

IMPLEMENTACIÓN DE UN PROTOTIPO EN SOFTWARE LIBRE PARA LA GESTIÓN DE PRÉSTAMO DE MATERIALES EN UNA INSTITUCIÓN EDUCATIVA, CASO DE ESTUDIO (UNIVERSIDAD DE CARTAGENA).

IMPLEMENTATION OF A PROTOTYPE IN FREE SOFTWARE FOR THE MANAGEMENT OF LOAN OF MATERIALS IN AN EDUCATIONAL INSTITUTION, CASE STUDY (UNIVERSITY OF CARTAGENA)

AUTOR

EVER JESÚS BLANCO MEZA
Ingeniero de Sistemas
*Universidad de Cartagena
Profesor cátedra
Facultad de Ingeniería de
Sistemas
blancoever@hotmail.com
COLOMBIA

AUTOR 2

DIOFANOR ACEVEDO CORREA
Doctorado en Ingeniería
*Universidad de Cartagena
Docente, Investigador Grupo de
Investigación NUSCA
Facultad de Ingeniería
diofanor3000@gmail.com
COLOMBIA

AUTOR 3

FREDDY MÉNDEZ ORTIZ
Magister en Software Libre
**Universidad Autónoma de
Bucaramanga
Profesor asistente, Grupo de
Investigación PRISMA
fmendez@unab.edu.co
COLOMBIA

*INSTITUCIÓN

Universidad de Cartagena
Unicartagena
Universidad Pública
Avenida del Consulado # Calle 30 No.
48 – 152
correo@unicartagena.edu.co
COLOMBIA

**INSTITUCIÓN

Universidad Autónoma de
Bucaramanga
UNAB
Universidad Privada
Avenida 42 No. 48 – 11
contactos@unab.edu.co
COLOMBIA

INFORMACIÓN DE LA INVESTIGACIÓN O DEL PROYECTO: El objetivo de la investigación fue desarrollar un sistema de gestión de préstamo de materiales a nivel de prototipo haciendo uso de herramientas de software libre que mejore los procesos internos en la Universidad de Cartagena.

RECEPCIÓN: Febrero 16 de 2015

ACEPTACIÓN: (mes/día/año)

TEMÁTICA: Ingeniería del Software

TIPO DE ARTÍCULO: Artículo de Investigación Científica e Innovación

RESUMEN ANALÍTICO

El presente trabajo describe el análisis, diseño e implementación de una herramienta de software que brinda asignaciones para la gestión de préstamos de materiales en los distintos laboratorios de una Institución de Educación Superior, para nuestro caso, la universidad de Cartagena. Dentro del desarrollo se determinan los

tiempos utilizados bajo la metodología del prototipo evolutivo el cual parte de la implementación inicial donde el usuario interviene con sus comentarios generándose así un proceso de refinamiento a través de las diferentes versiones hasta llegar a un sistema adecuado. Se presentan los resultados del desarrollo del proyecto, los cuales comprenden la implementación del software: Validando la calidad del mismo basándonos en la Norma ISO/IEC 25010–Modelo de calidad (2005) en las características de funcionabilidad y usabilidad, el último criterio soportado bajo la encuesta de una muestra de 8 usuarios extraída de una población de 450 estudiantes. Hubo cumplimiento con las necesidades establecidas en los requerimientos del producto.

PALABRAS CLAVES: Requerimientos, ingeniería de requerimientos, herramientas, técnicas.

ANALYTICAL SUMMARY

This paper describes the analysis, design and implementation of a software tool that provides allowances for loan management of materials in different laboratories of an institution of higher education, in our case, the University of Cartagena. Within the development time used under the methodology of evolutionary prototype which part of the initial deployment where the user interacts with your comments thus creating a refining process through the various versions until a suitable system are determined. Development outcomes of the project are presented, which include software implementation: based on the ISO/IEC 25010 standard under the criteria of functionality and usability, the last criterion supported under the survey of a sample of 8 users from a population of 450 students. Has complied with the requirements established in product requirements.

KEYWORDS: requirements, requirements engineering, tools, techniques.

INTRODUCCIÓN

El proceso de asignación de salones, materiales de laboratorios, recursos multimedios requieren tiempo y personal en las distintas universidades, la mayoría de estas instituciones emplea una heurística simple, asistida por un computador que consiste en una asignación basada en el orden de llegada de los listados de requerimientos de aulas para cada sesión que se desea impartir, carente del análisis de datos que permita medir el cumplimiento de los objetivos propuestos por las partes involucradas en la asignación de los recursos. [1] [4]

El problema de la programación de laboratorios consiste en la disposición de objetos dentro de un patrón en el tiempo o el espacio, de tal manera que se logre el cumplimiento de unos objetivos determinados y que las restricciones sobre la forma en que se deben disponer dichos objetos se satisfagan. [2] [3]

En cualquier proceso de programación existen tres elementos fundamentales: objetos, patrones y restricciones. Los objetos se refieren al recurso a organizar o disponer, los cuales son personas, laboratorios, salones de clases, entre otros; el patrón es el orden que deben seguir los eventos y se crea como parte del proceso de programación; y las restricciones se definen como las relaciones tangibles o intangibles entre objetos y representan las reglas que se deben seguir para la construcción de un patrón. Llamado usualmente como

Timetabling la asignación sujeta a restricciones de un conjunto de recursos a objetos específicos, siendo ubicados en espacios de tiempo, de tal forma que satisfaga en el mayor grado posible un conjunto de objetivos deseables. [4] [6] [10]

En [6] plantearon la solución del problema de programación óptima de horarios escolares de una escuela pública Colombiana. Consideraron no solo la asignación adecuada de salones y docentes sino también los ritmos cognitivos que presentan los estudiantes como el factor más importante en el proceso de optimización. Los resultados muestran una mayor eficiencia del algoritmo en comparación con otros aplicados al mismo problema y evaluados en la misma forma. Los grupos programados presentaron una reducción en los niveles de mortalidad académica en comparación con una programación horaria que no consideran los ritmos cognitivos de los estudiantes.

A lo largo de la historia, los laboratorios en las universidades siempre han necesitado herramientas que le permitan administrar su información de una manera eficiente. Algunos laboratorios en las universidades utilizan herramientas como sistemas de información de gran capacidad, pero estas no son de uso libre, también hay algunos que utilizan herramientas más sencillas, como hojas de cálculo, o incluso planillas o cuadernos. [3]

1. MARCO TEÓRICO

Para el desarrollo del prototipo se utilizaron herramientas de diseño como el lenguaje de modelado unificado (UML), este permite comprender de una mejor forma, especificar, construir y documentar un sistema a medida que avanza en su proceso de desarrollo. La principal razón por la cual se prefirió UML como lenguaje de modelado es que facilita el entendimiento de la información, la función y el comportamiento de un sistema, haciendo así más fácil y sistemático el análisis de los requerimientos. La diagramación de UML utilizada en el diseño del prototipo fue: Diagrama de casos de uso.

Para dar comienzo al proceso de desarrollo e implementación se acordó utilizar un patrón arquitectónico que nos permitiera tener un proyecto más organizado y estandarizar el proceso de programación para que en el futuro sea más fácil su mantenimiento. Para la construcción del prototipo, se utilizó la arquitectura modelo vista controlador, debido a las ventajas que ofrece como: Portabilidad, escalabilidad y facilidad de mantenimiento.

Para el desarrollo de este proyecto se usó la Programación Orientada a Objetos (P.O.O.). La principal razón por la cual se prefirió este tipo de programación es que nos da la oportunidad de tener aplicaciones modificables y fácilmente extensibles a partir de código reutilizable, disminuyendo el tiempo de programación gracias al nivel de potencia que nos brinda la herencia.

Las herramientas de implementación utilizadas fueron: HTML 5, CSS 3, PHP y el sistema gestor de bases de datos PostgreSQL.

2. METODOLOGÍA

Para implementar los servicios del prototipo para el sistema de gestión de préstamo de materiales en la Universidad de Cartagena se utilizó como metodología de desarrollo el prototipado evolutivo.

La principal razón por la cual se eligió la metodología de desarrollo de prototipo evolutivo es por lo que el sistema de gestión de préstamos de materiales en la Universidad de Cartagena está cambiando constantemente, presentando nuevas necesidades por parte de los usuarios o en su defecto modificando los servicios ya existentes, así de esta manera nos encontramos no frente a un sistema estático sino por el contrario ante un sistema dinámico que sufre cambios constantemente; el procedimiento seguido por la

metodología utilizada se describe en los siguientes items.

- Se puntualizaron las necesidades globales del software realizando reuniones entre los desarrolladores, los administradores de las salas de laboratorios y el curso muestra que serán los usuarios del sistema, estos estuvieron aportando ideas durante todo el desarrollo del mismo, se identificaron los requisitos y se determinaron los subprocesos donde era necesario hacer mayor énfasis.
- Se realizó el Diseño del Prototipo teniendo en cuenta primeramente los aspectos relevantes del código como por ejemplo, métodos de consultas, inserción, actualización y borrado en la base de datos; luego se prosiguió a diseñar la interfaces de usuario como son los formularios de entrada/salida de datos.
- El prototipo es evaluado por los desarrolladores del proyecto, los administradores de las salas de laboratorios y el curso muestra.
- Se produce un proceso interactivo en el que el prototipo es "depurado" (Refinamiento del prototipo) para que satisfaga las necesidades del usuario, al mismo tiempo que facilita a los desarrolladores una mejor comprensión de lo que hay que hacer para poder entregar el producto final requerido.

2.1 ANÁLISIS DE REQUERIMIENTOS

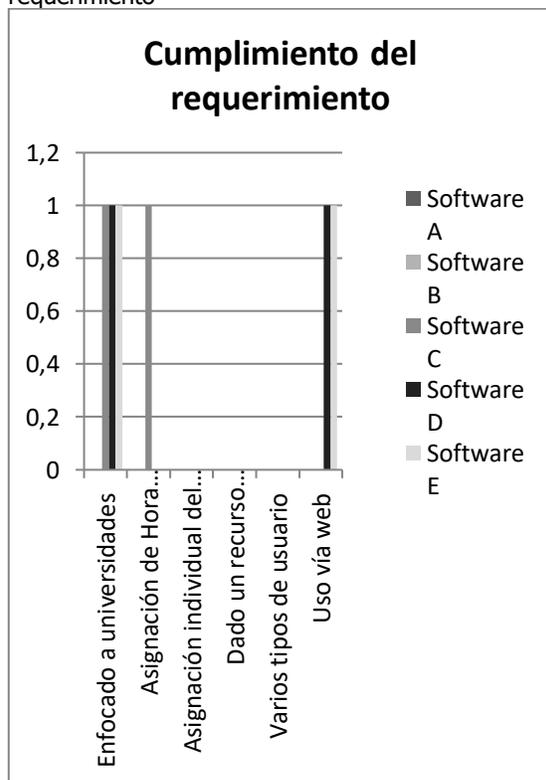
Uno de los elementos que se tienen en cuenta para el análisis de requerimientos es el considerado como el del requerimiento funcional para este se determinan las posibles transformaciones que se producen en un sistema debido a la entradas y como es su efecto en sus posibles salidas [7], para el estudio del prototipo en esta dimensión se procedió a hacer un análisis comparativo con cinco aplicativos, software relacionados primero bajo el licenciamiento de software libre y segundo bajo el concepto de la gestión de préstamos o asignación de recursos, enmarcado en la técnica utilizada en la ingeniería de requerimientos donde se consideran que para el desarrollo del prototipo se deben identificar los requerimientos que son conocidos, para luego señalar áreas en las que será necesario las posibles implementaciones [7], los criterios utilizados fueron: Los programas enfocados a universidades, asignación de horas específicas, asignación individual del recurso, dado un recurso

permite consultar su disponibilidad, permite tener varios tipos de usuario y uso vía web.

TABLA 1. Análisis de requerimientos.

Requerimiento	Software				
	A	B	C	D	E
Enfocado a universidades	NO	NO	SI	SI	SI
Asignación de Hora específica	NO	NO	SI	NO	NO
Asignación individual del recurso	NO	NO	NO	NO	NO
Dado un recurso consultar su disponibilidad	NO	NO	NO	NO	NO
Varios tipos de usuario	NO	NO	NO	NO	NO
Uso vía web	NO	NO	NO	SI	SI

FIGURA 1. Análisis de cumplimiento del requerimiento



La primera característica hace referencia si el aplicativo está enfocado a escuela u universidad como lo requerimos, la cumplen tres de ellos.

La segunda característica hace referencia si en el aplicativo la asignación del recurso se hace en

rangos de horas preestablecidas o se puede hacer en una hora cualquiera como lo requerimos, solamente la cumple uno de ellos.

La tercera característica hace referencia si en el aplicativo la asignación del recurso se hace a grupos previamente conformados o se puede hacer a una persona individual como lo requerimos, no la cumple ninguno de ellos.

La cuarta característica hace referencia si en el aplicativo dado un recurso se puede consultar su disponibilidad como es requerido, ninguno de ellos la cumple.

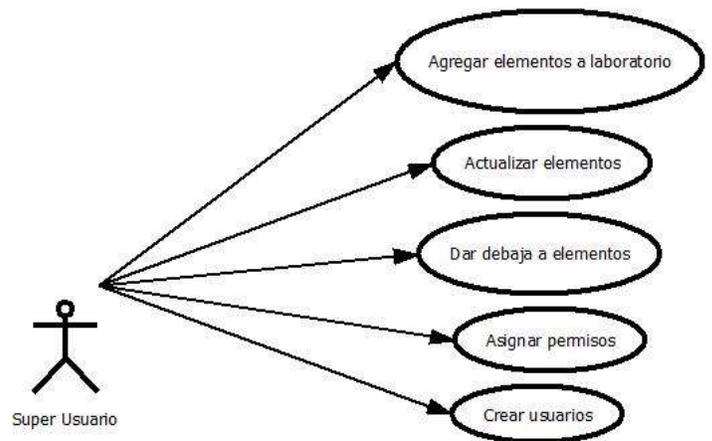
La quinta característica hace referencia si el aplicativo permite varios tipos de usuarios como es requerido, ninguno de ellos la cumple.

La sexta característica hace referencia si el aplicativo puede ser usado vía web como es requerido, la cumplen dos de ellos.

2.2 DIAGRAMAS DE CASOS DE USO

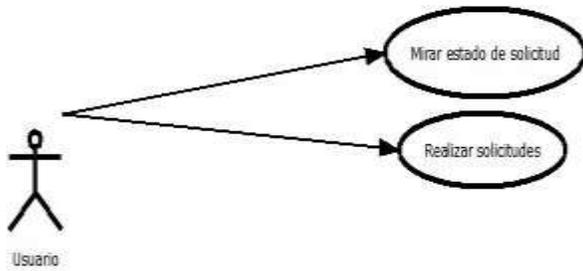
En las siguientes imágenes se muestran los diagramas para cada uno de los tres tipos de usuarios requeridos:

FIGURA 2. Diagrama de Casos de Uso: Súper Usuario.



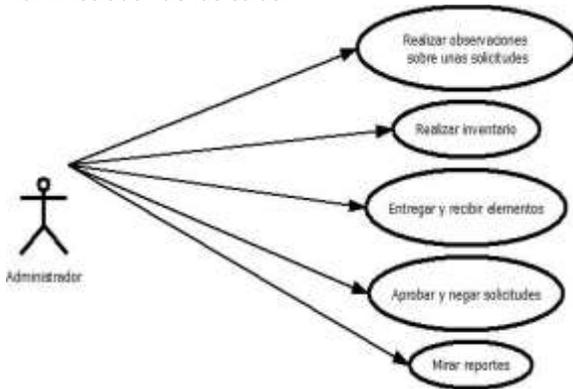
Fuente: Autores

FIGURA 3. Diagrama de Casos de Uso: Usuario Normal.



Fuente. Autores

FIGURA 4. Diagrama de Casos de Uso: Usuario Administrador de las salas.



Fuente. Autores

2.3 ARQUITECTURA Y CLASES IMPLEMENTADAS

En las siguientes imágenes se muestran las clases, modelos, vistas y controladores obtenidos durante su implementación:

FIGURA 5. Modelos



FIGURA 6. Vistas



FIGURA 7. Controlador



2.4 PUESTA EN MARCHA

La solución se encuentra albergada en un servidor VPS (Virtual Private Server) de buen rendimiento el cual se encuentra activo las 24 horas del día durante todo el año, evitando percances que podrían generar la intermitencia en el servicio, el link del prototipo es: www.sredweb.com

FIGURA 8. Vista de súper usuario

Sistema recursos

Bienvenido sr(a) **EVER BLANCO - Super usuario**

Administrar cuentas usuarios

+ Nueva cuenta

10 Entradas

Documento	Nombre	Tipo usuario	activo
12345	EVER	ADMINISTRADOR	si
123456	cread	ESTUDIANTE	si
123456789	EVER	SUPER USUARIO	si
104744194	MOSES ALEJANDRO	ESTUDIANTE	no

Mostrando 1 a 4 de 4 Entradas

Primer Ant 1 Sig Ultima

FIGURA 9. Vista de administrador

Imprimir inventario

Inventario

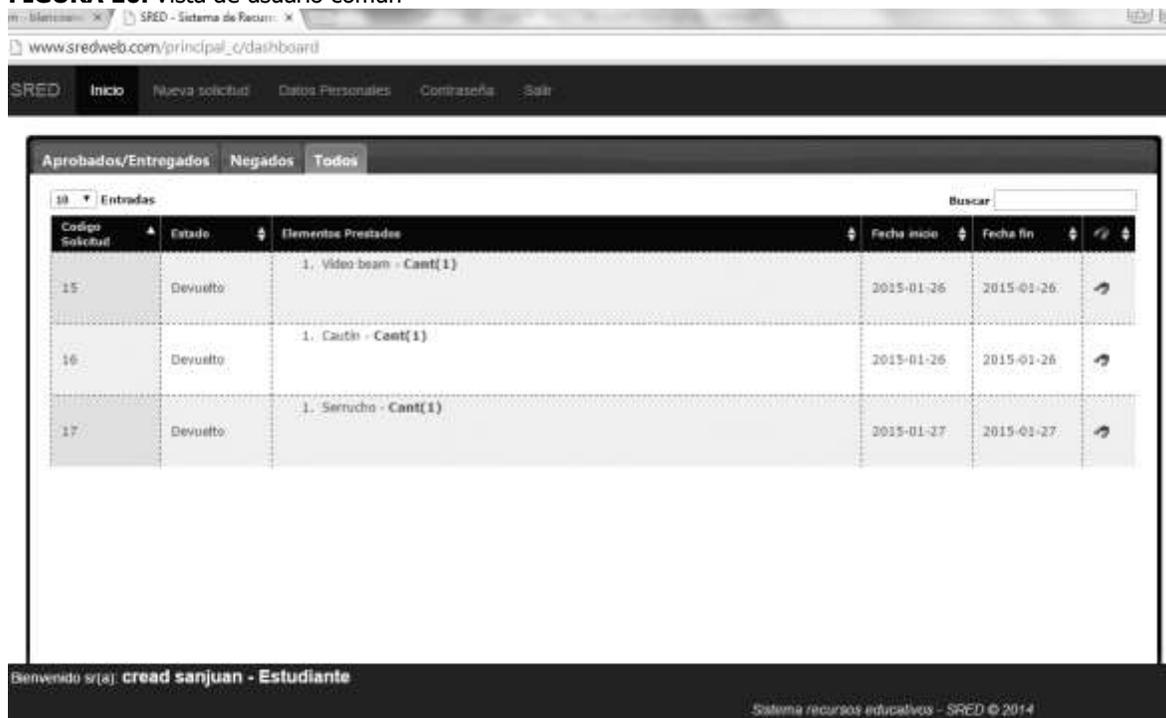
Item	Nombre Elemento	Disponibles	Prestados	Total
3	Semacho	10	0	10
4	Vulvo bono	10	0	10
5	Pizza	10	0	10
6	destornillador	10	0	10
7	Taladro	10	0	10
8	litonio	10	0	10
9	Sopleador	10	0	10
10	suaviz	10	0	10
11	puchador	10	0	10
12	extractor de slup	10	0	10
13	Cable	10	0	10

Mostrando 0 a 0 de 0 entrada

Bienvenido sr(a) **EVER BLANCO - Administrador**

Sistema recursos educativos - SRED © 2014

FIGURA 10. Vista de usuario común



2.5 EVALUACIÓN DE LA CALIDAD DEL PROTOTIPO

La calidad del prototipo se evaluó según la Norma ISO/IEC 25010–Modelo de calidad (2005) en sus características de funcionalidad, y usabilidad en cada una de sus cuatro subcaracterísticas: Apropiabilidad, facilidad de aprendizaje, operabilidad y atractibilidad.

La característica de funcionalidad fue evaluada cada vez que se desarrolló una parte del sistema con sus respectivas codificaciones se procedía a comprobar mediante pruebas unitarias que se cumplieran los requerimientos.

La metodología para la realización del test de usabilidad se hizo de forma similar al procedimiento hecho por Mercovich en el taller: Cómo hacer un test de usabilidad de un sitio [9]; la misma se presenta en las siguientes secciones:

- Planificación
- Test
- Análisis de los datos

Planificación

En esta etapa mediante un muestreo no probabilístico se procedió a seleccionar tres grupos

representativos para los tres tipos de usuarios: Uno como súper usuario del sistema integrado por dos personas, otro como administrador del sistema integrado por dos personas y un tercer grupo como usuario común integrado por ocho personas.

Se diseñaron las tareas que serían realizadas por cada grupo de participantes acorde a su tipo de usuario; como por ejemplo realizar solicitud para el usuario común, aprobar y negar solicitudes para el usuario administrador y asignar permisos para el súper usuario.

Se definió que el sitio de aplicación de la prueba sería en los laboratorios de informática de la universidad de Cartagena realizando previamente la reservación de la sala.

Test

En el cuestionario se aplicaron los siguientes tipos de preguntas:

De afirmación con grado de aceptación de 1 a 5, significando que 1 está en completo desacuerdo con la afirmación de la pregunta y 5 está completamente de acuerdo con la afirmación de la pregunta.

De SI, NO; significando con SI que está completamente de acuerdo con la afirmación de la pregunta y NO que está en completo desacuerdo con la afirmación de la pregunta.

3. RESULTADOS

Se realizaron pruebas de funcionalidad para cada uno de los casos de uso, cumpliéndose satisfactoriamente para todos los casos de uso.

El test de usabilidad aplicado a los tres tipos de usuarios para la subcaracterística de Apropiabilidad arrojó los siguientes resultados:

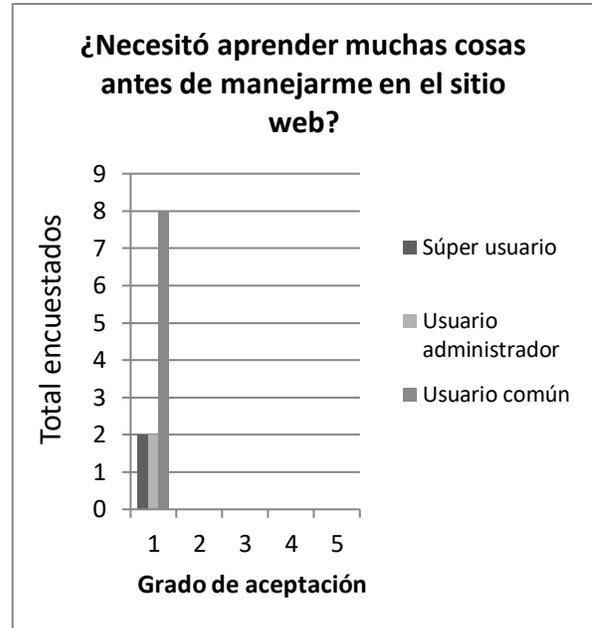
FIGURA 11. Resultados para la subcaracterística de Apropiabilidad



Definidos en el anterior gráfico el cual expresa que los súper usuarios ubicaron su grado de aceptación en el nivel 4, los usuarios administradores ubicaron su grado de aceptación en los niveles 4 y 5, 5 de los usuarios comunes ubicaron su grado de aceptación en el nivel 5, y 3 ubicaron su grado de aceptación en el nivel 4; se puede observar un alto grado de apropiabilidad en el sistema por parte de los usuarios encuestados ya que sus respuestas fueron ubicadas en los niveles 4 y 5 que son los grados más altos de aceptación.

El test de usabilidad aplicado a los tres tipos de usuarios para la subcaracterística Facilidad de aprendizaje arrojó los siguientes resultados:

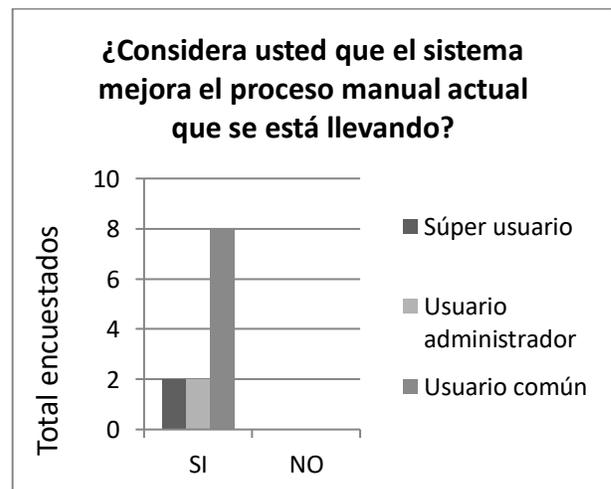
FIGURA 12. Resultados para la subcaracterística de Facilidad de aprendizaje



En esta grafica se observa como los súper usuarios consideran que no necesitan aprender muchas cosas para manejar el sistema, lo mismo consideraron los usuarios administradores y los usuarios comunes, ubicando sus respuestas en el nivel 1 que es el grado de aceptación más bajo a la afirmación que involucra tener que aprender muchas cosas para poder manejar el sitio web.

El test de usabilidad aplicado a los tres tipos de usuarios para la subcaracterística de Operabilidad arrojó los siguientes resultados:

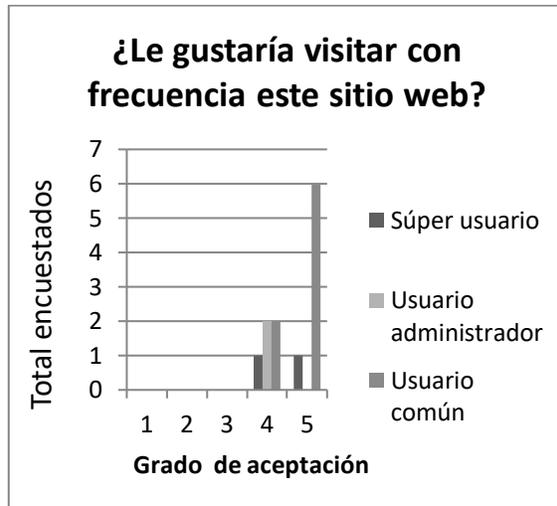
FIGURA 13. Resultados para la subcaracterística de Operabilidad



Las respuestas de los tres tipos de usuarios fueron de conformidad con respecto a la afirmación de que el sistema implementado mejora el proceso manual que hasta entonces se estaba realizando.

El test de usabilidad aplicado a los tres tipos de usuarios para la subcaracterística de Atractibilidad arrojó los siguientes resultados:

FIGURA 14. Resultados para la subcaracterística de Atractibilidad



El test arrojó altos índice en el gusto por la interacción de los usuarios, donde se puede apreciar que los súper usuarios ubicaron su grado de aceptación en el nivel 4 y 5, los usuarios administradores ubicaron su grado de aceptación en el nivel 5, 2 de los usuarios comunes ubicaron su grado de aceptación en el nivel 4, y 6 ubicaron su grado de aceptación en el nivel 5; se puede observar un alto grado de atractibilidad en el sistema por parte de los usuarios encuestados ya que sus respuestas fueron ubicadas en los niveles 4 y 5 que son los grados más altos de aceptación.

4. CONCLUSIONES

Los resultados permiten obtener las siguientes conclusiones: En la ingeniería del software la elección de una metodología es fundamental para la gestión y administración de un proyecto, de esta depende que el mismo tenga viabilidad o fracase, la metodología a utilizar va acorde al problema a resolver lo mismo que al tipo de empresa, como ejemplo si es una empresa grande o pequeña, al seleccionar correctamente la metodología a aplicar la misma traza un camino recto en el proceso de desarrollo evitando en lo posible encontrar problemas en la realización del proyecto, de esta

manera se pueden gestionar cada una de las etapas del proceso de desarrollo como partes de un engranaje y darles un tratamiento especial a cada una de ellas aplicándoles técnicas, métodos y utilizando herramientas específicas para cada etapa. La especificación de requerimientos es una de las etapas más importante en la construcción de cualquier sistema permite hacer gestión de tiempos, costos y por ende con base en estas variables la asignación de recursos dando viabilidad al proyecto. Además los requerimientos son un reflejo de las necesidades de los usuarios del sistema y por tanto deben quedar completos y bien especificados. Aplicando ingeniería de requerimientos se pudo determinar que los aplicativos actualmente desarrollados y que se relacionan con nuestro sistema no satisfacen la gran mayoría de nuestras necesidades y por tanto era más sencillo la implementación desde cero de un nuevo sistema que la adaptación de cualquier sistema al nuestro; de esta manera se dio solución a todas las necesidades de forma eficiente y eficaz, logrando una mejora notable en los procesos internos de la Universidad de Cartagena.

5. REFERENCIAS BIBLIOGRÁFICAS

- [1] Avella, P. y tres autores. (2007).A computational study of local search algorithms for Italian high-school timetabling. *Journal of Heuristics*, 13(6), 543-556
- [2] Flórez P. (2011). Revisión de algoritmos genéticos aplicados al problema de la programación de cursos universitarios. *Programación Matemática y Software*:3(1). 49-65
- [3] Burke E. y Petrovic S. (2002). Recent research directions in automated timetabling. *European Journal of Operational Research*. Vol. 140, No. 2. 266-280.
- [4] Carter M. y Laporte G. (1996). Examination timetabling: Algorithmic strategies and applications. *Journal of the Operational Research Society*. Vol.47, No 3. 373-383
- [5] Pressman, R. (2005). *Ingeniería del Software. Un enfoque práctico. Quinta Edición*. España: McGraw Hill.
- [6] Suárez, V.F., Guerrero, A. & Castrillón, O.D. (2013). Programación de Horarios Escolares basados en Ritmos Cognitivos usando un Algoritmo Genético de Clasificación No-dominada, NSGA-II Programación de Horarios Escolares. *Información Tecnológica Vol. 24(1)*, 103-114
- [7] Chavez, M.A. (2005). La ingeniería de requerimientos y su importancia en el desarrollo de

proyectos de software. *InterSedes: Revista de las Sedes Regionales*, vol. VI, núm. 10, 1-13, 2.

[8] McCall, J.A., Richards, P.K. & Walters, G.F. (1977). Factors in Software Quality, *RADC TR-77-369, US Rome Air Development Center Reports NTIS AD/A-049*, 014, 015, 055.

[9] Mercovich E. (2014). *Workshop: cómo hacer un test de usabilidad de un sitio*. Recuperado (2014, noviembre 30) de: <http://www.gaiasur.com.ar/infoteca/siggraph99/test-de-usabilidad-de-un-sitio.html#que-es-un-test-de-usabilidad>

[10] Tassopoulos, I. X., y G. Beligiannis, Using particle swarm optimization to solve effectively the school timetabling problem. *Soft Computing*: 16(7) 1229-1252 (2012).