, el nivel de zoom con que se está visualizando el mapa o si el usuario está visualizando o no los detalles de un producto turístico. La ubicación geográfica del centro de la porción de el mapa que está siendo visualizada y el nivel de zoom se utilizarán para completar, por cercanía, el conjunto de ítems que van a conformar la vecindad a partir de la cual se calculará la predicción de valoración para cada ítem, para luego obtener el listado de productos a recomendar; la información de si un usuario esta visualizando los detalles de un producto se utilizará para recomendar productos similares a este y, de esta forma, contribuir a mitigar el problema del arranque en frío. Es importante destacar que este tipo de información en particular no es persistente dado que describe una situación en un instante dado, esta información constituye una salida del componente Interfaz Gráfica de Usuario y entrada del componente Selector de Recomendación.

3.2 Selector de recomendación

La responsabilidad del selector de recomendación será la de analizar cada escenario en que se encuentre cada usuario en particular para decidir qué método de recomendación debe ser utilizado. Es en este componente donde se realiza la mayor parte de la tarea correspondiente a mitigar el problema del arranque en frío. El problema del arranque en frío viene dado cuando no se tienen suficientes valoraciones de un usuario o sobre un ítem de manera que no se puede realizar una predicción efectiva de valoración para uno y el cálculo de la vecindad para el otro.

De manera general, la información que sirve de entrada a este componente es la información contextual del usuario activo que se había mencionado anteriormente, referencia geográfica del centro de la región que está siendo visualizada, nivel del zoom y si está siendo visualizada la descripción detallada de un producto. A esta información contextual se le debe agregar la información de la cantidad de valoraciones que ha realizado el usuario activo, esta información aunque es situacional (puede variar de un instante de tiempo a otro) no la ofrece el componente Interfaz Gráfica de Usuario y debe ser obtenida a partir de los datos de las valoraciones almacenados por el sistema. La cantidad de valoraciones suficientes para ofrecer una predicción efectiva es un número que debe ser definido por aquel que se encuentre implementando la técnica pues depende de la situación particular (volumen de usuarios, ítems y valoraciones con que la organización debe trabajar). En cualquier caso, luego de definido el número, el trabajo con él es sencillo: si el usuario tiene las valoraciones necesarias se utiliza el método de recomendación ítem – ítem correspondiente, de lo contrario se debe utilizar un método de recomendación trivial.

Por otro lado, el dato de la referencia geográfica del centro de la región que está siendo visualizada debe ser utilizado, de ser necesario, para completar la vecindad de ítems

que permitirá calcular una predicción de valoración. Por ejemplo, se puede definir una distancia X y, de no encontrarse los k ítems valorados por el usuario activo que son necesarios para predecir una valoración a cualquier ítem no valorado, se completaría el número k de ítems utilizando aquellos que se encuentren dentro de un radio X a partir del punto que marca el centro de la región que se está visualizando y de los que se obtengan una mayor predicción de valoración para el usuario activo. Aplicar esta restricción puede incidir, en la calidad del funcionamiento del sistema de recomendación que se obtenga. Esto se debe a que a la relevancia que tiene el componente geográfico de los productos turísticos, por ejemplo: para un usuario que se encuentre hospedado o tenga reservación en un hotel, existe la probabilidad de que esté más interesado en los productos turísticos relativamente cercanos al dicho hotel. Por esta razón, se asume que si el usuario está visualizando el mapa con un nivel de zoom que se presume elevado es porque ese usuario tiene un interés especial en esa porción del mapa y luego tendría sentido usar este completamiento con ítems que se encuentren en una distancia razonablemente corta. Una vez más, el número que se defina para X y para el nivel de zoom depende de la situación en particular en la que se encuentre el equipo de desarrollo cuando esté implementando la técnica.

La información de si se está visualizando los detalles de un producto sirve para decidir que se va a recomendar en caso que no se tenga un número de valoraciones suficientes para efectuar una recomendación. Si se está visualizando la descripción detallada de un producto se puede asumir que este producto está recibiendo un nivel de atención significativo y en consecuencia se deben recomendar los productos que se encuentren en la vecindad inmediata (tengan un mayor coeficiente de similitud) de aquel que está siendo visualizado.

De manera general, en el cumplimiento de su responsabilidad, este componente, primero debe analizar si el usuario activo posee el número suficiente de valoraciones necesarias para realizar una recomendación. De no contar con las valoraciones suficientes se debe solicitar al Motor de Recomendación una recomendación trivial como lo sería el listado de ítems más populares (mejor valorados). En el caso de que el usuario activo no cuente con las valoraciones necesarias pero esté visualizando la información detallada de un ítem, se debe recomendar el listado con la vecindad inmediata de ese ítem. Por último, si el usuario cuenta con las valoraciones suficientes para obtener una recomendación, el componente debe analizar si el usuario está haciendo énfasis en una región específica del mapa para solicitar al Motor de Recomendación el listado de ítems a recomendar usando o no el completamiento de vecindad a partir de la distancia del centro de la región que está siendo visualizada.

3.3 Motor de recomendación

La responsabilidad del Motor de Recomendación es la de obtener el listado de ítems que serán mostrados al usuario prominentemente. Como su nombre lo indica, este componente es el responsable de hacer que la herramienta obtenida sea capaz de recomendar los ele-

mentos que son, presumiblemente, de interés para el usuario. Es en este componente donde se debe implementar el comportamiento derivado del razonamiento matemático usado para predecir la valoración que un usuario activo daría a un ítem no valorado, para luego mostrar prominentemente el listado de ítems que posean una predicción de valoración más elevada. Como se discutió anteriormente, se ha demostrado recientemente que el enfoque colaborativo es el más exitoso para efectuar recomendaciones al usuario. Principalmente porque ofrece varias y útiles ventajas y porque sus deficiencias pueden ser mitigadas con relativa facilidad. Usando este enfoque se persigue predecir las valoraciones, para los ítems de los cuales éstas no se conocen, utilizando las valoraciones que los usuarios de la comunidad ya han dado. Siguiendo esta línea de pensamiento, existen dos métodos para la implementación del enfoque colaborativo: el “Usuario - Usuario” y el “Ítem - Ítem”. En el contexto turístico cubano, la población de clientes (usuarios) es usualmente más voluminosa e inestable que la de los productos disponibles para el consumo (ítems) por esto se propone la implementación del método “Ítem - Ítem”.

Para cumplir con su responsabilidad, este componente debe tener disponible, como información de entrada, toda aquella asociada a las valoraciones otorgadas por los usuarios del sistema a los ítems disponibles. Además, debe tener disponible el componente geográfico de cada uno de los ítems, en caso que deba efectuar un completamiento de la vecindad usando la distancia de estos respecto a un punto (x;y). Este punto constituye la única información que deberá proveer el Selector de Recomendación al Motor de Recomendación, y lo hará solo en el caso que se desee obtener una lista de recomendación aplicando esta restricción en particular. En cualquier caso, la información de salida que debe proveer este componente es una lista con los elementos que deben ser mostrados prominentemente (recomendados) al usuario que será obtenida a partir del tipo de recomendación que haya solicitado, en particular, el Selector de Recomendación.

4 Pruebas

Para probar la técnica descrita se implementó un Sistema de Recomendación de Productos Turísticos aplicando las reglas definidas en esta. Luego se pobló una base de datos con 4000 usuarios y 1000 productos turísticos para, entonces, generar aleatoriamente valoraciones de esos 4000 usuarios sobre los 1000 ítems.

Para analizar la información que brinda como resultado la herramienta implementada se utilizaron indicadores de la calidad de las recomendaciones obtenidas como lo son la probabilidad de recomendaciones relevantes (Precisión) y la probabilidad de que los ítems relevantes sean recomendados (Recall).

4.1 Precisión y recall

La Precisión y el Recall se computan a partir de una tabla de 2 x 2 como se muestra en la tabla 1. El conjunto de ítems debe ser dividido en dos clases: ítems relevantes o no relevantes. En el caso de la técnica propuesta en el presente trabajo se considerarán relevantes para un usuario aquellos ítems que hayan obtenido una valoración de 4 o 5. Además se deben considerar los ítems que fueron recomendados (Seleccionados) al usuario, para el caso de la técnica se recomiendan todos aquellos que superen un umbral determinado. A su vez, en el ámbito de la implementación de la técnica, el valor umbral de predicción que debe superar un ítem para ser recomendado fue definido en 3.5 unidades.

Tabla 1. Datos de entrada para el cómputo de la Precisión y el Recall.

	Seleccionados	No Seleccionados	Total
Relevantes	Nrs	Nrn	Nr
Irrelevantes	Nis	Nin	Ni
Total	Ns	Nn	N

Luego, la precisión de las recomendaciones efectuadas a un usuario determinado, que representa la probabilidad de que un ítem recomendado sea relevante para el usuario, se obtiene como la división de la cantidad de ítems que fueron recomendados y son relevantes entre el total de ítems que fueron recomendados [24]:

$$P = \frac{N_{rs}}{N_s} \quad (3)$$

por otro lado el Recall representa la probabilidad de que un ítem relevante sea recomendado. Esta métrica se calcula como la división de la cantidad de ítems que fueron recomendados y son relevantes entre el total de ítems que son relevantes para el usuario [24] :

$$R = \frac{N_{rs}}{N_r} \quad (4)$$

aunque lo normal para un sistema de recomendación es predecir la valoración que le daría un usuario a cualquier ítem que este no haya valorado, a priori se puede observar que, para este tipo de prueba, se deben generar las predicciones de valoración para el conjunto de ítems valorados por cada usuario.

4.2 Resultados de las pruebas

Luego del cómputo de los valores de Precisión y el Recall para cada usuario, se obtuvo la media de estas métricas para el sistema usando solo el método de recomendación Ítem – Ítem por regresión, que fue implementado en el Motor de Recomendación. Luego se procedió a efectuar los mismos cálculos pero para cuando se utiliza la técnica como un todo, donde la diferencia en la recomendación marca la utilización del componente Selector de Recomendación perteneciente a la técnica propuesta. Las figuras 2 y 3 muestran una comparativa de estos resultados para Precisión y Recall del sistema respectivamente.

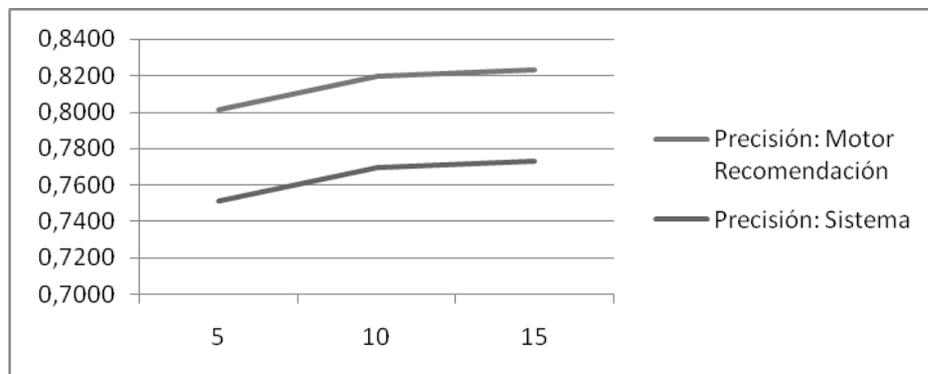


Fig. 2. Comparación de la precisión entre motor de recomendación y el sistema implementado.

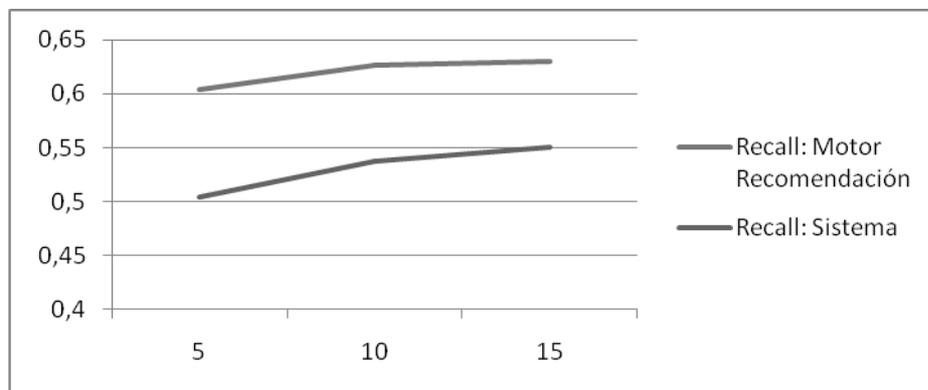


Fig. 3. Comparación del Recall entre motor de recomendación y el sistema implementado.

Como se puede apreciar, los resultados arrojados para la Precisión aumentan según aumenta el valor de k (5, 10, 15) para todos los casos. La probabilidad de que un ítem recomendado por el sistema sea relevante para el usuario, supera el 80 % para las 3 va-

riantes probadas en el caso del Motor de Recomendación en solitario y se mantiene sobre el 75 % cuando se utiliza el sistema como un todo. En el caso del Recall, la probabilidad de que un ítem relevante sea recomendado, también crece en relación directa con el valor de k y esta probabilidad se mantiene por encima del 60 % cuando se evalúa al Motor de Recomendación solamente y sobre el 50 % para la evaluación del sistema. Estos resultados se pueden catalogar de muy buenos para el sistema de recomendación obtenido pues permite afirmar que las recomendaciones obtenidas son probablemente útiles al usuario en cuestión. El decrecimiento de la Precisión y el Recall cuando se utiliza el sistema como un todo es un sacrificio aceptable teniendo en cuenta que, a priori, se pueden calcular un número significativamente mayor de predicciones de valoración en el espacio de predicción con lo que se logra una cobertura notablemente mayor.

5 Conclusiones

1. La importancia que, para Cuba, tiene la industria del turismo, hace de la personalización masiva de los contenidos que muestran las organizaciones de e-turismo cubanas una tarea necesaria.
2. La explotación de las características contextuales derivadas de la utilización de un GIS permite incidir positivamente en la cobertura del sistema sobre el espacio de predicción sin hacer grandes sacrificios en la precisión con que se calculan las predicciones de valoración.
3. La técnica propuesta define las características y el comportamiento que debe ser implementado para obtener recomendaciones de productos turísticos cubanos a partir de las preferencias y la información contextual de cada usuario.
4. Las pruebas realizadas demuestran que para el contexto turístico descrito se obtienen resultados positivos en la exactitud y el uso de las predicciones de valoración calculadas mientras se incide positivamente en la cobertura del sistema implementado, lo que permite afirmar que la aplicación de la técnica se logra obtener de manera eficaz el contenido que debe ser mostrado prominentemente a los usuarios de las organizaciones de e-turismo cubanas.

Referencias

1. Jannach, D., Zanker, J.M.M., Seidler, O.: Developing a conversational travel advisor with advisor suite. En: Information and Communication Technologies in Tourism, Proceedings, Frew, A.J., pp. 43–52, (2007)
2. Gao M. , Liu K., Wu Z.: Personalisation in web computing and informatics: Theories, techniques, applications, and future research, Information Systems Frontiers 12 607–629, (2010)

3. Adomavicius, G., Tuzhilin, A.: Towards the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering* 17 (6), 734–749, (2005)
4. Resnick P., Varian H.: Recommender systems, *Association for Computing Machinery. Communications of the ACM*. 40 p. 56–58, (1997)
5. Charou, E., Kabassi, K., Martinis, A. and Stefouli, M.: Integrating Multimedia GIS Technologies in a Recommendation System for Geo-tourism. En: Tsihrantzis, G. A. and Jain, L. C. (eds.). *Multimedia Services in Intelligent Environment*, Springer-Verlag Berlin, pp. 63-74, (2010)
6. Werthner, H.: *Intelligent Systems in Travel and Tourism*. Sweet,P. (2001) *Designing Interactive Value Development*. PhD. Thesis, Lund University, Lund Business Press, Sweden.
7. Minghetti, V. and Buhalis, D.: Digital divide in tourism. *Journal of Travel Research* 49, 267–281, (2004)
- Chiang, C. and Jang, S.C. The effects of perceived price and Brand image on value and purchase intention: leisure travelers' attitudes toward online hotel booking. *Journal of Hospitality and Leisure Marketing* 15(3), 49–69, (2006)
8. Berka, T., Plobnig, M.: *Designing Recommender Systems for Tourism*. En: 11th International Conference on Information Technology in Travel and Tourism (2011)
9. Benckendorff, P.: An exploratory analysis of traveler preferences for airline website content. *Information Technology and Tourism* 8, 149–159, (2006)
10. Gretzel, U., Yuan, Y.L. and Fesenmaier, D.R.: Preparing for the new economy: advertising strategies and changes in destination marketing organizations. *Journal of Travel Research* 39, 146–156, (2000)
11. Ricci, F.: Travel recommender systems. *IEEE Intelligent Systems* 17, 53–55, (2002)
12. Kaplanidou, K. and Vogt, C.: A structural analysis of destination travel intentions as a function of web site features. *Journal of Travel Research* 45, 204–216, (2006)
13. Lee, J., Soutar, G. and Daly, T.: Tourists search for different types of information: a cross-national study. *Information Technology and Tourism* 9, 165–176. Kim, W.G., Ma, X. and Kim, D.J. (2006) Determinants of Chinese hotel customers' e-satisfaction and purchase intentions. *Tourism Management* 27, 890–900, (2007)
14. Mahmood, T., Ricci, F.: Improving recommender systems with adaptive conversational strategies. En: C. Cattuto, G. Ruffo, F. Menczer (eds.) *ACM, Hypertext*, pp. 73–82, (2009)
15. Law, R. and Jogaratnam, G.: A study if hotel information technology applications. *International Journal of Contemporary Hospitality Management* 17, 170–180, (2005)
16. Burke, R.: Hybrid web recommender systems. En: *The AdaptiveWeb*, Springer Berlin / Heidelberg, pp. 377–408, (2007)
17. Ben-Shimon, D., Tsikinovsky, A., Rokach, L., Meisels, A., Shani, G., Naamani, L.: Recommender system from personal social networks. En: K. Wegrzyn-Wolska, P.S. Szczepaniak (eds.) *AWIC, Advances in Soft Computing*, vol. 43, pp. 47–55. Springer.
18. Arazy, O., Kumar, N., Shapira, B.: Improving social recommender systems. *IT Professional* 11(4), 38–44, (2007)
19. Manouselis, N., Vuorikari, R., Assche, F.V.: Collaborative recommendation of e-learning resources: an experimental investigation, *Journal of Computer Assisted Learning* 26. p. 227–242, (2010)

20. Redpath, J., Glass, D.H., McClean, S., Chen, L.: Collaborative filtering: The aim of recommender systems and the significance of user ratings, en: ECIR , LNCS 5993, pp. 394–406, (2010)
21. Konstan, J.A., Miller, B.N., Maltz, D., Herlocker, J.L., Gordon, L.R., Riedl, J.: GroupLens: applying collaborative filtering to usenet news. *Communications of the ACM* 40(3), 77–87, (1997)
22. Ricci, F., Rokach, L., Shaphira, B., Kantor, P.B.: A Comprehensive Survey of Neighborhood-based Recommendation Methods. En: *Recommender Systems Handbook*, Springer New York / Dordrecht / Heidelberg/ London, pp. 117-130, (2011)
23. Demers, M.: *Fundamentals of Geographic Information Systems*. John Willey & Sons, (2008)
24. Herlocker, J. L., J. A. Konstan, et al.: "Evaluating collaborative filtering recommender systems." *ACM Transactions on Information Systems (TOIS)* 22(1): 5-53, (2004)

Programación generativa en Java y herramientas de meta-programación

José Ismael Beristain-Colorado y Ulises Juárez-Martínez

Instituto Tecnológico de Orizaba
División de Estudios de Posgrado e Investigación
Orizaba, Ver., México
isma.sp6@gmail.com, ujuarez@ito-depi.edu.mx

Resumen. La programación generativa (PG) es un paradigma de desarrollo de software el cual modela e implementa familias de sistemas de modo que es posible que un sistema se genere automáticamente con base en una especificación definida; esto con el objetivo de conseguir alta intención, reutilización y adaptación sin comprometer el desempeño en tiempo de ejecución ni los recursos del software que se produce, solventando de esta manera la necesidad de adaptación de una aplicación a nuevos requerimientos. En este artículo se describe la situación actual de la PG en general y bajo el enfoque Java; también se describe el caso de estudio de una aplicación para química, ésta aplicación representa la creación de elementos químicos por medio de la creación de objetos; a la cual, utilizando distintas herramientas de meta-programación se le agrega la funcionalidad de representar la creación de moléculas con base en los elementos creados por el sistema original.

Palabras Clave: Java, Programación Generativa, Meta-Programación.

1 Introducción

Actualmente en el proceso de desarrollo de software, la etapa de mantenimiento presenta un gran reto debido al continuo cambio en los requerimientos, si bien, la programación orientada a objetos (POO) y programación orientada a aspectos (POA) proporcionan cierto nivel de encapsulación y modificación de programas con lo cual se facilita el reemplazo y mantenimiento de piezas de software, no son suficientes para solventar la necesidad de realizar las modificaciones a un sistema de manera automatizada y a tiempo de ejecución. La PG brinda el soporte adecuado para solventar esta necesidad.

La PG es un paradigma que permite el análisis, desarrollo y mantenimiento de aplicaciones de software, además de permitir la generación de programas con características muy similares a través de líneas de productos de software (LPS). De esta manera al tener tanta similitud, es posible mediante pequeñas modificaciones, el intercambio de las piezas

de software con que están contruidos dichos programas, del mismo modo gracias al análisis de las LPS se obtiene un amplio conocimiento del comportamiento de los sistemas, permitiendo prever cuáles son las piezas que tienen más tendencia a ser modificadas y cuáles son los posibles cambios que se les aplicarán. También es posible reutilizar elementos previamente contruidos para generar nuevos programas de manera automatizada. Para esto es necesario del soporte de distintas técnicas de programación y meta-programación. Las bases de la PG son:

Programación intencional: Es un paradigma cuyo objetivo es que la implementación de un sistema refleje el comportamiento específico (intención) que el programador tiene en mente.

POA: Es un paradigma de programación que permite una mejor modularización de cada una de las partes del sistema a través de la separación de asuntos, entrelazando (a tiempo de carga, compilación o ejecución) los módulos que sean necesarios para el correcto funcionamiento de un programa; a nivel de meta-programación es de gran utilidad para realizar transformaciones a las aplicaciones de software ya que permite la inserción y modificación de código usando aspectos, así como cambios a la estructura estática de los programas (introducciones).

Generadores: Un generador es un componente que obtiene como entrada una especificación (intención) de un requerimiento nuevo o cambiante y mediante distintas técnicas de meta-programación realiza de manera automática los cambios necesarios o genera nuevo código que satisfaga el cambio de requerimientos.

No obstante, a pesar de sus bondades, la programación generativa ha recibido poca atención por la industria de desarrollo de software en su aplicación práctica, ya que es poco común que en términos de desarrollo se contemple un enfoque de PG para la solución de problemas.

La contribución principal de este trabajo consiste en ilustrar mediante un caso de estudio y la utilización de diversas técnicas de meta-programación, las ventajas que presenta la implementación de distintas herramientas para la generación y transformación de programas bajo el enfoque Java; esto con el objetivo de que la industria de desarrollo de software tenga una idea más clara sobre este paradigma y lo lleve a la práctica reduciendo de esta manera el tiempo y costos invertidos a la fase de mantenimiento de software.

Este artículo está organizado de la siguiente forma: en la sección 2 se describe la situación actual de la PG desde un enfoque general. La sección 3 se refiere a la PG enfocada a las tecnologías de meta-programación que proporciona Java. La sección 4 describe el caso de estudio y muestra la implementación de cada una de las herramientas Java propuestas. La sección 5 muestra los resultados obtenidos mediante el caso de estudio. La sección 6 da pie a la discusión de ideas. Finalmente en la sección 7 se presentan las conclusiones y el trabajo a futuro.

2 Situación actual de la PG

La aplicación que se le da a la PG presenta un enfoque de autoconfiguración y generación automática de componentes, por ejemplo: en [1] se menciona que WebDSL es un lenguaje específico del dominio que provee a los desarrolladores conceptos de modelado de datos orientados a objetos con una implementación de persistencia en bases de datos relacionales; cuando el modelo de datos evoluciona, los datos necesitan migrarse, sin embargo, una migración a nivel de la base de datos rompe las abstracciones provistas por WebDSL. Por tanto se presenta un lenguaje específico del dominio para generar código que permita una evolución acoplada del modelo de datos relacional y el modelo de datos de la aplicación. Una aplicación práctica de la programación generativa se aprecia en [2] describiendo a *GeoGram* como un sistema que genera programas para cómputo geométrico; mediante la combinación de componentes de software genéricos realiza inferencias para derivar nuevos datos, introducir nuevos objetos basados en el razonamiento geométrico, filtra las opciones presentadas al usuario y genera el programa. En [3] se presenta un framework para monitorizar alguna propiedad del código generado en tiempo de ejecución aumentando de esta manera el nivel de abstracción con el que los desarrolladores analizan el código obtenido. En [4] se propone introducir la programación generativa en el área de generación de aplicaciones de interfaces de usuario gráficas, esto con el objetivo de demostrar que es posible generar aplicaciones personalizadas de forma automática con partes de interfaces gráficas de usuario mediante especificaciones abstractas. En [5] se mencionan los generadores de bibliotecas de alto rendimiento y las investigaciones que se han realizado comparando las características y conceptos necesarios para que un lenguaje de meta-programación permita la construcción sistemática de estos. En [6] se propone un proceso para el manejo de cambios a nivel de *features* en el desarrollo de software para solventar los problemas de ineficiencia en la comunicación, fallas en el código y altos costos de mantenimiento que se presentan en los proyectos de software debido a la diversidad de *stakeholders* y artefactos.

3 Programación generativa en Java

En el ambiente Java también se reportan trabajos relacionados con la programación generativa, por ejemplo: en [7] se modificó una Máquina Virtual de alto rendimiento para permitir cambios arbitrarios a definiciones de clases cargadas, permitiendo de esta manera actualizar clases en cualquier momento durante la ejecución del programa. En [8] se reporta una propuesta para la migración de bibliotecas Java mediante una herramienta que genera instancias de dichas bibliotecas, como resultado se comprueba que la generación automática de código es de gran ayuda para la migración de bibliotecas de una Máquina Virtual Java estándar a una empotrada. En [9] se presenta una herramienta basada en Java HotSpot la cual permite realizar cambios en tiempo de ejecución a las clases cargadas,

basándose en esta herramienta se desarrolló una versión mejorada de una parte del IDE NetBeans, permitiendo añadir componentes sin reiniciar la aplicación. En [10] se desarrolló una biblioteca de clases que genera el código necesario para que los programas escritos por los desarrolladores cumplan con los protocolos necesarios que establece cada *framework* que se necesita implementar, reduciendo costos al ahorrar el tiempo que invierte el desarrollador para aprender a utilizar cada *framework*.

A continuación se describen las herramientas de meta-programación que se encuentran bajo el enfoque Java: *Meta-AspectJ* es una herramienta de meta-programación que permite generar programas *AspectJ* sintácticamente correctos mediante plantillas de código, es una extensión de Java, por tanto es posible mezclar arbitrariamente código de Java con plantillas de código de *AspectJ* promoviendo una metodología que combina POA y PG, se utiliza para implementar generadores y de esa manera extender el poder de *AspectJ* [11]. *Jenerator* es un framework que provee mecanismos para crear clases, métodos, miembros e interfaces, permitiendo además modificar clases existentes e implementar conceptos abstractos como macros y aspectos [12]; sin embargo, *Meta-AspectJ* tiene un mayor soporte para la generación de código respecto a *Jenerator*, ya que al ser una extensión de *AspectJ* cuenta con todas las primitivas de corte y demás construcciones de alto nivel para la transformación de programas. *Spoonlet* es un componente de compilación distribuido como un paquete *.jar* el cual permite validar programas Java mediante análisis estático o transformar los programas mediante técnicas de PG y con base en una entrada produce código fuente adecuado para ser compilado; sin embargo, actualmente ya no se le da mantenimiento ni es compatible con versiones actuales de Eclipse [13]. *Javassist* es una herramienta de meta-programación que permite escribir meta-programas para definir clases automáticamente, simplificando la manipulación del bytecode, provee dos niveles de desarrollo: nivel de código fuente y nivel de bytecode [14].

4 Caso de estudio aplicando tecnologías Java

El caso de estudio consiste en agregar funcionalidad a un sistema con base en restricciones definidas en una especificación de requerimientos (intención). El sistema original por medio de clases y objetos representa la creación, el ordenamiento e introspección de elementos químicos con sus respectivas características. La nueva funcionalidad consiste en representar la creación generativa de moléculas utilizando los elementos químicos creados por el sistema original, tomando en cuenta las restricciones de composición de cada molécula. El diagrama de clases simplificado que representa la arquitectura general del sistema original se ilustra en la Fig. 1.

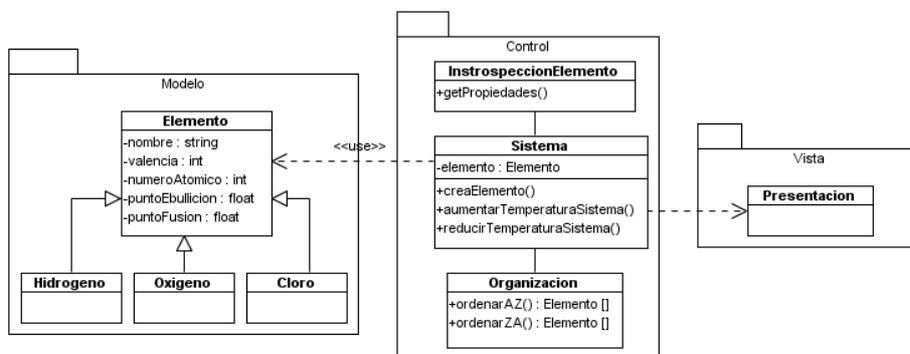


Fig. 1. Diagrama de clases del sistema original.

Para la implementación de la nueva funcionalidad se requieren mecanismos que permitan al usuario definir las restricciones de composición de las moléculas a generar, para esto se propone la utilización de *XML* (código 1). También se requiere que el sistema sea capaz de interpretar y adoptar las restricciones de composición establecidas; además es necesaria la obtención y manipulación de los elementos químicos que han sido creados por el sistema original, para esto se utiliza la exposición de contexto que ofrece *AspectJ* a través de su modelo de cortes. Otro mecanismo necesario es la generación e incorporación de las moléculas al sistema en ejecución, para esto se utilizan herramientas de meta-programación que permiten la generación de clases. Como se muestra en la Fig. 2, el esquema necesario para implementar la nueva funcionalidad utiliza los tres pilares de la programación generativa (POA, definición intencional y generación de código).

```

1 <Requerimiento>
2 <Componente nombre="Agua">
3 <Composicion>
4 <Elemento nombre="Hidrogeno" cantidad="2" valencia="1"></Elemento>
5 <Elemento nombre="Oxigeno" cantidad="1" valencia="-2"></Elemento>
6 </Composicion>
7 </Componente>
8 </Requerimiento>
    
```

Código 1. Restricciones de composición de la molécula de agua en el documento *XML*

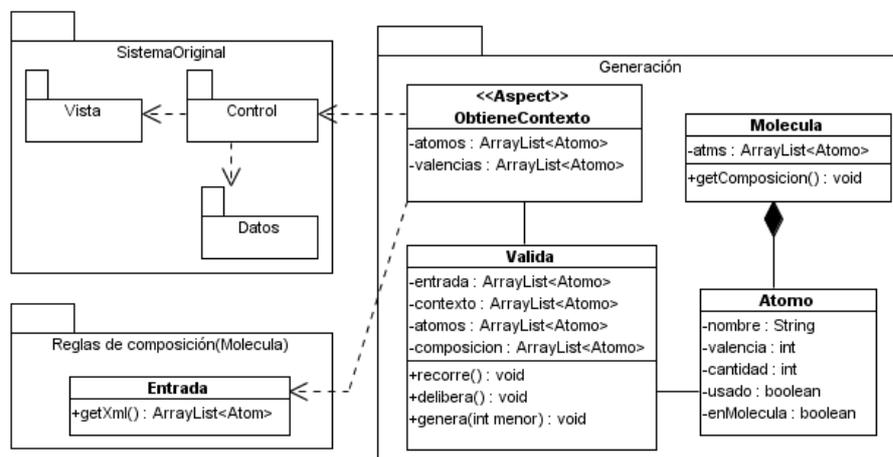


Fig. 2. Diagrama de clases para el subsistema de generación de las moléculas.

La clase “Molécula” que se debe generar consta de una estructura de datos que almacena los elementos químicos recibidos por el constructor y que son mostrados por el método “getComposicion()”. El mecanismo de generación de la clase “Molécula” se aplica utilizando distintas herramientas de meta-programación: *AspectJ*, *Javassist* y *MetaAspectJ*; con el objetivo de comparar cuál es la herramienta más adecuada dependiendo el tipo de problema a abordar.

4.1 Generación mediante *AspectJ*

Utilizando únicamente los mecanismos con los que cuenta *AspectJ* no es posible generar nuevos archivos *.class* por medio de instrumentación, sin embargo, el mecanismo de *introductions* permite definir nuevos elementos estructurales a una clase existente; en este caso como se muestra en el código 2, al sistema original se le agregan el campo, el constructor, y el comportamiento necesario que debe tener la clase *Molécula*.

```

1 public aspect GeneraMolécula{
2     ArrayList<Atomo>Sistema.atms;          //introducción del campo
3     public void Sistema.getComposicion(){ //introducción del método
4         System.out.println(atms);
5     }
6     public Sistema.new(ArrayList<Atomo> a){ //introducción
7         super();                          // del constructor
8         atms = a;
9         getComposicion();
10 }
11 }
    
```

Código 2. Introducciones de comportamiento de una molécula al sistema original

4.2 Generación mediante *Javassist*

La biblioteca de clases *Javassist* da un amplio soporte a la generación de clases en tiempo de ejecución; para la generación de la clase “Molécula” se utilizan construcciones de tipo *CtClass*, *CtField*, *CtConstructor* *CtMethod* que son representaciones que permiten definir clases, campos, constructores y métodos respectivamente. El código 3 muestra la generación de la clase molécula utilizando *Javassist*.

```

1 public static void genera() throws Exception{
2     ClassPool pool=ClassPool.getDefault();           //Representaciones de:
3     CtClass molecula=pool.makeClass("enJavassist.Molecula"); // Clase
4     CtField campo=CtField.make("Object [] atms;", molecula); // Campo
5     molecula.addField(campo);                       // Añade el campo a la clase
6     CtMethod metodo=CtNewMethod.make("public void ...",molecula); //Método
7     molecula.addMethod(metodo);                     // Añade el método a la clase
8     CtConstructor cons=CtNewConstructor.make("public Molecul...",molecula);
9     molecula.addConstructor(cons);                  // Añade el constructor a la clase
10    molecula.writeFile("bin/");                      // guarda el .class en disco
11 }

```

Código 3. Generación de la clase molécula mediante *Javassist*

4.3 Generación mediante *Meta-AspectJ*

Meta-AspectJ permite crear archivos *.class* por medio de la definición de tipos de árboles sintácticos que representan elementos del lenguaje Java “*quote (`)*”; además el elemento “*unquote (#)*” permite anexar un árbol sintáctico previamente definido a uno de mayor jerarquía, por ejemplo: añadir un método a una clase. El código 4 muestra la generación de la clase molécula utilizando *MetaAspectJ*.

```

1 public void genera() throws IOException{ // definiciones de:
2     infer pack=`[package metaAspectJ]; // paquete
3     infer campo=`[Object [] atms]; // campo
4     infer sentencia=`[for(int i=0; i<atms.length; i++) {...}];
5     infer metodo=`[public void getComposicion() {#sentencia}]; //método
6     infer clase=`[ #pack
7         public class Molecula{#campo //clase
8             public Molecula(Object[]a){atms = a;...} //constructor
9             #metodo
10        }];
11    String prog=clase.unparse();

```

```

12  FileWriter fw=new FileWriter ("C:/CasoDeEstudio/bin/metaAspectJ");
13  fw.write(prog); fw.close();
14  }

```

Código. 4. Generación de la clase molécula mediante *MetaAspectJ*

5 Resultados

La transformación del sistema se realizó en la etapa de mantenimiento, agregando la nueva funcionalidad, ya sea por medio de la creación de una nueva clase “Molécula” a tiempo de ejecución (*Javassist*, *MetaAspectJ*) o agregando el comportamiento necesario a una clase definida previamente (*AspectJ*); del mismo modo, la definición en *XML* de las restricciones de composición (intención) para las moléculas a crear fue efectiva ya que se realizaron varios experimentos con distintas restricciones que resultaron exitosos.

En cada uno de los experimentos con las distintas herramientas de meta-programación utilizadas se aplicó el mismo mecanismo para obtener e implementar las reglas de composición definidas en el documento *XML*.

Las tres herramientas de meta-programación con las que se realizó el caso de estudio cuentan con mecanismos de verificación de correctitud del código generado; en *AspectJ* mediante el compilador *ajc*; en *Javassist* con un compilador intermedio que verifica la correctitud de los elementos generados, y *MetaAspectJ* a través de las estructuras sintácticas que se definen para la generación de código. La Tabla 1 muestra las fortalezas que presenta cada herramienta de meta-programación en el aspecto generativo.

Tabla 1. Comparativa de herramientas de meta-programación

	AspectJ	Javassist	MAJ
Manejo de anotaciones	X	X	
Manejo de genéricos	X	X	
Representación de estructuras Java		X	X
Representaciones de estructuras de aspectos			X
Parametrización de elementos Java/AspectJ			X
Instrumentación de bytecode		X	

6 Discusión

El área de investigación y aplicación de la programación generativa es muy amplia, ya que incluye técnicas de programación orientada a aspectos, reflexión, programación intencional, generadores, componentes de software, entre otras. Sin embargo en el área de evolución de aplicaciones de software bajo el enfoque Java se alcanza a visualizar lo si-

guiente: Es necesario contar con mecanismos para la revisión y optimización del código que se modifica o genera, ya que implementar modificaciones a sistemas construidos previamente puede desencadenar en problemas a corto o largo plazo en términos de rendimiento o funcionalidad. También es necesario contar con un correcto y sistematizado control de versiones de los componentes que se utilicen, modifiquen y se produzcan a lo largo del proceso de evolución de un sistema de software, esto con el objetivo de tener un control de cada cambio que se realiza a cada componente y al sistema en su totalidad permitiendo obtener una traza completa de las modificaciones realizadas. Aunado a esto se aprecia la necesidad de un lenguaje que se especialice en la generación de programas y soporte las nuevas versiones del lenguaje Java.

Como se aprecia en la Fig. 3, la transformación de programas aplicando técnicas de PG en el proceso de desarrollo de software se realiza en la etapa de mantenimiento. Al presentarse requerimientos nuevos o cambiantes, se realizan las modificaciones necesarias o se construyen nuevos elementos en tiempo de ejecución para agregar la nueva funcionalidad esperada realizando la menor cantidad de cambios posibles al sistema original.

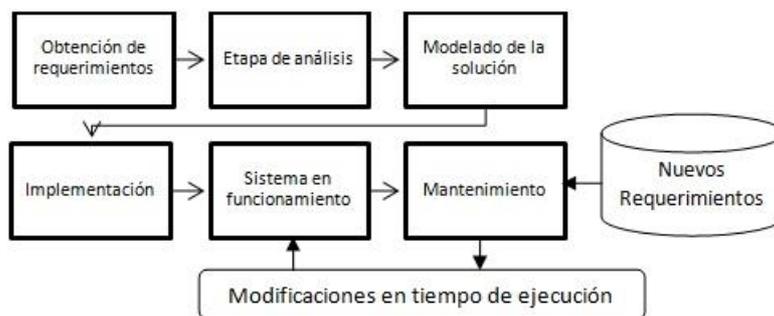


Fig. 3. Transformación de programas en el ciclo de vida de software.

7 Conclusiones y trabajo a futuro

La importancia de contar con un soporte de transformación de programas radica en la necesidad de implementar de manera natural, sencilla y lo más automáticamente posible nuevos requerimientos que sean necesarios a un sistema en la fase de mantenimiento de la ingeniería de software.

La generación y transformación de programas bajo el enfoque Java es posible gracias a las distintas herramientas de meta-programación que se reportan, sin embargo, es necesario contar con un mayor soporte que permita al programador una implementación transparente y lo más automatizada posible.

Los resultados obtenidos permiten detectar las capacidades de cada herramienta de meta-programación presentada, con esto es posible detectar, dependiendo la situación, la

herramienta o conjunto de herramientas necesarias para la transformación y generación de programas.

Como trabajo a futuro se considera realizar otros casos de estudio controlados y puntuales en la generación de programas para determinar los alcances de cada herramienta de meta-programación en el desarrollo de software.

Agradecimientos

Este trabajo cuenta con apoyo por parte del Consejo Nacional de Ciencia y Tecnología (CONACYT).

Referencias

1. Vermolen Sander D, Wachsmuth Guido and VisserEelco "Generating Database Migrations for Evolving Web Applications" in GPCE'11, Oregon, USA, pp. 83-92. (2011)
2. Li Yulin and Novak Jr. Gordon S."Generation of Geometric Programs Specified by Diagrams" in GPCE'11, Oregon, USA, pp. 63-72.(2011)
3. Esmailsabzali Shahram, Fischer Bernd and Atlee Joanne M. "Monitoring aspects for the customization of automatically generated code for big-step models" in 10th ACM international conference on Generative programming and component engineering, New York, USA, pp. 117-126 (2011)
4. Max Schlee and Jean Vanderdonckt, "Generative Programming of graphical user interfaces," in proceedings of the Working conference on Advanced visual interfaces, New York, pp. 403-406, ISBN:1-58113-867-9 (2004)
5. Ofenbeck Georg [et al.]"Spiral in Scala: Towards the Systematic Construction of Generators for Performance Libraries" in GPCE '13, Indianapolis USA, pp. 125-134 (2013)
6. Passos Leonardo, Krzysztof Czarnecki, Apel Sven, Wasowski Andrej et al, "Feature-Oriented Software Evolution" in proceedings of The Seventh International Workshop on Variability Modelling of Software-intensive Systems, Italy (2013)
7. Thomas Wurthinger, Christian Wimmer, and Lukas Stadler, "Dynamic code evolution for Java," in proceedings of the 8th International Conference on the Principles and Practice of Programming in Java, pp. 10-19, New York (2010)
8. Victor L. Winter and Azamat Mametjanov, "Generative programming techniques for Java library migration," in proceedings of the 6th international conference on Generative programming and component , 185 – 196, New York (2007)
9. Wurthinger Thomas [et al.] Applications of Enhanced Dynamic Code Evolution for Java in GUI Development and Dynamic Aspect-Oriented Programming in GPCE'10, pp. 123-126, Eindhoven, Holanda, (2010)
10. Chiba Shigeru, "Generative programming from a post object-oriented programming viewpoint," in proceedings of the international conference on Unconventional Programming Paradigms, pp. 355-366, Berlin (2004)
11. Huang Shan Shan, Zook David and Smaragdakis Yannis "Domain-specific languages and program generation with meta-AspectJ" in ACM Trans. Softw. Eng. and Methodology, Vol. 18-2, pp. 32, New York, USA, November (2008)

12. Volter M. and Gartner A. "Jenerator - Generative Programming for Java" in OOPSLA Workshop on Generative Programming, Tampa Bay, Florida (2001)
13. INRIA. Tutorial: Enhancing Java with Spoon. [Online]. <http://spoon.gforge.inria.fr/TutorialJDT/TutorialJDT>, (2013)
14. Chiba Shigeru, "Javassist—a reflection-based programming wizard for Java" in OOPSLA'98 Workshop on Reflective Programming in C++ and Java, pp. 92-115, Vancouver, BC, Canada (1998)

SISFIUX: adaptación de Feature-driven Development para el desarrollo de un sistema financiero para una universidad

César Ricardo Alducin Ruiz, Jorge Octavio Ocharán Hernández, Lizbeth A. Hernández González

Facultad de Estadística e Informática, Universidad Veracruzana
Av. Xalapa Esq. Manuel Ávila Camacho s/n CP. 91020

zs12015283@estudiantes.uv.mx, jocharan@uv.mx, lizhemandez@uv.mx

Resumen. El papel de las tecnologías de información en la economía actual ha crecido en importancia. Así, las organizaciones desarrolladoras de software contemporáneas se enfrentan a ambientes de negocio altamente cambiantes y a un incremento tanto en la complejidad de las tecnologías utilizadas para el desarrollo de software de calidad así como en la resolución de los puntos de vista de los diferentes interesados en el sistema. Por lo anterior, es deseable que los métodos de desarrollo de software sean adaptables para poder magnificar los beneficios para la organización a la que van dirigidas dichos sistemas. El presente trabajo describe la adopción y adaptación del enfoque de desarrollo basado en características tomando un sistema financiero como caso de estudio. La construcción de un modelo de dominio, la creación de la lista de características, la planeación del proyecto, así como el diseño y la construcción del sistema utilizando este método ágil son descritas y discutidas. Se concluye con una serie de lecciones aprendidas a partir de esta experiencia, las cuales buscan contribuir en la adopción exitosa de este tipo de métodos en proyectos futuros.

Palabras clave. Enfoque de desarrollo basado en características, Feature-Driven Development, FDD, desarrollo ágil de software.

1 Introducción

En desarrollo de software actual se encuentra ante diferentes retos. Los ambientes de negocio son altamente cambiantes, la complejidad del software y las tecnologías que se utilizan para desarrollarlo han aumentado tanto en escala como en alcance y se tienen que considerar los puntos de vista y necesidades, a veces en conflicto, de un mayor número de interesados (*stakeholders*) en el software [1]. Por otro lado, las metodologías de desarrollo de software guiadas por planes (*plan-driven development methodologies*) dificultan la

respuesta a los cambios en el entorno de negocio pues estas metodologías requieren de un conjunto completo de requerimientos de software que se anticipa de manera temprana y se trata de reducir el costo al eliminar el cambio en lugar de administrarlo durante el desarrollo y beneficiarse de él [2]. Ante tal situación, 17 profesionales destacados de la industria del desarrollo de software se reunieron en 2001 para discutir la situación y publicaron lo que se conocería como el “Manifiesto por el desarrollo ágil de software” [3]. En el manifiesto se definieron los valores que sustentan a los entonces nacientes métodos ágiles de desarrollo de software. Jim Highsmith define a la agilidad como la habilidad de crear y responder al cambio para obtener beneficios en un ambiente de negocios turbulento [4], haciendo así énfasis en el alcance de los objetivos de negocio de una organización.

A la fecha, existen diferentes métodos de desarrollo de software que se consideran ágiles de acuerdo a la literatura, destacando entre ellos *eXtreme Programming* (XP) [5], *Scrum* [6], *Dynamic Systems Development Method* (DSDM) [7], [8], *Adaptive Software Development* (ASD) [9], *Lean Software Development* [10], *Crystal Methods* [11] y *Feature-driven Development* (FDD) [12], [13]. Los métodos ágiles de desarrollo de software han sido adoptados incluso en grandes empresas desarrolladoras de software como Google, IBM, Microsoft, Nokia o Yahoo! [14]–[17] y ha sido objeto de estudio de diferentes investigadores en los últimos años [18]. Para el caso de México, además de la evidencia anecdótica, existen pocos trabajos que den cuenta de la adopción de los métodos ágiles para el desarrollo de software por parte de la industria y la academia [19] [20] [21] y para el caso particular de FDD aún es menor [22].

En el caso de estudio que aquí se presenta, se adoptó FDD para el desarrollo de un sistema financiero para una universidad. Para este proyecto, fue necesario adaptar algunos de los procesos que proponen los autores de este método de desarrollo ágil debido a la misma naturaleza del proyecto. Además de la adaptación de los procesos de FDD, se presenta el proceso de planeación para el desarrollo del Sistema Financiero para la Universidad de Xalapa (SISFIUX), el cual permitió asignar a cada una de las etapas el tiempo necesario para desarrollar la arquitectura de la aplicación, la lista de requerimientos, el modelo del dominio y el diagrama de clases, entre otros artefactos necesarios.

2 Contexto

La Universidad de Xalapa (UX) es una institución de educación superior privada. A nivel administrativo, la UX se encuentra dividida en diferentes departamentos los cuales son: rectoría, finanzas, administración escolar, jefaturas de escuelas, ingresos y contabilidad. Dentro del departamento de ingresos, se realizan diferentes procesos y actividades tales como la administración de egresos e ingresos, las proyecciones financieras de ingresos y la generación de reportes. Cabe destacar que dichas actividades no se encuentran bien definidas y son semi-asistidas por computadora (mediante el uso de hojas de cálculo, principalmente) lo que origina que el personal invierta una cantidad de tiempo y esfuerzo considerable, presentándose además errores de doble captura haciendo que la información sea

inconsistente, no se tenga en el momento y formato adecuado y se dificulte la toma de decisiones por parte de los interesados (Rector, Vicerrector financiero y de planeación y Jefaturas de Ingresos, Contabilidad, Egresos y Presupuestos). Por lo anterior, el desarrollo de un sistema financiero que apoye en la realización de estos procesos y actividades de manera automatizada al generar las proyecciones financieras de la universidad era deseable. Los beneficios que se podrían obtener de este sistema son el manejo de egresos, la generación de reportes financieros y la proyección presupuestal de la universidad de una manera más eficiente lo que repercutirá en un mejor control del ingreso que se obtiene, el contar la información necesaria para la toma de decisiones de manera oportuna y en el formato adecuado.

3 Enfoque de desarrollo basado en características

El enfoque de desarrollo basado en características o FDD por sus siglas en Inglés, es un método ágil y adaptativo para el desarrollo de software creado por Peter Coad y Jeff De Luca a partir de su experiencia en el desarrollo de un sistema bancario en Singapur en el que el dominio del problema era muy complejo [12], [13] y posteriormente formalizado por Steve Palmer [13]. FDD es un método ágil para el desarrollo de software el cual propone una serie de procesos para el desarrollo en los que se enfatizan las iteraciones cortas, la calidad en cada uno de sus pasos, la entrega frecuente de resultados tangibles y un seguimiento preciso y significativo del proyecto.

FDD propone cinco procesos secuenciales con criterios de entrada, tareas y criterios de salida bien definidos los cuales guían a los interesados en el proyecto (*stakeholders*) desde el modelado hasta la liberación del sistema. FDD incluye seis roles principales y otros roles de soporte, artefactos, objetivos y prácticas que necesita el sistema para ser construido y liberado (modelado de objetos de dominio, desarrollo por características, propiedad individual del código por clases, equipos por características, inspecciones, construcciones continuas, administración de la configuración y visibilidad/reporte de resultados) [23].

3.1 Modelo de procesos del enfoque de desarrollo basado en características

FDD propone un modelo de procesos minimalista en el que únicamente en cinco procesos se diseña y construye el sistema. Los procesos son: Desarrollar un modelo general, Construir una lista de características, Planear por característica, Diseñar por características y Construir por característica [23]. Cada uno de los procesos son representados en una descripción de una página utilizando el patrón Criterio de entrada-Tareas-Verificación-Criterio de salida.

Proceso 1: desarrollar de un modelo general

Se desarrolla un modelo general del dominio de la aplicación, el cual es realizado por los expertos del dominio y el arquitecto en jefe. En él, se busca establecer el alcance del sistema y contextualizar los requerimientos del cliente. FDD se apoya para este proceso de los denominados recorridos por el dominio o “*Walkthrough*”, el cual es descripción del dominio a alto nivel por parte de los expertos [23] y por el estudio de documentación relevante del dominio.

El resultado final que se obtiene de este proceso es un diagrama de clases con los tipos de objetos más significativos en el dominio del problema y las relaciones entre ellos complementado con un conjunto de diagramas de secuencia de alto nivel que muestran a los objetos interactuando unos con otros para cumplir sus responsabilidades [13]. La técnica que se sugiere para el desarrollo de este modelo general es “Modelando en color” [12] la cual usa cuatro arquetipos con diferentes colores para dividir las categorías de clases de la siguiente manera:

- Amarillo: Roles, usualmente una persona o una organización.
- Azul: Descripciones tipo catálogo.
- Verde: Una persona, lugar o cosa.
- Rosado: Un momento en el tiempo o un intervalo de tiempo asociado con un proceso del negocio.

Proceso 2: construir una lista de características

En este proceso se descompone la funcionalidad del dominio, a partir del modelo y diagramas creados en el Proceso 1, en un conjunto de características priorizadas. Una “característica” se define como una pieza de funcionalidad del sistema pequeña, entregable y de valor para el cliente que puede ser implementada en no más de dos semanas [13]. La división del dominio es desarrollada por el equipo de lista de características (expertos del dominio, programadores en jefe y arquitecto en jefe). Para la construcción de la lista de características se tiene que identificar las áreas del dominio, las actividades de negocio y los pasos de las actividades del negocio o características. Cada actividad del negocio puede estar constituida por un número variable de características. FDD recomienda un formato para la descripción de una característica el cual facilita su mapeo entre objetos y métodos [12].

Proceso 3: plan por características

La planeación de características incluye la creación de un plan de alto nivel en donde se asignan estas a los programadores en jefe, de acuerdo a su complejidad y dependencia entre ellas [23]. El equipo de planeación, el cual es integrado por el administrador del proyecto, el programador en jefe y los programadores, planean el orden en el cual se desarrollaran las características de acuerdo a dependencias, riegos, complejidad, balanceo de trabajo y prioridades del cliente. Posteriormente, las actividades son asignadas a equipos por características, liderados por un programador en jefe y cada clase en específico es

asignada a un desarrollador el cual a partir de ese momento será el responsable de dicha clase durante todo el desarrollo (*class owner*) [4].

Proceso 4: diseño por característica

A partir de la planeación por características, el programador en jefe y su equipo de característica produce los diagramas de secuencia para las características asignadas. Posteriormente el programador en jefe refina el modelo de objetos (dominio) con base en el contenido de los diagramas de secuencia al mismo tiempo que el equipo por características escribe las clases y prólogos de los métodos. Finalmente, se realiza una verificación del diseño a través de una inspección del diseño por parte de equipo de característica y el resto del equipo de desarrollo [4]. Hay que destacar que tanto este proceso como el Proceso 5 son iterativos.

Proceso 5: construcción de características

El equipo de características toma el paquete de diseño creado en el proceso anterior y realiza las siguientes actividades [4]: implementación de las clases y métodos, inspección del código producido, realización de pruebas de unidad y promoción de las clases para la construcción de una versión del sistema (*software build*). Al igual que el proceso anterior, se trata de un proceso iterativo.

4 Caso de estudio: SISFIUX

Se desarrollaron cada una de los procesos propuestos por FDD en el proyecto SISFIUX. En las siguientes subsecciones se habla a detalle de cada uno de los procesos, las tareas realizadas, los artefactos construidos así como las adaptaciones que se hicieron a este método de desarrollo ágil de software.

4.1 Modelo del dominio general

De acuerdo con FDD el primer proceso está dedicado al modelo de dominio, por lo que fue necesario realizar una reunión con los expertos del dominio y el arquitecto en jefe. Los procesos que se llevan a cabo en el departamento de ingresos de la Universidad de Xalapa que necesitan ser automatizados y que se toman como base para la elaboración del modelo de dominio general son: Proyecciones financieras de ingresos, Reportes de ingresos y Reportes de Vales (ver figura 1).

Cráterios de entrada

Los participantes para este proceso fueron son los siguientes:

- Expertos del dominio: Jefa del departamento de ingresos de la Universidad.
- Usuarios: Cajeras del departamentos de ingresos de la Universidad.

- Programador en jefe: Ingeniero en sistemas, encargado de adaptar FDD al desarrollo de SISFIUX.
- Dueño de las clases: Igual que el anterior.
- Desarrollador: Igual que el anterior.

Tareas

Formar el equipo de modelado: Para este caso de estudio se conformó de los expertos del dominio, programador jefe y desarrolladores y como un elemento externo el administrador del proyecto.

Recorrido del dominio: El experto del dominio explica el área en general que fue modelada al programador en jefe.

Estudiar los documentos: Los documentos que se estudiaron y que como se comento en la sección de contexto, son las hojas de calculo que ocupan los usuarios para realizar los reportes y proyecciones.

Desarrollar el modelo del dominio: El modelo inicial del dominio se desarrollo con la ayuda del experto del dominio y el arquitecto en jefe utilizando la técnica de modelado a color “UML-color”. Esta técnica permitió identificar los elementos del modelo de una manera mas entendible para el experto del dominio ya que esta técnica permite identificar roles, descripciones, momentos e intervalos y lugares o cosas[12].

Refinar el modelo del dominio: Se refinó el modelo de dominio considerando los comentarios del experto del dominio.

Escribir las notas del modelo: Se escribieron las notas correspondientes para refinar el modelo de dominio.

Verificación: Se realizó de manera conjunta entre experto del dominio y el equipo de modelado para verificar el modelo de dominio construido.

Criterios de salida: El resultado de este proceso es el modelo de dominio en forma de: Diagrama de clases, Métodos y atributos identificados y Diagramas de secuencia.

4.2 Lista de características

Una vez que se ha construido el modelo de dominio general, la siguiente fase es la de construir una lista de características, considerando el modelo de dominio se identifica que el sistema debe contar con los módulos principales tales como: proyecciones financieras, reportes de ingresos y reporte de vales. Para ello se construye la lista de características considerando áreas, subáreas y características.

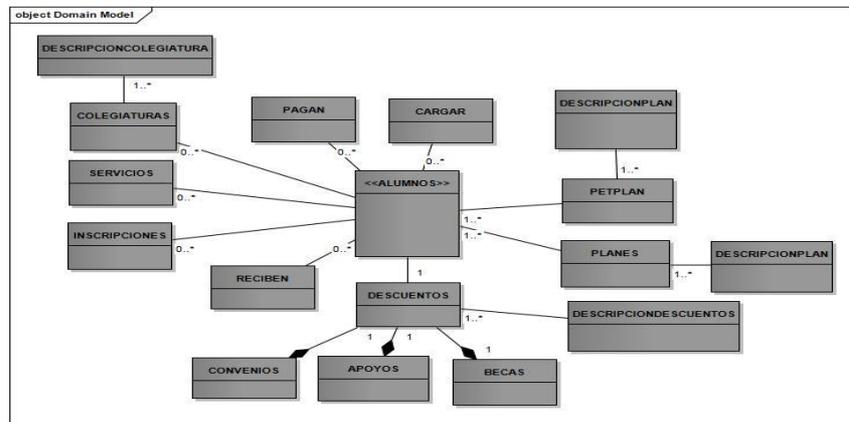


Fig. 1. Modelo del dominio parcial.

Criterios de entrada: Se seleccionaron los expertos del dominio, el programador en jefe y el arquitecto en jefe. En este caso fueron las mismas personas del Proceso 1.

Tareas:

Formar el equipo de características: el equipo de características se integro por el experto del dominio y el programador en jefe.

Construir la lista de características: se construye la lista de características bajo el siguiente formato establecido por parte del autor de FDD.

Ejemplo de una característica del sistema financiero:

“Calcular el total de pagos Semestre completo por modalidad”

Verificación: La verificación de la lista de características se llevó a cabo por parte del experto del dominio el cual también realizo una priorización de las características considerando aquellas que le son de mayor importancia:

Interna: verificación interna por parte del experto del dominio, quien apoyó al programador en jefe para refinar la lista de características y hacer los comentarios oportunos, esto permitió que la lista de características tenga validez, sea consistente, este completa, sea realista acorde con las necesidades del experto del dominio y los interesados y verificable.

Externa: verificación externa por parte del programador en jefe, para este caso de estudio ya no se tuvo que realizar una modificación al modelo de dominio.

Criterios de salida: el resultado del proceso es una lista de características por área, subáreas y actividades del negocio (ver figura 2).

ID	Características
1	Calcular el total de vales por subcuenta
2	Calcular el total de ingreso por folio de cobro
3	Calcular el total de vales por cuenta contable
4	Calcular el total de saldos de vales
5	Calcular el total de vales por unidad presupuestal
6	Calcular el total de vales pagados
7	Calcular el total de detalle de vales pagados
8	Calcular el total de vales de un usuario "X"
9	Calcular el total de abonos a los vales de un usuario "X"
10	Calcular el total de ingresos acumulados
11	Calcular el ingreso de las formas de pago por semana
12	Calcular el ingreso por forma de pago y por usuario
13	Calcular el flujo de efectivo diario
14	Calcular el total de la forma de pago efectivo
15	Calcular el total de la forma de pago tarjeta

Fig. 2. Lista parcial de características.

4.2.1 Plan por características

Una vez construida la lista de características se estableció un plan por características que consiste en calendarizar la forma en la que serán construidas estas. Como parte de las adaptaciones que se realizaron al proceso de FDD se adoptó SCRUM como marco de trabajo para la administración del proyecto mas adelante se detalla como se llevo a cabo esta adaptación.

Criterios de entrada: Lista de características construida en el proceso 2.

Tareas

Formar el equipo de planeación: Se integro por el programador en jefe y el administrador del proyecto.

Determinar la secuencia de desarrollo: El equipo de planeación consideró los siguientes elementos para determinar la secuencia en que serán construidas las características.

- Priorización del experto del dominio
- Dependencia entre características y clases
- Balanceo de carga de trabajo de los dueños de las clases
- La complejidad de las características

Asignar un conjunto de características a los programadores en jefe: El equipo de planeación asigna a los programadores en jefe las actividades del negocio basándose en la secuencia de desarrollo, dependencia entre características y la complejidad de las mismas. Hay que considerar que los programadores en jefe también son dueños de las clases y que posteriormente estos asignaran a los programadores.

Asignación de las clases a los desarrolladores: El equipo de planeación asignó las clases que se identificaron para ello se realizó un formato que se adapto a la metodología y que se muestra parcialmente en la tabla 1.

Cada característica no debe de llevar más de 2 semanas realizarla si eso pasara entonces hay que dividir la característica en otras más pequeñas. FDD propone que se refine el modelo de dominio general aún y cuando ya se encuentre en la fase de planeación. Por lo cual si el programador en jefe decide regresar al proceso uno esta en libertad de hacerlo. En la tabla 2 se observar el plan por características.

Verificación: El equipo de planeación verifico que los formatos reportados no presenten inconsistencias y sea claros para los miembros del equipo que llevó a cabo el proceso FDD.

Criterios de salida: Los criterios de salida que se tuvieron después de realizar este proceso FDD al proyecto, fue los formatos que me permiten saber cuales son las actividades del negocio, a quien fueron asignadas las clases, así como la calendarización de las mismas. Como parte de las adaptaciones que se realizaron a esta metodología en este proceso en particular es el uso de Scrum considerando el *product backlog* y los *sprints*, quedando de la siguiente manera: en el *product backlog* se integro el conjunto de características divididas por *sprints* y cada sprint tuvo una duración de 3 semanas calendario.

Tabla 1. Extracto de asignación de clases a los desarrolladores

ID	Descripción	Programadores en Jefe	Dueños de las clases
1	Calcular el total de vales por subcuenta	I. S. C. César Ricardo	I. S. C. César Ricardo
2	Calcular el total de ingreso por folio de cobro	I. S. C. César Ricardo	I. S. C. César Ricardo
3	Calcular el total de vales por cuenta contable	I. S. C. César Ricardo	I. S. C. César Ricardo
4	Calcular el total de saldos de vales	I. S. C. César Ricardo	I. S. C. César Ricardo
5	Calcular el total de vales por unidad presupuestal	I. S. C. César Ricardo	I. S. C. César Ricardo
6	Calcular el total de vales pagados	I. S. C. César Ricardo	I. S. C. César Ricardo
7	Calcular el total de detalle de vales pagados	I. S. C. César Ricardo	I. S. C. César Ricardo
8	Calcular el total de vales de un usuario "X"	I. S. C. César Ricardo	I. S. C. César Ricardo
9	Calcular el total de abonos a los vales de un usuario "X"	I. S. C. César Ricardo	I. S. C. César Ricardo
10	Calcular el total de ingresos acumulados	I. S. C. César Ricardo	I. S. C. César Ricardo
11	Calcular el ingreso de las formas de pago por semana	I. S. C. César Ricardo	I. S. C. César Ricardo
12	Calcular el ingreso por forma de pago y por usuario	I. S. C. César Ricardo	I. S. C. César Ricardo
13	Calcular el flujo de efectivo diario	I. S. C. César Ricardo	I. S. C. César Ricardo
14	Calcular el total de la forma de pago efectivo	I. S. C. César Ricardo	I. S. C. César Ricardo
15	Calcular el total de la forma de pago tarjeta	I. S. C. César Ricardo	I. S. C. César Ricardo

Tabla2. Extracto del plan parcial por Características

ID	Descripción	Diseño		Inspección del Diseño	
		Plan	Actual	Plan	Actual
1	Calcular el total de vales por subcuenta	16/09/2013	01/03/2014	01/04/2014	15/04/2014
2	Calcular el total de ingreso por folio de cobro	16/09/2013	01/03/2014	01/04/2014	15/04/2014
3	Calcular el total de vales por cuenta contable	16/09/2013	01/03/2014	01/04/2014	15/04/2014
4	Calcular el total de saldos de vales	16/09/2013	01/03/2014	01/04/2014	15/04/2014
5	Calcular el total de vales por unidad presupuestal	16/09/2013	01/03/2014	01/04/2014	15/04/2014
6	Calcular el total de vales pagados	16/09/2013	01/03/2014	01/04/2014	15/04/2014
7	Calcular el total de detalle de vales pagados	16/09/2013	01/03/2014	01/04/2014	15/04/2014
8	Calcular el total de vales de un usuario "X"	16/09/2013	01/03/2014	01/04/2014	15/04/2014
9	Calcular el total de abonos a los vales de un usuario "X"	16/09/2013	01/03/2014	01/04/2014	15/04/2014
10	Calcular el total de ingresos acumulados	16/09/2013	01/03/2014	01/04/2014	15/04/2014
11	Calcular el ingreso de las formas de pago por semana	16/09/2013	01/03/2014	01/04/2014	15/04/2014
12	Calcular el ingreso por forma de pago y por usuario	16/09/2013	01/03/2014	01/04/2014	15/04/2014
13	Calcular el flujo de efectivo diario	16/09/2013	01/03/2014	01/04/2014	15/04/2014
14	Calcular el total de la forma de pago efectivo	16/09/2013	01/03/2014	01/04/2014	15/04/2014
15	Calcular el total de la forma de pago tarjeta	16/09/2013	01/03/2014	01/04/2014	15/04/2014

4.3 Diseño por características

Una vez construido el plan por características se realizó un diseño por características el cual consistió en que el programador en jefe seleccionara un conjunto de características para formar un paquete de trabajo, para ellos se crearon los paquetes de la siguiente manera: Proyecciones financieras, Reportes de egresos y Reporte de ingresos.

Criterios de entrada: El proceso de planeación se completó correctamente en el proceso anterior.

Tareas

Formar el equipo de características: El programador en jefe identificó el conjunto de características que deberán ser desarrolladas y que tienen en común clases para conformar un diseño de paquete.

Revisión del modelo de dominio: El experto del dominio revisó el modelo del dominio para identificar las clases que serán construidas, esta tarea solo es para que el experto esté enterado de cuáles son las características que serán diseñadas.

Estudiar los documentos de referencia: los documentos que se estudiaron y que como se comentó en la sección de contexto, son las hojas de cálculo que ocupan los usuarios para realizar los reportes y proyecciones.

Desarrollar los diagramas de secuencia: se desarrollan los diagramas de secuencia por diseño de paquete para la construcción de una o de un conjunto de características.

Refinar el modelo de dominio: no se tuvo que refinar el modelo del dominio, en este proceso cuatro.

Escribir los prólogos de las clases y métodos: los desarrolladores escribieron los prólogos de las clases y métodos utilizando una herramienta para la construcción de software (IDE Netbeans).

Verificación: La decisión de inspeccionar el diseño de las características es del programador en jefe, para este caso de estudio se realizó esta tarea para evitar incongruencias en la lista y planeación de características.

Criterios de salida: El resultado de este proceso es el diseño por paquetes de las características que se van a construir.

4.4 Construcción por características

Una vez diseñadas e identificadas las características en el modelo de dominio se comenzó con la programación de las clases que fueron asignadas. Los programadores desarrollan las características con las tecnologías que se decidieron implementar, el programador aquí encuentra una relación entre las clases y características, ya que una clase se conforma de una o más características.

Criterios de entrada: El diseño por características ha sido completado e inspeccionado por parte del programador en jefe.

Tareas

Implementación de clases y métodos: Se realizó la implementación utilizando tecnología Java, por ser orientado a objetos y el manejo de clases.

Inspección del código: Inspección de código por parte del programador en jefe.

Pruebas de unidad: Se realizaron las pruebas de unidad utilizando JUnit, por cada característica construida se realizó su respectiva prueba de unidad.

Promoción de la característica: Una vez que se construía una característica, se verificaba y se validaba con el cliente, la clase se promovía para formar parte de la siguiente versión del sistema.

Verificación

Pruebas de unidad: Se utilizó JUnit para esta tarea y un plan de pruebas de unidad.

Pruebas de integración: Se utilizó un plan de pruebas de integración.

Pruebas de aceptación de usuario: Se utilizó un plan de pruebas de aceptación.

Criterios de salida: El resultado de este proceso fue: Clases y métodos exitosos con código inspeccionado, características promovidas por el experto del dominio y Características evaluadas por el cliente y usuarios finales.

5 Lecciones aprendidas

El aplicar esta metodología ágil en un contexto en el cual no se ha aplicado algún proceso de la Ingeniería de Software de manera formal, implica establecer procesos para la definición de las actividades que se realizan. Guiar a un equipo de desarrollo para la construcción de un software a través de un proceso definido como FDD permite establecer una forma de trabajo en el cual los resultados se ven tangibles en un corto plazo. Seguir una metodología ágil implica generar los artefactos y actividades propuestas por esta de manera sistemática, además de lograr una implementación que apoye el desarrollo de un software que cumpla con los requerimientos del cliente y de los estándares definidos.

6 Conclusiones y trabajo futuro

En el presente artículo se describe la adaptación de FDD para el desarrollo del SISFIUX. En la adaptación de esta metodología ágil, se tuvieron que realizar adaptaciones para lograr producir un software que cumpliera con los requerimientos del cliente. Para lograr la adaptación de FDD se tuvo que conformar un equipo multidisciplinario, lo que permitió contar con un conjunto de opiniones que permitieron un mejor desarrollo del software.

El programador en jefe, los dueños de las clases y los programadores tuvieron que estudiar los documentos necesarios para un mejor entendimiento del sistema a desarrollar, conocer los procesos y ayudar a sistematizar estos al personal del departamento de ingresos de la Universidad es una de las lecciones que deja la adaptación de FDD. Las adaptaciones a esta metodología para complementar cada una de las actividades que propone FDD, permitió generar nuevos artefactos complementarios a los FDD. El presente proyecto al ser de naturaleza transaccional se adaptó a la metodología porque precisamente el origen de FDD proviene de un proyecto financiero.

SISFIUX permite la elaboración de proyecciones financieras para conocer los ingresos que se percibirán por conceptos de pago de servicios en los distintos niveles educativos de la UX. Gracias a la implementación de FDD para el desarrollo de SISFIUX se logró poner énfasis en las etapas de diseño y construcción, ya que son las que principalmente promueve esta metodología ágil. SISFIUX permite además de la generación automática de los procesos antes mencionados, contar con un sistema que está desarrollado aplicando las técnicas y herramientas de la Ingeniería de Software.

Al ser un sistema financiero, las pruebas jugaron un papel muy importante ya que nos permiten saber que el software que se está desarrollando cumple tanto con los requerimientos del cliente. Es por ello que se realizaron pruebas de unidad en cada construcción de la lista de características utilizando pruebas de unidad, pruebas de integración y pruebas de aceptación de usuario.

Referencias

- [1] S. Nerur, A. Cannon, V. Balijepally, and P. Bond, Towards an understanding of the conceptual underpinnings of agile development methodologies, in *Agile Software Development*, Springer, pp. 15–29, (2010).
- [2] J. Highsmith, A. Cockburn, C. Consortium, and A. Cockburn, Agile software development: the business of innovation. *Computer* (Long Beach, Calif.), vol. 34, no. 9, pp. 120–127, Sep.(2001).
- [3] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mallor, K. Shwaber, and J. Sutherland. *The Agile Manifesto* (2001).
- [4] J. Highsmith, *Agile software development ecosystems*. Addison-Wesley Professional, 2002.
- [5] K. Beck, *Extreme Programming Explained: Embrace Change*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., (2000).
- [6] K. Schwaber and M. Beedle, *Agile software development with scrum*. Pearson Education International, (2002).
- [7] J. Stapleton, *DSDM, Dynamic Systems Development Method: The Method in Practice*. Addison-Wesley, (1997).
- [8] J. Stapleton, *DSDM: Business focused development*. Pearson Education, (2003).
- [9] J. A. Highsmith, *Adaptive Software Development: A Collaborative Approach to Managing Complex Systems*. Dorset House Pub., (2000).
- [10] M. Poppendieck and T. Poppendieck, *Lean Software Development: An Agile Toolkit*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., (2003).
- [11] A. Cockburn, *Crystal clear: a human-powered methodology for small teams*. Pearson Education, (2004).
- [12] B. P. Coad, E. Lefebvre, and J. De Luca, *Java Modeling in Color with UML: Enterprise Components and Process*. (1999).

- [13] S. R. Palmer and M. Felsing, A practical guide to feature-driven development. Pearson Education, (2001).
- [14] J. Sutherland and K. Schwaber, The scrum papers: nut, bolts, and origins of an Agile framework. 2001. [Online]. Available: <http://jeffsutherland.com/ScrumPapers.pdf>. [Accessed: 02-Jun-(2014)].
- [15] E. Woodward, S. Surdek, and M. Ganis, A practical guide to distributed Scrum. Pearson Education, (2010).
- [16] M.-W. Chung and B. Drummond. Agile at Yahoo! From the Trenches. In Agile Conference, 2009. AGILE '09., 2009, pp. 113–118, (2009).
- [17] M. Laanti, O. Salo, and P. Abrahamsson, Agile Methods Rapidly Replacing Traditional Methods at Nokia: A Survey of Opinions on Agile Transform5ation. *Inf. Softw. Technol.*, vol. 53, no. 3, pp. 276–290, (2011).
- [18] T. Dingsøy, S. Nerur, V. Balijepally, and N. B. Moe. A Decade of Agile Methodologies: Towards Explaining Agile Software Development. *J. Syst. Softw.*, vol. 85, no. 6, pp. 1213–1221, Jun. (2012).
- [19] J. A. Perez-Torres and M. Mejia. Software Development Using Agile Methodologies: An Airline Case. In *Proceedings of the Sixth Mexican International Conference on Computer Science*, pp. 129–135, (2005).
- [20] C. Enríquez-Ramírez and P. Gómez-Gil. Análisis empírico sobre la adopción de las metodologías ágiles en los equipos de desarrollo de software en empresas mexicanas. In *Tópicos selectos de tecnologías de la información y las comunicaciones. Memorias del Congreso Nacional y Congreso Internacional de Informática y Computación 2012*, (2012).
- [21] Aceves-Ortega H. M.: Agile Methods in Mexico. 2014. [Online]. Available: <http://www.scrumalliance.org/community/articles/2014/february/success-story-agile-methods-in-mexico>.
- [22] L. C. A. Gutiérrez, E. S. C. Castro, and M. R. Hurtado. A Mexican Experience Redesigning a Software Development Process Using XP, FDD and RUP.
- [23] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta, Agile software development methods: Review and analysis, (2002).

Desarrollo de la versión móvil para la red social FEIBook

Luis Abraham Zavaleta Ibarra, Gerardo Contreras Vega, Karen Cortés Verdín, Juan Carlos Pérez Arriaga

Universidad Veracruzana, Facultad de Estadística e Informática,
Xalapa, Ver., México

luis.ibarra310@gmail.com, gcontreras@uv.mx, kcortes@uv.mx, juaperez@uv.mx

Resumen. Las redes sociales han transformado la manera en como las personas se comunican y se relacionan en la actualidad. Su impacto ha sido tan grande que cada vez es más común observar a las personas haciendo uso de las redes sociales. Actualmente se tiene claro que las redes sociales han llegado para quedarse y que su presencia es cada vez más habitual, pero el acceso a las redes sociales a través de los dispositivos móviles dentro del ámbito educativo es un tema que muy pocos han abordado en la actualidad. Por otra parte, hoy en día los dispositivos móviles ofrecen a los usuarios muchas funcionalidades y características que hace pocos años solo era posible encontrar en equipos más complejos y costoso, como las computadoras de escritorio o laptops. Haciendo uso de las redes sociales en combinación con tecnología móvil es posible crear aplicaciones para su uso en el ámbito universitario. La Facultad de Estadística e Informática de la Universidad Veracruzana desarrolló recientemente una red social para el seguimiento de egresados denominada FEIBook. Con esta red se busca mantener contacto con los egresados y que éstos proporcionen información de su desempeño laboral para realizar los estudios que correspondan. Es así, que en búsqueda de obtener una mayor participación en la red social, se desarrolla la versión móvil de dicha red.

Palabras Clave: Redes sociales, Seguimiento de Egresados, Android, Servicios Web, Móviles.

1 Introducción

El estudio de seguimiento de egresados es una estrategia evaluativa que posibilita conocer la situación profesional o laboral de un egresado de una carrera profesional con la finalidad del mejoramiento institucional [1]. Los estudios de seguimiento de egresados son importantes ya que permiten conocer la situación profesional del egresado. Con esta información se determina la pertinencia del programa educativo, se actualizan los progra-

mas de estudio y se evalúa la calidad de las estrategias educativas y del cuerpo docente de la institución.

Estudios realizados en el año 2012 por la *Salzburg Academy of Global Media Literacy* [12] revelan que México es el segundo país del mundo con más usuarios universitarios conectados en redes sociales, con un promedio de tres horas diarias. Se destaca que el 66% de los estudiantes reportan el uso de teléfonos celulares y teléfonos tipo *Smartphone* para conectarse a las redes sociales, bajando así el uso de computadoras de escritorio e incluso *laptops*. El artículo asegura que publicar estados y comentar actualizaciones y contenidos de amigos son las actividades más realizadas por los usuarios universitarios.

En la Facultad de Estadística e Informática (FEI) de la Universidad Veracruzana se ha pensado resolver la necesidad del seguimiento de egresados mediante el uso de redes sociales y de la tecnología móvil. Actualmente ya se tiene una red social, que lleva por nombre FEIBook [2][8][9]. Esta red social tiene como objetivo principal crear un vínculo con los egresados de la FEI para así crear un directorio actualizado de egresados. Algunas de las actividades que puede realizar actualmente un usuario de FEIBook son: compartir lo que están pensando, agregar fotografías, agregar o modificar su perfil de usuario, comentar una publicación, hacer amistad con usuarios de la red y enviar mensajes privados a sus amigos.

Uno de los problemas actuales de FEIBook es el desinterés de los egresados en actualizar sus datos y situación laboral. Es por esto que se decidió realizar una versión móvil de la actual red social y es así que se desarrolla una versión de FEIBook para dispositivos móviles que cuenten con Sistema Operativo Android [13].

Este documento presenta el desarrollo de FEIBook versión móvil y se encuentra organizado de la siguiente manera: el apartado dos describe las redes sociales móviles, tipos y su aplicación en diversos contextos. A continuación, se describe FEIBook. En el cuarto apartado se presenta el desarrollo de FEIBook móvil, incluyendo las pruebas realizadas. Por último, se presentan las conclusiones y se describen los trabajos futuros.

2 Redes sociales móviles

Las redes sociales móviles son un medio de comunicación que mezcla la tecnología móvil y la infraestructura de Internet. Las redes sociales móviles (*Mobile Social Networks*) o MSN hacen referencia a todos los elementos que permiten el desplazamiento, carga y consumo de los sitios de redes sociales alojados en la web a través de un dispositivo móvil. Las redes sociales móviles pueden ser divididas en cinco tipos diferentes[11]:

- Los Medios Sociales son redes sociales móviles no profesionales para subir fotos digitales, mensajes escritos y documentos (ej. *Blogs*), sonido y vídeo.
- Los Perfiles son páginas de contenido dinámico que pueden ser actualizados por el autor, mejorando así la interacción entre el usuario y la red social.

- Los Portales de Comunicación son aquellos destinados a que un usuario se suscriba a una aplicación para obtener el contenido de un grupo o comunidad.
- Las Comunidades, dentro del contexto de las redes sociales móviles, se definen como comunidades de lazos interpersonales que promueven la sociabilidad, apoyo, información, sentido de pertenencia e identidad social.
- La Mensajería Social para redes sociales móviles se refiere a una herramienta libre que permite a las personas intercambiar mensajes con grupos (comunidades) o particulares, en ocasiones combinando mensajes de texto plano, pero casi siempre utilizando la plataforma web y el navegador.

Con el fin de tener servicios que cubran las necesidades de los usuarios de redes sociales móviles y evitar las interfaces de usuario excesivamente complejas o redundantes, los creadores de las comunidades móviles deben identificar las necesidades de una sociedad. Por lo regular una red social móvil atiende un conjunto específico de necesidades, aunque en ocasiones los servicios de una RSM logran satisfacer más necesidades de una comunidad, esto es debido a los servicios que implementan [11].

El entretenimiento y la curiosidad son necesidades a cubrir en las RSM, mismas que se complementan con la necesidad de hacer amigos y pertenecer a un grupo. Algunas redes sociales móviles buscan o promueven la descarga de contenido, la compra o venta de artículos, la competencia entre los usuarios y en ocasiones generar ingresos directos o indirectos para los usuarios. Las comunidades de entretenimiento casi siempre permiten a los usuarios compartir sus experiencias y recomendaciones mediante votos o comentarios.

Una RSM puede ayudar a sus miembros a desarrollarse y crecer profesionalmente. Existen comunidades móviles diseñadas para el intercambio y presentación de información referente a la formación profesional y laboral. De esta manera las comunidades móviles cubren las necesidades profesionales de la comunidad. Un ejemplo es la red social LinkedIn.

3 FEIBook

FEIbook es la red social de seguimiento de egresados de la Facultad de Estadística e Informática [2][13]. Elgg [4] es el CMS que soporta el desarrollo de FEIBook. Elgg fue creado por Ben Wedmuller y David Tosh. Elgg crece y mejora gracias al desarrollo comunitario de programadores en todo el mundo. En agosto de 2008 recibió el premio como la mejor plataforma social de trabajo [10]. Elgg sigue el patrón Modelo Vista Controlador (MVC) [3]. En este patrón, el modelo es un conjunto de clases que define la forma en la que funciona la plataforma y almacena en la base de datos. La vista maneja la interfaz del usuario. El controlador es el intermediario entre la vista y el modelo, recibe las órdenes del usuario a través de la vista, traduce dichas instrucciones al modelo y recibe la respuesta de éste para enviarla a la vista.

Para el desarrollo de FEIBook, se adaptó Elgg conforme a los requisitos para la red social de la FEI: publicar fotos, publicar un cambio de estado, enviar mensajes privados, compartir archivos, chat entre usuarios de la red, crear y modificar perfil para cada usuario, crear grupos privados y compartir contenidos con otras redes sociales. La adaptación de Elgg se llevó a cabo mediante el desarrollo de *plugins*. Los *plugins* que son fáciles de incorporar gracias a la arquitectura tipo MVC de Elgg. Un *plugin* es una aplicación que se agrega o se elimina hasta obtener todas las funcionalidades deseadas de Elgg [6]. Todos los *plugins* que se anexan a Elgg también deben seguir el estilo MVC. A continuación, se explican cada uno de los elementos de Elgg.

3.1 El modelo de Elgg

El modelo de Elgg recibe el nombre de *core* Elgg o núcleo. Dicho *core* está diseñado en unidades atómicas llamadas unidades. Las clases más empeladas y las más importantes, son las siguientes [5]:

- ElggEntity: Representa cualquier elemento de la red social, se encarga de otorgar los permisos de acceso, almacena el usuario propietario de esa entidad, asigna el identificador principal.
- ElggObject: Esta entidad representa todos los objetos arbitrarios como: los archivos subidos, los favoritos de los usuarios, los mensajes publicados.
- ElggUser: Se encarga de los usuarios, sus datos y permisos.
- ElggSite: La entidad representa los datos de la red social como el nombre, la dirección Web, el dominio.
- ElggGroup: Controla los grupos de usuarios creados.
- Metadatos: Contiene información que describe un objeto por ejemplo las rutas de los archivos.
- Anotaciones: Contiene información general agregada por terceros, como lo son los comentarios de una publicación.
- ElggRelationship: Se encarga de unir las entidades de acuerdo a su interacción, por ejemplo la pertenencia de un usuario a un grupo.

3.2 La vista de Elgg

Las vistas de Elgg se encargan de mostrar desde el diseño de las páginas hasta los formularios en el navegador y también se pueden realizar vistas avanzadas. Algunos ejemplos de vistas avanzadas son [5]: HTML estándar, HTML optimizado para accesibilidad, *wid-gets* embebidos e Internet móvil.

3.3 El controlador de Elgg

El controlador dentro de Elgg corresponde a Elgg *Action*. Una acción es el código que se ejecuta cuando un usuario realiza algo dentro de Elgg[7]. Se considera acción a cualquier actividad que puede realizar el usuario como iniciar sesión, crear y/o eliminar un comentario. La acción procesa el formulario o petición enviada por el usuario, realiza los cambios necesarios en la base de datos y envía una respuesta al usuario, proporcionando de esta manera la interactividad del usuario con el modelo. Todas las acciones se desarrollan en scripts PHP por separado y el nombre del archivo debe ser del mismo nombre que la acción que puede realizar. Todas las acciones son ejecutadas por un controlador central llamado *action_handler*.

3.4 Elgg realización de *plugins*

El *core* de Elgg proporciona las funcionalidades básicas para una red social, sin embargo, el administrador tiene la posibilidad de construir la red social en base a sus necesidades, lo cual se logra mediante el uso de *plugins*. Los *plugins* permiten agregar nuevas funcionalidades, por ejemplo agregar un chat, un *blog* e inclusive cambiar toda la vista de Elgg. Estos *plugins* son reconocidos por el *core* después de colocarlos dentro del directorio */mod*, pero no es hasta que son configurados por el administrador que serán visibles en la red social. Los *plugins* son reconocidos por el *core* sólo si están desarrollados bajo las pautas designadas por Elgg [6] y si han sido creados siguiendo el patrón MVC. Las pautas para desarrollar un *plugin* para Elgg son las siguientes: 1) Conservar el esqueleto o estructura por defecto, el cual puede consultarse en la documentación de Elgg; 2) Conservar los nombres y 3) Seguir el MVC de Elgg.

4 Desarrollo de FEIBook móvil

La finalidad principal de hacer una versión móvil de la red social de la FEI es dar a los egresados la oportunidad de acceder desde cualquier dispositivo móvil con sistema operativo Android. Así, la FEI podrá contar con información necesaria para el seguimiento de egresados. El desarrollo de la versión móvil de FEIBook, consistió en la creación y uso del API FEIBook_WebServices para la comunicación entre el cliente y el servidor, así como en la creación de una aplicación móvil para los dispositivos con sistema operativo Android. A continuación, se explica con mayor detalle el desarrollo.

4.1 Plugin FEIBook_WebServices

Elgg permite exponer algunas funcionalidades para poder ser consumidas por aplicaciones de terceros. Por otra parte, dentro de la comunidad de Elgg encuentran algunos *plugins* de

Servicios Web que logran exponer algunas funcionalidades. Sin embargo, después de analizar y probar estos, se llegó a la conclusión que ninguno cubría las necesidades de FEIBook móvil o estaban creados para otras versiones de Elgg y no eran compatibles con FEIBook. Es así como se decide crear un *plugin* propio que cubra las necesidades requeridas y que sea compatible con la versión de Elgg 1.7.17 la cual está siendo usada en FEIBook. Este *plugin* se denomina FEIBook_Webservices.

Los Servicios Web surgen por la necesidad de ofrecer aplicaciones distribuidas haciendo uso de software que pueda ser ejecutado en diferentes sistemas operativos y arquitecturas, utilizando diferentes lenguajes de programación y herramientas.

Actualmente existen varias técnicas de arquitectura software para la creación de servicios web: RPC (*Remote Procedure Call*), SOA (*Service Oriented Architecture*) y REST (*REpresentational State Transfer*). Se elige REST para la implementación del *plugin* ya que se pueden trabajar sistemas de hipermedia distribuidos como la Web y por su funcionamiento independiente que garantiza la independencia entre servidores y clientes. REST, se basa en estándares como HTTP, URI y XML, entre otros.

FEIBook_WebServices, es un *plugin* que únicamente usa el controlador de Elgg. Es así, que sirve como una interfaz entre el cliente móvil y el servidor. De esta manera, recibe las peticiones del cliente y responde estas peticiones con base en los criterios establecidos por el propio Elgg *core*.

Los pasos seguidos para la construcción del *plugin* son los siguientes:

1. Se crea un nuevo directorio dentro del directorio *mod* de Elgg. El directorio creado lleva por nombre FEIBook_WebServices y con este nombre será identificado y reconocido por el motor de Elgg.
2. Se crea el archivo *manifest.xml* dentro del directorio FEIBook_WebServices, este archivo contiene información del *plugin* como nombre del *plugin*, autor, versión, licencia, entre otros datos que pueden ser añadidos.
3. Dentro del directorio FEIBook_WebServices se crea un nuevo archivo que lleva por nombre *start.php*. Este archivo contiene las funciones que serán expuestas mediante la función *expose_function()*, y que permite a otras aplicaciones interactuar con la red social.

Posteriormente, se debe comprobar que el núcleo de Elgg reconoce *plugin* y que fue habilitado correctamente. Para esto, se ingresa a FEIBook con una cuenta de administrador y, en la sección de herramientas de administración, se habilita el *plugin* FEIBook_WebServices que, por defecto, aparece deshabilitado. Entonces, Elgg notifica con un mensaje si el *plugin* fue reconocido y habilitado correctamente.

Al agregar el *plugin* FEIBook_WebServices se pone una capa al MVC de Elgg (FEIBook), modificando así su arquitectura. En la figura 1, en la siguiente página, se presenta la arquitectura modificada empleando un diagrama de paquetes de UML. Se puede observar como el *plugin* FEIBook_WebServices interactúa con el controlador, dando así acceso a la funcionalidad de FEIBook.

4.2 Aplicación FEIBook móvil

La aplicación FEIBook móvil es la responsable de consumir las funcionalidades expuestas por el *plugin* FEIBook_WebServices. Por cuestiones de tiempo, se opta por desarrollar esta aplicación móvil en Android, ya que es un sistema operativo popular con un SDK fácil de implementar y de utilizar.

Android es un sistema operativo libre que no necesita ningún tipo de hardware especial para desarrollar aplicaciones. Actualmente Android tiene mayor popularidad y demanda en comparación con iOS y es más común observar a estudiantes y maestros de la FEI utilizar dispositivos con esta plataforma.

Para que el usuario pueda acceder a las opciones ofrecidas por la aplicación móvil de FEIBook, debe hacer uso de un menú de navegación llamado *Navigation Drawer*, ubicado en el lado izquierdo de la pantalla. El menú de navegación de FEIBook móvil cuenta con las siguientes opciones:

- El Muro.
- Mi Perfil.
- Amigos.
- Buscar personas.
- Mensajes.
- Ayuda.
- Cerrar sesión.
- Notificaciones.

El intercambio de información entre la aplicación FEIBook y el servidor se realiza mediante el protocolo HTTP. Además, FEIBook móvil hace uso de SSL para mantener la seguridad cuando un usuario accede a la red social. Para poder hacer uso del certificado dentro de la aplicación móvil de FEIBook y tener una conexión segura se prepara el certificado y se añade a la aplicación, para que esta pueda hacer uso de él al momento de realizar una conexión HTTPS. FEIBook móvil conecta con el servidor de FEIBook para poder realizar las tareas que el usuario solicita. Además, realiza peticiones de manera periódica para poder notificar al usuario de los cambios y noticias más recientes en la red social. Esta comunicación se realiza a través de Internet ya sea mediante una conexión Wi-Fi o por datos móviles.

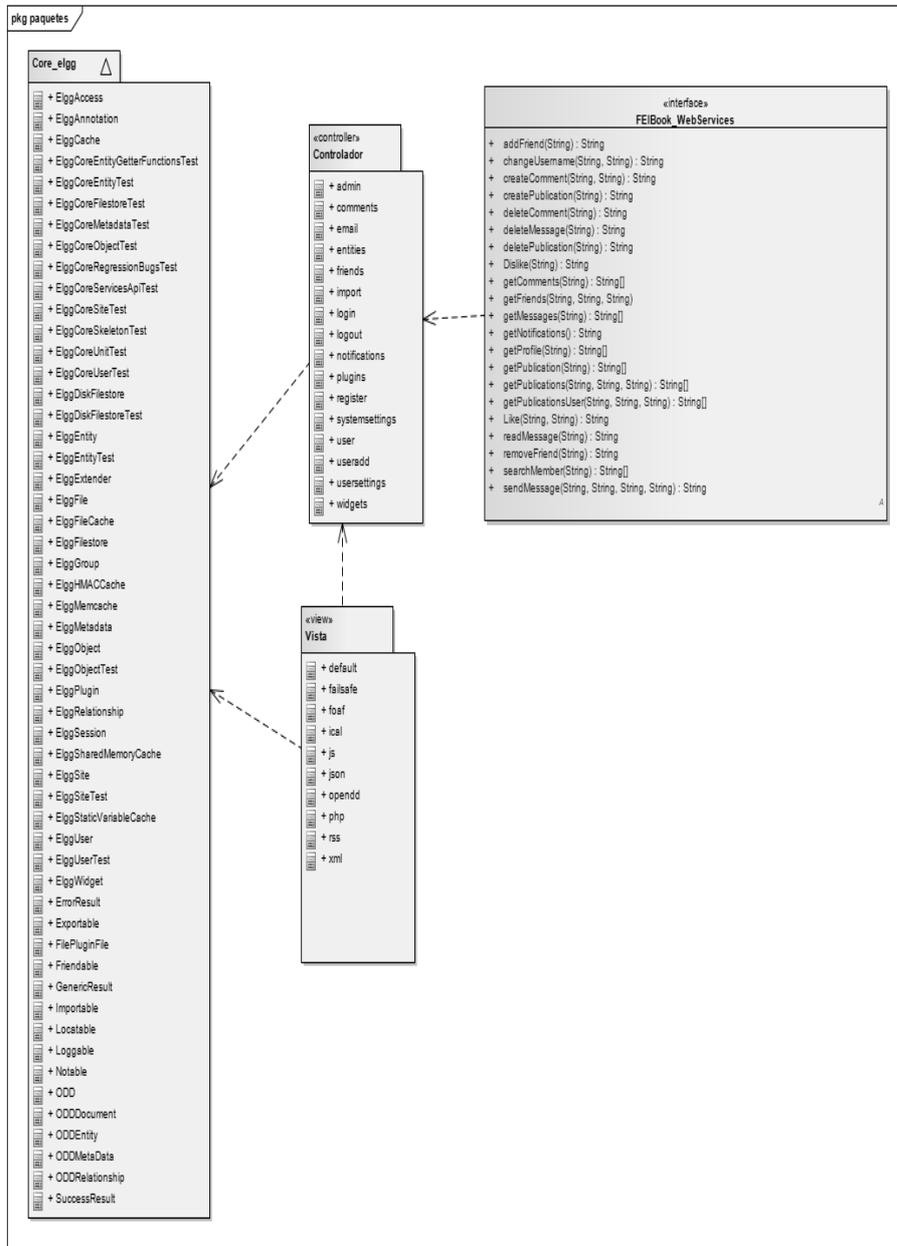


Fig. 1. Arquitectura de FEIBook modificada.

4.3 Pruebas

Las pruebas realizadas a FEIBook móvil se basan en casos de uso [13]. El proceso de pruebas consideró los tres pasos fundamentales a realizar: entorno, desarrollo y ejecución. En la tabla 1, abajo, se presentan los 11 casos de prueba de la aplicación, para realizar las pruebas, se subió a un servidor remoto una copia de la red FEIBook con las mismas características que la versión en producción. Participaron 15 estudiantes de la FEI, realizándose las pruebas en dos días. Cada estudiante contó con dos horas para realizar los casos de pruebas planteados. Las versiones de los sistemas operativos Android variaba de la 4.0.4 a la 4.4.2. La versión más común, con 7 dispositivos, fue la 4.1.2.

Tabla 1. Casos de prueba

ID	Nombre del caso de prueba	Propósito de la prueba
P1	Instalar la aplicación.	La aplicación FEIBook móvil puede ser descargada del servidor e instalada en el dispositivo móvil.
P2	Ingresar a FEIBook móvil con datos erróneos.	Se impide el acceso a usuarios que proporcionan datos incorrectos.
P3	Ingresar a FEIBook móvil.	Los usuarios ingresan cuando proporcionan nombre de usuario y contraseña correctos.
P4	Publicar en el muro.	La aplicación permite a los usuarios realizar publicaciones en el muro de FEIBook.
P5	Comentar a una publicación.	Los usuarios comentan las publicaciones del muro.
P6	Agregar a un usuario como amigo.	Buscar un miembro de la red social FEIBook y que es posible agregarlo a su lista de amigos.
P7	Enviar mensaje a un usuario.	Los usuarios pueden enviar mensajes privados a otros usuarios.
P8	Revisar mensaje privado.	El usuario puede revisar sus mensajes privados y puede responder a un mensaje enviado por otro usuario.
P9	Eliminar comentario de una publicación.	Comprobar que el usuario puede eliminar los comentarios hechos en una publicación
P10	Eliminar usuario de la lista de amigos.	La aplicación FEIBook permite a los usuarios eliminar usuario de FEIBook de su lista de amistades.
P11	Cerrar sesión en FEIBook móvil	Comprobar que el usuario puede cerrar la sesión creada en la aplicación móvil de FEIBook.

En la tabla 2, abajo, se listan los casos de prueba con resultados fallidos.

Tabla 2. Casos de prueba con resultados fallidos.

ID	# re-sultados fallidos	Comentarios
P4	1	Un usuario reportó: “La aplicación se detuvo y nunca cargó el muro. Después de volver a abrir la aplicación cargó el muro y se pudo publicar”.
P9	6	Seis usuarios reportaron el error: “La aplicación FEIBook se detuvo”. Un usuario comentó: “No aparece el cesto de basura” Un usuario comentó: “Se borró el comentario pero no hubo ningún diálogo”.
P10	9	Siete usuarios reportaron: “Se eliminó el amigo pero sigue apareciendo en mi lista de amigos, pero ya no puedo dar clic en el nombre.” Dos usuarios reportaron: “La aplicación FEIBook se detuvo”.

Para corregir los defectos encontrados en la aplicación, se revisaron los resultados fallidos de los casos de prueba, así como la información de depuración generada por la plataforma Android en los dispositivos de los estudiantes, dando como resultado las siguientes actividades para realizar las correcciones:

- Para evitar los problemas de carga de la aplicación, se realizó la optimización de los recursos gráficos utilizados por la plataforma disminuyendo el peso de los mismos.
- La detención abrupta de la aplicación se solucionó disminuyendo la instanciación de objetos innecesarios.
- Para corregir el defecto de la lista de amigos, se implementó la notificación de cambios a la lista de amigos en la aplicación para refrescar la lista y no mostrar los amigos eliminados.

5 Conclusiones

En este trabajo se presentó el desarrollo de FEIBook versión móvil. FEIBook es una red social construida para llevar a cabo el seguimiento de egresados de los programas académicos de la Facultad de Estadística e Informática. Esta red social fue desarrollada empleando Elgg, un CMS que sigue el patrón MVC. Con el fin de aprovechar el acceso a dispositivos móviles que actualmente se tiene y facilitar así la actualización de los datos de los egresados en la red social, se propone realizar la versión móvil de FEIBook para el

sistema operativo Android. Se busca entonces alguna manera de aprovechar la arquitectura tipo MVC de FEIBook, exponiendo los métodos a través de algún tipo de interfaz. Aunque Elgg cuenta con un mecanismo para exponer sus servicios, y existen comunidades con algunas soluciones, no existe alguna que sea compatible con FEIBook. Se construye entonces FEIBook_WebServices, un *plugin* (API REST) que se comunica directamente con el controlador de Elgg y permite hacer uso de toda a funcionalidad de la red social. Se desarrolla, además, la aplicación FEIBook móvil para Android, que usa los servicios expuestos por el *plugin*. Los resultados de las pruebas permiten asegurar que el *plugin* y la aplicación funcionan correctamente. La principal contribución de este trabajo es integrar las redes sociales y la tecnología móvil en una aplicación para el ámbito universitario. En [13] se presenta una revisión detallada de redes sociales en el ámbito universitario. Sin embargo, al momento de realizar este trabajo se desconoce la existencia de RSM en el ámbito universitario.

Los trabajos futuros propuestos a partir de este trabajo, son los siguientes:

En primer lugar mejorar la usabilidad de la interfaz gráfica de la aplicación. Aunque se diseñó y desarrolló una interfaz usable y accesible para los usuarios, la usabilidad no era un objetivo primario. Es así que, queda por realizar el diseño y desarrollo de una interfaz altamente usable para los usuarios de la red social.

Un segundo trabajo es que la aplicación móvil para Android permita subir y visualizar fotografías y videos así como interactuar en los grupos a los que se pertenece. Por último, queda pendiente realizar pruebas de seguridad, de desempeño y de resistencia que, por cuestiones de tiempo, no fue posible realizar en este trabajo.

Referencias

1. Barrón M.C.: Los estudios de seguimiento de egresados en el período 1992-2002, en Educación, trabajo, ciencia y tecnología, Colección la Investigación Educativa en México 1992-2002, libro 6. COMIE, México (2003)
2. Barrera Gallegos S.: Actualización de la Red Social de la Facultad de Estadística e Informática, Tesis de Licenciatura. Universidad Veracruzana, Xalapa, Ver. México (2013)
3. Buschmann, F., Rohnert, H., & Stal, M: Pattern-oriented software architecture. Volume 1, a system of patterns, J. Wiley and Sons. New York (1996)
4. Elgg <http://elgg.org>
5. Engine Elgg Documentation <http://docs.elgg.org/wiki/Engine/Views>
6. All site plugins, themes, languages packs: The Elgg Community <http://community.elgg.org/plugins>
7. Forms+actions, Elgg Documentation <http://learn.elgg.org/en/latest/guides/actions.html>
8. González Jiménez B.: ReSoSE: Red Social para el Seguimiento de Egresados, Tesis de Licenciatura. Universidad Veracruzana, Xalapa, Ver. México (2012)
9. González-Jiménez B., Gotreras-Vega G., Cortés-Verdín M.: Redes Sociales para el Seguimiento de Egresados. En: Research in Computer Science, Special Issue: Advances in Intelligent

- and Information Technologies, vol. 60, pp.239-249, Centro de Investigación en Computación, Instituto Politécnico Nacional. México (2012)
10. InfoWorld <http://www.infoworld.com/d/open-source/infoworld-announces-our-2008-best-open-source-awards-065>
 11. Lane, N.: Mobile Social Networking, White Paper (2009). http://www.telecoms.com/files/2009/05/buongiorno_final-fmt_nl-3110-f.pdf
 12. The Salzburg Academy on Media and Global Change. <http://www.salzburgglobal.org/current/sa-b.cfm>
 13. Zavaleta Ibarra L. A.: Desarrollo de FEIBook versión móvil, Tesis de Licenciatura. Universidad Veracruzana, Xalapa, Ver. México (2014)

Desarrollo de una arquitectura orientada a servicios para un prototipo de una línea de productos de software

Ramón Gómez-Romero, Karen Cortés Verdin, Juan Carlos Pérez Arriaga, Ángeles Arenas Valdés

Universidad Veracruzana, Facultad de Estadística e Informática,
Xalapa, Ver., México

ultra_rgr@gmail.com, kcortes@uv.mx, juaperez@uv.mx, aarenas@uv.mx

Resumen. Una arquitectura orientada a servicios o SOA (*Service Oriented Architecture*) es una arquitectura de software que proporciona servicios que corresponden a funcionalidad de negocio reutilizable a través de interfaces bien definidas. SOA surge ante la necesidad de poder emplear funcionalidad ya disponible en sistemas heterogéneos distribuidos. Para desarrollar una SOA, existen diversas opciones tecnológicas que permiten implementar servicios web. Un servicio web permite definir componentes o servicios reutilizables que representan tareas de negocio. En este trabajo se presenta el desarrollo de una arquitectura orientada a servicios para una aplicación denominada PAAC. PAAC es el prototipo de una herramienta para registro de actividades y productos académicos que pretende servir de base para el desarrollo de una Línea de Productos de Software (LPS) y sigue un estilo Modelo Vista Controlador (MVC). Dadas las ventajas que ofrece SOA, se decide implementar este estilo arquitectónico para PAAC como una opción en la futura implementación de la mencionada LPS. Adicionalmente, es posible constatar la reutilización de la lógica de negocios de PAAC, gracias a la implementación del MVC.

Palabras Clave: Arquitectura de Software, SOA, Servicios Web, Líneas de Productos de Software, Android.

1 Introducción

Una Línea de Productos de Software (LPS) “es un conjunto de sistemas de software intensivo que comparten un conjunto de características común y administrado las que satisfacen las necesidades específicas de un segmento del mercado o misión y que se desarrollan a partir de un conjunto común de activos esenciales de una manera prescrita” [1]. Con un enfoque de Línea de Productos de Software se busca la reutilización proactiva y planeada, mejorando así la calidad y reduciendo costos y el tiempo de introducción al mercado. La arquitectura de una LPS es quizá el artefacto o activo más importante, ya que permite la

reutilización y cumplir con los requisitos de calidad. La arquitectura, además, es la base común de cada uno de los productos que conforman la LPS. Una Arquitectura de Software se define como "una descripción de más alto nivel de los subsistemas y sus componentes y las relaciones que existen entre ellos" [2]. Una arquitectura orientada a servicios (SOA) tiene como base la implementación de servicios para realizar tareas que lleven a cabo los requisitos del negocio. En este modelo de arquitectura se tienen tres elementos principales: el consumidor de los servicios, el proveedor de los servicios y el registro o repositorio de los servicios que hace posible que el consumidor haga uso de los servicios del proveedor.

Desde hace algún tiempo, se tiene en desarrollo una herramienta para el registro de productos y actividades académicas (denominada PAAc) para los profesores de la Facultad de Estadística e Informática de la Universidad Veracruzana. Con el objetivo de construir una Línea de productos, se han construido dos prototipos funcionales de la herramienta. Estos dos prototipos, además de ayudar en la especificación de requerimientos para la LPS, han permitido probar tecnología para el desarrollo de la misma. A fin de explorar el uso de una arquitectura orientada a servicios como una opción para el desarrollo de una LPS, se decide, a partir de la versión más reciente de PAAc obtener una SOA. Así, aquí se presenta el desarrollo de la arquitectura orientada a servicios para PAAc. Este documento se encuentra organizado de la siguiente manera: en el segundo apartado se define la arquitectura SOA y sus principales características; en este mismo apartado, además, se describen los servicios web que contribuyen a la implementación de SOA; en el tercer apartado, se presenta el desarrollo de la arquitectura SOA para PAAc, mostrando las diversas vistas que describen esta arquitectura; en el cuarto apartado, se describen las pruebas realizadas a la implementación de SOA y, por último, se dan las conclusiones y los trabajos futuros.

2 Arquitectura orientada a servicios (SOA)

La arquitectura de software de un programa o sistema de cómputo es la estructura o estructuras del sistema que comprende los elementos de software, las propiedades externamente visibles de dichos elementos y relaciones entre ellos" [3]. Los elementos externamente visibles corresponden a los componentes de la arquitectura y se pueden definir como un elemento del sistema en el que se encapsulan servicios y funcionalidades que son accesibles mediante una interfaz. Las relaciones entre los elementos o componentes se encuentran definidos en la interfaz del componente. Como ya se dijo en la Introducción, una arquitectura de software es la base común de los productos de una LPS, promueve la reutilización y es el artefacto en el que se construye la calidad del software.

Una arquitectura orientada a servicios (SOA) trata resolver dos grandes temas que las organizaciones dedicadas al desarrollo de software enfrentan: heterogeneidad y cambio. Heterogeneidad en los sistemas y tecnologías con las que fueron desarrollados dichos

sistemas; cambio en las necesidades del mercados y la tecnología misma que ofrece cada vez un desarrollo más acelerado. Un aspecto característico de una SOA es que sigue un modelo arquitectónico de capas al agregar una capa de servicios de tecnología a la capa a la lógica de negocios deseada. Así, se cuenta con las implementaciones necesarias para las aplicaciones de los clientes [6].

Al desarrollar una SOA, se tiene una manera de diseñar, desarrollar y administrar sistemas de tal forma que:

- Los sistemas proporcionan funcionalidad del negocio a través de interfaces bien definidas.
- Los consumidores de servicios se construyen empleando funcionalidad provista por servicios ya disponibles.
- Hay una clara separación entre la interfaz del servicio y la implementación del mismo.
- La infraestructura permite el descubrimiento, composición e invocación de servicios.
- Los protocolos son predominantemente intercambios basados en mensajes.

La estructura de SOA se muestra en la figura 1. Puede observarse que dicha estructura se divide en dos partes. Al lado izquierdo, los elementos que se encargan de la funcionalidad; mientras que al lado derecho, lo elementos que tratan con la calidad del servicio. A continuación, se describen cada uno de ellos [6]:

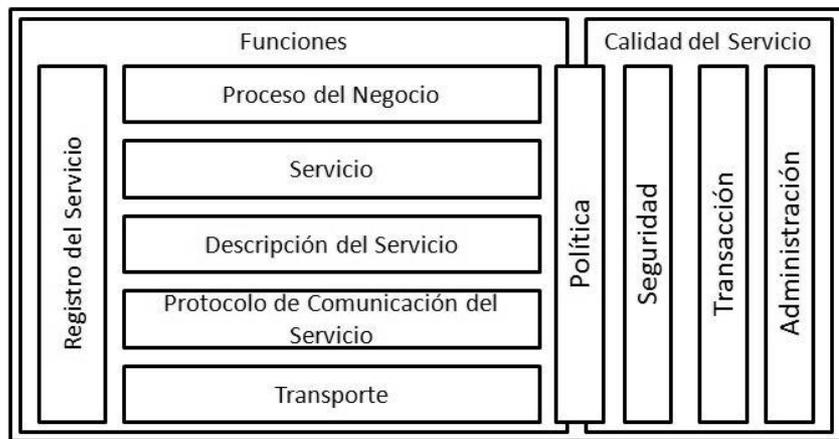


Fig. 1. Elementos de SOA.

- Funcionalidad:
 - Transporte.- Es encargado de trasladar las solicitudes de servicio del consumidor al proveedor de servicios y las respuestas del proveedor al consumidor.

- Protocolo de comunicación del servicio.- Mecanismo de comunicación acordado para que el proveedor de servicios y el consumidor intercambien lo que se solicita y lo que se devuelve.
 - Descripción de servicio.- Esquema que define el servicio, cómo debe ser llamado y los datos que se requieren para llamarlo de manera exitosa.
 - Servicio.- Se describe el servicio que está disponible para su uso.
 - Proceso del negocio.- Servicios que se llaman en un orden en particular, conforme a ciertas reglas a fin de satisfacer un requisito de negocio.
 - Registro del servicio.- Es un repositorio que contiene los servicios y las descripciones de datos a emplear por los proveedores para publicar sus servicios y por los consumidores para encontrar servicios disponibles.
- Calidad de servicio:
 - Política.- Son las condiciones o reglas que el proveedor define para los consumidores de servicios. Hay políticas que se refieren a aspectos funcionales y a aspectos de calidad de servicio.
 - Seguridad.- Conjunto de reglas que pueden ser aplicadas a la identificación, autorización y control de acceso de los consumidores de servicio.
 - Transacción.- Atributos que pueden aplicarse a un conjunto de servicios para que obtengan un resultado consistente.
 - Administración.- Es el conjunto de atributos para administrar los servicios provistos o consumidos.
 - Un Servicio Web o *Web Service* "es un conjunto de aplicaciones o de tecnologías con capacidad para inter-operar en la Web" [4]. Los servicios web, emplean una serie de protocolos y estándares para intercambiar datos entre aplicaciones que pueden estar desarrolladas en distintos lenguajes de programación y ser ejecutadas en cualquier plataforma y que se encuentran distribuidas en redes como Internet. Los Servicios Web utilizan protocolos abiertos para el transporte de la información sobre internet, el más utilizado en HTTP aunque existen otros que también pueden ser utilizados, JMS(*Java Message Service*) y SMTP (*Simple Mail Transfer Protocol*) son algunos ejemplos.

En cuanto a las tecnologías para comunicación de servicios web, se tienen RPC (*Remote Procedure Call*), SOAP (*Simple Object Access Protocol*) y REST (*REpresentational State Transfer*). REST que es la tecnología empleada en este trabajo y constituye un estilo de arquitectura para sistemas hipermedia distribuidos en la web, generalmente llamados RESTful. REST es un conjunto de principios que permiten diseñar arquitecturas en red y está basado en estándares como: HTTP, URI, de representación de recursos (XML, JSON, HTML, entre otros.) y tipos MIME. Para desarrollar aplicaciones RESTful simples, ligeras y rápidas se recomienda seguir estos principios:

- Identificación de Recursos a través de URIs

- Interfaz uniforme
- Mensajes auto-descriptivos
- Interacciones con estado a través de hipervínculos.

3 Prototipo PAAC

PAAC (Productos y Actividades Académicas) es una herramienta que permite el registro de los productos y actividades de los académicos de la Facultad de Estadística e Informática de la Universidad Veracruzana. Se pretende realizar una LPS ya que se ha encontrado variación entre las actividades y productos que un académico o investigador realiza y desea administrar en determinado momento. La primera versión se realizó en 2011[7] y consistió en el desarrollo de un prototipo a emplear para las actividades de captura, análisis y especificación de los requerimientos del dominio. Este primer prototipo consideraba siete casos de uso y empleaba las tecnologías JSF1.2 y páginas JSP bajo un modelo MVC (Modelo Vista Controlador) [2].

En el 2012 se desarrolla la segunda versión del prototipo [5]. En esta segunda versión se hicieron adecuaciones a la base de datos, mejoras a los casos de uso y se incluyeron nuevas tecnologías, bajo el mismo modelo MVC. Además de las mejoras, en este segundo prototipo se buscaba evaluar nueva tecnología para su posterior inclusión en el desarrollo de la LPS. Es así que se usaron JavaServer Faces 2.0, ADF Faces, las páginas JSF y la tecnología AJAX. En la figura 2 de la siguiente página puede apreciarse el diagrama de paquetes correspondiente al MVC de la segunda versión de PAAC, donde:

- Paquete JavaBeans: Contiene todos los JavaBeans que se utilizan el prototipo y constituyen el modelo del MVC.
- Paquete Managed Beans: Contiene todos los controladores de JSF que intervienen entre las vistas y el modelo .
- Paquete Páginas JSF: Contiene todas las páginas (vistas) del prototipo.

La arquitectura SOA que se desarrolló para PAAC está compuesta por siete recursos que a su vez están compuestos por métodos a los cuales se puede acceder a través de una URL. Estos métodos cumplen la funcionalidad de los 21 casos de uso que se implementaron en la segunda versión de PAAC. Para la implementación de esta arquitectura se empleó:

- JEE. - Plataforma para desarrollo de aplicaciones en Java. JEE fue empleada para desarrollar PAAC y se emplea la implementación de Servicios Web.
- JDeveloper 11g. - Es el ambiente de desarrollo empleado..
- HTTP. - Canal para el transporte de mensajes.
- REST.- Para intercambio de mensajes, es ligero y está bien acoplado con HTTP.

- JSON. - Formato ligero de intercambio de datos, que soporta un gran volumen de información, además de ser fácil de leer y escribir.

En la figura 3 de la siguiente página, pueden observarse los recursos de los Servicios Web del paquete Web Service que están disponibles en la red para ser consumidos por un cliente. En esta figura se pueden observar:

- Paquete JavaBeans: Todos los JavaBeans que se utiliza en la herramienta PAAC.
- Paquete Managed Beans: Todos los controladores de JSF que intervienen entre las vistas y los modelos (JavaBeans).
- Paquete Páginas JSF: Todas las vistas (páginas) de la versión Web
- Paquete Web Service: Todos los recursos de los Servicios Web que están disponibles en la red para ser consumidos por un cliente.

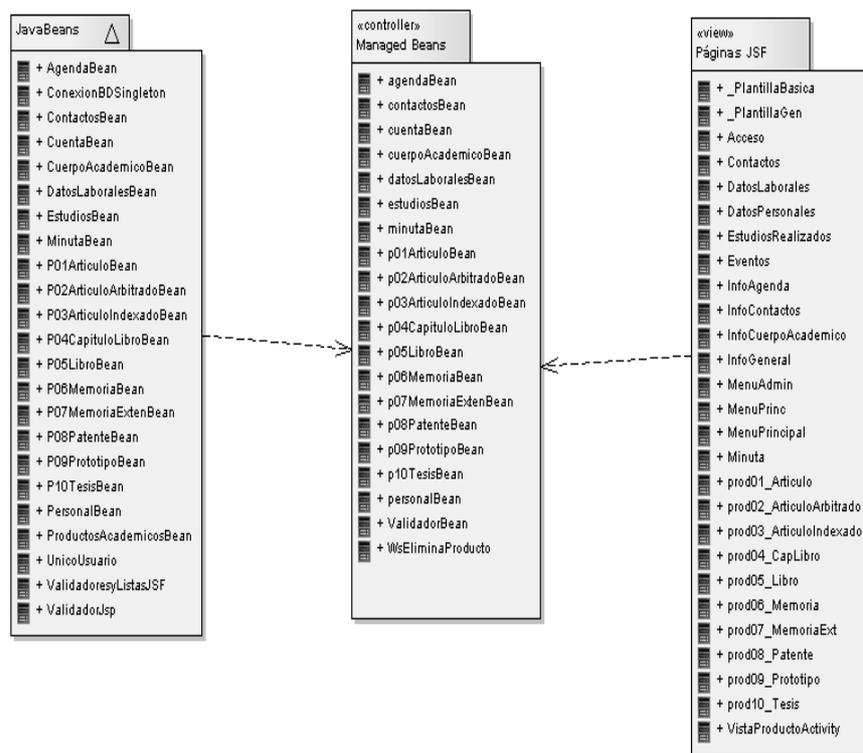


Fig. 2. MVC de PAAC v2.

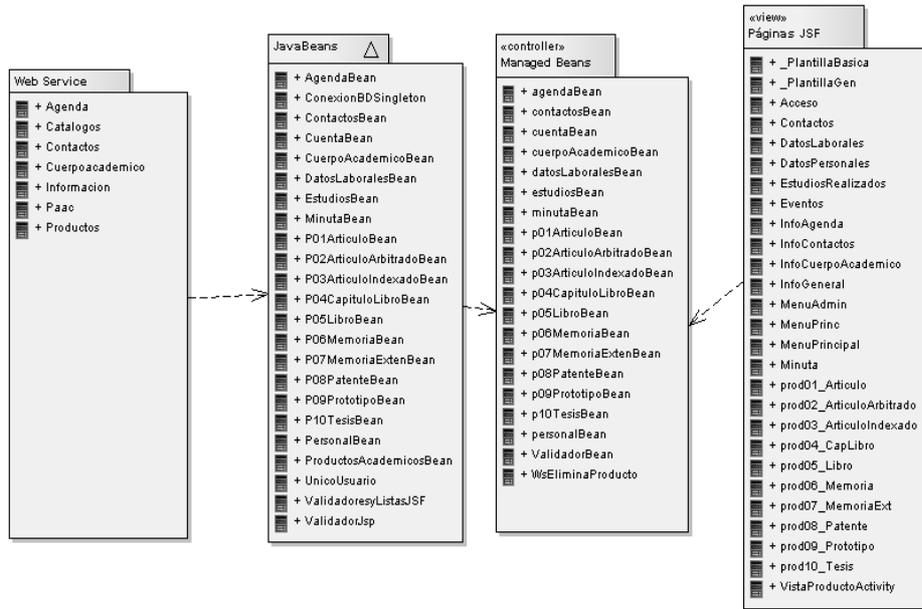


Fig. 3. SOA de PAAc v2.

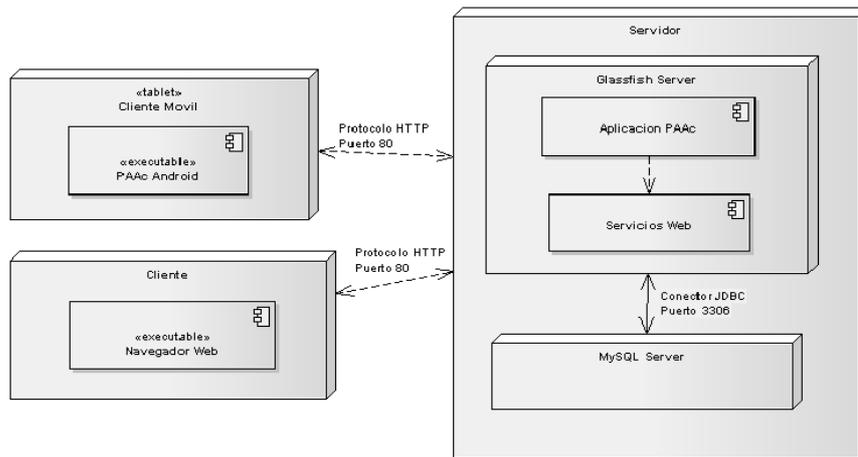


Fig. 4. Modelo de despliegue de la arquitectura.

En la figura 4, muestra el modelo de despliegue en el que se distribuyen los diversos elementos de la arquitectura en el hardware; puede observarse:

- Cliente: PC donde el usuario ejecuta el navegador web, para ingresar al sistema. Se conecta al servidor mediante el protocolo HTTP utilizando generalmente el puerto 80.
- Cliente móvil: Tableta donde el usuario ejecuta la aplicación PAAc basada en Android, para interactuar con la información del sistema. Se conecta al servidor mediante el protocolo HTTP utilizando generalmente el puerto 80.
- Servidor: Se ejecuta Glassfish Server, que es un servidor de aplicaciones en el cual se despliega la aplicación PAAc y los Servicios Web. Éstos a su vez interactúan directamente con PAAc para ser consumidos por un cliente móvil. También se conectan con el servidor MySQL, por medio del conector JDBC (Java DataBase Connector) a través del puerto 3306 (puerto utilizado por defecto en MySQL).

Dentro de la vista dinámica, se desarrollaron los diagramas de secuencia correspondientes pero no se muestran en este trabajo por limitaciones de espacio. Estos diagramas muestran la interacción entre los componentes o elementos del sistema cada vez que un cliente realiza la invocación del servicio web correspondiente.

Se desarrolla una aplicación móvil en sistema operativo Android, denominada Android PAAc para llevar a cabo las pruebas de SOA. Puede observarse la distribución de este cliente móvil en el modelo de la figura 5 arriba.

4 Pruebas

Al desarrollar software orientado a objetos guiado por casos de uso, éstos se convierten en la unidad mínima de funcionalidad, por lo tanto las pruebas realizadas se basan en los casos de uso. En las pruebas de la primera y segunda versión de PAAc se consideró detectar defectos de: configuración, de funciones faltantes, de uso incorrecto o inconsistente de archivos, de violaciones a la integridad de datos y validaciones. Se decidió entonces emplear el mismo conjunto de pruebas para la aplicación Android PAAc y la implementación de la SOA. Así, se corrobora que se cumplan con las mismas funciones ya disponibles en las versiones anteriores [5] y, de esta manera, verificar que no hubiera errores en la implementación de los servicios web. Las pruebas por casos de uso realizadas fueron las siguientes:

- Ingresar al sistema
- Actualizar datos personales
- Registrar estudio realizado
- Actualizar estudio realizado
- Registrar datos laborales
- Actualizar datos laborales
- Registrar producto

Para cada uno de los casos de uso se consideraron los cursos exitoso y fallido. Estos casos de uso y las pruebas correspondientes fueron exactamente los mismos que en las versiones anteriores, lo cual da la ventaja de: 1) aplicar pruebas de regresión sin esfuerzo adicional, 2) reducir el tiempo de pruebas con resultados confiables, 3) verificar la funcionalidad de los servicios web. No se cuenta con datos estadísticos de las pruebas pero éstas pueden resumirse en la tabla 1 para las tres versiones:

Tabla 1. Pruebas por caso de uso.

Versiones de PAAc/ Pruebas por casos de uso	1a. versión	2a. versión	Android (servicios web)
Ingresar al sistema	Sin defectos	Sin defectos	Sin defectos
Actualizar datos personales	Sin defectos	Sin defectos	Sin defectos
Registrar estudio realizado	Sin defectos	Sin defectos	Sin defectos
Actualizar estudio realizado	Sin defectos	Sin defectos	Sin defectos
Registrar datos laborales	Sin defectos	Sin defectos	Sin defectos
Actualizar datos laborales	Sin defectos	Sin defectos	Sin defectos
Registrar producto	Sin defectos	Sin defectos	Sin defectos

Después de realizar las pruebas a la Aplicación PAAc de la plataforma Android, se dieron los siguientes resultados:

- La comunicación entre la aplicación de Android PAAc y los servicios web que se encuentran en el servidor, se llevó a cabo sin ningún problema por medio del protocolo HTTP.
- La funcionalidad de los servicios web probados en cada caso de uso resulto satisfactoria, ya que se llevaron a cabo las mismas funciones que las que se tenían en las versiones anteriores. No se detectaron defectos y los resultados de las operaciones fueron correctos.
- Como era de esperarse, se comprobó que las validaciones que se hacen en la versión anterior, son muy dependientes de la tecnología con que se implementó, específicamente el framework ADF Faces (correspondiente a la vista del MVC). Por lo tanto, hubo que implementar las validaciones correspondientes en la aplicación Android PAAc.

La aplicación Android PAAc cumplió con los resultados esperados en las pruebas ya que se verificó que la implementación de los servicios web fuera correcta y completa con respecto a la lógica de negocio original. Además, se comprobó que hubo reutilización del código de la versión anterior y que los servicios web logran ofrecer una funcionalidad completa a la aplicación cliente en Android.

5 Conclusiones y trabajos futuros

En este trabajo se presentó el desarrollo de una arquitectura orientada a servicios para la herramienta PAAC. PAAC es un prototipo de una LPS para el registro de productos y actividades académicas para la Facultad de Estadística e Informática de la Universidad Veracruzana y sigue una arquitectura MVC. A continuación se presentan las conclusiones y trabajos futuros correspondientes a este trabajo.

5.1 Conclusiones

Mediante la implementación de servicios web se obtiene una arquitectura orientada a servicios para PAAC. Es así que se desarrolla la capa tecnológica que interactúa directamente con el modelo del MVC de PAAC. Para este desarrollo se emplean servicios web en java, utilizando REST y JSON. Se desarrolla, además, una aplicación en Android, denominada Android PAAC, para realizar las pruebas de los servicios web. Se realizaron las pruebas basadas en los casos de uso originales de PAAC, aplicándolas en la aplicación en Android. Se obtienen los siguientes resultados:

- La interoperabilidad con sistemas ya existentes, en este caso PAAC, logrando la reutilización de la lógica de negocios previamente implementada sin mayores problemas.
- La implementación de una Arquitectura SOA a través de servicios web.
- Se desarrolló una aplicación en la plataforma Android que, a través de la SOA, permite utilizar la funcionalidad de la herramienta PAAC.

La aportación de este trabajo está encaminada a abrir paso al desarrollo de diferentes sistemas o aplicaciones para diferentes plataformas. Aunque en esta versión se utilizó la plataforma Android como consumidor del servicio, cualquier tecnología que pueda consumir los servicios web creados en este trabajo podrá implementar toda la funcionalidad de la herramienta PAAC. Es por esto la importancia utilizar una Arquitectura Orientada a Servicios, que permite reutilizar componentes, tener interoperabilidad y ser escalable, ya que otra arquitectura no hubiese podido dar los mismos beneficios.

5.2 Trabajos futuros

Como trabajos futuros, se tienen los siguientes:

1. Incorporar funcionalidades en la aplicación de Android PAAC, correspondiente a los casos de uso faltantes a la versión anterior de esta herramienta. Los casos de uso implementados y probados para las tres versiones (incluyendo la presente versión) se mostraron en la Tabla 1. Sin embargo, en el caso de la segunda versión se tienen disponibles más casos de uso que no fueron incluidos en la presente versión.

2. Realizar pruebas de seguridad y rendimiento a través de Android PAAc para detectar defectos y mejoras a la arquitectura SOA y MVC, según corresponda.
3. Implementar, a partir de las tecnologías que se tienen, una plataforma común para poder llevar a cabo una Línea de Productos de Software. El desarrollo de esta plataforma común requerirá la integración de prácticas de SOA y LPS, lo cual, se está convirtiendo en un nuevo paradigma de desarrollo en el ámbito de las LPS.

Referencias

1. Clements P. C. & Northrop, L. M.: Software Product Lines: Practices and Patterns, Addison-Wesley. Boston, MA, USA (2001)
2. Buschmann, F., Rohnert, H., & Stal, M.: Pattern-oriented software architecture. Volume 1, a system of patterns, J. Wiley and Sons. New York (1996)
3. Bass L., Clements P., Kazman R.: Software architecture in Practice, Addison-Wesley, Boston, MA, USA (2003)
4. Guía Breve de Servicios <http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>
5. Hernández Pérez, M.: Evaluación y evolución del prototipo de la herramienta PAAc (productos y actividades académicas), Tesis de Licenciatura. Universidad Veracruzana, Xalapa, Ver. México: a (2012)
6. IBM:Patterns: Service-OrientedArchitecture and Web Services, IBM Redbooks. USA (2004)
7. Ruiz Tlapa, R., & Vargas Alonso, M.: Desarrollo de un prototipo funcional de apoyo a cuerpos académicos para una línea de producto de software. Tesis de Licenciatura Universidad Veracruzana, Xalapa, Ver. Mexico (2011)

Software para evaluación de construcciones sustentables: caso práctico

Eunice Villicaña Ortiz, Ubaldo R. Montes Juárez, Olga E. Serena Ibáñez,
Ana G. Pérez Aguilar, Gabriela Martínez Rosiles, María R. Zarate Nava,
Jesús L. López Hernández

Universidad Tecnológica del Centro de Veracruz,
Av. Universidad No.350, Carretera Federal Cuitláhuac-La Tinaja, Congregación Dos Caminos,
Cuitláhuac, Veracruz, C.P. 94910 México

{soloenergias, ubaldo.montes.juarez, olga.serena.ibanez, anagabrielperezaguilar, gmrnov92,
ma.reina.zarate, jleonardoh}@gmail.com

Abstract. From the energy efficiency's point of view in construction, buildings are open thermodynamic systems, where the dull and translucent walls swap energy with the environment, therefore they're not in thermal balance, this state suggest us the necessity to establish mechanisms to know their behavior before the external and internal variations. Because of that, the conception of a building, from the design to the habitability, an energy and comfort analysis should be considered, due to the climatic and geographic conditions of the place where the house is meant to be built, that will allow to identify the thermodynamic behavior and that will anticipate the energy consumption that will demand the system. SICODES is a software that virtually evaluates the behavior of the covering transmittance of a building according to the user specifications and climatic conditions of the place. At the end, SICODES establishes the level of energy efficiency of the construction.

Palabras clave. Ingeniería de software, sustentabilidad, evaluación, simulación.

1 Introducción

El aumento del consumo energético en las últimas décadas ha impactado en el medio ambiente, se estima que el 75% de emisiones de Gas de efecto invernadero (GEI) son atribuibles al sector energético [7]. En este sentido, el sector de la construcción ha contribuido de sobremanera, si se consideran los procesos que se realizan desde el diseño, la construcción y habitabilidad de las obras.

Se estima que en promedio una casa para cuatro personas tiene un consumo de 300 KWh/mes, esto supone un consumo medio anual de 3.6 MWh [2]. Estos datos reflejan un importante consumo para satisfacer las necesidades habituales, no obstante pueden

establecerse estrategias de ahorro como el cambio de luminaria, el establecimiento de horarios de cargas eléctricas, el correcto dimensionado del sistema eléctrico, etc., que si bien permiten reducir el consumo, por si solas no pueden considerarse como las únicas alternativas.

La aplicación de la eficiencia energética en edificios antes del diseño es un tema que cobra importancia, y que debe ser atendido en el corto plazo. Determinar la demanda energética que tendrá un edificio permitirá realizar adecuaciones en cuanto a orientación, morfología y materiales a emplear con el fin de reducir el consumo energético dado por la iluminación, calefacción o refrigeración. Por ello el desarrollo de herramientas y técnicas que permitan prever el comportamiento energético de un edificio, es un tema de suma importancia. El desarrollo de edificaciones energéticamente sustentables en México apenas se empieza a trabajar, por lo cual aún no se ha tenido un avance significativo, pues aún se carece de la cultura del cuidado del medio ambiente y la edificación bioclimática. Algunos países han desarrollado herramientas de este tipo, tal es el caso de España, país que integró un documento que contiene las herramientas de cálculo para disminuir la demanda energética a través de técnicas de refrigeración, calefacción, iluminación, diseño de cerramientos, etc. Esta información se encuentra contenida en el Código Técnico de la Edificación, mismo que es utilizado para los programas informáticos CALENER y LIDER los cuales se enfocan en la certificación energética de edificios [2],[3].

El desarrollo de una aplicación que permita simular en entorno 3D la envolvente de un edificio y evaluar su comportamiento a través del análisis de transferencia térmica de sus cerramientos, tales como paredes, techos y suelos, todo ello en función a las condiciones climáticas del lugar donde ha de ser construida, será de gran impacto tecnológico y ambiental en México, dado que no existe un software de simulación y evaluación de construcciones. El presente artículo describe el caso práctico Simulador de Construcciones Energéticamente Sustentables (SICODES).

2 Aplicación de la metodología

La metodología empleada para el desarrollo del caso práctico SICODES, Object Modeling Technique (OMT) se basa en el análisis y diseño orientado a objetos. [8]. La gran virtud que aporta OMT, es su carácter de ser abierta (no propietaria), su objetivo es desarrollar un modelo de lo que va a realizar el sistema, el modelo se expresa en términos de objetos y de relaciones entre ellos, muestra un flujo dinámico de control y las transformaciones funcionales.

2.1 Fase de análisis

El propósito de la fase radica en entender y modelar el problema en el dominio de la aplicación.

Para obtener un modelo de dominio que se encarga de mostrar las propiedades más importantes de la problemática identificada, se diseña un conjunto de diagramas (casos de uso, secuencia, estado, clases, flujo del sistema, diccionario de datos) que permitan representar el modelo de dominio. A continuación la Figura 1, ejemplifica la interacción entre el usuario y el sistema.

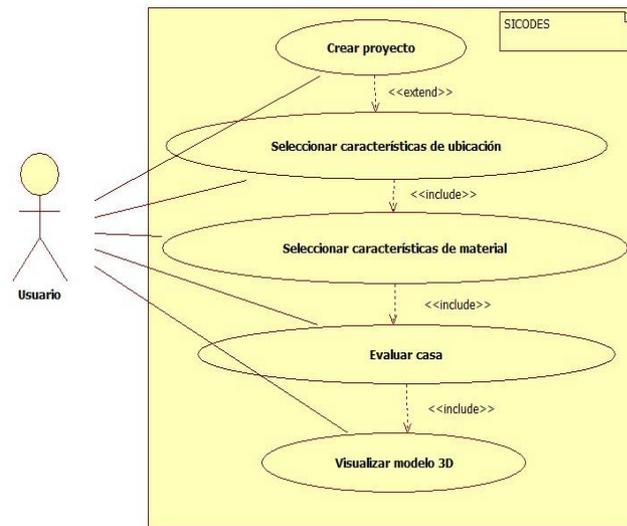


Fig. 1. Diagrama de caso de uso de la aplicación SICODES.

2.2 Fase de diseño del sistema

El propósito de la fase radica en definir la arquitectura del sistema y sirve para la tomar decisiones estratégicas. Con base en el análisis de la fase anterior, se diseña una arquitectura software para la implementación en este caso práctico; sirve la arquitectura para examinar la propuesta de construcción de SICODES, estos procesos permiten formular ciertas estrategias y decisiones arquitecturales de apoyo en esta fase, así como en el resto de desarrollo de la aplicación.

2.2.1 Arquitectura SICODES

Para el desarrollo, la aplicación basa su implementación en la arquitectura Model-View-Controller (MVC), ya que por sus características proporciona escalabilidad, robustez y adaptabilidad a un modelo orientado a objetos, propuesto por OMT. Al ser una

arquitectura adaptable permite la integración de nuevas capas que admitan la escalabilidad de la misma, la capa evaluación complementa la funcionalidad de esta aplicación.

A continuación se describen las capas que integran la *Arquitectura SICODES*:

- **Vista:** Contiene todos los componentes visuales e interfaces de la aplicación, que permiten capturar la información necesaria para efectuar los procesos de diseño. Los materiales 3D que se utilizan para visualizar la edificación son integrados en esta capa.
- **Modelo:** Las clases que funcionan como enlace entre la aplicación y la base de datos se encuentran integradas en esta capa, además de contener las clases que permiten modelar la información relacionada con la zona geográfica, las condiciones climáticas elementales y los materiales que se emplearán en el diseño de la edificación.
- **Controlador:** Es la capa de comunicación entre la vista y el modelo, es decir que la información ingresada por el usuario es validada y enviada al modelo para hacer actualizaciones a la base de datos de la aplicación. También permite la comunicación con la capa de evaluación, para realizar los cálculos pertinentes en el proceso de evaluación.
- **Evaluación:** El proceso de evaluación de las características de una edificación es integrado en esta capa, la cual incluye las clases, los métodos y fórmulas para evaluar la eficiencia energética, de acuerdo a las características seleccionadas dentro de SICODES.

Las fórmulas implementadas en esta capa, permiten evaluar con base a las características del espesor y conductividad térmica de los materiales, determinando la resistencia térmica total de la edificación. Además de incorporar en la evaluación factores como: ventanas, puertas y zona geográfica de localización.

Las fórmulas tomadas para el desarrollo de esta capa, se basan en varios métodos simplificados que se pueden emplear para el cálculo de los parámetros característicos de los diferentes elementos que componen la envolvente térmica de la edificación.[4]. La fórmula 1 muestra la expresión matemática para determinar la resistencia térmica total por pared.

$$R_t = R_{si} + R_1 + R_2 + \dots + R_n + R_{se} . \quad (1)$$

Una vez calculada la resistencia total de cada pared se procede a determinar la temperatura interior que tendrá la edificación de acuerdo al tipo de cercamientos ingresados por el usuario, pero primero se calcula la temperatura por cada pared de la siguiente manera:

$$\text{Temperatura por pared} = \text{Resistencia total por pared} - \text{Temperatura de la ciudad en grados Kelvin.} \quad (2)$$

Una vez que se tiene la temperatura por cada pared, se obtiene un promedio el cual se convierte en la temperatura aproximada que podría llegar a generar una edificación. Ya que se ha obtenido la temperatura promedio de la edificación se procede a determinar la diferencia de temperaturas que existirá entre la temperatura de la ciudad con la de la edificación y en base a este dato se muestra la gráfica indicando si la diferencia de temperaturas es acorde a la zona que el usuario seleccionó para construir la edificación.

La adaptación de la arquitectura MVC y la incorporación de la capa de evaluación permiten la robustez y escalabilidad de SICODES, la cual se expone en la Figura 2.

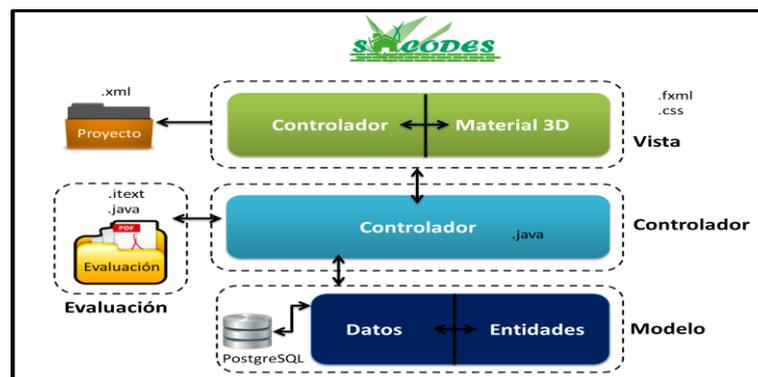


Fig. 2. Arquitectura de SICODES basada en MVC.

2.3 Fase de diseño de objetos

El propósito de la fase radica en definir, refinar y optimizar el modelo de análisis, agregando conceptos de programación.

Los algoritmos que definen la funcionalidad de la capa de evaluación en la arquitectura SICODES se desarrollaron en esta fase, es decir, se planeó y se diseñó los algoritmos utilizados por la aplicación para el proceso de evaluación de una edificación. Además, se

buscó la optimización de clases, para que sea más eficiente el modelo que se realizó anteriormente en la fase de análisis. El diseño de objetos se centra en las estructuras de datos y algoritmos que son necesarios para implementar cada clase que se ha planteado ya en fases anteriores.

2.4 Fase de implementación

El propósito de la fase radica en definir implementar las clases de objetos en un lenguaje de programación. Empleando la arquitectura SICODES, se generaron 4 paquetes que contienen las clases, modelos y recursos necesarios para la funcionalidad de la aplicación. En la capa vista se modelaron 27 interfaces, 3 hojas de estilo y 48 recursos relacionados con los modelos 3D; en modelo se modelaron 39 clases necesarias para poder realizar conexión a base de datos y se llevó a cabo la implementación del controlador con 22 clases y en la capa de evaluación se tienen las clases clave de la aplicación, en ella se modelaron cada una de las formulas determinantes para la transmitancia térmica.

El resultado final de esta fase de implementación es una gráfica que indica la diferencia de temperaturas entre la zona geográfica y la temperatura que se genera en la edificación, dependiendo de la zona se dan recomendaciones para ajustar los materiales de cómo está constituida la edificación, dicha grafica se muestra en la Figura 3.

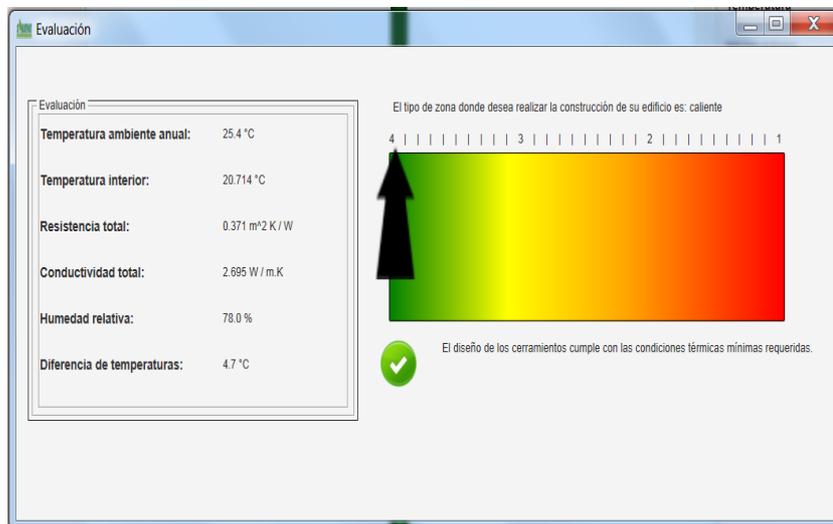


Fig. 3. Gráfica de evaluación de la edificación bajo la aplicación SICODES.

3 Resultados

La implementación de OMT facilitó el modelado de los objetos y actores que interactúan en SICODES, este punto fue fundamental para la construcción del mismo. La versión 3.0 de la aplicación permite simular una edificación de uno o dos niveles y cuatro paredes por nivel, además de la selección de los materiales para los cerramientos y la localización geográfica; con esta información se evalúa la transmitancia térmica y se realiza un análisis del confort térmico.

La Figura 4, muestra el proceso básico que se modeló bajo OMT de la simulación y evaluación de una edificación en la aplicación SICODES [4].

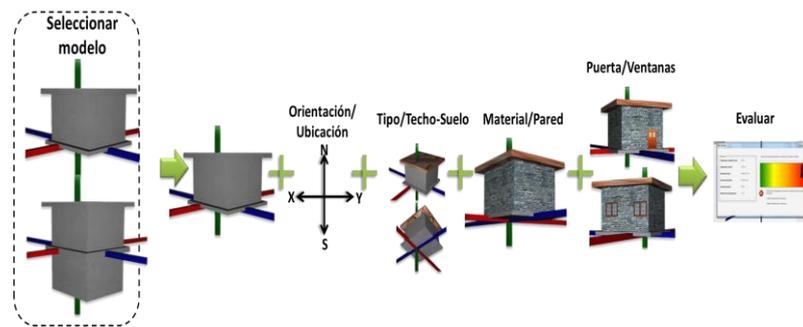


Fig. 4. Flujo básico para el diseño y evaluación de una edificación en SICODES.

Como se observa en la Figura 4, se brinda al usuario la posibilidad de elegir el tipo de edificación que desea simular y personalizarla al seleccionar características como: puertas, ventanas, zona geográfica, material de techo, material del suelo y material de construcción, factores que son importantes para poder determinar el nivel de eficiencia térmica en la evaluación de la edificación.

El uso de la Arquitectura SICODES permitió estructurar los elementos y componentes de la aplicación, brindando la característica de escalabilidad a la aplicación, dada la condición de ampliar su funcionalidad en fases posteriores. Otra de las ventajas, es que el estilo de arquitectura fue dividida en módulos, para implementar la codificación separada y posteriormente ser integrada, facilitando el trabajo en equipo y división de tareas.

4 Trabajos relacionados

4.1 LIDER

LIDER es una aplicación informática que se puede emplear para verificar, de forma temporal y bajo ciertas condiciones técnicas, las exigencias de demanda energética establecidas en el Código Técnico de la Edificación de España. Esta herramienta está diseñada para realizar la descripción geométrica, constructiva y operacional de los edificios, así como para llevar a cabo los cálculos de demanda energética de los edificios [2],[3].

4.2 CALENER VyP

El programa CALENER VyP (Vivienda y Pequeño edificio terciario) es una de las herramientas informáticas y legislativas que da cuerpo al método general de la Certificación Energética de España. Con ella, se pueden calificar todos los edificios de viviendas, y aquellos edificios del sector terciario, cuyas instalaciones térmicas y de agua caliente sanitaria, se puedan simular con alguno de los modelos que el programa trae adscritos [2],[3].

4.3 Ecotect

El software de análisis de diseño sustentable Ecotect es una herramienta de diseño sustentable de edificios que ofrece un amplio rango de simulación y funcionalidad de análisis de energía de las construcciones que puede mejorar el desempeño de los edificios existentes y diseños de nuevos edificios. Capacidades de análisis de energía, agua y emisiones de carbono integradas con herramientas que permiten visualizar y simular el desempeño de un edificio dentro del contexto de su entorno [1].

4.4 ESP-R

ESP-R es un programa que permite simular dinámicamente el comportamiento energético de una vivienda o construcción, lo cual permite verificar en fase de diseño las cargas térmicas, tanto de calefacción como de refrigeración. Esto es lo básico que se puede desarrollar con ESP-R ya que también permite el análisis de flujos de aire y aún análisis de CFD (Computer Fluid Dynamic) [5].

4.5 MIT DESIGN ADVISOR V1.1

Plataforma Web del MIT que evalúa el comportamiento energético de edificios. Esta herramienta permite evaluar las condiciones de iluminación, ventilación, gasto de energía y confort de un edificio en forma rápida e intuitiva [6].

5 Conclusiones

La aplicación de OMT para el desarrollo de SICODES ha permitido generar una aplicación que permite la simulación de una edificación en un entorno 3D y simular el comportamiento de la envolvente de un edificio a partir de las características consideradas por el usuario tales como orientación, climatología, grosor de cerramientos y tamaño de las paredes y material con el cual estará construida. Dentro del equipo de desarrollo la metodología fue fácil de entender e implementar dado que se trabajó con un lenguaje orientado a objetos y gran parte de los artefactos generados son propios del paradigma de programación.

El desarrollo de una aplicación de tal magnitud permitió modelar la Arquitectura SICODES, que facilito el manejo y separación del proyecto en paquetes de trabajo que se traduce en una mejor administración de los recursos para el desarrollo del mismo. SICODES es un proyecto que se encuentra en su tercera fase, la visión es ampliar su funcionalidad para ofrecer mayor impacto en el sector de edificación sustentable en México.

Desde un enfoque sustentable, la construcción debe reducir el impacto medioambiental que produce, es por ello que en los últimos años se ha incrementado el interés por parte de empresas de la industria de la construcción para edificar de forma sustentable. SICODES es una herramienta que permite medir que tan eficiente térmicamente, será una casa antes del proceso de construcción. SICODES busca ser la herramienta que permita a los constructores evaluar y prever el consumo energético de las nuevas edificaciones con el fin de disminuir el consumo energético y el impacto medioambiental en México.

Referencias

1. Autodesk, <http://usa.autodesk.com/ecotect-analysis/>, recuperado 18 de Julio (2014)
2. CENEAM. (s.f.). Recuperado el 5 de Julio de 2014, de CENEAM: <http://www.magrama.gob.es/ca/ceneam/formacion-ambiental/cursos-y-postgrados/internet/certificacion-energetica-edificios-metodo-lider.aspx>, (2014)
3. Código tecnico (s.f.) : http://www.codigotecnico.org/web/recursos/aplicaciones/contenido/texto_0002.html, Julio (2014)
4. Documento de Apoyo al Documento Básico DB-HE Ahorro de energía. En M. d. Fomento, Cálculo de parámetros característicos de la envolvente. España.
5. Energy Systems Research Unit. Energy Systems Research Unit. Obtenido de ESP-r: <http://www.strath.ac.uk/esru/>, (2000)
6. Massachusetts Institute of Technology. MIT Design Advisor. Obtenido de MIT Design Advisor: <http://designadvisor.mit.edu/design/>, (2009)

7. Trabajo de Investigación: Estudio de la Energía Solar en México. Departamento de Energía, Universidad de Oviedo. España A Survey of Augmented Reality. *Presence*, 355-385, (2009)
8. Universidad Nacional Autónoma de México. (n.d.). Metodología OMT. Recuperado el 25 de Julio de 2014 de Posgrado en Ciencia e Ingeniería de la Computación: <http://www.mcc.unam.mx/~cursos/Objetos/Omt/omt.html>, (2014)

Metodología ágil scrumban en el proceso de desarrollo y mantenimiento de software de la norma moprosoft

David Ibarra Guzmán, Ulises Castañeda Islas, Carlos Pérez Corona, Blanca Estela Pedroza Méndez

Instituto Tecnológico de Apizaco, División de Estudios de Posgrado e Investigación, Tlaxcala, México

ibarra_david50@hotmail.com,sesilucasta@gmail.com,
cperez_corona@hotmail.com,thelismedina@hotmail.com

Resumen. En este artículo se propone el uso de la metodología ágil Scrumban en las actividades del proceso de Desarrollo y Mantenimiento de Software (DMS) de la norma mexicana MoProSoft para la industria de desarrollo de software, con la finalidad de utilizar las mejores prácticas de ambas metodologías (Scrum y Kanban) durante las diferentes fases del proceso de desarrollo de software que describe la categoría Operación (OPE) de la norma MoProSoft. La propuesta es utilizar un tablero de control o sistema de tarjetas que permita visualizar de manera fácil y práctica el avance de las actividades de un proyecto, las personas que están trabajando en cada una de las tareas, así como en qué fase y estado se encuentran cada una de éstas con el apoyo de tarjetas visuales.

Palabras clave: MoProSoft, Scrum, Kanban, Scrumban, Metodologías ágiles.

1 Introducción

Hoy en día las empresas que desarrollan software se enfrentan a grandes retos tanto de demanda como de calidad en sus productos, el uso de las tecnologías de la información en la industria de cualquier tipo se ha vuelto una herramienta de primera necesidad y entre la sociedad un estilo de vida, la mayoría dependemos de alguna forma de tecnología, sin embargo, a pesar de que existen muchas empresas alrededor del mundo que se dedican a satisfacer la necesidad de contar con sistemas o aplicaciones de software que procesen la información, no todas lo hacen de forma correcta, ordenada y con calidad. Actualmente existe un alto índice de fracaso en los proyectos de desarrollo de software, las empresas no alcanzan a terminar en tiempo y dentro de los costos planificados por falta de implementación de un modelo de procesos de calidad. Según cifras de la organización “The Standish Group” en su estudio de proyectos que realizó en 2013, determinó que el 18% de

los proyectos examinados fallaron, el 43% no cumplió los requisitos y solamente el 39% tuvo éxito [7].

En México, son pocas las empresas de desarrollo de software que cumplen con sus compromisos de fechas de entrega y satisfacen las necesidades del cliente una vez que terminan el producto de software, la mayoría de éstas no cumplen con el calendario, los costos de producción de una aplicación de software se elevan, la funcionalidad que requiere el cliente no es totalmente satisfactoria y los problemas en la calidad del software terminan por generar demasiados problemas durante las diferentes etapas del desarrollo de software. Por esta razón se propone el uso de metodologías ágiles, como Scrum y Kanban durante las actividades que establece cada una de las fases del proceso de Desarrollo y Mantenimiento de Software (DMS) de la norma MoProSoft.

2 Moprosoft

En 2002 se crea el modelo MoProSoft para las pequeñas y medianas empresas (PyMEs) con el objetivo de guiar y mejorar los procesos que se implementan para el desarrollo de software en México [1]. Como se muestra en la figura 1, este modelo consta de tres categorías y considera nueve procesos:

1. Categoría de alta dirección (DIR). Esta categoría aborda las prácticas relacionadas con la gestión del negocio. Proporciona los lineamientos a los procesos de la Categoría de Gerencia y se retroalimenta con la información generada por ellos [2].
2. Categoría de gerencia (GER). Esta categoría aborda las prácticas de gestión de procesos, proyectos y recursos en función de los lineamientos establecidos en la Categoría de Alta Dirección. Proporciona los elementos para el funcionamiento de los procesos de la Categoría de Operación, recibe y evalúa la información por éstos y comunica los resultados a la Categoría de Alta Dirección [2].
3. Categoría de operación (OPE). Categoría de procesos que aborda las prácticas de los proyectos de desarrollo y mantenimiento de software (DMS). Esta categoría realiza las actividades de acuerdo a los elementos proporcionados por la Categoría de Gerencia y entrega a ésta la información y productos generados [2].

En este trabajo se hace referencia específicamente al proceso de Desarrollo y Mantenimiento de Software (DMS) que se encuentra en la categoría de Operación (OPE). El propósito del DMS es la realización de las actividades de análisis, diseño, construcción, integración y pruebas de productos de software nuevos o modificados cumpliendo con los requerimientos especificados.

El proceso de Desarrollo y Mantenimiento de Software se compone de uno o más ciclos de desarrollo, en donde cada ciclo está compuesto por diferentes fases como se muestra en la tabla 1.

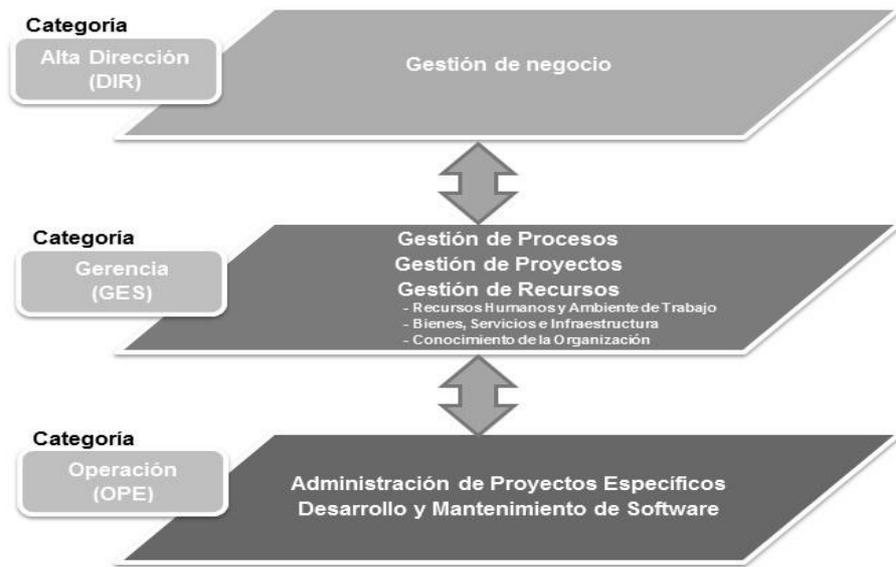


Fig. 1. Estructura de procesos del MoProSoft.

Tabla 1. Fases del DMS.

Fase	Descripción
F1:	Realización de la fase de inicio
F2:	Realización de la fase de requerimientos
F3:	Realización de la fase de análisis y diseño
F4:	Realización de la fase de construcción
F5:	Realización de la fase de integración y pruebas
F6:	Realización de la fase de cierre

3 Scrum

Scrum es un modelo que organiza a las personas en equipos pequeños, interdisciplinarios y auto-organizados [3], divide el trabajo en una lista de entregables pequeños y concretos [3], llamados sprint e incrementos, asigna a cada actividad de la lista un orden de prioridad que se determina en colaboración con el cliente, en base a la revisión realizada de un entregable después de cada iteración y además estima el esfuerzo relativo que tiene cada actividad. Los roles más importantes de Scrum son: Product Owner (Propietario del producto), Scrum Master (facilitador) y Team (equipo de desarrollo) como se muestra en la figura 2. En la tabla 2 se muestra la descripción de roles que Scrum utiliza para el desarrollo de un proyecto de software.

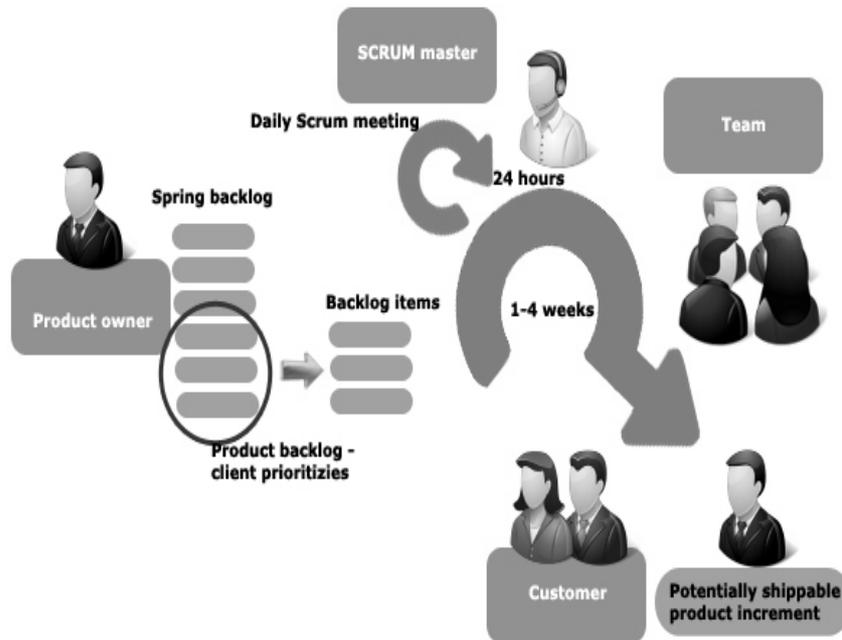


Fig. 2. Metodología ágil Scrum.

Tabla 2. Roles de Scrum.

Nombre	Descripción
Product Owner	Es la persona que se encarga del backlog

	(lista de requerimientos de un proyecto), es el cliente o un representante.
Scrum Master	Es la persona que se encarga de que las reglas se hagan efectivas, no es el líder del equipo.
Team	Es el equipo de desarrollo.

Scrum divide el tiempo en iteraciones cortas de longitud fija (generalmente de 1 a 4 semanas), con código potencialmente entregable y demostrado después de cada iteración y optimiza el proceso mediante una retrospectiva después de cada iteración [3].

En la Tabla 3 se muestran los productos que utiliza Scrum: Product Backlog, Sprint Backlog, Increment of potentially shippable product functionality y Burndown Chart.

Tabla 3. Productos de Scrum.

Nombre	Descripción
Product Backlog	Es una lista de los requerimientos del proyecto para realizar un producto de software completo y funcional, la lista tiende a cambiar con el tiempo de acuerdo a las condiciones.
Sprint Backlog	Es una lista de tareas que el equipo de desarrollo define para las iteraciones o sprint.
Increment of potentially shippable product functionality	Es un producto de software que cumple con todos los requerimientos funcionales para el sprint.
Burndown Chart	Es una gráfica donde se muestra el trabajo a lo largo del sprint.

4 Kanban

El método Kanban en el desarrollo de software fue originado en 2004, cuando David J. Anderson [6] estaba ayudando a un pequeño equipo de TI de Microsoft que estaba funcionando inadecuadamente.

El método Kanban en el desarrollo de software maneja equipos de proyectos para visualizar el flujo de trabajo, limita el trabajo en progreso (WIP) en cada etapa del flujo de trabajo y mide el tiempo de ciclo (lead time) como se muestra en la tabla 4.

Tabla 4. Principales características de Kanban.

Característica	Descripción
Visualiza el flujo de trabajo	Divide el trabajo en bloques, escribe cada elemento en una tarjeta y se coloca en el tablero. Utiliza columnas con nombre para ilustrar dónde está cada elemento en el flujo de trabajo.
Limita el trabajo en curso (WIP)	Asigna límites concretos al número de elementos que pueden estar en progreso en cada estado del flujo de trabajo.
Mide el tiempo de ciclo (lead time) medio para completar un elemento	Optimiza el proceso para que el lead time sea tan pequeño y predecible como sea posible.

El tablero Kanban proporciona visibilidad del proceso del software, en el muestra el trabajo asignado para cada desarrollador, comunica claramente las prioridades y resalta los cuellos de botella, así el equipo se concentra en resolver los problemas que bloquean el proceso y restauran el flujo productivo. En la figura 3 se muestra un ejemplo de un tablero básico Kanban.

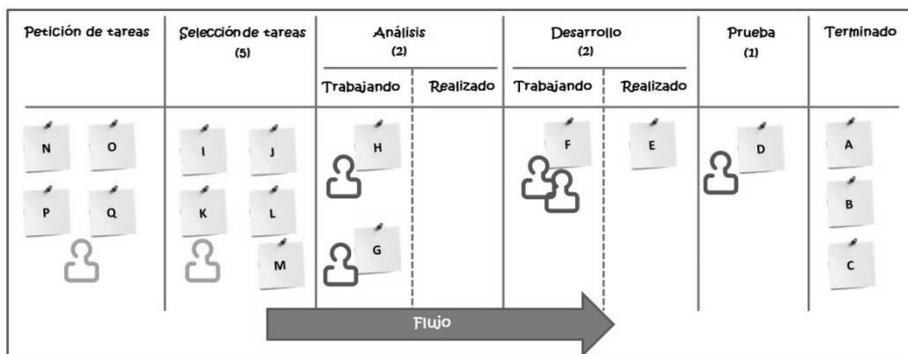


Fig. 3. Tablero Kanban.

Kanban limita el trabajo en curso de acuerdo a la capacidad del equipo, que equilibra la demanda contra el rendimiento del trabajo liberado por el equipo. Esto ayuda a visualizar los problemas del proceso, minimiza los defectos y mantiene un flujo estable. Al limitar el trabajo en curso se consigue un ritmo de desarrollo sostenible, elevando la calidad de los productos y un mayor rendimiento de los integrantes del equipo. El flujo estable y la calidad en el producto ayuda a reducir el tiempo de ciclo (lead time), generando la libe-

ración de entregables de forma más regular que incrementa la confianza del cliente hacia la empresa desarrolladora de software [5].

5 Moprosoft-Scrumban

La propuesta que se pretende implementar es la combinación de las mejores prácticas de desarrollo de software de las metodologías ágiles Scrum y Kanban, con la finalidad de apoyar al proceso de Desarrollo y Mantenimiento de Software (DMS) dentro de la empresa Miracle Business Network S.A. de C.V. que actualmente se encuentra certificada en el nivel 2 de la norma Moprosoft.

En base a las fases del proceso DMS de MoProSoft descritas en la parte superior, se propone un tablero de control Kanban como se muestra en la figura 4. Debe tomarse en cuenta los recursos humanos programados o planeados por el Responsable de Administración del Proyecto Específico (RAPE) y el rol de cada integrante del equipo de desarrollo que especifica la norma MoProSoft para el proceso DMS (ver figura 5), se establece el límite de trabajo en curso (WIP) en cada etapa del flujo de trabajo de un sprint o iteración para un entregable.

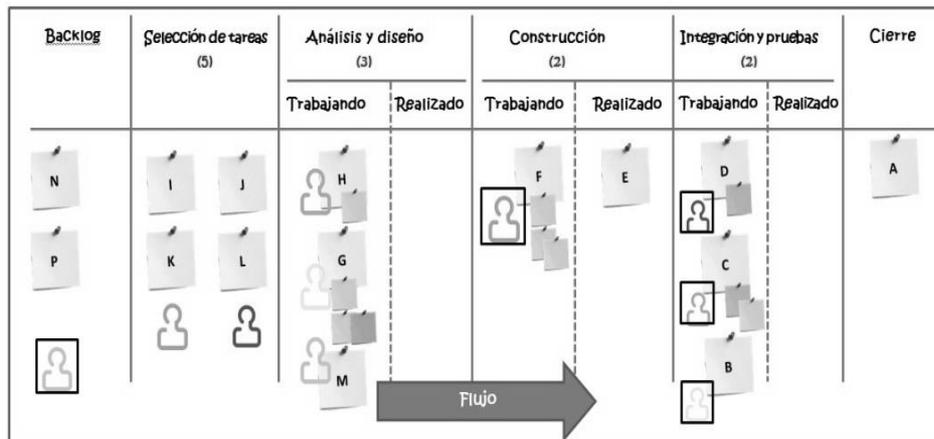


Fig. 4. Tablero MPS-Scrumban.

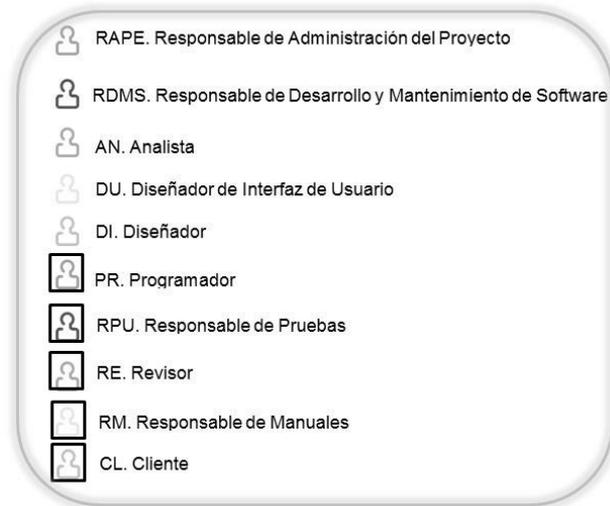


Fig. 5. Roles del proceso Desarrollo y Mantenimiento de Software MoProSoft.

Cada tarjeta visual debe contener la siguiente información: descripción de la actividad, fecha en que el elemento se agrega al tablero, fecha límite de entrega (si aplica), el nivel de prioridad y el nombre de la persona que trabaja en la actividad en el momento actual [4] como se muestra en la figura 6. Los integrantes del equipo, al momento de comenzar a realizar una actividad que se encuentre en la columna de selección de tareas deben tomar en cuenta la información sobre el dato de prioridad que se muestre en la tarjeta, se debe considerar en primer lugar las actividades marcadas como “Urgente”, posteriormente las de prioridad “Alta”, sino existe ninguna tarjeta con ese tipo de prioridades se debe considerar aquellas en donde la fecha límite de entrega está en riesgo y finalmente las actividades más antiguas, como se detalla en la tabla 5.



Fig. 6. Tarjeta visual para tablero Kanban.

Una historia o característica de lo que el cliente o propietario del producto necesita, consta de un conjunto de tareas que se tienen que hacer para poder implementar dicha característica en el producto final que se entrega al cliente. Las características (historias) son productos entregables. Estas fluyen a través del tablero de izquierda a derecha y su estado en el flujo de trabajo está indicado por la columna en donde esta se encuentra. El límite de trabajo en curso (WIP) en cada columna aplica solo para las características, no para las tareas [4].

Tabla 5. Prioridad de actividades.

Prioridad	Descripción
Urgente	Debe ser inmediatamente atendida y mantenida en constante movimiento por cada etapa del tablero. Se puede interrumpir el trabajo que se esté realizando en ese momento e incluso cambiar el límite del trabajo en curso (WIP) si es necesario.
Alta	Debe ser atendida una vez que se termine la actividad que se esté realizando en ese momento, este tipo de prioridad no puede cambiar el límite del trabajo en curso (WIP).
Fecha de límite	Se debe atender solamente si la fecha de límite de entrega está en riesgo.
Fecha en que se agrega al tablero	Se debe atender la que tenga fecha de puesta en el tablero más antigua.

Las tareas forman parte de una característica (historia), es decir, una característica puede tener una o más tareas que tiene que realizar un integrante del equipo para poder ser implementada. Las tareas no fluyen a través del tablero, pero su estado es indicado por tarjetas de colores [4], como se muestra en la figura 7, tarjeta en color azul indica que se trata de una tarea que se lleva a cabo para una determinada característica, verde indica tarea completada y roja tarea bloqueada, todas estas durante la etapa de **Construcción**; naranja indica que se trata de un defecto encontrado en la etapa de **Integración y Pruebas**.

Cuando una característica llega a la etapa de **Construcción - Realizado** entonces todas las tareas son apartadas. Si existen defectos encontrados durante la etapa de **Integración y Pruebas** entonces tarjetas de defectos en color naranja son agregadas dentro de columna **Trabajando** de dicha etapa.

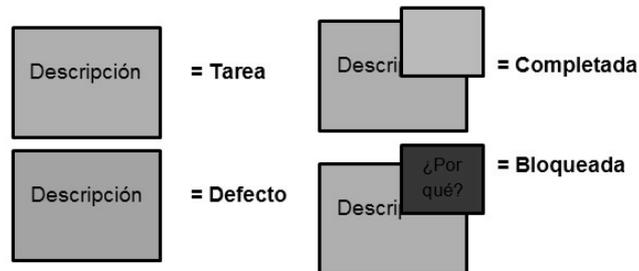


Fig. 7. Tarjetas de colores que indican el estado de las tareas.

6 Conclusiones

Las metodologías ágiles para desarrollo de software son herramientas que ayudan a organizar, administrar y controlar mejor las actividades que se llevan a cabo para la implementación de un sistema o aplicación de software. Con la proliferación de empresas de consultoría y desarrollo de software en México, es primordial que una empresa tenga ventaja competitiva en el mercado de las tecnologías de la información y comunicación con respecto a las demás, para irse posicionando en el medio y posteriormente convertirse y/o mantenerse como líder en soluciones que se ofrecen a los clientes, realizando el trabajo dentro del tiempo y costo planeado, cumpliendo con las necesidades y expectativas de los usuarios finales y entregando productos de calidad. Como dice Henrik Kniberg, “lo importante no es la herramienta con que se empieza, sino la forma en que se mejora el uso de esta y además se expande el conjunto de herramientas con el tiempo”, por tal motivo en este trabajo se ha propuesto el uso de las mejores prácticas de dos metodologías de desarrollo ágil, Scrum y Kanban bajo la norma MoProsoft para empresas mexicanas y de esta manera competir en proyectos de negocios globalizados con cualquier empresa del mundo.

Referencias

1. H. Oktaba, et al.: MoProSoft: Process Model to Industry Software, Ministry of Economy, Mexico (2005).
2. M. Dávila Muñoz, H. Oktaba: Especialización de MoProSoft basada en el Método Ágil Scrum (MPS-Scrum),
3. H. Kniberg , M. Skarin: Kanban and Scrum - making the most of both, USA (2009)

4. H. Kniberg: Kanban kick-start example, Noviembre 2009. [En línea]. Disponible: <http://www.crisp.se/gratis-material-och-guider/kanban>. [Último acceso: 25 octubre 2013].
5. M. Ovais Ahmad, J. Markkula, M. Ovio: Kanban in software development: A systematic literature review, 2013 39th Euromicro Conference Series on Software Engineering and Advanced Applications, (2013).
6. D. J. Anderson: Kanban: Successful Evolutionary Change for Your Technology Business. Sequim, WA: Blue Hole Press, (2010).
7. The Standish Group International: CHAOS Manifesto 2013, 2013. [En línea]. Disponible: <http://www.standishgroup.com>, (2013).

Especificación de requerimientos con Áncora y el estándar 830

Nora Isela Zamora Aguilar¹, Ángeles Arenas Valdés¹, Karen Cortés Verdín¹, Juan Carlos Pérez Arriaga¹

¹ Universidad Veracruzana, Facultad de Estadística e Informática,
Xalapa, Ver., México

nizaguilar@hotmail.com, aarenas@uv.mx, kcortes@uv.mx, juaperez@uv.mx

Resumen. Una de las fases iniciales en la ingeniería de software es la obtención de requerimientos. Éstos son elementos medulares en el desarrollo de software, por lo cual es de gran importancia el llevar una metodología para su obtención y análisis; ya que de lo contrario, la lista de requerimientos puede ser tan errónea que causaría el fracaso total del proyecto. Se destaca la importancia e impacto de la ingeniería de requerimientos en la ingeniería de software; así como la descripción y ventajas al desarrollar las actividades metodológicamente. Se aplica la metodología Áncora para la obtención y análisis de requerimientos en un caso de estudio, y se muestra la Especificación de Requerimientos de Software (ERS) utilizando el estándar IEEE 830, complementándolo con los artefactos que se generan en la metodología utilizada.

Palabras Clave: Áncora, Requerimientos de software, IEEE 830

1 Introducción

La especificación de requerimientos permite realizar una descripción de lo que hará el software para cubrir las necesidades del usuario y garantizar así su calidad. La finalidad del presente documento es resaltar la importancia de seguir una metodología y apegarse a estándares, que aseguren el desarrollo de software de calidad. organizándolo de la siguiente manera: en la segunda sección se presentan los fundamentos de la ingeniería de software; en la tercera sección se describe la metodología Áncora y se indica en qué fase de la ingeniería de requerimientos se aplican las herramientas y artefactos de ésta metodología; en la cuarta sección se describe el estándar IEEE 830; en la quinta sección, se desarrolla la plantilla de especificación de requerimientos combinando Áncora y el estándar IEEE 830, la cual fue utilizada en un caso de estudio; por último en la sexta sección se describen los resultados obtenidos al aplicar la metodología en un caso de estudio.

2 Ingeniería de requerimientos

Un punto de referencia en la ingeniería de software es SWEBOK[1], cuya finalidad es proporcionar un compendio que muestra el conocimiento de dicha área, el cual se resume en 10 áreas de conocimiento de los conceptos básicos: Requerimientos de Software; Diseño de software; Construcción de Software; Pruebas de Software, Mantenimiento de Software; Gestión de Configuración del Software; Gestión de la Ingeniería de software; Procesos de Ingeniería de Software; Herramientas y métodos de la Ingeniería del Software y Calidad.

De todas las áreas antes mencionadas la primera corresponde a la identificación, definición, o clasificación de requerimientos, descripción de la importancia de los requerimientos cuantificables, las diferencias entre producto y proceso; y distingue entre sistema y software.

Los requerimientos describen los modelos de los procesos, actores, procesos de apoyo y gestión de procesos, calidad y mejora de procesos. El proceso de evaluación de los documentos de requerimientos permite garantizar la definición del sistema adecuado; es decir lo que el usuario espera; también se consideran prototipos y validación de los modelos, así como las pruebas de aceptación.

De acuerdo a [2], el proceso de ingeniería de requerimientos está conformado por cuatro actividades, en cada una de ellas se genera un documento, como lo muestra la figura 1.

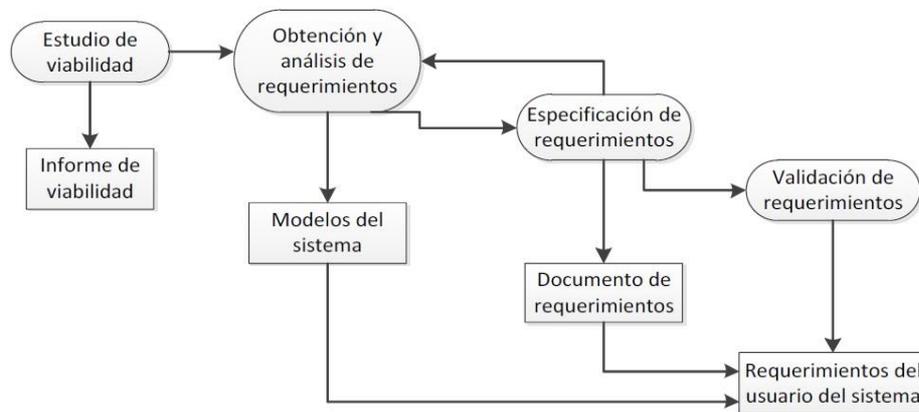


Fig. 1. Procesos de Ingeniería de requerimientos [2].

La especificación de requerimientos hace referencia a la elaboración de un documento con la finalidad de validar los requerimientos; este documento incluye los requerimientos del usuario y los requerimientos del sistema [2].

3 Áncora como metodología para la obtención de requerimientos

Áncora es una metodología que ayuda al análisis de requerimientos de software, es el acrónimo de análisis de requerimientos de software conducente al reuso de artefactos [3]. Esta metodología permite conocer claramente qué es lo que se quiere; así como el elaborar representaciones que sean claras para el usuario y poder continuar sin errores en las siguientes etapas de la ingeniería de software.

Considerando la figura 1, la cual muestra el proceso de ingeniería de requerimientos, Áncora se aplica en la segunda etapa, la cual corresponde a la obtención y análisis de requerimientos.

3.1 Ciclo de vida de Áncora

El ciclo de vida de Áncora está conformado por cinco etapas, en cada una de ellas se generan artefactos, tal como se muestra en la figura 2.

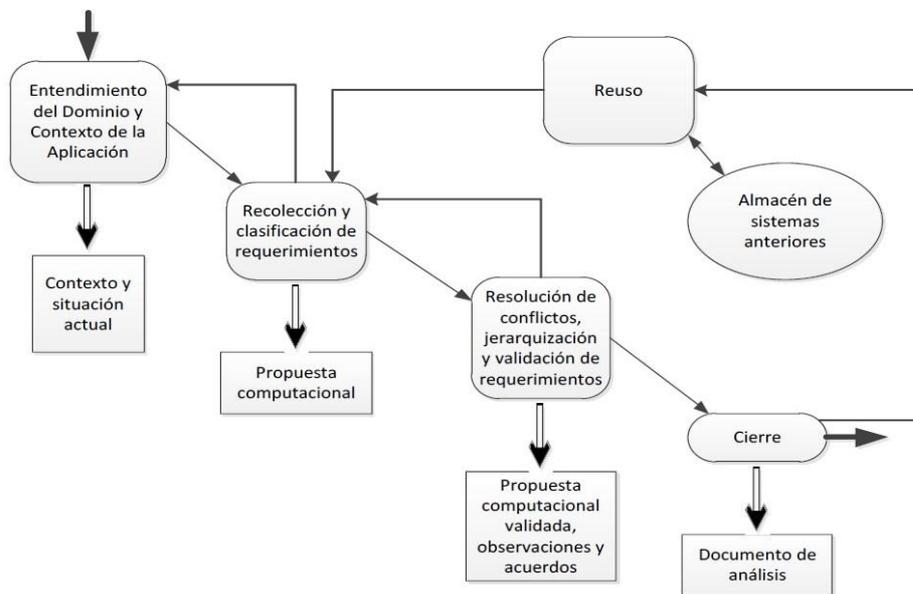


Fig. 2. Ciclo de vida de Áncora [3].

Entendimiento del Dominio y Contexto de la aplicación. El objetivo de esta etapa es identificar y aclarar las metas, dominio, actividades, situación actual del sistema y justificación del nuevo software. Las actividades que se llevan a cabo son lectura de material, entrevistas informales con el cliente, observación a usuarios, construcción de redes semánticas naturales del sistema; así como la construcción y aplicación de encuestas de actitud.

Recolección de requerimientos. En esta etapa el objetivo es desarrollar la propuesta computacional, construyendo el guión, biblioteca de desarrollo y aplicación de puntos de función.

Reuso de requerimientos. Esta etapa solo puede ser incluida si existe una base de datos con requerimientos o elementos de otros sistemas que hayan sido aprobados con anterioridad.

Resolución de conflictos, priorización y validación de requerimientos. El objetivo es resolver posibles incongruencias o conflictos entre los requerimientos identificados, asignar la prioridad a cada uno de ellos y validarlos con el usuario. Se realizan dos actividades la preparación de la reunión de reflexión y diseño, y la reunión como tal.

Cierre. Es la última fase del ciclo de vida, en ella se realizan actividades para conectar las siguientes etapas de la ingeniería de software; como el ratificar los puntos de función, envío de comunicados para los puntos pendientes y la conexión con la metodología de desarrollo seleccionada

3.2 Herramientas de Áncora

En Áncora se cuenta con diferentes herramientas, las cuales han sido tomadas de áreas distintas, como el teatro, la psicología, administración y por supuesto computación.

Las herramientas utilizadas en esta metodología son:

- Guiones y diálogos: son utilizados para la obtención de requerimientos, permitiéndole al analista tener una clara comunicación con los usuarios, así como el mostrar detalles iniciales a los diseñadores. Generando como resultado el guión de la propuesta computacional.
- Tablas y glosarios: Son utilizadas para poder reducir ambigüedades en la especificación de requerimientos por el uso del lenguaje natural; así como para aclarar términos computacionales o del área de aplicación del software.
- Modelo de datos Entidad-Relación: Permite la delimitación correcta de las fronteras del sistema [4], y el modelado de la base de datos.
- Análisis de puntos de función: Es utilizado para calcular los costos para un nuevo software, utiliza cinco indicadores para identificar cada uno de los elementos del software.
- Redes semánticas naturales: Son utilizadas para identificar los conceptos de los usuarios y eliminar las ambigüedades entre éstos.

- Técnicas de grupo nominal: Su finalidad es la toma de decisiones organizacionales para dar solución a la problemática.
- Reunión de reflexión y diseño: Tiene como objetivo obtener los modelos que permiten identificar lo que se requiere que el software realice. Se divide en cinco etapas: exploración del ambiente, análisis interno del sistema, diseño del software ideal, identificación de obstáculos y oportunidades, selección de líneas de acción y diseño de líneas de acción.

Beneficios de utilizar Áncora.

Áncora es metodología que utiliza recursos de fácil comprensión para el usuario, quien puede o no estar familiarizado con los términos técnicos del área de computación, lo que permite una participación activa. El uso de los guiones, utilizados en el teatro y literatura facilitan al usuario la comprensión del desarrollo de cada uno de los procesos, cómo éstos pueden ser implementados en un sistema automatizado y quienes son los usuarios (actores) que intervienen en ellos.

Los recursos y dinámicas empleados para la elaboración de redes semánticas permiten identificar de forma rápida aquellas acciones que son relevantes en los procesos que se implementarán en el software, también ayudan enormemente para la eliminación de ambigüedad en los conceptos involucrados en el contexto de éste.

Las reuniones de reflexión y diseño son de gran apoyo para reunir a los involucrados con el software a desarrollar, proporcionando una guía puntal para que éstas sean lo más efectivas posibles y obtener el mayor provecho de ellas sin necesidad de requerir mucho tiempo.

Todo lo anterior proporciona las herramientas necesarias para la obtención de requerimientos de forma metodológica y certera de lo que realmente necesita el usuario, ya que también facilita el cercamiento con él.

Siguiendo las fases de la metodología Áncora y generando los artefactos antes mencionado, se obtienen los requerimientos asegurando así que el software cubra todas las necesidades del usuario.

4 Estándar IEEE 830 para la especificación de requerimientos

La falta de herramientas automatizadas y poca habilidad de los ingenieros de software, en la creación de una buena especificación de requerimientos en lenguaje natural, han sido un obstáculo en el proceso eficiente de la ingeniería de requerimientos, por lo que se han desarrollado algunas propuestas, entre las que se encuentra un software Tailandés [5], NALASS [6] y herramientas que permiten la creación y verificación de especificación de requerimientos desde puntos de vista diferentes [7].

El estándar IEEE 830-1998 [8], “Recomendaciones prácticas para la especificación de requerimientos”, fue desarrollado por la IEEE y la IEEE-SA (Standards Association),

indica la estructura y organización de toda la información que debe incluirse en un buen documento de especificación de requerimientos de software. Los objetivos son: Ayudar a los clientes de software para describir con precisión lo que quieren; y a las personas para lograr: Desarrollar una estructura estándar para la especificación de requerimientos de software (ERS) en sus organizaciones; Definir el formato y contenido de las especificaciones de requerimientos de software y Desarrollar elementos de apoyo adicionales, lista de control de calidad de ERS o un manual del mismo.

La especificación de requerimientos de software obliga a los involucrados en el desarrollo del software a considerar todos los requerimientos de forma rigurosa antes de iniciar el diseño y codificación del mismo, con la finalidad de evitar el rediseño, proporcionando las bases necesarias para la estimación de tiempo y costo, referencias de verificación y validación.

En el 2011 surge el estándar ISO/IEC/IEEE 29148[9], en el cual unifica los procesos y productos involucrados en la ingeniería de requerimientos durante todo el ciclo de vida de los sistemas y software, es el resultado de la unión de los siguientes estándares: ISO/IEC 12207:2008, ISO/IEC 15288:2008, ISO/IEC/IEEE 15289:2011, ISO/IEC TR 19759, IEEE Std 830, IEEE Std 1233, IEEE Std 1362, ISO/IEC TR 24748-1, e ISO/IEC/IEEE 24765.

Es importante mencionar que uno de los puntos importantes del ISO/IEC/IEEE 29148[9], es que se mencionan los procesos y dentro de éstos las actividades que se deben llevar a cabo para una buena obtención de requerimientos, que es precisamente una de las ventajas de utilizar una metodología, en este caso *Áncora*, por lo que con los artefactos que se generan a partir de ésta, se complementa una de las plantillas propuesta por el estándar IEEE 830 y se muestra su aplicación en un caso de estudio.

5 Especificación de requerimientos de Software utilizando *Áncora* y el estándar IEEE 830

El caso de estudio analizado para la elaboración de la plantilla de especificación de requerimientos, corresponde a un software para la admisión y control escolar para el Instituto Educativo Panamericano (SIIEP-Primaria), en el nivel primaria, cuyo objetivo principal del software es automatizar los procesos de preinscripción e inscripción que actualmente realizan de forma manual y en el cual intervienen muchas personas al inicio del nuevo ciclo escolar primaria.

Como inicio de este proyecto se llevaron a cabo las actividades que permiten enlazar los requerimientos identificados con el resto de las fases y actividades de la ingeniería de software; en esta ocasión se realizó el documento de especificación de los requerimientos tomando como guía el estándar IEEE 830-1998[8] quedando la plantilla del documento con se muestra en la Tabla 1.

Es importante resaltar que el documento de Especificación de Requerimientos de Software, debe cumplir las siguientes características de calidad [10], las cuales son: Completo, Consistente, Exacto, Modificable, Clasificado, Comprobable, Trazable, Inequívoco, Válido y Verificable y que fueron consideradas al momento de elaborar el documento de especificación de requerimientos.

Tabla 1. Plantilla para la especificación de requerimientos.

Sección	Descripción	Artefactos de Ancora
1. Introducción		
1.1. Propósito	Define el propósito y a quién va dirigido el documento de especificación de requerimientos.	Entrevista
1.2. Ámbito del Sistema	Proporciona el nombre futuro del sistema, lo que hará o no, beneficios, objetivos y metas que se esperan alcanzar.	Entrevista
2. Descripción general del sistema		Red semántica natural
2.1. Contexto del sistema		Red semántica natural
2.2. Definiciones, Acrónimos y Abreviaturas	Incluye la introducción y vista general de las especificaciones.	Tablas y glosarios
2.3. Situación actual	Conceptos básicos que se utilizan.	Guión de la situación actual
2.4. Problemática	Funciones principales que actualmente se realizan.	Red semántica natural
2.5. Descripción de usuarios	Descripción del papel que desempeña cada uno de los usuarios.	Perfilado de usuario
2.6. Suposiciones y dependencias		
2.7. Restricciones	Problemática presente en el proceso actual.	
2.7.1. Entorno operativo		
2.8. Propuesta computacional		Guión de propuesta computacional
3. Descripción de requerimientos	Describe los requisitos del software que permita satisfacer los requerimientos del	
3.1. Interfaces externas		Prototipo

3.2. Requerimientos funcionales	sistema y satisfaga las necesidades del usuario. Incluye modelos que permitan dar continuidad a la ingeniería de software.	Casos de uso
3.3. Diseño de casos de uso		
3.4. Requerimientos de usabilidad		Red semántica natural y reunión de reflexión
3.5. Requerimientos de rendimiento		
3.6. Requerimientos de base de datos		Modelo de datos
3.7. Restricciones de diseño		Reunión de reflexión
3.8. Atributos del sistema		Reunión del reflexión
3.9. Otros requerimientos		Reunión del reflexión
4. Casos de prueba		Describe el plan de pruebas que se van llevar a cabo. Incluye los casos de prueba unitarios; así como los de integración.
4.1. Pruebas unitarias		
4.2. Pruebas de integración		
5. Apéndices	Información adicional que da soporte a los puntos anteriores	Entrevistas con los usuarios, redes semánticas, encuestas de actitud, gráficas, etc.

6 Resultados

La metodología de Áncora fue aplicada en el desarrollo software SIIEP-Primaria del Instituto Educativo Panamericano S. C., siguiendo todas las actividades necesarias, utilizando las herramientas propias de la metodología para generar los siguientes artefactos:

- Red semántica natural
- Encuesta de actitud
- Guión de la situación actual
- Guión de la propuesta computacional
- Perfilado de usuarios
- Casos de uso
- Modelo de datos

- Prototipo
- Especificación de usabilidad
- Cálculo de Puntos por casos de uso
- Planteamiento de casos de prueba

Se realizaron tres reuniones de Reflexión y Diseño de las cuales se obtuvo la priorización de requerimientos, se establecieron aspectos técnicos para la plataforma de desarrollo y algunas características no funcionales pero de importancia para el usuario final.

Con toda esta información fue posible generar un Documento con la especificación de los Requerimientos de Software (ERS) que cumplía las características mencionadas en la sección cinco (Tabla 2). Dicho documento permitió la fácil transición a las siguientes fases de desarrollo, ya que el sistema estaba bien delimitado, con los requerimientos claramente especificados, un conjunto de datos bien definidos en un modelo preliminar de datos y el prototipo que cubría en su totalidad los casos de uso. Todo lo anterior fue aprobado en una reunión de Reflexión y diseño, por lo tanto solo quedaba colocar el código correspondiente a cada uno de los eventos que conformaban el prototipo.

Tabla 2. Características del documento de ERS.

Característica	Validación realizada
Correcto	Cada requerimiento fue revisado y aceptado por los involucrados con el software en las reuniones de reflexión. Cada requerimiento tiene una descripción de lo que debe cumplir, definidas las entradas y salidas; y la relación con otros requerimientos
Sin ambigüedades	Cada requerimiento se apega a lo establecido en el proceso y los conceptos empleados fueron unificados con en los glosarios, acrónimos y abreviaturas
Completo	El documento contiene toda la información necesaria para comprender las necesidades del cliente y pasar a las siguientes fases de desarrollo.
Consistente	Los requerimientos son revisados en las reuniones de reflexión y diseñados para asegurar que los requerimientos no se contradigan Cada requerimiento tiene una descripción de lo que debe cumplir, definidas las entradas y salidas; y la relación con otros requerimientos
Importante y/o estable	Cada requerimiento fue priorizados por los usuarios en las reuniones de reflexión y agrupados de acuerdo a la funcionalidad y secuencia establecida por el proceso
Verificable	Cada uno de los requerimientos se incluyeron en los casos

	de uso, los cuales fueron revisados y aceptados por los usuarios desde la definición hasta su funcionalidad validada con la planificación y ejecución de pruebas
Modificable	Cada requerimiento tiene un identificador y se describe cada uno por separado para referirse a él se manera no ambigua
Trazable	Cada requerimiento es trazable a las siguientes fases, por ejemplo a los casos de prueba

7 Conclusiones y trabajos futuros

El documento muestra la importancia de la ingeniería de requerimientos para desarrollar correctamente un software que satisfaga las necesidades del usuario.

Describe las fases de la metodología Áncora, la cual proporciona herramientas y artefactos que son de gran utilidad en la definición de requerimientos.

También se presenta la estructura del documento de definición de requerimientos, la cual se elaboró tomando como guía el estándar IEEE 830-1998, complementándolo con diversos artefactos de Áncora; la estructura diseñada fue utilizada como el documento en el que se formalizaron los requerimientos del usuario y que sirvieron como base para dar continuidad a la ingeniería en el desarrollo del sistema SIIEP-Primaria.

El utilizar Áncora y el estándar IEEE830, permitieron el acercamiento y correcta comunicación con el usuario (siendo esta una prioridad para la metodología), facilitando la definición, claridad y priorización de los requerimientos; así como la identificación y establecimientos de procesos previos y necesarios antes de automatizar un proceso.

La aportación principal de este documento es mostrar la importancia de la ingeniería de requerimientos utilizando metodologías y estándares que facilitan la comunicación con el usuario; quien al estar involucrado y comprender la importancia de realizar todo el proceso de forma metodológica lo que da valor a cada una de las fases a desarrollar; así como unificar los conceptos utilizados en la organización.

Al concluir el presente documento, se tienen como trabajos futuros:

- El análisis de herramientas automatizadas para la especificación de requerimientos en lenguaje natural, que permita agilizar la elaboración del documento
- Elaboración de la plantilla de especificación de requerimientos, incorporando aspectos para la unificación de procesos y producto.

Referencias

1. A. Abran, J. W. Moore, P. Bourque, & R. Dupuis, (Edits), Guide to the Software Engineering Body Knowledge, Published by IEEE, Los Alamitos, (2004)
2. I. Sommerville, Ingeniería de Software, Séptima edición, Pearson Educación, S. A., Madrid España, (2006)
3. Sumano, Ma, de los Ángeles, Áncora: Análisis de requerimientos de software conducente al reuso de artefactos, Universidad Veracruzana, Xalapa, Ver., México, (2006)
4. Kendall, K. E., & Kendall, J. E., Análisis y diseño de sistemas, 6° edición, Pearson Educación de México, México, (2005)
5. Thongglin, K., Cardey, S., & Greenfield, P., Thai Software Requirements Specification Pattern. SoMeT 2013, 12th IEEE International Conference on Intelligent Software Methodologies, Tools and Techniques, (2003)
6. Georgiades, M., & Andreou, A., Automatic Generation of a Software Requirements Specification (SRS) Document. 10th International Conference on Intelligent Systems Design and Applications, (2010)
7. Matsuo, Y., Ogasawara, K., & Ohnishi, A., Software Requirements Specifications. 978-1-4244-4840-1/101 ©2010 IEEE, (2010)
8. IEEE, Institute of Electrical and Electronics Engineers, Especificación de Requerimientos según el estándar de IEEE-830-1998, Recuperado el 03 01, 2013, desde Institute of Electrical and Electronics Engineers: www.ieee.org, (1998)
9. ISO/IEC 2011, IEEE2011, International Standard ISO/IEC/IEEE 29148, Systems and software engineering – Life cycle processes – Requirements engineering, IEEE Standards Activities Department, Piscataway, NJ, USA, (2011)
10. INKtopia Limited., TECHwhirl. Recuperado el 13 de Julio de 2014, de Helping the world, explorer the art of content and communications: <http://techwhirl.com/writing-software-requirements-specifications/>, (2014)

Análisis del estado actual de certificaciones CMMI-DEV ver. 1.3 año 2013 y 2014, a nivel Mundial y en México

Yesenia Nohemí González Meneses, Nayely Yaline León Padilla, José Juan Hernández Mora, María Guadalupe Medina Barrera

Instituto Tecnológico de Apizaco, Departamento de Sistemas y Computación, Apizaco, Tlaxcala
yeseniaglez@hotmail.com, salome31_7@hotmail.com, jjhmora@msn.com,
lupita_medina@hotmail.com

Resumen. En este artículo, se presenta un análisis de las organizaciones desarrolladoras de software que a nivel mundial han sido evaluadas de acuerdo al modelo CMMI, en el apartado de CMMI-Dev en su versión 1.3 en los años 2013 y 2014. Este trabajo da como resultado el ranking mundial por país en cuanto a certificaciones en CMMI-DEV ver. 1.3 en sus cinco niveles de madurez, lo que refleja el interés de que las empresas desarrolladoras de software tienen en mejorar la calidad en sus procesos para el desarrollo de software y lo importante que en la actualidad es el hecho de obtener una certificación avalada internacionalmente. También se presenta un estudio enfocado a las empresas mexicanas que lograron certificarse en CMMI, en estos mismos años, apoyadas en algunos casos por programas gubernamentales que están impulsando la certificación internacional a través de la promoción de convocatorias que les ofrecen diferentes apoyos, encaminados al mejoramiento de la calidad de sus procesos.

Palabras clave: CMMI-DEV, mejora de procesos, certificaciones de calidad, desarrollo de software.

1. Introducción

Hoy en día, el desarrollo de software con calidad es una necesidad que las empresas desarrolladoras de software tienen, demandado por sus mismos clientes, que esperan que se les garantice la calidad en sus productos y que a la vez cumplan con especificaciones internacionales. Por muy pequeñas que estas empresas sean, no se pueden dar el lujo de sacrificar la calidad, existe una alta competencia de industrias extranjeras como las de los

países de India, China y EEUU, debido a que estos países tienen altos índices de certificaciones de calidad y mejora de procesos, lo que les permite garantizar a sus clientes la calidad de sus productos.

La competitividad de las empresas debe basarse en mejoras e innovación de sus procesos, que brinden una propuesta de valor a sus clientes. Es por ello que las empresas buscan una alternativa en la mejora continua de sus procesos clave mediante una estrategia de calidad que incluya la implantación de algún modelo o estándar de calidad formal y reconocido, como es el modelo CMMI (*Capability Maturity Model Integration*).

2. Breve historia de CMMI

El Instituto de Ingeniería de Software (SEI, Software Engineering Institute), fue fundado en el año de 1984 mediante el financiamiento del Departamento de Defensa de Estados Unidos, es administrado por la Universidad de Carnegie Mellon. Este instituto fue creado con el objetivo de desarrollar modelos de evaluación y mejora en el desarrollo de software, que dieran solución a los problemas de planeación y desarrollo de software, durante la construcción de sistemas militares del ejército de Estados Unidos.

Desde su creación el SEI, se ha centrado en proporcionar la base para mejorar el desarrollo de software, basado sobre el principio de que la calidad del producto depende principalmente de la calidad de los procesos empleados en su desarrollo y considerando a las tareas de desarrollo del software como una serie de procesos que se puedan definir, medir y controlar [1].

El SEI ha creado un conjunto de modelos utilizados para incrementar la calidad del software, entre los cuales se encuentran CMMI-DEV, CMMI-ACQ, CMMI-SVC, SCAMPI E IDEAL. Los modelos de CMMI son colecciones de buenas prácticas que ayudan a las organizaciones a mejorar sus procesos [1].

2.1. Marco CMMI

El marco CMMI brinda la organización necesaria para establecer los modelos del CMMI y los componentes de evaluación de CMMI. El marco CMMI permite el uso de múltiples modelos, para ello aloja componentes comunes a todos los modelos CMMI aplicables a cualquier modelo. El material común es nombrado “CMMI Model Foundation” o “CMF”, los componentes del CMF son utilizados para conformar todos los modelos generados a partir del marco CMMI, esos componentes son armonizados con el material aplicable a un área de interés para crear un modelo. La colección de componentes que son utilizados para construir modelos, materiales de entrenamiento y materiales en el área de interés son llamados “constelación”.

Las constelaciones que han sido definidas para CMMI actualmente son tres [2]:

- + CMMI para el desarrollo (CMMI-DEV), brinda las guías necesarias para medir, controlar y gestionar los procesos de desarrollo.
- + CMMI para la adquisición (CMMI-ACQ), brinda una guía para controlar y gestionar la adquisición de productos y servicios que cumplan con las necesidades del cliente.
- + CMMI para los servicios (CMMI-SVC), brinda guías para aquellos que proveen servicios dentro de las organizaciones y a clientes externos.

CMMI-DEV

El modelo CMMI para Desarrollo, brinda un conjunto de guías completas e integradas para desarrollar productos y servicios, además de brindar orientación para aplicar las buenas prácticas CMMI en una organización de desarrollo. Las buenas prácticas del modelo se centran en las actividades para desarrollar productos y servicios de calidad con el objetivo de cumplir las necesidades de clientes y usuarios finales [2].

CMMI-DEV contiene 22 áreas de proceso [2], un área de proceso es un conjunto de prácticas que se relacionan y que cuando se implementan en conjunto satisfacen una serie de metas consideradas importantes para mejorar esa área.

Las 22 áreas de proceso se presentan a continuación por orden alfabético de sus acrónimos en inglés [3]:

- | | |
|--|--|
| 1. Análisis Causal y Resolución (CAR). | 12. Monitorización y Control del Proyecto (PMC). |
| 2. Gestión de Configuración (CM). | 13. Planificación del Proyecto (PP). |
| 3. Análisis de Decisiones y Resolución (DAR). | 14. Aseguramiento de la Calidad del Proceso y del Producto (PPQA). |
| 4. Gestión Integrada del Proyecto (IPM). | 15. Gestión Cuantitativa del Proyecto (QPM). |
| 5. Medición y Análisis (MA). | 16. Gestión de Requisitos (REQM). |
| 6. Definición de Procesos de la Organización (OPD). | 17. Desarrollo de Requisitos (RD). |
| 7. Enfoque en Procesos de la Organización (OPF). | 18. Gestión de Riesgos (RSKM). |
| 8. Gestión del Rendimiento de la Organización (OPM). | 19. Gestión de Acuerdos con Proveedores (SAM). |
| 9. Rendimiento de Procesos de la Organización (OPP). | 20. Solución Técnica (TS). |
| 10. Formación en la Organización (OT). | 21. Validación (VAL). |
| 11. Integración del Producto (PI). | 22. Verificación (VER). |

3. Objetivo y alcance del estudio

El objetivo de este trabajo es presentar un análisis de la información estadística de organizaciones que han sido evaluadas por el CMMI Institute, en la constelación de CMMI-DEV versión 1.3 en los años 2013 y 2014.

En la página oficial del CMMI Institute en “Published Apraisal Results” [4] se encuentran publicados los resultados de las evaluaciones realizadas a las organizaciones que implantaron algún nivel de CMMI, en cualquiera de sus constelaciones y versiones (ya sea versión 1.2 y 1.3). En esta página se pueden consultar resultados por modelo/constelación, nivel de madurez, año (que va desde el año 2011 hasta los resultados más actuales de 2014) y país.

Este trabajo se centrará en los resultados de las evaluaciones de 2013 en CMMI-DEV ver. 1.3, debido a que son los datos más actuales y completos hasta el momento. También se analizarán de una manera breve los cuatro primeros lugares en certificaciones en lo que va del año 2014.

4. Análisis estadístico

Al finalizar el año 2013, existen 1389 certificaciones activas en CMMI- DEV ver 1.3 alrededor del mundo, estas certificaciones están repartidas en 53 países. Debido a que existen diferentes tipos de constelaciones, algunas empresas pueden tener más de una certificación activa, también se da el caso de que existan diversas certificaciones compartidas entre diferentes países, un ejemplo de ello es la organización Tata Consultancy Services Limited (TCS) que está presente en los siguientes países: Argentina, India, Brasil, China, Colombia, Estados Unidos, Paraguay, Chile, Ecuador, México y Uruguay. En este estudio la certificación cuenta por separado para cada país debido a que la certificación incluye una evaluación por país de manera independiente [4].

A continuación en la Tabla 1, se muestran el número total de certificaciones obtenidas por país, divididas por niveles de madurez.

Tabla 1. Listado de certificaciones CMMI-DEV ver 1.3 año 2013, por país.

<i>Núm.</i>	<i>País</i>	<i>Nivel 5</i>	<i>Nivel 4</i>	<i>Nivel 3</i>	<i>Nivel 2</i>	<i>Total por País</i>
1	China	31	26	558	6	621
2	EEUU	20	2	157	72	251
3	India	36	0	115	4	155
4	México	4	1	27	17	49
5	Brasil	3	1	18	14	36

6	España	4	0	10	21	35
7	República de Corea	2	2	16	11	31
8	Japón	0	5	14	4	23
9	Francia	0	0	6	12	18
10	Colombia	5	0	9	2	16
11	Turquía	1	0	11	1	13
12	Tailandia	2	0	6	3	11
13	Taiwán	0	1	9	1	11
14	Italia	0	0	7	3	10
15	Argentina	2	0	4	3	9
16	Perú	0	0	8	1	9
17	Canadá	1	0	4	3	8
18	Portugal	2	0	3	3	8
19	Vietnam	2	0	5	0	7
20	Alemania	0	0	2	4	6
21	Chile	2	0	1	3	6
22	Egipto	1	0	3	1	5
23	Australia	1	0	2	1	4
24	Filipinas	3	0	1	0	4
25	Malaysia	1	0	3	0	4
26	Reino Unido	0	0	3	1	4
27	Bélgica	0	0	1	2	3
28	Pakistán	0	0	2	1	3
29	Singapur	1	0	2	0	3
30	Hong Kong	1	0	0	1	2
31	Luxemburgo	0	0	1	1	2
32	Países bajos	0	0	0	2	2
33	Rusia	2	0	0	0	2
	Total por Nivel	132	38	1016	205	1389

Como se puede apreciar en la Tabla 1, el país con mayor número de certificaciones totales es China con un total de 621, le siguen Estados Unidos con 251, India con 155, México con 49 y Brasil con 36 certificaciones. Los países que solo obtuvieron una certificación en el año 2013 en CMMI-DEV ver. 1.3, aunque no aparecen listados en la tabla por cuestiones de espacio fueron: Arabia, Austria, Bangladesh, Bulgaria, Chipre,

Ecuador, Grecia, Indonesia, Israel, Jordán, Kenia, Kuwait, Letonia, Polonia, Sir Lanka, Suecia, Suiza, Paraguay, Rumania y Uruguay.

Con lo que respecta al número de certificaciones en el Nivel 5 de madurez, existen 132 a nivel mundial, el país que ocupa el primer lugar en lo que se refiere a certificaciones en este nivel es India con 36, en segundo lugar se encuentra China con 31, Estados Unidos ocupa la tercera posición con 20 certificaciones, en cuarto lugar se encuentra Colombia con 5 certificaciones y en la quinta posición se encuentran México y España con 4 certificaciones, en la sexta posición se encuentra Brasil y Filipinas con 3 certificaciones, en séptimo lugar se encuentran República de Corea, Tailandia, Argentina, Portugal, Vietnam, Chile y Rusia con 2 certificaciones y los países que cuentan con una certificación son Paraguay, Uruguay, Ecuador, Rumania, Israel, Egipto, Australia, Canadá, Turquía, Singapur y Hong Kong [4]. Esta información se lista en la Tabla 2.

Tabla 2. Listado de certificaciones en el nivel 5 de CMMI-DEV ver. 1.3 año 2013, por país.

<i>Posición</i>	<i>País</i>	<i>Certificaciones en Nivel 5</i>
1	India	36
2	China	31
3	EEUU	20
4	Colombia	5
5	México	4
5	España	4

Una de las empresas más importantes a nivel mundial es Tata Consultancy Services Limited, México, que es parte fundamental de TCS Latinoamérica, y es brazo de negocios de TCS que opera a lo largo de toda la región latinoamericana, con Centros de Entrega Global (GDC), en Argentina, Brasil, Chile, Colombia, Ecuador, Perú y Uruguay en el nivel 5 de CMMI, el estándar de calidad más alto de la industria de software; cuenta con centros de servicios de tercerización de procesos de negocios en Chile y Uruguay.

Todos estos Centros de Entrega Globales cuentan con certificación en altos niveles de calidad y servicio, así mismo, la empresa cuenta con un sistema interno de calidad, IQMS alineado a estándares internacionales como: *ISO 9001*, *ISO 20000*, *ISO 27000*, *TL 9000*, y a los modelos *CMMI Nivel 5* y *PCMM*. Además la empresa opera cumpliendo los estándares más altos de infraestructura y seguridad física, de redes, del personal, y de sistemas [5].

Algunos de sus principales clientes son: Aeromexico, IMSS, Bank of America, GE, JCI, Ceridian, Verizon, Banco Santander, BAC Credomatic, Banco Interamericano de Desarrollo, Banamex y América Móvil, entre otros [5].

Tabla 3. Listado de certificaciones en el nivel 4 de CMMI-DEV ver. 1.3 año 2013, por país.

<i>Posición</i>	<i>País</i>	<i>Certificaciones en Nivel 4</i>
1	China	26
2	Japón	5
3	EEUU	2
3	República de Corea	2
4	México	1
4	Brasil	1
4	Taiwán	1

En lo que respecta a certificaciones en el nivel 4 de madurez, China se encuentra en primer lugar con 26 certificaciones, le sigue Japón con 5, Estados Unidos y Republica de Corea con 2, y finalmente México, Brasil y Taiwán con 1 respectivamente [4]. La información se muestra en la Tabla 3.

En las estadísticas internacionales del nivel 3 de madurez en CMMI-DEV ver 1.3, el primer lugar lo ocupa India con 115 certificaciones, en segundo lugar le sigue México con 27, en tercer lugar con 18 certificaciones se encuentra Brasil, en cuarto lugar se encuentra República de Corea con 16 certificaciones y Japón ocupa el quinto lugar con 14 certificaciones [4], esto lo podemos observar en el listado de la Tabla 4.

Tabla 4. Listado de certificaciones en el nivel 3 de CMMI-DEV ver. 1.3 año 2013, por país.

<i>Posición</i>	<i>País</i>	<i>Certificaciones Nivel 3</i>
1	India	115
2	México	27
3	Brasil	18
4	República de Corea	16
5	Japón	14
6	Turquía	11
7	España	10

En lo que corresponde al nivel 2 de madurez en CMMI-DEV ver 1.3, los primeros seis lugares corresponden a los siguientes países, el primer lugar en certificaciones es Estados Unidos con 72, seguido por España con 21, México con 17, Brasil con 14, Francia con 12 y República de Corea con 11 certificaciones [4], esto se muestra en la Tabla 5.

Tabla 5. Listado de certificaciones en el nivel 2 de CMMI-DEV ver. 1.3 año 2013, por país.

<i>Posición</i>	<i>País</i>	<i>Certificaciones Nivel 2</i>
1	EEUU	72
2	España	21
3	México	17
4	Brasil	14
5	Francia	12
6	República de Corea	11

4.1. Análisis estadístico de certificaciones en CMMI-DEV ver. 1.3 en México en el año 2013

La constante competencia en el mercado de desarrollo de software a nivel mundial, ha impactado considerablemente a las empresas mexicanas dedicadas a este rubro, motivándolas a mejorar sus procesos y sus productos de software. México no se queda atrás con respecto a la implantación del Modelo CMMI-DEV ver 1.3, como se puede apreciar en la Tabla 1, en la que México ocupó el cuarto lugar a nivel mundial en certificación en CMMI-DEV ver 1.3, en el año 2013.

A continuación se presentan en la Tabla 6 un listado de las empresas mexicanas certificadas en CMMI-DEV ver 1.3, que obtuvieron alguna certificación, el nivel de madurez que alcanzaron y el estado de la República Mexicana en el que están establecidas [4].

Tabla 6. Empresas Mexicanas Certificadas en CMMI-DEV ver. 1.3, año 2013.

<i>Nombre de la Empresa</i>	<i>Nivel</i>	<i>Estado</i>
ECARESOFTE MEXICO S.A. DE C.V.	5	Nuevo León
IBM	5	Jalisco
TATA CONSULTANCY SERVICES LIMITED	5	Estado de México
TECNOCOM	5	Queretaro

INNOVACIÓN INTELIGENTE S. DE R.L. DE C.V.	4	Jalisco
APPLIED PROTOCOL INTERFACES, S.A. DE C.V.	3	Sinaloa
ASTECI S.A. DE C.V.	3	Estado de México
AXXIS SOLUCIONES S.A. DE C.V.	3	Estado de México
AZERTIA TECNOLOGIAS DE LA INFORMACIÓN MÉXICO, S.A. DE C.V. UNA EMPRESA DEL GRUPO INDRA	3	Estado de México
BANCO BASE	3	Nuevo León
BIOXOR S. DE R.L. DE C.V.	3	Jalisco
BSD ENTERPRISE, S.A. DE C.V.	3	Nuevo León
CORPORACIÓN DE SERVICIOS EN TECNOLOGÍAS DE LA INFORMACIÓN S.A. DE C.V.	3	Estado de México
DELOITTE CONSULTING GROUP S.C.	3	Estado de México
EBCOMM S.A. DE C.V.	3	Nuevo León
EC SERVICES, S.A. DE C.V.	3	Jalisco
EMERGYS MÉXICO S.A. DE C.V.	3	Jalisco
HEWLETT PACKARD ENTERPRISE SERVICES	3	Chihuahua
HILDEBRANDO	3	Estado de México
IBM DE MEXICO, COMERCIALIZACION Y SERVICIOS, S. DE R.L. DE C.V.	3	Ciudad de México, D.F.
INFOVIEWS S.A. DE C.V	3	Estado de México
INSTITUTO TECNOLÓGICO SUPERIOR DEL SUR DE GUANAJUATO (ITSUR)	3	Guanajuato
IT SKILL, S.A. DE C.V	3	Estado de México
LUMINA AMÉRICAS S.A.	3	Ciudad de México, D.F.
SOFTNET SOLUCIONES, S.A. DE C.V.	3	Nuevo León
SOLUTIA INTELLIGENCE S.A. DE	3	Jalisco

C.V.		
SYE SOFTWARE S.A. DE C.V.	3	Jalisco
TATA CONSULTANCY SERVICES DE MÉXICO S.A. DE C.V.	3	Estado de México
TECNOLOGÍA DE GESTIÓN Y COMUNICACIÓN S.A. DE C.V.	3	Chihuahua
UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN	3	Nuevo León
VEDASOLUTIONS PROVIDER, S.C.	3	Estado de México
WORLD SOFTWARE SERVICES GROUP S.A. DE C.V.	3	Nuevo León
ALDEASOFT	2	Sinaloa
ALGORIA SOFTWARE, S.C.	2	Sinaloa
AXA SEGUROS, S.A. DE C.V.	2	Ciudad de México, D. F.
CONECTIVIDAD Y TELECOMUNICACIÓN, S.A. DE C.V.	2	Guanajuato
CONSULTORES PLASCENCIA SC.	2	Jalisco
DESPACHO DE ASESORÍA FINANCIERA INTEGRAL S.C.	2	Sinaloa
DESARROLLOS MULTIMEDIALES INTERACTIVOS, S.A. DE C.V.	2	Jalisco
GB NETWORKS S.A. DE C.V.	2	Jalisco
IB - MED S.A. DE C.V.	2	Jalisco
INNOVAWEB SA DE CV	2	Sinaloa
MASSOLUCIONES SOFTWARE S.A. DE C.V.	2	Guanajuato
MESSOFT SYSTEMS S.A DE C.V.	2	Jalisco
MINDTEC	2	Sinaloa
S&C CONSTRUCTORES DE SISTEMAS, S.A. DE C.V.	2	Ciudad de México, D.F.
TEQUI SOFT TECH, SAPI DE C.V.	2	Jalisco
WERNER PEGASUS, S. DE R.L. DE C.V.	2	Jalisco
ZONA HS, S.A. DE C.V.	2	Sinaloa

Como podemos observar las cuatro empresas certificadas en el nivel 5, se encuentran ubicadas en los estados de la República México con mayor desarrollo económico, la empresa ECARESOFTE MEXICO S.A. DE C.V., es una empresa mexicana de tecnología de información especializada en el sector salud, que se ubica en Nuevo León; también encontramos a IBM en Jalisco y desde luego como ya la habíamos mencionado a nivel internacional la empresa TATA CONSULTANCY SERVICES LIMITED, con su corporativo en nuestro país, en el Estado de México.

Finalmente también encontramos en el nivel 5 a la empresa TECNOCOM, es una multinacional española de las tres primeras empresas del sector de la tecnología de la información en España, su presencia regional incluye además de España y México, Portugal y otros países de América Latina (Chile, Colombia, Perú, Brasil, Paraguay y República Dominicana) y en EEUU, Miami.

De la Tabla 6, después de hacer un análisis discreto, podemos concluir que el estado con más empresas certificadas en CMMI-DEV ver. 1.3, es Jalisco con 14, le sigue el Estado de México con 11, Nuevo León con 7, al igual que Sinaloa, después se ubica en cuarto lugar con 4 certificaciones, el Distrito Federal (D.F.), con 3 se encuentra Guanajuato, con 2, Chihuahua y en última posición con 1 certificación ubicamos a Querétaro. Esta información estadística esta concentrada en la Tabla 7.

Tabla 7. Número de certificaciones en CMMI-DEV ver. 1.3 por Estados en la República Mexicana.

<i>Estado</i>	<i>Certificaciones</i>
Jalisco	14
Estado de México	11
Nuevo León	7
Sinaloa	7
Ciudad de México, D.F.	4
Guanajuato	3
Chihuahua	2
Querétaro	1
Total	49

4.2. Análisis estadístico de certificaciones en CMMI-DEV ver. 1.3 a nivel mundial y México en el año 2014

En el mes de agosto de 2014 se realizó el estudio sobre las certificaciones logradas de CMMI-DEV ver. 1.3, el cual arrojó, que en lo que va del año 2014, existen 725 certificaciones mundiales, los datos fueron obtenidos de la página oficial del CMMI Institute del “Published Appraisal Results List 2014” [6], y el estudio dio como resultado la Tabla 8, donde se ubican los cuatro países con mayor número de certificaciones a nivel mundial, cabe destacar que México ocupa el tercer lugar mundial en certificaciones del modelo CMMI-DEV ver. 1.3 y el primer lugar lo ocupa China con 339 certificaciones de un total de 725.

Tabla 8. Listado de Países con mayor número de Certificaciones CMMI-DEV Ver. 1.3, año 2014.

<i>País</i>	<i>Certificaciones</i>
China	339
E.U.	121
México	120
India	82

En lo que se refiere a México el estudio dio como resultado el resumen que se presenta para los años 2013 y 2014 en la **Tabla 9**, donde se concentran el número de certificaciones por nivel de madurez en los últimos dos años.

Tabla 9. Listado de Certificaciones de empresas Mexicanas en CMMI-DEV ver. 1.3 año 2013 y 2014.

<i>Año</i>	<i>Nivel 2</i>	<i>Nivel 3</i>	<i>Nivel 4</i>	<i>Nivel 5</i>	<i>Total</i>
2014	52	47	7	14	120
2013	17	27	1	4	49

Los datos obtenidos de las certificaciones del año 2014, fueron considerados hasta el 15 de agosto de 2014, y en comparativa con los datos obtenidos del año 2013, el crecimiento en Certificaciones CMMI-DEV Ver.1.3 ha sido importante, ya que en el año 2013 se lograron únicamente 49 certificaciones en los diferentes niveles de madurez y en el transcurso del año 2014 se tienen registradas 120 certificaciones, si esta tendencia continua, al finalizar el año México habrá obtenido aproximadamente 180 certificaciones y abra aumentado su certificaciones 250% más a comparación del año anterior.

Esto tiende a ser posible si consideramos que en este año deben ser evaluadas las empresas favorecidas por la Convocatoria PROSOFT 2013-1, 2013-2 [7,8], que a través

del Gobierno Federal y de la Secretaría de Economía obtuvieron apoyos económicos para empresas que realizan actividades vinculadas al sector de tecnologías de información (TI), para: **certificación de competencias**; creación de empleo de alta especialización; proyecto productivo; **certificación organizacional**; vinculación academia-industria; innovación; comercialización, entre otros.

Muchas de las empresas mexicanas que han logrado alguna certificación en CMMI al menos del 2013 a la fecha, ha sido por medio de programas de financiamientos que ofrece el Gobierno Federal a través de la Secretaria de Economía, pues todo lo que implica obtener una certificación, desde la capacitación, la implementación del modelo y finalmente la evaluación tienen un alto costo sobre todo para PyMES (Pequeñas y Medianas Empresas), de tal forma que estos apoyos buscan fomentar la creación, desarrollo, consolidación, viabilidad, productividad, competitividad y sustentabilidad de las empresas del sector de tecnologías de información y servicios relacionados, dirigidos a crear una cultura de Calidad.

5. Resumen de resultados

En este artículo se presentó el estudio de certificaciones en CMMI-DEV ver 1.3 logradas por organizaciones de todo el mundo, estos resultados se agruparon en 53 países y nos dio como resultado el ranking mundial. Los resultados de la investigación incluyen el número de certificaciones por país en cada uno de los niveles de madurez, que van del nivel 2 al nivel 5, con los que podemos observar el gran avance que tiene el país de China en cuanto a madurez en sus organizaciones, ya que cuenta con 621 empresas certificadas, también se hace notoria la participación de Estados Unidos con 251, India con 155 y México 49.

Cabe resaltar que México obtuvo el primer lugar en lo que se refiere a países de América Latina en cuanto a certificaciones en CMMI-DEV ver.1.3 en el año 2013.

En comparativa con el año 2013, en lo que va del año 2014, México ha logrado 120 certificaciones, ocupando el tercer lugar a nivel mundial, en segundo lugar se encuentra India con 121, en primer lugar se encuentra China con un total de 339 certificaciones.

6. Conclusiones

Los resultados demuestran que cada día crece más el número de empresas a nivel mundial preocupadas por mejorar sus procesos y realizar productos de calidad, que les garanticen la preferencia y permanencia en el mercado, fue quizás inesperado el haber encontrado a México en el tercer lugar a nivel mundial con 120 certificaciones, sin duda ha aumentado considerablemente el número de empresas certificadas en CMMI, esto puede ser una consecuencia del apoyo que actualmente esta ofreciendo el Gobierno Federal a través de la Secretaría de Economía a las empresas desarrolladoras de Software, por ello es indispensable que las organizaciones no pierdan de vista la mejora sus

procesos y la importancia de las Certificaciones Internacionales, para que México pueda seguir estableciéndose como punta de lanza a nivel mundial en cuanto a las certificaciones en CMMI. Es importante considerar ahora una evaluación sobre los resultados de los apoyos brindados por las convocatorias de la Secretaría de Economía a las empresas beneficiadas con la finalidad de conocer el Nivel de Madurez que alcanzaron, la Certificación obtenida, los problemas a los que se enfrentaron, como los resolvieron y las estrategias tomadas para lograr la Certificación.

Referencias

1. Velthuis, M. G., Rubio, F. O., Guzman, I. G., & Pino, F.: Calidad de Sistemas de Información. Mexico, D.F. Alfa Omega RA-MA. (2012)
2. CMMI Institute: Reporte técnico CMMI-DEV VER. 1.3. 2010. [En línea]. disponible: <http://cmmiinstitute.com/assets/Spanish%20Technical%20Report%20CMMI%20V%201%203.pdf>. [Último acceso: 18 de febrero de 2014].
3. Clearmodel, C.I. CMMI Institute: Capability Maturity Model Integration. [En línea]. Disponible: <http://cmmiinstitute.com/>. 2014. [Último acceso: 15 Agosto 2014].
4. CMMI-Institute: Published Appraisal Results List, 2013. [En línea]. Disponible: <https://sas.cmmiinstitute.com/pars/pars.aspx>. [Último acceso: 16 enero 2014].
5. CMMI-Institute: Published Appraisal Results List. 2014. [En línea]. Disponible: <https://sas.cmmiinstitute.com/pars/pars.aspx>. [Último acceso: 15 Agosto 2014].
6. Limited, T. C.: TCS IT Services, Consulting and Business Solutions. 2014. [En línea]. Disponible: <http://www.tcs.com/worldwide/es/es/mexico/Pages/default.aspx>, America Latina, Mexico. [Último acceso: 11 abril 2014].
7. Secretaría de Economía. Prosoft 3.0: CONVOCATORIA PROSOFT 2013-1, 2013. [En línea]. Disponible: <http://www.prosoft.economia.gob.mx/ro2013/ConvocatoriaPROSOFT20131a.pdf>. [Último acceso: 15 Agosto 2014].
8. Secretaría de Economía. Prosoft 3.0: CONVOCATORIA PROSOFT 2013-2, 2013. [En línea]. Disponible: <http://www.prosoft.economia.gob.mx/ro2013/Convocatoria%20PROSOFT%202013-2.pdf>. [Último acceso: 15 Agosto 2014].

Propuesta de Sistema Web Personalizable para el Control de Reservaciones en Hoteles

Betzabet García-Mendoza, Carlos R. Jaimez-González

Departamento de Tecnologías de la Información,
Universidad Autónoma Metropolitana, Unidad Cuajimalpa,
Av. Vasco de Quiroga 4871, Col. Santa Fe Cuajimalpa, C.P. 05300, México, D.F.
209363465@alumnos.cua.uam.mx, cjaimez@correo.cua.uam.mx

Resumen. La planificación de viajes está dominada actualmente por los recursos en línea, tales como agencias de viajes, sitios web de opiniones de viaje y hoteles, redes sociales, entre otros. Cuando se ha decidido el lugar a visitar, el siguiente paso es buscar dónde hospedarse y dado que el primer medio de búsqueda es el Internet, no hay mejor oportunidad para los hoteles que incluir un motor de reservaciones en su sitio web, ofreciendo comodidad a sus clientes y abarcando un amplio mercado de ventas. En este artículo se presenta una propuesta de un sistema web personalizable para facilitar a administradores de hoteles pequeños el control de reservaciones de sus habitaciones, gestionando las ventas desde su propio sitio web; así como para facilitar y garantizar de inmediato las reservaciones para sus huéspedes potenciales. Adicionalmente, el sistema web será visualizado automáticamente en diferentes dispositivos móviles, siguiendo el patrón de diseño web adaptable.

Palabras Clave. Diseño Web Adaptable, Control de Reservaciones, Sistema Web Personalizable, Hoteles.

1 Introducción

La planificación de viajes está dominada actualmente por los recursos en línea: agencias de viajes, sitios web de opiniones de viaje, sitios web de opiniones de hoteles, redes sociales, sitios web de hoteles, etc. Por ejemplo, el 93% de los viajeros a nivel mundial dice que sus decisiones de reserva se ven influenciadas por las opiniones en línea [1]. Cuando una persona ha decidido el lugar a visitar, el siguiente paso es buscar dónde hospedarse y dado que el primer medio de búsqueda es el Internet, no hay mejor oportunidad para los hoteles que promocionar sus servicios por Internet. En la actualidad existe una amplia variedad de agencias de viajes, buscadores de hoteles y otras opciones, que permiten localizar hoteles en todo el mundo; sin embargo, la información del hotel que se muestra en estos sitios web es muy escasa y poco útil.

Por otro lado, el 72% de los viajeros a nivel mundial afirman que la posibilidad de efectuar reservaciones a través de un dispositivo móvil es muy útil, y sólo el 25% de los alojamientos permite esta funcionalidad [1]. Es por ello que incluir un motor de reservaciones en el sitio web de un hotel es la mejor opción para los hoteles que desean vender de manera fácil y directa, ofreciendo comodidad a sus clientes y abarcando un amplio mercado de ventas: el Internet.

Los sitios web de hoteles tienen como objetivo principal ofrecer el servicio de alojamiento, además de otros servicios generales. Estos sitios se dividen en secciones, donde se describen los diferentes servicios con los que cuenta el hotel, mediante galerías fotográficas, así como secciones de contacto y solicitudes de reservación.

En este artículo se presenta una propuesta de un sistema web personalizable para facilitar a administradores de hoteles pequeños el control de reservaciones de sus habitaciones, gestionando las ventas desde su propio sitio web; así como para facilitar y garantizar de inmediato las reservaciones para sus huéspedes potenciales.

El resto del artículo está organizado de la siguiente manera. La sección 2 describe la estructura general, el tipo de información y las funcionalidades de algunos sitios web de hoteles que se visitaron. En la sección 3 se analizan tres herramientas para personalización de sitios web para hoteles. La sección 4 presenta la propuesta del sistema web que se está desarrollando, junto con la metodología y los avances que se tienen. Finalmente, en la sección 5 se presentan las conclusiones y el trabajo futuro.

2 Estructura de Sitios Web de Hoteles

Con el objetivo de conocer la estructura, tipo de información y funcionalidades con las que cuentan los sitios web de hoteles, se visitaron seis sitios web de hoteles ubicados en lugares turísticos de México: H1) Las cúpulas en Malinalco, Estado de México [2]; H2) La casa del laurel en Taxco, Guerrero [3]; H3) El encanto en Bernal, Querétaro [4]; H4) Hotel Malinalco en Malinalco, Estado de México [5]; H5) Casa abierta en Valle de Bravo, Estado de México [6]; y H6) Hotel Cristal en Chignahuapan, Puebla [7].

La Tabla 1 muestra un comparativo de la información que se ofrece en cada uno de los sitios web de los hoteles visitados; la paloma indica que el hotel tiene la información o sección en su sitio web, mientras que el tache indica que no la tiene. A continuación se proporciona una breve descripción de la información contenida en los sitios web de los hoteles analizados.

Tabla 1. Información contenida en los sitios web de los hoteles analizados.

<i>Información / Secciones</i>	<i>H1</i>	<i>H2</i>	<i>H3</i>	<i>H4</i>	<i>H5</i>	<i>H6</i>
Inicio / Acerca / Quiénes somos	✓	✓	✓	✓	✓	✓
Habitaciones	✓	✓	✓	✓	✓	✓
Reservaciones	✓	✓	✓	x	✓	✓
Servicios	✓	✓	✓	✓	✓	✓
Galería	✓	x	x	✓	✓	✓
Instalaciones	✓	x	✓	x	✓	x
Paquetes / Promociones	x	x	x	x	x	✓
Tarifas	✓	✓	✓	x	✓	✓
Contacto	x	x	✓	✓	✓	✓
Ubicación	✓	✓	✓	✓	✓	✓
Atracciones	✓	✓	✓	✓	x	✓

Inicio / Acerca / Quiénes somos. En esta sección se encuentra una breve descripción del hotel, donde se menciona la historia del hotel, cuántos años lleva funcionando, el número de habitaciones con que cuenta, sus servicios, atractivos turísticos, etc.

Habitaciones. En esta sección se encuentra la información de los diferentes tipos de habitaciones que ofrece el hotel, tal como fotografías y descripción de cada habitación, servicios con que cuenta cada habitación y el tipo de mobiliario que posee.

Reservaciones. Esta sección se compone de un formulario, donde el usuario introduce sus datos y los datos de la habitación que desea reservar. Se le indica al usuario que se verificará la disponibilidad y se le enviará posteriormente un correo electrónico con la confirmación de su reservación.

Servicios. En esta sección se describen los servicios con los que cuenta el hotel en general; pueden ser servicios básicos como: Internet, estacionamiento, cafetería, etc.; o servicios extras como: restaurante-bar, salón de fiestas, sala de negocios, etc.

Galería. La galería es una sección que puede ser un menú o estar incluida en alguna otra sección. En ella se muestran fotografías del hotel.

Instalaciones. La sección de instalaciones es una combinación de las secciones de servicios y galería; se muestran con imágenes las diferentes áreas del hotel.

Paquetes / Promociones. Esta sección se compone de paquetes o promociones definidos por el hotel. Los paquetes pueden contener diferentes servicios extras añadidos a la reservación de una habitación; mientras que las promociones pueden ser descuentos en las reservaciones en ciertas fechas.

Tarifas. La sección de tarifas muestra una tabla con los precios de cada tipo de habitación, los cuales pueden depender de las temporadas que tengan definidas en el hotel. También se tienen tarifas de los servicios extra que ofrece el hotel.

Contacto. Es una sección con un formulario y/o con información de contacto del hotel, tal como: dirección, teléfono y correo electrónico.

Ubicación. En esta sección se muestra la dirección del hotel y un mapa (Google maps en varios de los casos) de ubicación del hotel.

Atracciones. La sección de atracciones contiene información de los principales atractivos turísticos del lugar, así como información de paquetes turísticos.

El enfoque principal de esta propuesta de sistema web son los hoteles pequeños ubicados en lugares turísticos, los cuales en la actualidad carecen de sistemas de información para el control del registro de sus reservaciones.

3 Herramientas Existentes

En esta sección se describen tres herramientas que permiten personalizar sitios web para hoteles con motor de reservaciones en línea: *PRO Internacional* [8], *Creowebs* [9] y *Obehotel* [10].

3.1 PRO Internacional

De acuerdo a su sitio web, *PRO Internacional* [8] es una empresa dedicada a brindar servicios web integrales, que van desde la creación de sitios web con alto grado de funcionalidad e interactividad, hasta acciones de compromiso en redes sociales y campañas de posicionamiento en buscadores. Entre los sitios web con alto grado de funcionalidad que crea *PRO Internacional*, se encuentran los sitios web para hoteles con módulo integrado de reservaciones de habitaciones en línea.

Estos sitios web cuentan con las siguientes funcionalidades: reserva de habitaciones en línea; tarifarios por temporadas; galería de fotos; suscripción a boletín electrónico; comentarios (opcional); encuestas; buzón de sugerencias.

Cabe señalar que *PRO Internacional* no es una herramienta gratuita, y no cuenta con demostración.

3.2 Creowebs

Creowebs [9] es una herramienta web que permite crear sitios web a partir de una variedad de plantillas predeterminadas, las cuales están diseñadas para diferentes tipos de sitios web, tales como blogs, sistemas de reservaciones para hoteles o restaurantes, etc. *Creowebs* proporciona los siguientes módulos para creación de sitios web: plantillas; importar contenido; editor de imágenes; blog y noticias; redes sociales; videos; música; galerías de fotos; formulario de contacto; compartir archivos; pagos por PayPal; Uso de Google AdSense; reservaciones; Uso de Google Maps; encuestas; visualización de ruta de navegación; versión para dispositivos móviles; subdominios;

Para conocer más sobre el funcionamiento de esta herramienta web, sus ventajas y desventajas, se creó una cuenta en el sitio web y se procedió a crear un sitio web para un

hotel. Se realizaron diferentes pruebas para conocer a fondo cada una de sus funcionalidades; por ejemplo, se cambió la tipografía, se editó la información de ciertas secciones del sitio, se modificaron los formularios prediseñados, se dieron de alta habitaciones y temporadas, así como los precios correspondientes en cada una de ellas; y sobre todo, se realizaron consultas de disponibilidad y peticiones de reservaciones. Como ejemplo, en la Figura 1 se muestra la interfaz para registrar una nueva habitación en la herramienta *Creowebs*.

Nombre	Modelo cama	Adultos	Niños	Número
Uso individual	individual	1	0	15
Doble	doble	2	0	10
Familiar	doble + cama supletoria	2	1	5

Fig. 1. Registro de nueva habitación en Creowebs.

3.3 ObeHotel

De acuerdo a sus creadores, *ObeHotel* [10] es el conjunto de las distintas soluciones, desarrollos y aplicaciones tecnológicas dirigidas a los hoteles para su comercialización y distribución en línea, orientadas a mejorar tanto los procesos de los hoteleros como la experiencia de los clientes. *ObeHotel* ofrece cuatro diferentes servicios: pasarela de reservaciones; web completa; channel manager; y máxima visibilidad.

El servicio de *web completa* de *ObeHotel* es el que está relacionada con esta propuesta, por lo cual fue el que se analizó. Las principales características de *web completa* son las siguientes: diseño personalizado; creación y configuración de cualquier tipo de habitación; reservación de múltiples habitaciones; gestión de ofertas y paquetes; posicionamiento en buscadores; diferentes tarifas para el sitio web y para los canales de venta; código de rastreo para monitorear visitas; integración con Google Analytics y Google Adwords; módulo web de opiniones de clientes.

ObeHotel ofrece un sitio web de demostración, en el que se pueden probar las funcionalidades que ofrece hacia los huéspedes potenciales. En este sitio de prueba se puede verificar disponibilidad; reservar habitaciones; conocer los tipos de habitaciones; los ser-

vicios, paquetes y ofertas; mapa de ubicación del hotel; un formulario de contacto; tour fotográfico y un video tour. Como ejemplo, en la Figura 2 se puede observar el formulario para verificar la disponibilidad de habitaciones.



Fig. 2. Formulario de disponibilidad en ObeHotel.

Debido a que el sistema no es gratuito y no cuenta con una demostración en la que se pueda interactuar directamente con el sistema en la parte de administración, no se pudieron probar los módulos que corresponden al usuario administrador.

3.4 Comparación

En la Tabla 2 se muestra una comparación de funcionalidades entre las tres herramientas mencionadas anteriormente: *PRO Internacional*, *Creowebs* y *ObeHotel*; la paloma indica que la herramienta tiene la funcionalidad, mientras que el tache indica que no la tiene. A continuación se muestra una descripción de las características mostradas en la Tabla 2, las cuales se utilizaron para evaluar las herramientas analizadas.

Tabla 2. Características para evaluar las herramientas analizadas.

Características	<i>PRO Internacional</i>	<i>Creowebs</i>	<i>ObeHotel</i>
Gestión de habitaciones	✓	✓	✓
Reservación múltiple	✗	✗	✓
Tarifas por temporadas	✓	✓	✗
Gestión de ofertas y paquetes	✗	✗	✓
Mapa de disponibilidad	✓	✓	✓
Comentarios	✓	✓	✓
Encuestas	✓	✓	✗
Buzón de sugerencias	✓	✗	✗
Plantillas prediseñadas	✓	✓	✗
Formularios personalizables	✗	✓	✗

Estadísticas	x	x	✓
Versión móvil	x	✓	x

Gestión de habitaciones. Funcionalidad para permitir al administrador la creación, modificación y eliminación de habitaciones.

Reservación múltiple. Funcionalidad para permitir la reservación múltiple en una sola transacción por el mismo huésped potencial.

Tarifas por temporadas. Es la funcionalidad para registrar tarifas por habitación según la temporada del hotel; dichas temporadas son definidas por el administrador con fechas de inicio y fin sin posibilidad de traslapes entre ellas.

Gestión de ofertas y paquetes. Funcionalidad que permite la creación, modificación y eliminación de ofertas y paquetes. Las ofertas son descuentos o promociones en el precio de los servicios; y los paquetes son un grupo de servicios en conjunto.

Mapa de disponibilidad. Esta funcionalidad permite visualizar el mapa de disponibilidad con la relación de habitaciones disponibles y ocupadas. En algunos casos se muestra gráficamente y en otros es una tabla del total de las habitaciones y su estado.

Comentarios. Funcionalidad para incluir una sección que permita a los huéspedes publicar comentarios sobre su estancia en el hotel.

Encuestas. Funcionalidad que permite al administrador del hotel realizar encuestas de satisfacción de la estancia de los huéspedes.

Buzón de sugerencias. Esta funcionalidad permite incluir un buzón de sugerencias con un formulario, donde el huésped puede emitir una sugerencia al hotel y ésta se envía al correo electrónico del administrador.

Plantillas prediseñadas. Funcionalidad que permite al administrador del hotel elegir una plantilla prediseñada para su sitio web.

Formularios personalizables. Es la funcionalidad que permite al administrador del hotel personalizar formularios; añadiendo y eliminando campos.

Estadísticas. Funcionalidad que permite al administrador conocer información relevante sobre las visitas al sitio web.

Versión móvil. Esta funcionalidad permite visualizar el sitio web desde un dispositivo móvil, ajustando el diseño del sitio web según la resolución del dispositivo.

4 Propuesta

En esta sección se presenta la propuesta del sistema web, la metodología a seguir, así como los avances que se han alcanzado. La propuesta es un sistema web personalizable para hoteles con motor de reservaciones, utilizando las siguientes tecnologías: Java, HTML, CSS y JavaScript para la creación del sistema web; MySQL como manejador de bases de datos relacionales; y siguiendo una metodología iterativa e incremental para el levantamiento de requerimientos, análisis y diseño del sistema.

El administrador podrá personalizar el sistema web según sean las características de su hotel: habitaciones, precios por temporada, servicios, paquetes, promociones, información de contacto, atracciones, así como gestionar las reservaciones. El huésped potencial podrá consultar la información completa del hotel; servicios, promociones, paquetes, atracciones, información de contacto, así como realizar su reservación. La interfaz del huésped potencial se diseñará utilizando el patrón de diseño web adaptable (*Responsive Web Design*), para adaptar la interfaz según el dispositivo que utilice el usuario para acceder al sitio web, como se muestra en la Figura 3.

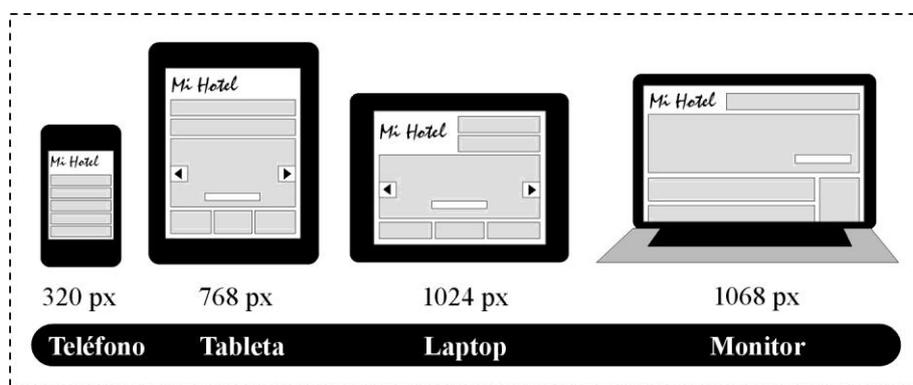


Fig. 3. Diseño web adaptable.

4.1 Metodología

Para la creación del sistema web se plantearon las siguientes actividades.

Identificar usuarios. En esta etapa se realizarán entrevistas a administradores de hoteles con el objetivo de conocer las actividades que realizan para la administración del hotel y las características de éste. También se realizarán entrevistas a huéspedes potenciales para conocer los medios de búsqueda y reservación que utilizan, así como sus preferencias al viajar. Se analizarán los resultados y se definirán dos perfiles de usuario: administrador y huésped potencial; así como las características del hotel.

Realizar levantamiento de requerimientos. En esta etapa se elegirá a un hotel para realizar el levantamiento de requerimientos, con el objetivo de adquirir la información necesaria para desarrollar el sistema web.

Diseñar e implementar un módulo para administración del hotel. En esta etapa se diseñará e implementará un módulo que permita la administración de las reservaciones. Las funciones para llevar a cabo la administración de reservaciones del hotel son el mapa de disponibilidad y reservaciones, entre otras.

Diseñar e implementar un módulo que permita la personalización del portal web. En esta etapa se diseñará e implementará un módulo que permita al administrador del

hotel personalizar el portal web con las características particulares del hotel. Se permitirá la creación, modificación y eliminación de habitaciones, temporadas, paquetes y promociones; así como encuestas y formularios.

Diseñar e implementar un módulo para reservaciones en línea. En esta etapa se diseñará e implementará un módulo que permita reservar una o varias habitaciones por medio del portal web; así como el envío automático de un correo electrónico al usuario que reservó, con los datos de su reservación y la confirmación de la misma.

4.2 Avances de la Propuesta

Dentro de los avances de esta propuesta, ya se identificaron a los dos principales usuarios del sistema web propuesto, a través de entrevistas: administradores y huéspedes potenciales. También se realizó un primer levantamiento de requerimientos con el administrador de un hotel; análisis y diseño con diagramas de clases, de colaboración y de secuencia; así como un mapa de navegación y un conjunto de interfaces que ya han sido validadas con el usuario administrador. En la Figura 4 se muestra el mapa de navegación del sistema, con el administrador, y el huésped potencial.

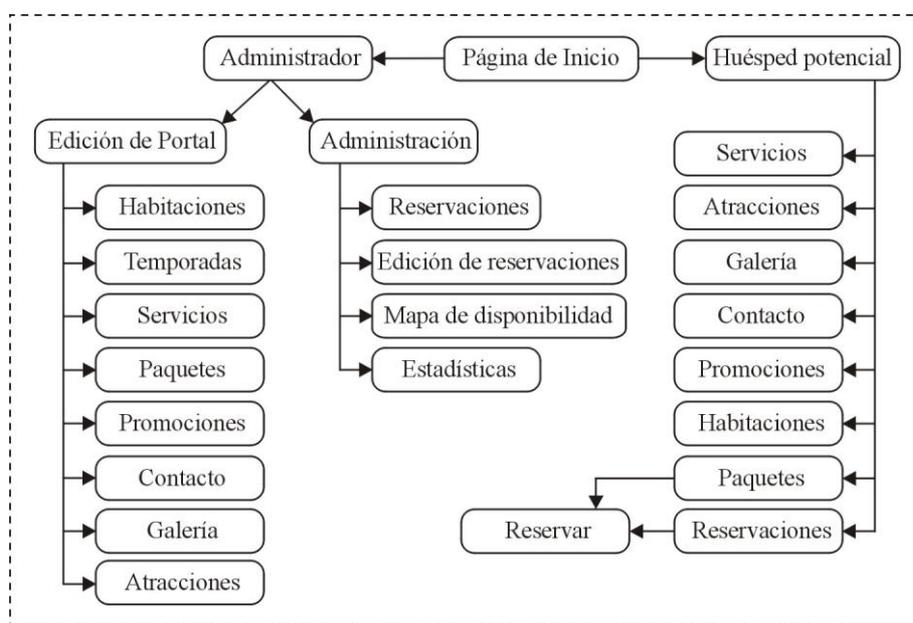


Fig. 4. Mapa de navegación.

El mapa de navegación muestra el conjunto de páginas web a las que tiene acceso cada usuario del sistema web. El administrador puede acceder a dos secciones principalmente: la edición del portal, donde puede realizar altas, bajas y cambios de habitaciones, temporadas, servicios, paquetes, promociones, contacto, galería, y atracciones; y la administración, la cual incluye el registro y edición de reservaciones, mapa de disponibilidad y estadísticas. El huésped potencial puede acceder a las siguientes páginas web: servicios, atracciones, galería, contacto, promociones, habitaciones, paquetes, y puede realizar reservaciones de habitaciones y paquetes.

Como ejemplo, en la Figura 5 se muestra la interfaz gráfica del mapa de disponibilidad de las habitaciones de un hotel, dentro del sistema web. Cabe señalar que las interfaces gráficas desarrolladas son una versión preliminar; falta incorporar las plantillas personalizables, las cuales proporcionarán creativos diseños para la visualización del sistema web en computadoras de escritorio, laptops, y en dispositivos móviles como tabletas y teléfonos inteligentes.

		< JULIO >																														
		M	I	J	V	S	D	L	M	I	J	V	S	D	L	M	I	J	V	S	D	L	M	I	J	V	S	D	L	M	I	J
Habitación 1		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Habitación 2		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Habitación 3		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Habitación 4		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Habitación 5		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Fig. 5. Mapa de disponibilidad del sistema web.

5 Conclusiones y Trabajo Futuro

El propósito de este trabajo es generar un sistema web personalizable para facilitar a administradores de hoteles pequeños el control de reservaciones de sus habitaciones, gestionando las ventas desde su propio sitio web; así como para facilitar y garantizar de inmediato las reservaciones para sus huéspedes potenciales. Adicionalmente, el sistema web será visualizado automáticamente en diferentes dispositivos, tales como computadoras de escritorio, laptops, tabletas y teléfonos inteligentes, siguiendo el patrón de diseño web adaptable.

En el artículo se presentó un análisis de herramientas existentes similares, y se proporcionó una comparativa de funcionalidades. Dentro del avance de la propuesta presentada, ya se han identificado a los dos usuarios de este sistema web, y se han creado sus perfiles: administrador y huésped potencial. También se ha realizado un primer levantamiento de

requerimientos con un administrador de un hotel; análisis y diseño con sus correspondientes diagramas de clases, de colaboración y de secuencia; así como un mapa de navegación y un conjunto de interfaces gráficas que ya han sido validadas con el usuario administrador.

Dentro del trabajo futuro, es necesario completar el diseño e implementación del sistema web, al mismo tiempo de mantener una estrecha comunicación con el administrador del hotel para su validación. También es necesario realizar pruebas de funcionalidad y de usabilidad, tanto con administradores de hoteles, como con huéspedes potenciales. Finalmente, se realizarán ajustes al sistema después de las pruebas realizadas, y se pondrá en funcionamiento en los hoteles para los cuales se está diseñando.

Referencias

1. Trip Barometer: Encuesta realizada a hoteleros y viajeros. Disponible en: http://www.tecnohotelnews.com/wp-content/uploads/2013/03/Infografia_TripBarometer.jpg. Último acceso en Junio de (2014)
2. Sitio web del hotel Las cúpulas. Disponible en: <http://lascupulas.com.mx/>. Último acceso en Junio (2014)
3. Sitio web del hotel La casa del laurel. Disponible en: <http://www.hotelentaxco.com/es/index.html> Último acceso en Junio (2014)
4. Sitio web del hotel El encanto. Disponible en: <http://elencantobernal.com/acerca>. Último acceso en Junio (2014)
5. Sitio web del hotel Hotel Malinalco. Disponible en: <http://www.hotelmalinalco.com.mx/>. Último acceso en Junio (2014)
6. Sitio web del hotel Casa abierta. Disponible en: <http://www.casabierta.com.mx/index.html>. Último acceso en Junio (2014)
7. Sitio web del hotel Hotel Cristal. Disponible en: <http://www.hotelcristalchignahuapan.com.mx/>. Último acceso en Junio (2014)
8. Sitio web de PRO Internacional. Disponible en: <http://www.prointernacional.com/>. Último acceso en Junio de (2014)
9. Sitio web de Creowebs. Disponible en: <https://creowebs.com/>. Último acceso en Junio (2014)
10. Sitio web de ObeHotel. Disponible en: <http://www.obehotel.com/>. Último acceso en Junio (2014)
11. Tendencias en diseño web. Disponible en: <http://www.tecnohotelnews.com/2013/07/las-tendencias-de-futuro-en-diseno-web-para-hoteles/>. Último acceso en Junio (2014)
12. Responsive Web Design. Disponible en: <http://www.designtribe.ie/news/24/61/New-Responsive-B-B-Hotel-website>. Último acceso en Junio (2014)
13. Motores de reservaciones. Disponible en: <http://www.tecnohotelnews.com/2013/05/motores-de-reservas-flexibles-e-integrados/>. Último acceso en Junio (2014)
14. Infografía. Disponible en: <http://www.alojapro.com/blog/wp-content/uploads/2014/01/infografia.jpg>. Último acceso en Junio (2014)

Análisis y evaluación de marcos de trabajo para el desarrollo de aplicaciones y servicios en la nube PAAS

Rafael Fernández Domínguez ^{1,*}, Giner Alor Hernández ¹, Leticia Dávila Nicanor ²

¹División de Estudios de Posgrado e Investigación Instituto Tecnológico de Orizaba, Veracruz, México.

rafafallo1@gmail.com, galor@itorizaba.edu.mx

²Maestría en ciencias de la computación Centro Universitario UAEM Valle de México, Atizapan de Zaragoza, Edo de México, México.

letidn@yahoo.com.mx

Resumen. Hoy en día, las tecnologías para la Internet aumentan a pasos agigantados, provocando la existencia de un gran número de proveedores de servicios en la nube como: Google App Engine, Windows Azure, Amazon E2C, entre otros. Esto a su vez hace que las empresas adoptantes sean más exigentes al adquirir un servicio. En este sentido, este trabajo propone un análisis comparativo entre los diferentes marcos de trabajo para el desarrollo de servicios y aplicaciones en la nube, considerando aspectos de calidad de servicio (QoS) como: Accesibilidad, Confiabilidad, Escalabilidad, Elasticidad, Disponibilidad, Adaptabilidad, Portabilidad y Tiempo de Respuesta, que permita a los clientes seleccionar un marco de trabajo para el desarrollo de aplicaciones de acuerdo a sus necesidades.

Palabras Clave: Cloud Computing, QoS, PaaS, Frameworks.

1 Introducción

La computación en la nube (Cloud Computing) se define como un modelo que permite el acceso de red conveniente y bajo demanda, de un conjunto compartido de recursos informáticos configurables (por ejemplo, redes, servidores, almacenamiento, aplicaciones y servicios) que son rápidamente aprovisionados y liberados con un mínimo esfuerzo de gestión o interacción de servicios [1,2]. Es una tecnología orientada a la Web, y surge para combinar ciertos aspectos para la mejora en el almacenamiento y ventajitas de las empresas, como la reducción de costos, de producción y de mantenimiento. Se confunde fácilmente con el Grid Computing que surge a mediados de 1990, se deriva de la red de energía eléctrica para destacar sus características como la capacidad de penetración, sencillez y fiabilidad. La demanda a gran escala de aplicaciones científicas requieren más po-

der de cómputo que un grupo dentro de un mismo dominio (por ejemplo, un instituto) podría proporcionar; y debido a la inter-conexión rápida por Internet, institutos científicos fueron capaces de compartir y agregar recursos geográficamente distribuidos, incluidos los sistemas de racimo, las instalaciones de almacenamiento de datos y fuentes de datos pertenecientes a diferentes organizaciones [3]. Grandes empresas de Software forman parte de esta tecnología brindando diversos servicios a los usuarios, estos servicios se clasifican en tres tipos diferentes: PAAS, SAAS e IAAS [1, 2].

En este caso la Plataforma como servicio (PaaS) ofrece un entorno de desarrollo como un servicio, donde las aplicaciones se desarrollan utilizando un conjunto de lenguajes de programación y herramientas. Estos servicios incluyen almacenamiento de desarrollo, integración, pruebas o recursos para completar el ciclo de vida de servicios [1]. El consumidor no gestiona ni controla la infraestructura de nube subyacente, incluyendo la red, servidores, sistemas operativos, o el almacenamiento, pero tiene el control sobre las aplicaciones implementadas y posiblemente de alojamiento de aplicaciones configuraciones de entorno [2];

Este análisis comparativo tiene la finalidad de dar soporte a los nuevos clientes del cómputo en la nube al elegir un marco de trabajo para el desarrollo de aplicaciones, al presentar una comparación entre 5 marcos de trabajo bajo el soporte de atributos de calidad de servicio. La calidad del servicio (QoS), la cual es definida por el *ITU Telecommunication Standardization Sector (ITU-T)* como "El efecto colectivo de la eficiencia de servicios que determinan el grado de satisfacción de un usuario del mismo servicio." [4].

2 Trabajos relacionados

En años recientes se han propuesto diversos trabajos enfocados a la computación en la nube, a continuación se presentan algunos de ellos. En [5] se presenta un estudio de diversos servicios en la nube. Se identifican y explican sus principales características, posteriormente se organizan estas características y se propone una taxonomía con estructura de árbol basándose en taxonomías y en tablas de comparación existentes. En [6] se plantea el desarrollo de un marco de trabajo llamado (SMI) Service Measure Index, que consiste en la aplicación de medidas coherentes y significativas que se diseñaron para permitir la comparación de los actuales servicios basados en la nube o proveedores de servicios disponibles. En [7] se presenta de manera general el concepto, la historia, las ventajas y desventajas de la computación en la nube, así como el esfuerzo para crear una normalización de la misma. En [8] se examinan algunas perspectivas de Computación en la nube así como algunas características esenciales de la misma, basado en definiciones dadas por el NIST (National Institute of Standard and Technology) c En [9] se propone un marco de trabajo y un mecanismo que mida la calidad y dé prioridad a servicios en la nube, ya que debido a la gran diversidad de servicios en la nube disponibles es difícil decidir qué servicio utilizar y en qué se basa para su selección, puesto que hoy en día no existe algún marco de trabajo que permita evaluar dichos servicios. En [10] se propone una arquitectura para la asignación de recursos orientada a la alta calidad del servicio y el cumplimiento

del Service Level Agreements (SLA). En [11] se proporciona información a detalle de criterios de seguridad que los próximos clientes de los servicios en la nube considerarán para seleccionar de manera eficiente una nube pública o un Cloud Service Provider (CSP) que satisfaga sus necesidades. En [12] se plantea el objetivo de dar una mejor comprensión del tema identificando y clasificando las principales preocupaciones y soluciones de seguridad en la nube, de igual manera, se propone una taxonomía dando una visión general de lo que es la seguridad actualmente en la computación en la nube. En [13] se revisan los desafíos relacionados con los conceptos de confianza y gestión de Service Level Agreements (SLA) y se analizan los marcos existentes de los acuerdos de nivel de servicio en diferentes ámbitos, como los servicios Web y la red. En [14] se propone un marco de trabajo de comparación que contiene las características de Computación en la nube. Este marco de trabajo se aplica en cuatro modelos de procesos establecidos para la ingeniería de requerimientos que son RUP, XP, Volere y VModel, con el fin de estudiar a qué grado se aplican a los requisitos específicos de las soluciones basadas en la nube.

3 Atributos de calidad aplicables a servicios en la Nube

Los atributos de calidad son aquellos puntos que brindan a los usuarios la confianza al seleccionar un marco de trabajo para el desarrollo en la nube. La universidad Carnegie Mellon en Silicon Valley California dio comienzo a un consorcio que definiera un marco de trabajo que señale atributos y métricas para la evaluación de calidad para seleccionar el servicio y el proveedor más conveniente. Este consorcio lleva por nombre “*Cloud Service Measurement Initiative Consortium*” (CSMIC). Está conformado por un conjunto de instituciones de todo el mundo que abogan por mejorar el cómputo en la nube. Este consorcio desarrolló el “*Service Measurement Index*” (SMI), que es un marco de trabajo que proporciona un índice de medición de los servicios en la nube basado en métricas orientada a la calidad del servicio [15]. Al identificar los atributos de calidad, se llevó a cabo la selección de los más relevantes con base en el soporte que proporciona la literatura actual al respecto de ellas (artículos científicos y libros) y a continuación se describen:

- **Adaptabilidad:** La capacidad del proveedor de servicios para adaptarse a los cambios en las necesidades del cliente.
- **Disponibilidad:** La adecuación de la ventana de la disponibilidad del servicio, así como la probabilidad de que en realidad será siempre la ventana de disponibilidad a los clientes.
- **Elasticidad:** La capacidad de un proveedor de servicio en la nube, para ajustar el consumo de recursos para un servicio a una velocidad suficientemente rápida para satisfacer la demanda del cliente.
- **Tiempo de respuesta del Servicio:** Un indicador del tiempo entre cuando se solicita un servicio, y cuando la respuesta está disponible.

- **Portabilidad:** La capacidad de un cliente para mover fácilmente un servicio de un proveedor de servicio en la nube a otro.
- **Escalabilidad:** La capacidad de un proveedor de servicios cloud para aumentar o disminuir la cantidad de servicios disponibles.
- **Accesibilidad:** El grado en que un servicio es operado por los usuarios
- **Confiabilidad:** Refleja la medida de cómo un servicio funciona sin falla bajo condiciones dadas durante un periodo de tiempo determinado[15]

Después de analizar 40 diferentes propuestas de solución de varios autores para llevar a cabo la evaluación de los marcos de trabajo basados en métricas de calidad, se propusieron las siguientes formas de evaluación para realizar el análisis, las cuales se muestran en la tabla 1

Table 1. Formas de evaluación propuestas para la evaluación de las métricas de calidad seleccionadas.

Atributo	Propuesta de evaluación (métrica)
Accesibilidad	$Ac = F \tag{1}$ <p>Donde F representa la facilidad que poseen los proveedores de servicios en la nube para que nuevos usuarios se adapten a ellos.</p>
Portabilidad	$P = m \tag{2}$ <p>En la cual se representa la cantidad de marcos de trabajo (m) que soportan aplicaciones desarrolladas en otros marcos</p>
Escalabilidad	$\frac{\sum_{i=1}^n c_i}{n} \tag{3}$ <p>Donde c representa los resultados en milisegundos obtenidos en la evaluación sobre n que representa la cantidad de campos utilizados para la sumatoria</p>
Adaptabilidad	$A = TPI - TPA \tag{4}$ <p>Diferencia de tiempo promedio al adaptar la aplicación a nuevos cambios, donde: TPI=Tiempo promedio inicial.</p>
Elasticidad	$\frac{\sum_{i=1}^n c_i}{n} \tag{5}$ <p>Se calcula la sumatoria de los resultados obtenidos en milisegundos(c) desde i que es el valor inicial hasta n que representa el total</p>

Atributo	Propuesta de evaluación (métrica)
de campos utilizados en cada.	
Disponibilidad	$\frac{\sum c}{n} \quad (6)$ <p>Donde se refleja el tiempo promedio al realizar la sumatoria de c que representa los milisegundos obtenidos como resultado sobre la cantidad de campos utilizados para la sumatoria n.</p>
Tiempo de Respuesta	$\sum T_i / n \quad (7)$ <p>Con esta fórmula se obtiene el tiempo promedio de respuesta al realizar la sumatoria de los tiempos pintados como resultado (T_i) sobre la cantidad de campos utilizados (n) en la sumatoria.</p>
Confiabilidad	$C = cm \quad (8)$ <p>Donde cm es la cantidad máxima de interacciones soportadas en los diferentes marcos de trabajo.</p>

4 Marcos de trabajo para el desarrollo de servicios y aplicaciones para la computación en la nube PAAS

Los marcos de trabajo orientados al desarrollo de aplicaciones para la nube permiten la creación de aplicaciones capaces de desplegarse en las mismas. Existe gran variedad de marcos de trabajo, a continuación se presentan los seleccionados para este trabajo con una breve descripción:

Google App Engine: Es una oferta PaaS, que proporciona soporte para Python y Java. AppEngine permite crear aplicaciones Web escalables, sin necesidad de hardware complejo subyacente y capas de software. Google abstrae esas capas y permite concentrarse plenamente en la aplicación [16].

Microsoft Azure: La plataforma de servicios Azure es la oferta PaaS de Microsoft. Azure tiene una API específica para almacenar y recuperar datos llamados Services de SQL. El sistema subyacente para estos servicios SQL es Microsoft SQL Server. Aunque no todas las funcionalidades de Microsoft SQL Server están expuestas a través de la API, el usuario ejecuta las operaciones y el uso de un lenguaje de con-sulta SQL restringido [17].

Heroku: Es una muy poderosa plataforma basada en Web que proporciona soporte para Ruby, JavaScript y Java Web. Se despliega directamente en un entorno PaaS, permite la migración rápida desde el desarrollo a la etapa de implementación. También contiene un editor de colaboración en tiempo real para su uso con un máximo de cinco personas [18].

Amazon Elastic MapReduce: Es un servicio Web de Amazon (AWS), que utiliza Hadoop para ofrecer una funcionalidad MapReduce. En el fondo sí se basa principalmente en otros dos servicios AWS: Elastic Compute Cloud (EC2) y Simple Storage Service (S3) [19].

CloudFoundry: Es una plataforma de código abierto como un servicio. Entorno ofrecido por VMware, proporciona el entorno para albergar múltiples lenguajes y marcos en una pila abierta de aplicaciones de software que se ejecutan tanto fuera como dentro del firewall. Las principales características de Cloud Foundry son: Elección de los marcos de desarrollo, elección de servicios de infraestructura de aplicaciones, y la elección de las nubes [20].

5 Análisis comparativo

En esta sección se describen los diferentes casos de estudio propuestos para llevar a cabo el desarrollo de las evaluaciones de la calidad de servicio de los marcos de trabajo en cada uno de los atributos de calidad seleccionados para este trabajo, de la misma manera se presentan unos ejemplos de gráficas de resultados obtenidos en las evaluaciones desarrolladas.

Accesibilidad.- En el caso de la accesibilidad, se desarrolló una aplicación que tiene lugar siempre al comienzo en la vida del desarrollador de software, un simple y sencillo “Hola mundo” en cada plataforma soportada por cada marco de trabajo; para llevar a cabo la evaluación se consideró una escala de Likert de 3 valores con los siguientes elementos: (1) “Alto”, el cual significa que cuenta con sencillez en el proceso de navegación, fácil acceso a las herramientas para desarrollo, depuración y despliegue, documentación abundante para comprender y aprender cómo funciona dicho marco; (2) “Medio” que es el nivel que contempla una navegación sencilla, se obtienen las herramientas proporcionadas fácilmente pero no cuenta con la documentación apta para principiantes; (3) “Bajo”. En este nivel se encuentran los marcos de trabajo más complejos ya que su navegación es agotadora y las herramientas que proporciona son demasiado complejas para utilizarlas si no se tiene capacitación previa.

Portabilidad.- En el caso de la portabilidad, se desarrolló una aplicación en los diferentes marcos de trabajo, se empaquetaron como proyectos y se dispuso a transferir las aplicaciones de un marco de trabajo a otro y, de esa forma, evaluar el soporte de cada marco de trabajo para aceptar aplicaciones desarrolladas en otros ambientes de desarrollo de aplicaciones.

Tiempo de respuesta. Al llevar a cabo la evaluación del tiempo de respuesta de los marcos de trabajo para el desarrollo de aplicaciones se desarrolló una aplicación Web en cada marco de trabajo, en la cual se realizó un proceso de conexión a la base de datos correspondiente a cada marco de trabajo y realizar la inserción, modificación y elimina-

ción de un registro simultáneamente y obtener así los tiempos de respuesta del proceso de cada iteración a la base de datos.

Disponibilidad.- Al llevar a cabo la evaluación de la Disponibilidad, se desarrolló una aplicación Web en cada marco de trabajo en la cual se consume el WS basado en SOAP, el cual devuelve una instancia con los datos climatológicos según la ciudad que el usuario solicite. Mediante un formulario, se envía una clave ZIP de la ciudad en EE.UU. de la que el usuario requiera conocer la información climatológica y la recepción de la información que se muestra en la pantalla del navegador. Esta aplicación al mismo tiempo de solicitar y recibir la información, también calcula el tiempo del proceso de petición y de respuesta del WS.

Escalabilidad.- Al desarrollar una aplicación Web en cada uno de los marcos de trabajo seleccionados. La escalabilidad consta de incrementar la funcionalidad de la aplicación, es decir, hacer que uno de sus componentes crezca funcionalmente en carga. La evaluación consiste en consumir 5 servicios Web SOAP diferentes, comenzando con uno e ir incrementando los servicios consumidos hasta llegar a 5, esto se realiza con la opción de un control "select" en HTML, donde el usuario elige con cuantos WS quiere calcular el tiempo de respuesta. En la figura 2 se presentan los resultados obtenidos en la evaluación realizada a los marcos de trabajo para obtener los niveles de escalabilidad reflejados por cada uno en la siguiente grafica donde se observa la potencia que ofrece Azure al presentar mejores tiempos de respuesta.

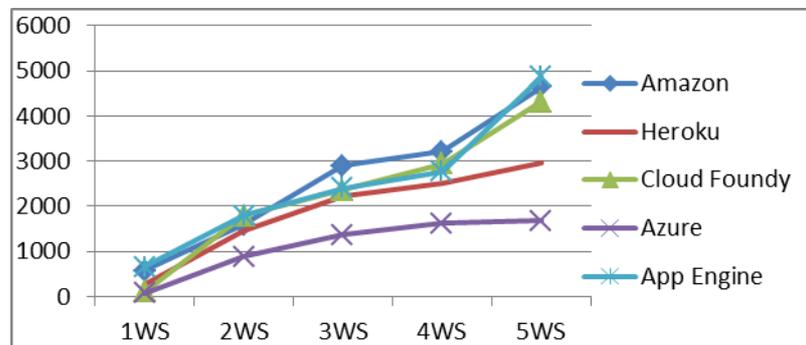


Fig. 1. Resultado de escalabilidad en los marcos de trabajo

Elasticidad.- Se desarrolló una aplicación Web en cada marco de trabajo que incremente la capacidad de memoria al concretizar una clase un número determinado de veces, en la cual se consumen 2 servicios Web SOAP y son almacenados los datos en una base de datos calculando el tiempo demorado para que se concluya el proceso.

En la figura 2 se observan los resultados obtenidos al evaluar la elasticidad en los marcos de trabajo reflejando a Azure como la mejor opción.

Confiabilidad.- Para este caso de estudio, se desarrolló una aplicación Web que consume un WS tantas veces como sea soportado por el marco de trabajo hasta que se obtenga un fallo por parte del mismo mediante un proceso simple, en el cual se inserta la cantidad de iteraciones requeridas por el usuario en una caja de texto, al dar clic en el botón “calcular” realiza un ciclo con la cantidad de iteraciones solicitadas y en cada una busca concretizar una clase.

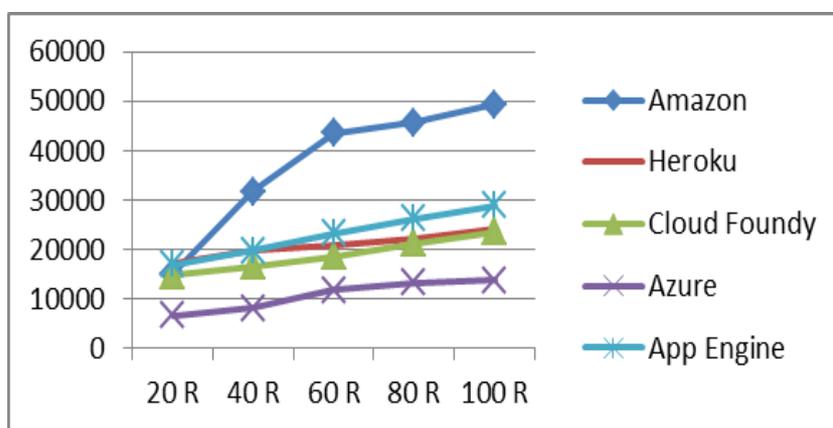


Fig. 2. Resultado de la Elasticidad en los marcos de trabajo

En la tabla 2 se muestran los resultados de las pruebas realizadas sobre los marcos de trabajo, en la cual se presentan las métricas de calidad seleccionadas para este trabajo de tesis y los marcos de trabajo seleccionados para ser evaluados.

Cada sección se encuentra dividida por métrica de calidad en la parte superior de la tabla, las cuales se subdividen en 3 niveles: “BUENO, MEDIO, MALO”. Estos niveles definen como:

- “Bueno”: como el marco de trabajo que obtuvo un resultado sobresaliente a los demás alcanzando niveles más altos en las evaluaciones desarrolladas.
- “Medio”: como el marco de trabajo que obtuvo un nivel medio en los resultados obtenidos durante las evaluaciones.

- “Malo”: implica que obtuvieron el resultado más bajo durante la evaluación. Estos niveles permiten conocer qué marcos de trabajo mantienen un nivel más apto para los usuarios que desean iniciar el uso de los mismos.

Table 2. Matriz de resultados

GAE	CF	AZURE	AMAZON	HEROKU		
X	X	X	X	X	BUENO MEDIO MALO	Accesibilidad
X	X	X	X	X	BUENO MEDIO MALO	Portabilidad
X		X		X	BUENO MEDIO MALO	Disponibilidad
X	X	X			BUENO MEDIO MALO	Adaptabilidad
	X	X		X	BUENO MEDIO MALO	Confiabilidad
	X	X		X	BUENO MEDIO MALO	Tiempo de Respuesta
X	X	X			BUENO MEDIO MALO	Escalabilidad
		X	X	X	BUENO MEDIO MALO	Elasticidad

6 Trabajo a Futuro y Conclusiones

Los marcos de trabajo para el desarrollo de aplicaciones y servicios en la nube enfrentan la problemática de la poca existencia de información respecto a la calidad del servicio en el cómputo en la nube, además de que no hay un precedente de la existencia de material que reúna información de diversos marcos de trabajo o que comparen el soporte ofrecido por cada uno con base en métricas de calidad especializadas, siendo una buena aportación el presente proyecto. Llegando a la conclusión de que falta mucho por hacer en el ámbito del cómputo en la nube y se recomienda analizar otros criterios de calidad semejantes a los analizados en el presente trabajo como capacidad, flexibilidad, funcionabilidad, entre otros, y con ello aportar productos académicos que apoyen la estandarización de la calidad del servicio en la nube.

Referencias

1. María del Pilar Salas, Luis Omar Colombo, Cloud Computing: A Review of PaaS, IaaS, SaaS Services and Providers. Lámpakos No. 7, Medellín, Colombia, pp. 47-57 (2012)
2. Mark Lee Badger, Timothy Grance, Robert Patt-Corner, Jeffery M. Voas. Cloud Computing Synopsis and Recommendations, Technical Report. NIST, Gaithersburg, MD, United States, pp 0-81 (2012)
3. Christof Weinhardt, Arun Anandasivam, Benjamin Blau, et al., Cloud Computing – A Classification, Business Models, and Research Directions, Business & Information Systems Engineering volumen 5, Karlsruhe, Germany. pp 392-399 (2009)
4. Jan Øyvind Aagedal, Quality of Service Support in Development of Distributed Systems, Faculty of Mathematics and Natural Sciences, University of Oslo, March 9, (2001)
5. C.N. Höfer, G. Karagiannis, Cloud computing services: taxonomy and comparison, In processing of the GLOBECOM Workshops (GC Wkshps), IEEE, Springer. Enschede, the Netherlands. pp 81-94. (2010)
6. Jane Sigel, Jeff Perdue, Cloud Services Measures For Global Use, The Service Measure Index (SMI), In Proceedings of the Service Research and Innovation Institute Global Conference, IEEE. Mountain View, CA, USA, pp. 411-415, (2012)
7. Ling Qian, Zhiguo Luo, Yujian Du, Leitao Guo, Cloud Computing: An Overview, in proceeding of the Cloud Computing, First International Conference, CloudCom, Springer-Verlag Berlin Heidelberg. Beijing, China. pp. 626-631, (2009)
8. Vic (J.R.) Winkler, Chapter 2- Cloud Computing Architecture, Securing the Cloud, Cloud Computer Security Techniques and Tactics, Elsevier. Waltham, Massachusetts, USA, pp 29–53, (2011)
9. Saurabh Kumar, Steve Versteeg, Rajkumar Buyya. A framework for ranking of cloud computing services. Future Generation Computer Systems, Volume 29, Elsevier. Melbourne, Australia. Pp 1012–1023, (2012)

10. Kouessi Arafat Sagbo, Pélagie Houngue. Quality architecture for resource allocation in cloud computing. In Proceedings of the First European conference on Service-Oriented and Cloud Computing (ESOCC'12), Flavio Paoli, Ernesto Pimentel, and Gianluigi, (2012)
11. Vic (J.R.) Winkler. Chapter 8 – Security Criteria: Selecting an External Cloud Provider. *Securing the Cloud, Cloud Computer Security Techniques and Tactics*, Elsevier. Waltham, Massachusetts, USA, pp 211–232, (2011)
12. Nelson Gonzalez, Charles Miers, Fernando Redigolo, Marcos Simplicio, Tereza Carvalho, Mats Naslund, Makan Pourzandi. A quantitative analysis of current security concerns and solutions for cloud computing, in Proceedings of the IEEE Third International Conference on Cloud Computing Technology and Science CLOUDCOM '11. Washington, DC, USA, pp 231-238, (2011)
13. Mohammed Alhamad, Tharam Dillon,, Elizabeth Chang, A Survey on SLA and Performance Measurement in Cloud Computing. In processing of the On the Move to Meaningful Internet System (OTM2011). Springer, Australia, pp. 469–477, (2011)
14. Stefan Wind, Schrödl, Requirements Engineering for Cloud Computing: A Comparison Framework. In Proceedings of The Workshop on Information Systems Economics (WISE2010), Springer-Verlag Berlin Heidelberg. Augsburg, Germany, pp. 404-415, (2010)
15. Jane Sigel, Jeff Perdue, A Cloud Services Measures For Global Use, The Service Measure Index (SMI). In Proceedings of the Service Research and Innovation Institute Global Conference, IEEE. Mountain View, CA, USA, pp. 411-415, (2012)
16. Google team. 26-mar-2012. Google App Engine, 23-Feb-2013, <https://developers.google.com/appengine/?hl=es>, (2012)
17. Microsoft team. 23-Feb-2013, Windows Azure: Plataforma en la nube de Microsoft., 23-Feb-2013, <http://www.windowsazure.com/es-es/>
18. Heroku team. 23-Feb-2013, Heroku Cloud Application Platform, 23-Feb-2013, <http://www.heroku.com/>, (2013)
19. Amazon Web Services team, Amazon Elastic MapReduce (Amazon EMR), 23-Feb-2013 <http://aws.amazon.com/elasticmapreduce/>, (2011)
20. VMware team. 24-Ene-2013, Cloud Foundry, 23-Feb-2013, <http://www.cloudfoundry.com/> , (2013)

SBDA-GRID: implementación de un servicio de base de datos en una arquitectura grid

Nancy Ocotitla Rojas, Axel E. Moreno Cervantes, Rogelio Castro González

Escuela Superior de Cómputo-IPN, Av. Juan de Dios Batíz, esquina con Miguel Othón de Mendizábal, México, D.F., 07738. México
nanwen1@gmail.com

Resumen. En el presente trabajo se propone la implementación de un sistema llamado SBDA-GRID que consiste en la implementación de una arquitectura Grid para el desarrollo de un prototipo que permite brindar servicios de bases de datos a través de Internet. Así como el desarrollo de una aplicación web de prueba que brinda los servicios de control de acceso a usuarios y el servicio de creación, modificación y consulta a bases de datos (con estructura select atributo(s) from tabla(s) where condición(es)) con un máximo de dos condiciones. Usando además recursos de diferentes equipos para efectuar las operaciones que se soliciten.

Palabras Clave: GRID, Sistemas Distribuidos, Servicio Web, Bases de Datos.

1 Introducción

Durante mucho tiempo, se ha pretendido lograr que el procesamiento de la información no se realice en una sola computadora si no que se lleve a cabo en diferentes máquinas, incluso estando en lugares muy lejanos. Es por ello, que nace el concepto de computación distribuida ésta hace referencia a cualquier sistema en el que varios componentes autónomos cada uno con capacidad individual se comunican entre sí, afectando mutuamente su comportamiento [1].

Hace algunos años para poder realizar grandes cálculos o procesamiento de gran cantidad de información de cualquier proyecto de investigación, era necesario tener un supercomputador que fuera capaz de soportar trabajos complejos y de gran escala, el cual por sus características tienen un costo demasiado elevado. Pero en la actualidad gracias a los avances de la tecnología, las computadoras personales están llegando a tener mayor poder de procesamiento. Es por ello que se agrega el concepto de computación Grid, que se define como una tecnología nueva e innovadora, una nueva forma de computación distribuida. Los nodos se agrupan más allá del dominio de una red LAN (*Local Area Network*, Red de Área Local). La Grid es un concepto muy ambicioso, consiste en poder

procesar operaciones desde un nodo aunque éste no cuente con los recursos suficientes debido a que será compartida entre los recursos de toda la Grid [2] [3].

La tecnología Grid permite que diversas instituciones puedan unir sus recursos computacionales para obtener uno más poderoso. Con la ayuda de esta tecnología se pueden adelantar proyectos de investigación en países subdesarrollados los cuales no cuentan con el capital suficiente para adquirir supercomputadoras que almacenen y procesen inmensas cantidades de datos.

SBDA-GRID hace uso de la tecnología Grid, aprovechando las características mencionadas para la gestión de bases de datos relacionales, desarrollando herramientas de software que permite al usuario controlar la gestión de bases de datos en un servicio web, las operaciones que puede realizar el usuario son el control de acceso a usuarios y consultas básicas con estructura: select atributo(s) from tabla(s) where condición(es), con un máximo de dos condiciones. Se crean certificados X.509, que es un estándar para infraestructuras de claves públicas que garantizan la seguridad del sistema.

2 Estado del arte

Se analizaron los siguientes sistemas que involucran conceptos de Grid y Bases de Datos obteniendo los siguientes resultados:

1. **Oracle Solaris 11.** Permite a los clientes ejecutar sus aplicaciones en nubes privadas, híbridas o públicas. Permite a los usuarios de entornos corporativos colaborar con las personas adecuadas dentro de su empresa o del entorno de la misma. Se planea integrar perfiles de usuario, grupos, actividades, actualizaciones de estado, foros de discusión, intercambio de documentos, mensajería instantánea, email, notificaciones en tiempo real [4].
2. **Base de Datos Espaciales.** Es un sistema de hardware, software y procedimientos diseñados para realizar la captura, almacenamiento, manipulación, análisis, modelación y representación de datos referenciados espacialmente con el propósito de proporcionar el soporte para la resolución de problemas complejos de planificación y gestión [5].

Los sistemas no implementan específicamente la creación de bases de datos relacionales para el usuario final. SBDA-GRID pretende proporcionar al usuario la posibilidad de habilitar roles de usuario, crear bases de datos relacionales y permitir a cada usuario la gestión de las mismas mediante la arquitectura Grid.

3 Metodología

En la figura 1 se muestra la arquitectura de SBDA-GRID.

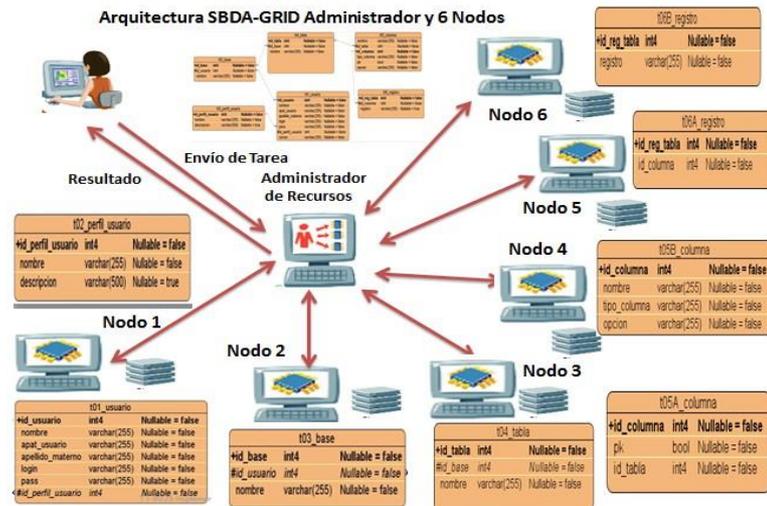


Fig. 1. Arquitectura de SBDA-GRID.

El funcionamiento de SBDA-GRID consiste en el envío de la tarea por parte del usuario a través de una aplicación web mediante el servicio web que es el encargado de establecer la comunicación con el administrador de recursos, éste se encarga de enviar las tareas a los nodos que se encuentren disponibles mediante los protocolos del GRID, la cual una vez procesada regresa al nodo administrador quien es el que envía el resultado final al usuario.

La arquitectura está compuesta por seis nodos clientes y un nodo servidor el cual contiene toda la base de datos que actúa como base de reserva en caso de que no se encuentre disponible alguno de los nodos para la realización del trabajo.

SBDA-GRID se compone de dos módulos principales los cuales son: el módulo GRID y el módulo Cliente.

A. El módulo Grid se compone de las conexiones a las bases de datos de cada uno de los nodos y la creación de los certificados para el manejo de la seguridad de la arquitectura Grid. El nodo administrador necesita crear una autoridad certificadora que es la que le permite implementar la seguridad en el Grid. Esto se logra haciendo uso de los certificados de seguridad X.509 [6]. Observe figura 2.



Fig. 2. Un certificado digital.

La arquitectura presenta una mayor carga de procesamiento en los nodos 1, 3, 4, y 5 (observe figura 1). Debido a que el almacenamiento de los nodos contiene los registros y las tablas de las bases de datos, por consiguiente en ellos habrá un mayor número de conexiones realizadas a dichos nodos. Lo que se propone es minimizar dichas conexiones reuniendo los datos necesarios en una sola petición. Por esta razón se empleó el tipo de fragmentación vertical.

Se produjeron varios fragmentos agrupando un subconjunto de los atributos de una relación. A cada subconjunto se le asignó la llave primaria de la relación. El objetivo de la realización de la fragmentación vertical consiste en particionar la relación en un conjunto de relaciones más pequeñas de manera que las aplicaciones se ejecutarán sobre un fragmento, minimizando así el tiempo de ejecución de las consultas [6]. Además al tratar con relaciones más pequeñas, se ve reducido considerablemente el número de accesos.

SBDA-GRID empleó un enfoque de división, pues la fragmentación se basó en el comportamiento de acceso de las consultas que se realizaban sobre los atributos.

La fragmentación de las tablas columna y registro se realizó de la siguiente forma:

Nodo3: t04_tabla → id_tabla, id_base, nombre

T05A_columna → id_columna, pk, id_tabla

Nodo 4: t05B_columna → id_columna, nombre, tipo_columna, opción.

Nodo 5: t06A_registro → id_reg_tabla, id_columna.

Nodo 6: t06B_registro → id_reg_tabla, registro

Donde la tabla columna y registro son:

t05_columna → nombre, id_tabla, id_columna, tipo_columna, pk, opción

t06_registro → id_reg_tabla, id_columna, registro.

Un submódulo del sistema consiste en la ejecución de consultas de la forma:

```
select atributos(s), from tabla(s), where condición(es);
```

Por ejemplo: dada la siguiente consulta:

```
select nombre, boleta from alumno where id=5;
```

En donde la petición es llegada al administrador de la Grid. Éste se encarga de verificar si alguno de los nodos se encuentra disponible. Consultado a las bases primero, que es en donde se realizó la consulta. Luego las tablas (alumno) en otro nodo, columnas

en dos nodos diferentes (nombre, boleta) y finalmente los registros (resultado) que se encuentran en los nodos restantes.

Las funciones principales del módulo Grid son:

1. La actualización de los nodos conectados a la red.
2. La ejecución de las operaciones a las bases de datos.
3. La comunicación con el cliente.
4. El monitoreo de los nodos disponibles.
5. Ejecución de submódulo consulta en los nodos de la Grid.
6. Envío de las operaciones en cada nodo disponible.

En la figura 3 se muestra la distribución del contenido de la base en los nodos de la GRID sin considerar el nodo usuario.

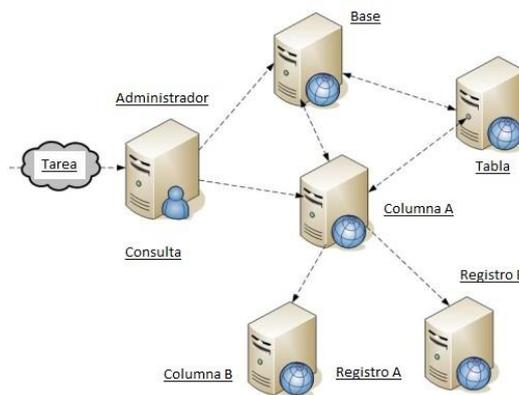


Fig. 3. Realización de operaciones de consulta.

B. El módulo cliente se conforma de las vistas del usuario final en el navegador web. Una aplicación web que hace uso de la conexión mediante el servicio web para extraer los datos que desee el usuario final.

En la figura 4 se muestra la relación de los componentes de SBDA-GRID: la vista de usuario, la WEB y la GRID.

La aplicación web puede estar en el servidor de Grid, se encargará de establecer la comunicación con el servicio web y mostrar los resultados al usuario, mediante archivos realizados en Java y lenguaje HTML. Además se realizó el manejo de sesión de los usuarios para controlar el acceso a las bases de datos específicas a un usuario y su tipo.

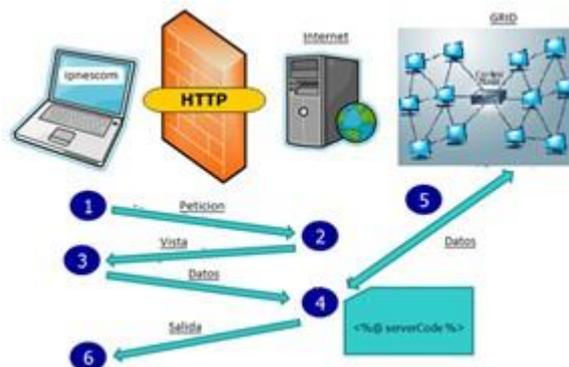


Fig. 4. Componentes SBDA-GRID.

La aplicación web hace uso de las clases stubs del servicio web para crear objetos que permiten la conexión con la arquitectura donde solamente crea las vistas para mostrarle los datos finales al usuario sin necesidad de realizar ninguna conexión a la Base de Datos.

Las funciones principales del módulo Cliente son:

1. Mostrar las vistas al usuario.
2. Extraer la información del usuario
3. La comunicación con el servidor grid.
4. Mostrar el resultado al usuario.

4 Resultados

Para verificar el correcto funcionamiento de SBDA-GRID se realizaron pruebas de consistencia de las bases repartidas en los nodos, dado que la consistencia de los datos fue programada ésta debe conservar los datos actuales al detener un nodo y volverlo a ejecutar. Es decir, debe mantener los datos de la base de datos de respaldo que se encuentran en el nodo principal. Se logró que la base de los nodos coincidiera con los registros de las demás bases en su totalidad.

Se tomó la velocidad que se tarda desde la petición del usuario en el cliente llevando a cabo operaciones en varios nodos y en el nodo principal arrojando el tiempo de respuesta de 800 y 900 milisegundos

Por parte del cliente se verificó la consistencia de los datos, tanto de la parte de Gestión de Usuario como de la de Gestión de Bases de Datos. Para verificar que la información mostrada en el cliente coincidiera con los datos que se encuentran en la base de datos. En la figura 5 se puede observar las bases de datos desde la vista del cliente.

Se realizaron pruebas de procesamiento, para las cuales, se utilizó el comando “top”, que muestra un listado en tiempo real de los procesos que se están ejecutando en el sistema, especificando además el porcentaje de CPU y el porcentaje memoria (KB) que se está utilizando.

En el servidor el uso de CPU es de 0 %, debido a que todo el procesamiento lo lleva a cabo el nodo1. En el nodo 1 (prueba 2), se obtuvo un promedio de 8% de uso del CPU y de uso de memoria un promedio de 8.66%.



Fig. 5. Visualización de bases de datos.

El procesamiento varía dependiendo del tipo de operación que se envía, tanto al Sistema Gestor de Base de Datos como al contenedor, aunque estos valores siempre tienden a un valor semejante. Sin embargo, el procesamiento se lleva a cabo en los nodos a los que se envíe la tarea, bajando la carga de trabajo al nodo administrador. La tabla 1 muestra la comparación de procesamiento de SBDA-GRID.

Tabla 1. Procesamiento de CPU.

Contenedor de Servicios Web					
	Memoria Virtual	Memoria Física	Memoria Compartida	%CPU	%MEM
Servidor	1265m	193m	9548	0	8.6%
Nodo 1 (prueba 1)	727m	171m	9496	7	8.6%
Nodo1 (prueba 2)	730m	171m	9499	8	8.6%
Nodo1 (prueba 3)	729m	172m	9500	9	8.8%
			PROMEDIO	8	8.66%

Los resultados obtenidos de las pruebas realizadas fueron satisfactorios cumpliendo con los requerimientos en su totalidad.

5 Conclusiones y trabajos a futuro

Se ha podido comprobar las ventajas que brindan las herramientas que emplean procesamiento en diferentes equipos, logrando disminuir la carga de las tareas de una sola máquina. Se elaboró un cliente web funcional, eficiente y principalmente transparente para el usuario. La tecnología Grid además del procesamiento que ofrece, una de las grandes ventajas es que permite el almacenamiento de información en los nodos; parte que se utiliza en la aplicación desarrollada. Se solventó la creación de bases de datos en la aplicación desarrollada, manipulando los metadatos de las mismas. Entonces podemos decir que se logró la gestión de BD y además minimizar los costos de procesamiento y almacenamiento.

Los trabajos a futuro para SBDA-GRID consisten en: Realizar consultas más elaboradas que impliquen el uso de operadores de MySQL y añadiendo más condiciones. Agregar nodos dinámicamente a la Grid para aumentar el aprovechamiento de recursos. Realizar respaldo de las bases de datos creadas por el usuario en la aplicación web.

Agradecimientos. Nosotros agradecemos a la ESCOM-IPN por las facilidades otorgadas para el desarrollo del SBDA-GRID: Implementación de un Servicio de Base de Datos en una Arquitectura Grid.

Referencias

- 1 IEEE (2005). Grid Architecture Storage Utilising Grid Computing for Dynamic Data Storage. 2da. Edición, IEEE Computer Society Washington, DC, USA, (2005)
- 2 N. Soriano. Computación distribuida en Grid y en la nube. [online] Retrieve from: <http://www.suite101.net/article/computacion-distribuida-en-grid-y-en-nube-a1937>. [Accessed: August (2012)].
- 3 IEEE COMPUTING. Grid Service Architecture plan gaining momentum. [online] Retrieve from: <http://cs.cinvestav.mx/jorge/bib/webservices/gridserplan.pdf>. [Accessed: August (2012)].
- 4 Oracle Solaris 11. The First Cloud OS. [online] Retrieve from: <http://www.oracle.com/us/products/servers-storage/solaris/solaris11/overview/index.htm>. [Accessed: August (2012)].
- 5 Uso de Grid Computer en la creación de BD espaciales. [online] Retrieve from: <http://es.scribd.com/doc/59521334/Grid-Computer-y-Bd-Espaciales>. [Accessed: March (2012)].
- 6 Ramez A. Elmasri, Shamkant B. Navathe . Fundamentos de Sistemas de Bases de Datos. 3era Edición. Editorial Pearson, (2007)

Reviewing Committee

Juan Manuel Ahuactzin Larios
Laura Alonso I Alemany
Jorge Buenabad Chávez
Marco Antonio Castro Liera
Guillermo De Ita Luna
Luis Gerardo De La Fraga
Andrés Gago Alonso
Kimberly Garcia
Jesús González Bernal
Crispin Hernández Hernández
Sonia G. Mendoza Chapa
Ma. Del Carmen Mezura Godoy

Roberto Morales Caporal
Wendy Moreno Peña
Perfecto Malaquias Quintero Flores
Federico Ramírez Cruz
Orion Reyes
Sebastián A. Ríos Pérez
Omar Sandre Hernandez
Grigori Sidorov
Humberto Sossa Azuela
Juan Vargas
Cornelio Yáñez Márquez
José Luis Zechinelli Martin

Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
octubre de 2014
Printing 500 / Edición 500 ejemplares