

Captura de requerimientos mediante ingeniería inversa de interfaces gráficas de usuario¹

Software requirements elicitation using reverse engineering of graphical user interfaces

Carol Liset Jaimes Vega², Fernando Antonio Rojas Morales³

¹Universidad Industrial de Santander, Escuela de Ingeniería de Sistemas e Informática
Bucaramanga, Colombia.

¹Artículo recibido en Marzo de 2016; artículo aceptado en mayo de 2016

Citación del artículo: Jaimes, C. Rojas, F. (2016). Captura de requerimientos mediante ingeniería inversa de interfaces gráficas de usuario. I+D Revista de Investigaciones, 7(1), 61-77.

Resumen

La Captura de Requerimientos es una actividad fundamental al momento de encontrar y entender las funcionalidades requeridas y las necesidades de los usuarios. Este artículo presenta una técnica para la captura de requerimientos, aplicando ingeniería inversa a la interfaz gráfica de usuario (GUI) de sistemas heredados. La técnica busca, dentro de los componentes de la GUI, el lenguaje de la interfaz y sus patrones de diseño respuestas sobre la funcionalidad de la aplicación heredada, convirtiéndose en una herramienta útil para el analista a la hora de obtener un conocimiento previo del dominio y las necesidades de los usuarios.

La técnica denominada ReFree (Reverse Engineering for Requirements Elicitation) se expone por medio de dos ejemplos de aplicación, para los que no se cuenta con acceso a su código fuente, datos o información sobre diseño y construcción. Lo interesante es que a pesar de la naturaleza totalmente diferente de los dominios de aplicación, el análisis realizado a las interfaces gráficas arroja resultados prometedores en cuanto a los requerimientos funcionales recuperados.

Palabras clave: captura de requerimientos, interfaces gráficas de usuario, ingeniería inversa, características,

ingeniería de software.

Abstract

Requirements elicitation is one of the fundamental activities to finding and understanding the required functionality and user needs. This article presents a technique for requirements elicitation from legacy systems applying Reverse Engineering to its Graphical User Interfaces (GUIs). The technique looks at the components of the GUI, the interface language and design patterns, answers about the functionality of the legacy application, making it a useful tool for the analyst when obtaining prior domain knowledge and the user needs.

This article presents a technique called ReFree (Reverse Engineering Requirements for Elicitation) using two examples of application. For both cases, there is not access to its source, data or information about its design and construction. What is interesting is that despite the completely different nature of application domains, the graphical interfaces analysis yields the promising results regarding the recovered functional requirements.

Keywords: requirements elicitation, graphical user interfaces, reverse engineering, features, software engineering.

¹Artículo de Investigación, enfoque cualitativo, resultado de un proyecto de investigación culminado, perteneciente al área de Ingeniería de Sistemas, sub área de Ingeniería de Software, desarrollado en el GIIB Grupo de Investigación en Ingeniería Biomédica y fue financiado por la Universidad Industrial de Santander de la ciudad de Bucaramanga (Colombia). Dirección Cra. 27 calle 9, PBX:(57) (7) 6344000 2337. Fecha de inicio: Enero 2014, fecha de terminación: Noviembre 2015.

²Ingeniera de Sistemas, Universidad Industrial de Santander. Investigador del grupo: GIIB Grupo de Investigación en Ingeniería Biomédica. Universidad Industrial de Santander de la ciudad de Bucaramanga (Colombia): Dirección Cra. 27 calle 9, PBX:(57) (7) 6344000 ext. 2489. Correo electrónico institucional: carol.jaimes@correo.uis.edu.co

³Ingeniero de Sistemas, Universidad Industrial de Santander. Magister en Ciencias Computacionales, Instituto Tecnológico de Monterrey – Universidad Autónoma de Bucaramanga. Profesor Asociado - investigador del grupo: GIIB Grupo de Investigación en Ingeniería Biomédica. Universidad Industrial de Santander de la ciudad de Bucaramanga (Colombia): Dirección Cra. 27 calle 9, PBX:(57) (7) 6344000 2337. Correo electrónico institucional: frojas@uis.edu.co.

Introducción

La ingeniería de Software se ha organizado en once áreas de conocimiento con el fin de estandarizar el proceso de software (Bourque & Dupuis, 2004). Sin embargo, por décadas el desarrollo de software ha enfrentado incontables fracasos (ACIS, 2012; The Standish Group International, 2013). Esta crisis del software (Dijkstra, 1972) se deriva de prácticas irregulares durante el desarrollo de la aplicación. Estas prácticas producen graves fallas que, en la mayoría de los casos, tienen origen en la captura de requerimientos, siendo así los más costosos de corregir cuando se detectan en etapas posteriores del proceso de software (Briski et al., 2008).

Es por esto que en el contexto de la ingeniería de software, la captura de requerimientos se considera una actividad esencial, esta hace parte de la especificación del software, la cual se define como la etapa donde los servicios del sistema, constantes y metas, son establecidas mediante consultas hechas a los interesados en el sistema (Sommerville, 2011), es decir que durante esta etapa se produce el insumo para el resto del proceso de software.

Para obtener un correcto conocimiento del dominio y una comprensión total de las necesidades de los interesados aún hay algunas limitaciones por vencer las cuales se relacionan con la adquisición correcta del conocimiento del dominio y con la interacción efectiva con los interesados (Lamsweerde, 2009); sin embargo, a pesar que la mayoría de metodologías proponen actividades de intensa interacción con el usuario, las experiencias en proyectos de software de los autores de este trabajo y una previa exploración realizada en empresas de la región deja entrever que la práctica en la industria presenta debilidad en la dedicación de tiempo de parte de los usuarios para la realización de estas actividades.

De aquí surge una clara pregunta de investigación ¿es posible dar al analista una herramienta que le permita conocer previamente aspectos del dominio y de las necesidades de los usuarios con el fin de mejorar la calidad de tiempo de interacción con el usuario a la hora de realizar la captura de requerimientos?

Teniendo en cuenta el escenario actual de desarrollo de software, es un hecho que en la mayoría de organizaciones ya se cuenta con aplicaciones, lo que da origen a la oportunidad de realizar una de las actividades más importante dentro de la captura de requerimientos: la adquisición del conocimiento del dominio, infiriéndolo desde dichas aplicaciones (conocidas como software heredado), apoyándose en un enfoque práctico de la ingeniería inversa.

En este trabajo de investigación se busca dar una nueva perspectiva a la captura de requerimientos y a las hasta ahora fuentes identificadas y usadas para recolectar estos requerimientos. Se presenta una alternativa que basada en enfoques clásicos permite explotar el software heredado y a través de este reconocer elementos de funcionalidad relacionados con el dominio de la aplicación.

Para lograr esto, la GUI se somete a un análisis detallado que se logra a partir de la observación y registro de los elementos que la componen, y el reconocimiento del lenguaje implícito que esta contiene. Lo anterior se complementa con el reconocimiento de patrones de diseño de la interfaz gráfica, lo cual arroja información complementaria sobre la funcionalidad del sistema bajo análisis y permite derivar un listado de requerimientos expresado en un formato de lenguaje natural, entendible para todos los interesados.

Método

El desarrollo de la presente investigación se aborda desde un enfoque cualitativo debido a que la ingeniería de software es una actividad intensamente humana (Crawford, Soto, de la Barra, Crawford, & Olguín, 2014), por lo anterior se estudian principios fundamentales de ingeniería de software relacionados con aspectos como el lenguaje del dominio, el diseño de la interfaz y la forma de navegación de un usuario dentro de una aplicación.

ReFree se basa fundamentalmente en el análisis del lenguaje incorporado en la GUI y las acciones que realizan los usuarios con el sistema. Basándose en los principios dados por (Abbott, 1983) en cuanto al uso del lenguaje para diseñar aplicaciones partiendo de descripciones redactadas en lenguaje natural. ReFree registra palabras clave y acciones individuales que permiten encontrar funcionalidades específicas así como apropiarse del vocabulario y dominio de la aplicación.

Además de tener en cuenta el uso del lenguaje, ReFree añade al análisis de la interfaz elementos importantes en el diseño de GUIs, estos son los modelos de navegación y los modelos de interacción. Los modelos de navegación permiten organizar y dar una estructura a las vistas que se van a analizar, de acuerdo a como el usuario navega a través de la aplicación. Los modelos de interacción establecen consistencia a través del artefacto de software, y determinan como los usuarios se mueven a través y entre las diferentes piezas de funcionalidad. La mayoría de las vistas de una aplicación usan por lo menos uno de los patrones mostrados en la figura 1 o una combinación de los mismos (Tidwell, 2011). Estos patrones permiten predecir de manera general la funcionalidad básica de la vista que se va a analizar.

Figura 1. Lista esquemática de los patrones de diseño de GUI agrupados en los distintos modelos de interacción a los que pertenecen.

Patrone de diseño de GUI
Mostrar una sola cosa
Multiples espacios de trabajo
Vistas Alternativas
Mostrar una lista de cosas
Destacado, búsqueda y navegación
Corriente de noticias
Administrador de imágenes
Proveer herramientas para construir una cosa
Multiples espacios de trabajo
Vistas alternativas
Lienzo y paleta
Facilitar una tarea
Asistente
Editor de con figuración

Fuente: Los autores

En cuanto a la forma final que van a tener los requerimientos obtenidos, se selecciona el sistema de características de software descrito por la metodología FDD (Feature Driven Development) (Palmer & Felsing, 2002) ya que este se enfoca especialmente en las funcionalidades del sistema y se presenta en un lenguaje natural (Brett D. McLaughlin, 2006).

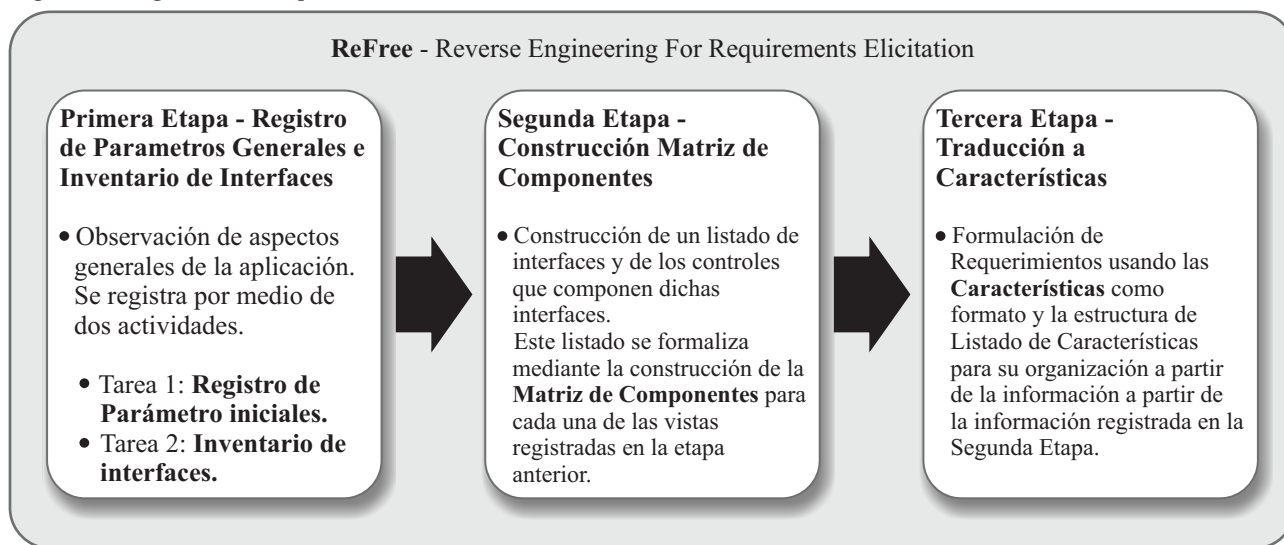
La evolución del desarrollo de la técnica se conduce mediante iteraciones definidas que van arrojando versiones iniciales de la misma, sin embargo, se tienen en cuenta los siguientes hitos como las condiciones que debe cumplir la técnica tanto en sus características propias como en las situaciones donde es aplicable: (a) Ninguna intervención ni observación del código fuente. (b) Poca o ninguna interacción con los usuarios. (c) Ninguna interacción con las personas involucradas en el diseño. (d) Inexistencia de documentación sobre el diseño o de las modificaciones hechas a través del tiempo en la aplicación. (e) Una aplicación ejecutable que está cumpliendo parcialmente con las necesidades del negocio y que es utilizada, pero que ya requiere ser reemplazada por una que cumpla con esas necesidades ya satisfechas y con las más nuevas, sean funcionales o de tecnología. (f) Una aplicación ejecutable de la cual se tiene conocimiento, se usa de manera efectiva en un dominio determinado y de la cual se quiere conocer qué requerimientos puede satisfacer. (g) Obtener consideraciones hechas en el diseño en un formato natural y de fácil entendimiento para los interesados.

A continuación se presenta una descripción de la técnica obtenida y de los resultados de su aplicación.

ReFree (Reverse Engineering for Requirements Elicitation)

El enfoque propuesto para la captura de requerimientos a partir de sistemas heredados consta de tres etapas fundamentales. La primera es el registro de parámetros iniciales e inventario de interfaces, la segunda es la construcción de las matrices de componentes y la tercera es la traducción a características, ver figura 2.

Figura 2. Diagrama de bloques técnica ReFree.



Fuente: Los autores

En las secciones subsiguientes se describen cada una de las etapas de ReFree al ser aplicada sobre los sistemas VTiger y K2MedicalWEB.

ReFree sobre VTiger y K2MedicalWEB

Sistema VTiger

VTiger es un sistema de administración de la relación con los clientes o CRM (Client Resource Management) que cuenta con una versión paga y otra versión open source, esta última puede ser fácilmente descargada, instalada y ajustada a las necesidades de una empresa. La versión utilizada en este ejercicio es la open source 6.1.0(VTiger, 2015).

Sistema K2MedicalWeb

El sistema K2MedicalWEB es un sistema de información para la gestión de servicios médicos actualmente utilizado en la Cruz Roja Colombiana Seccional Santander, gracias a la colaboración de la empresa MTI Manejo de Tecnologías de la Información S.A.S. quien para esta fecha se encarga de la gestión de sistemas en dicha institución fue posible llegar a conocer la situación de esta aplicación y las necesidades de la organización las cuales demuestran la necesidad de una técnica que permita recuperar información sobre la aplicación heredada.

Las problemáticas que en este momento aquejan tanto a la aplicación K2MedicalWEB y a la organización en la cual esta implementada y en continuo uso son: (a) Ausencia de soporte por parte de los desarrolladores o distribuidores de la aplicación. (b) Algunos de los módulos de la aplicación están comenzando a presentar fallas. (c) Evolución en los procedimientos realizados por la organización lo que requiere la actualización del sistema. (d) Crecimiento de la organización (usuarios y clientes) lo que demanda mejora en el manejo de los datos. (e) No es posible realizar copias de seguridad de los datos almacenados por el sistema de información. (f) La organización no cuenta con un acceso

pleno a los datos almacenados, ni al diseño de la base de datos. (g) La organización no cuenta con fuentes de la aplicación. (h) El sistema parece estar siendo utilizado parcialmente debido a la imposibilidad de realizar ajustes y a la falta de conocimiento sobre sus características. (i) El sistema presenta comportamientos erráticos esporádicamente. (j) No existe ningún tipo de documentación sobre descripción de la aplicación, configuración o diseño del sistema. (h) Usuarios no disponibles, no involucrados y en especial dispersos en distintas regiones del departamento.

Lo anterior es un panorama desalentador para el futuro de esta aplicación y en especial para la seguridad y eficiencia de los procesos desarrollados, en este caso se requiere con urgencia un análisis de la aplicación existente. Se busca tener una idea más general de la aplicación y en especial determinar las funcionalidades que brinda el sistema.

ReFree Primera Etapa: Parámetros generales e inventario de interfaces

En esta primera etapa deben desarrollarse dos tareas: el Registro de Parámetros Iniciales y el Inventario de Interfaces.

Registro de parámetros iniciales

El registro de parámetros iniciales es una forma ordenada y simple de registrar lo que parece obvio. Estos datos preliminares dan una idea al analista del dominio en el que se adentra a través de la interacción con esta aplicación. La información obtenida en esta primera etapa proviene de la interacción con algún usuario o encargado, y en la mayoría de los casos es la única información que se obtiene a través de entrevistas breves. Lo anterior se consigna en el Formato de Parámetros Iniciales. En la tabla 1 se observa el formato de VTiger y en la tabla 2 el obtenido para el K2MedicalWEB.

Tabla 1. *Formato de parámetros iniciales sistema VTiger*

Formato Parámetros Iniciales	
Nombre de la aplicación: VTiger	Fecha: 05/04/2015
Tipo de aplicación: Web	
S.O sobre el cual se ejecuta: Windows 8 en el navegador Google Chrome	
Propósito u objetivo de la aplicación: La aplicación es una CRM (<i>Client Resource Management</i>) de código abierto, es decir un software para la administración de la relación con los clientes, su objetivo principal es gestionar las relaciones con los clientes y las actividades que conlleven esta relación, como por ejemplo órdenes de compra, facturas, productos, oportunidades, presupuesto, etc. .	
Quiénes son los usuarios de la aplicación: Empleados, empresas interesadas en el manejo asistido de la relación con el cliente. Sin embargo para este caso el uso es dado únicamente por el desarrollador de este proyecto.	



Forma en que se usa la aplicación: En este momento se está usando como una de las plataformas de prueba para la técnica ReFree.

Dónde se usa la aplicación: Uso en un único computador personal.

Número de Módulos para analizar:
Uno (1) – Modulo Calendario

Número de Formatos para analizar:
Once (11)

Observaciones: Para efectos del desarrollo del ejemplo se tienen en cuenta solo algunas interfaces, estas están determinadas por el árbol de interfaces que se desarrolla más adelante.

Nota: *Información obtenida a través de una entrevista breve. ** Cantidad de módulos y formatos (vistas) para realizar el análisis con ReFree.

Tabla 2. *Formato de parámetros iniciales sistema K2MedicalWEB.*

Formato Parámetros Iniciales	
Nombre de la aplicación: K2MedicalWEB	Fecha: 16/09/2015
Tipo de aplicación: Web	
S.O sobre el cual se ejecuta: La aplicación está publicada en el servidor web IIS de un Windows Server 2008.	
Propósito u objetivo de la aplicación: * Administración de la información relacionada a los servicios médicos prestados por la Cruz Roja Colombiana en sus principales sedes en el departamento de Santander.	
Quiénes son los usuarios de la aplicación: * Funcionarios del sector salud y administrativos vinculados con los procesos de la Cruz Roja Colombiana en la seccional Santander. Más específicamente médicos, enfermeras, contabilidad y administración.	
Forma en que se usa la aplicación: * Se usa desde consultorios y zonas de información. Los usuarios registran los datos de los pacientes. Los supervisores y coordinadores consolidan y analizan la información. Además se generan reportes a las entidades de vigilancia y control.	
Dónde se usa la aplicación: * La aplicación se unas en las seccionales de Bucaramanga y Barrancabermeja (Santander - Colombia).	
Número de Módulos para analizar: ** 6 Módulos	Número de Formatos para analizar:** Cincuenta y dos (52)
Observaciones: El análisis de esta aplicación involucra los seis módulos principales presentados al inicio de la misma, para todos los módulos se llega a un primer nivel de profundidad y para el módulo de pacientes se analiza hasta el segundo nivel.	

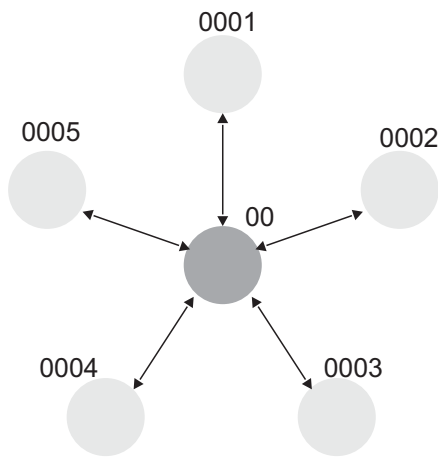
*Nota:**Información obtenida a través de una entrevista breve. ** Cantidad de módulos y formatos (vistas) para realizar el análisis con ReFree.

Inventario de interfaces

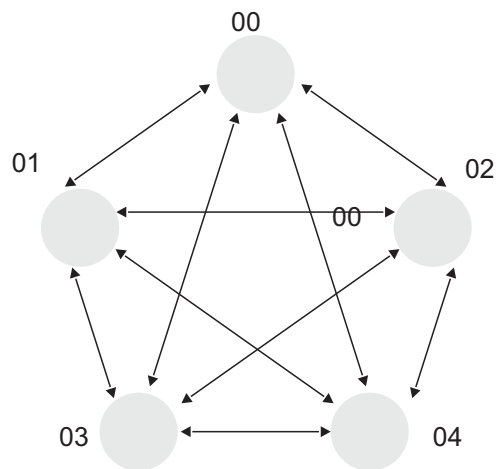
El inventario de interfaces consiste en la recopilación y organización de las interfaces que van a ser sometidas a análisis con ReFree. En esta tarea se asignan los identificadores a cada una de las interfaces, se sugiere tener en cuenta la plantilla de modelos de navegación figura 3.

Esta guía muestra los modelos de navegación más comunes en los sistemas software y las diferentes maneras para asignar los identificadores únicos de las interfaces que conforman la aplicación. El uso de estos modelos e identificadores no es obligatorio, el analista puede usarlos o adaptarlos de acuerdo a sus necesidades.

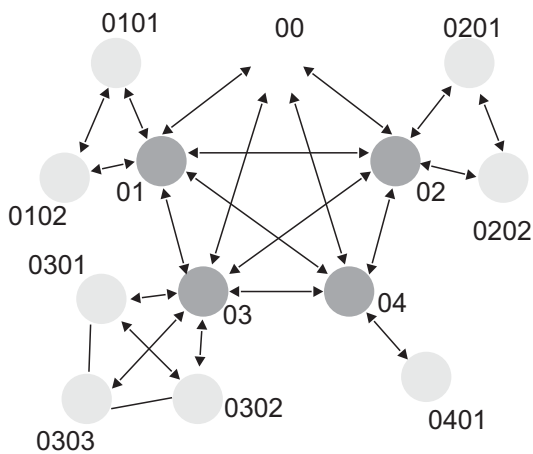
Figura 3. Identificadores para las vistas a analizar con ReFree usando los modelos de navegación.



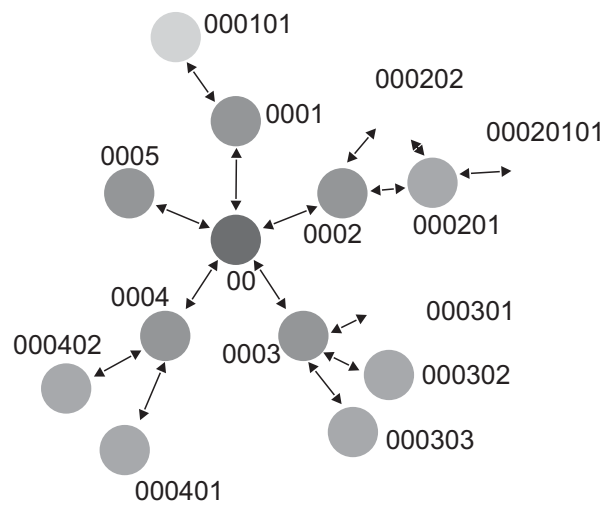
Identificadores modelo
 Concentrador y Radios



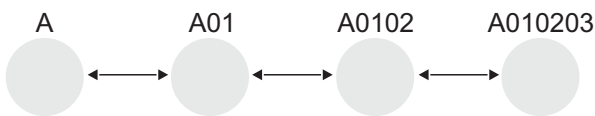
Identificadores modelo
 Totalmente Conectado



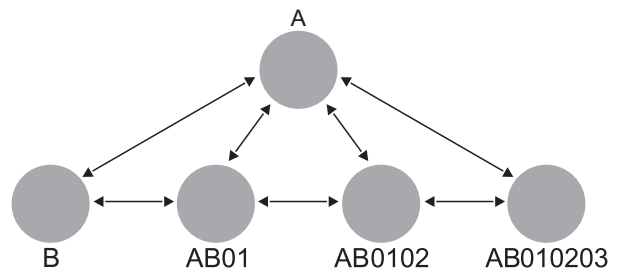
Identificadores
 Multinivel Caso A



Identificadores
 Multinivel Caso B



Identificadores modelo
 Paso a Paso



Identificadores modelo
 Piramide

Fuente: Los autores



El listado de interfaces permite no solo registrar los identificadores y nombres de las interfaces, también permite a través de otras herramientas simples como la estructura, los colores y hasta el tipo de letra usado, registrar eventos importantes y repetitivos dentro del sistema. Los

errores, la falta de datos, y los diseños poco eficientes son reconocibles y documentados de manera práctica y sencilla. En la tabla 3 y 4 se visualizan segmentos de las listas de interfaces realizadas para VTiger y K2MedicalWEB respectivamente.

Tabla 3. Segmento listado de interfaces VTiger

00				
Inicio				
01	0101	<i>0104</i>	<i>010401</i>	<i>01040101</i>
Contenedor	Calendario	<i>Contenedor</i>	<i>Creación Rápida</i>	<i>Creando Nuevo</i>
Calendario	Semana	<i>Creación Rápida</i>	<i>Evento</i>	<i>Evento</i>
		<i>Evento/Tarea</i>		
			<i>010402</i>	<i>01040102</i>
			<i>Creación Rápida</i>	<i>Creando Nuevo</i>
			<i>Tarea</i>	<i>Calendario</i>
	0102	<i>0104</i>	<i>010401</i>	<i>01040101</i>
Calendario	Calendario	<i>Contenedor</i>	<i>Creación Rápida</i>	<i>Creando Nuevo</i>
Mes	Mes	<i>Creación Rápida</i>	<i>Evento</i>	<i>Evento</i>
		<i>Evento/Tarea</i>		
			<i>010402</i>	<i>01040102</i>
			<i>Creación Rápida</i>	<i>Creando Nuevo</i>
			<i>Tarea</i>	<i>Calendario</i>
	0103	<i>0104</i>	<i>010401</i>	<i>01040101</i>
Calendario	Calendario	<i>Contenedor</i>	<i>Creación Rápida</i>	<i>Creando Nuevo</i>
Día	Día	<i>Creación Rápida</i>	<i>Evento</i>	<i>Evento</i>
		<i>Evento/Tarea</i>		
			<i>010402</i>	<i>01040102</i>
			<i>Creación Rápida</i>	<i>Creando Nuevo</i>
			<i>Tarea</i>	<i>Calendario</i>
	0104	<i>010401</i>		
Contenedor	Contenedor	<i>Creación Rápida</i>		
Creación Rápida	Creación Rápida	<i>Evento/Tarea</i>		
Evento/Tarea	Evento/Tarea			
a	a			

Nota: Segmento de la lista de interfaces de VTiger, los elementos cursivos indican la repetición de aparición de ciertas vistas a medida que se navega en la aplicación. Fuente: Autores del Proyecto.

Tabla 4. Segmento listado de interfaces K2MedicalWEB

0 Login	00 Página Principal - Usuarios	0001 Agenda Citas	
		0002 Listado Pacientes	Bloque 01
			000201 Nuevo Paciente
			000202 Editar Historia Clínica - Datos Básicos
			000202A Editar Historia Clínica – Visitas
			000202B Editar Historia Clínica – Procedimientos
			000202C Editar Historia Clínica - Imágenes
			000202D Editar Historia Clínica - Informes
			000202E Editar Historia Clínica - Certificados
			000202F Editar Historia Clínica - Antecedentes
			000202G Editar Historia Clínica - Odontogramas
			000202H Editar Historia Clínica - Reconocimientos Laborales
		0003 Nueva Cita Paso 1: Seleccionar Paciente	
		0004 Informe de Servicios por Médico	
		0005 Resumen Citas	
		0006 Informe Procedimientos por Especialista	

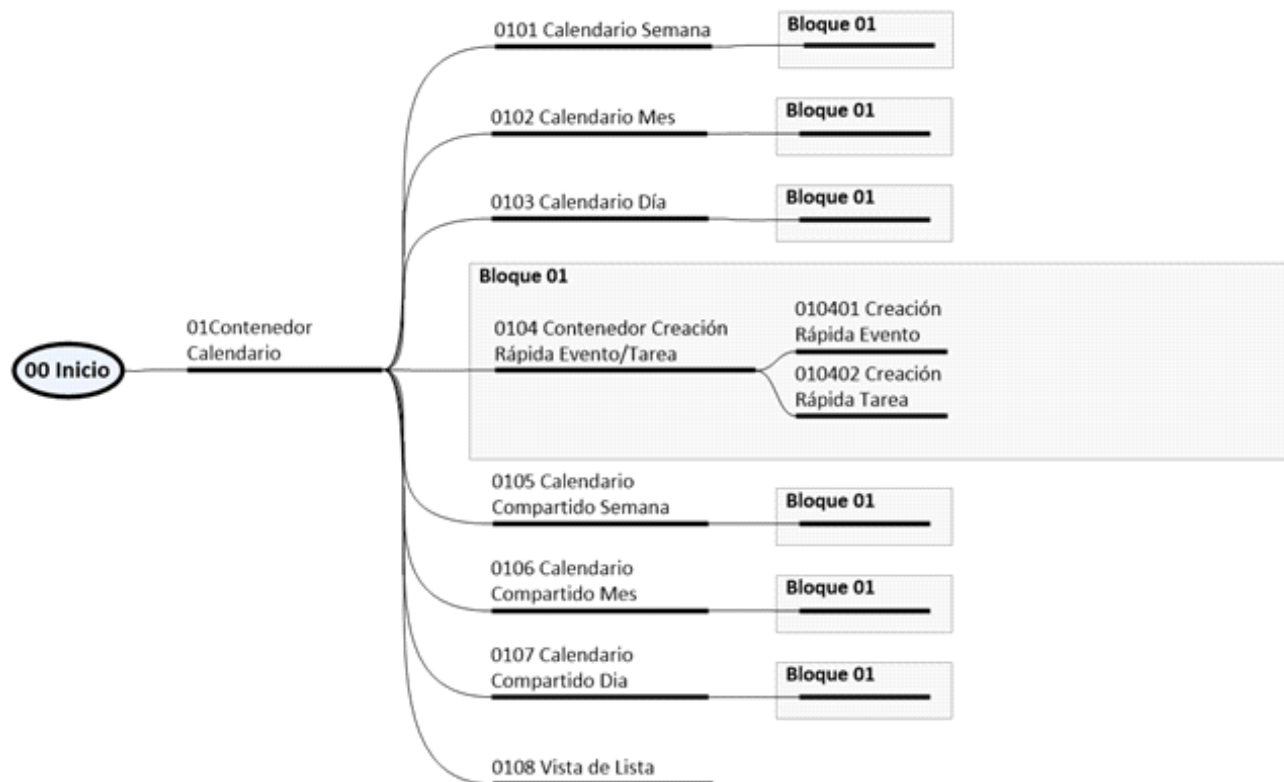
Nota: Segmento de la lista de interfaces de K2MedicalWEB, los elementos en negrilla indican que este tipo de interfaces tienen acceso limitado por el tipo de usuario provisto debido a que es un sistema del sector médico.

A continuación se construye el árbol de interfaces que se deriva de la lista de interfaces y es una representación visual de la misma y permite al analista comprender de una manera

esquemática los distintos lugares que va a recorrer en su análisis y cómo se vinculan entre ellos. El árbol para el sistema VTiger se muestra en la figura 4 y para K2MedicalWEB en la figura 5.

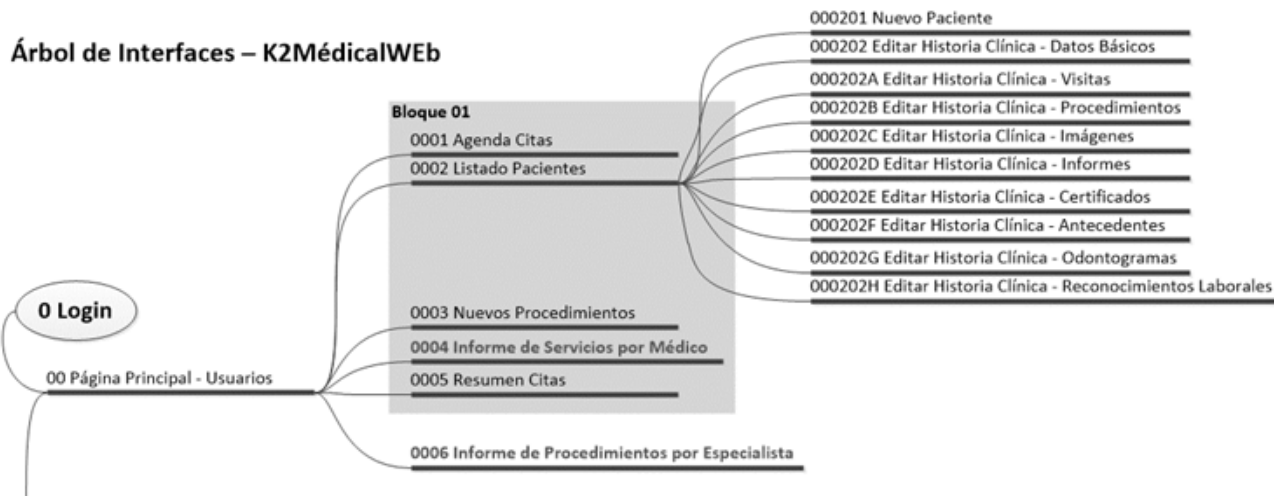


Figura 4. Árbol de interfaces construido para el análisis de VTiger, en este se visualizan los distintos niveles de profundidad que tiene la aplicación y la forma en que se navega a través de las diferentes vistas del módulo Calendario. Fuente: Autores del Proyecto.



Fuente: Los autores

Figura 5. Segmento árbol de interfaces construido para el análisis de K2MedicalWEB, en este se visualizan los distintos niveles de profundidad que tiene la aplicación y la forma en que se navega a través de las diferentes vistas del módulo Usuarios. Fuente: Autores del Proyecto.



Fuente: Los autores



ReFree Segunda Etapa: construcción matriz de componentes

La matriz de componentes es una manera ordenada y sistemática de registrar lo observable en la interfaz gráfica

de una aplicación. Para cada una de las vistas incluidas dentro del inventario de interfaces del paso anterior, se debe hacer un registro detallado de todos los elementos que la componen. Para esto es necesario completar la Matriz de Componentes, el formato de dicha matriz se muestra en la

Figura 6. Formato para el registro de cada uno de los componentes de la GUI, del verbo o acción asociado al componente, el nombre o elemento sobre el que se realiza la acción y la descripción de la acción. Fuente: Autores del Proyecto.

Matriz ID	Nombre Modulo/Vista				
Modelo de Interacción/ Patrón	Descripción				
Componente	Imagen Componente	Verbo	Nombre	Función	Observaciones

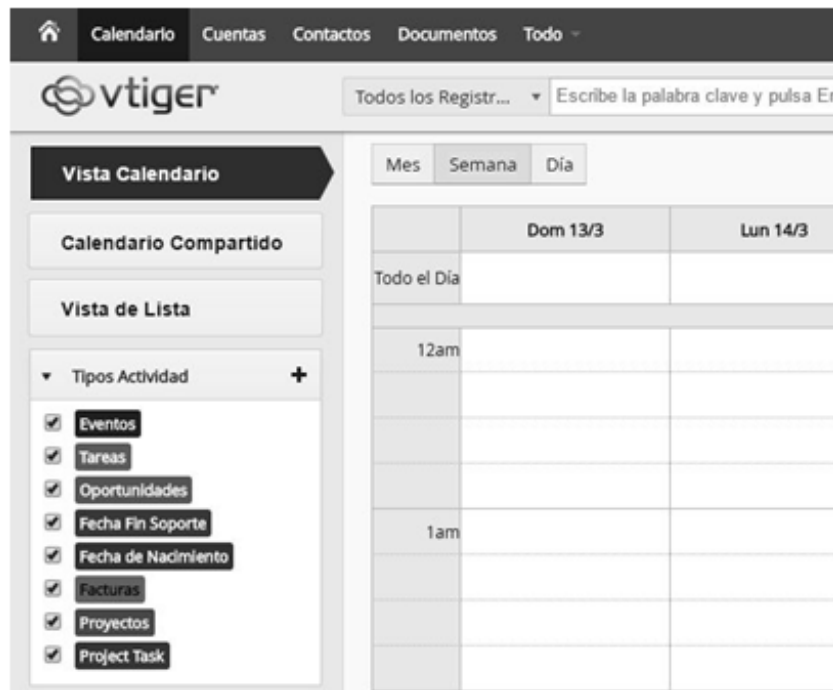
Fuente: Los autores

figura 6.

En la matriz de componentes se registra el patrón de interacción de la vista en análisis (patrones dados en la figura 7), este permite clasificar y describir de lo que de manera general lo que sucede en una determinada interfaz. A su vez se reconocen los Nombres y las Funciones que ocurren dentro de la interfaz, además de los datos manipulados por el sistema.

Para VTiger se registran un total de once vistas para analizar pero debido a las similitudes en algunas de ellas se construyen solo seis matrices de componentes en vez de once para evitar trabajo redundante. La vista “01 Contenedor Calendario” de VTiger de la figura 7 tiene la matriz de componentes de la tabla 5.





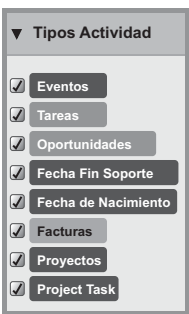
Figura 7. Segmento vista versión open source de VTiger instalado localmente. Copyright 2015 Sistema VTiger.



Fuente: Los autores



Tabla 5. Segmento listado de interfaces VTiger

Matriz	Nombre Modulo/Vista	Contenedor Calendario			
ID01					
Modelo de Interacción/ Patrón Vistas' Alternativas	Descripción La aplicación permite que el usuario escoja entre vistas alternativas que son substancialmente diferentes de la vista por defecto. La interfaz cuenta con un "interruptor" para cambiar el modo de visualización. En estas vistas alternativas alguna información puede ser añadida y alguna puede ser retirada, sin embargo el contenido básico sigue siendo el mismo. En este caso el contenedor presenta vistas alternativas del Calendario y Calendario Compartido, los calendarios se usan para el manejo de una agenda del usuario y otra compartida con otros usuarios.				
Componente	Imagen Componente	Verbo	Nombre	Función	Observaciones
Button		Mostrar	Calendario	Muestra dentro de la interfaz 01 la vista el calendario.	Vista desplegada y por defecto: 0101
Button		Mostrar	Calendario Compartido	Muestra dentro de la presente interfaz (01) la vista de calendario compartido semanal	Vista desplegada: 0105
Button		Mostrar	Listado Actividades	Muestra dentro de la presente interfaz (01) el listado de Eventos/Tareas del usuario.	Vista Desplegada: 0108
Collapsible Panel		Mostrar	Detallado de las actividades registradas o modificadas recientemente en el calendario.	El panel muestra una lista de los nombres de las actividades al dar clic en los nombres se despliega una vista con la información detallada de la actividad.	Este es un acceso rápido a las últimas actividades agregadas o modificadas.
Collapsible Panel		Mostrar/ Ocultar	Tipos de Actividad	El panel contiene un grupo de Check Box para seleccionar los tipos de actividad que quieren ser vistos en el calendario.	

Notas:

Nota: Matriz de componentes para la vista de la figura 7, sistema Vtiger.

Para esta segunda etapa en el caso del K2MedicalWEB

se construyeron cincuenta y dos matrices de componentes, una para cada una de las interfaces registradas en la primera etapa de ReFree. Para ilustrar este proceso se muestra en la figura 8 la interfaz identificada como 0 Login y en la tabla 6



Figura 8. Vista de Login del sistema K2MedicalWEB. Copyright 2015 Sistema K2MedicalWeb.


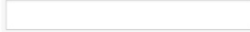
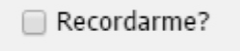
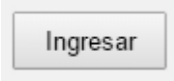


Fuente: Los autores

Tabla 6. Matriz de componentes 0 interfaz Login K2MedicalWEB

Matriz ID 0	Nombre Modulo/Vista	Login			
Modelo de Interacción/ Patrón	Descripción				
	No se selecciona ningún patrón de interacción debido a la sencillez de la vista.				
Componente	Imagen Componente	Verbo	Nombre	Función	Observaciones
Image		Mostrar	Logos	Muestra el nombre de la aplicación y la institución en la que esté implementada.	Carácter informativo.
Label		Mostrar	Tipo de Información	La etiqueta indica el tipo de información que está siendo requerida por los campos que rodea.	Carácter informativo.



<i>Text Box</i>		Ingresar	Nombre	Espacio para ingresar el nombre de usuario.	
<i>Text Box</i>		Ingresar	Contraseña	Espacio para ingresar la contraseña	
<i>Check Box</i>		Seleccionar	Recordar Usuario	Permite seleccionar la opción de mantener la sesión iniciada.	
<i>Button</i>		Validar	Información	Realiza la validación de los datos ingresados en las cajas de texto por el usuario.	Si la información es válida el sistema muestra la vista 00 Página Principal

Nota:

El sistema no cuenta con algún mecanismo de ayuda para recuperar la contraseña.
 Nota: Matriz de componentes para la vista de la figura 8, sistema K2MedicalWEB.

la matriz de componentes obtenida a partir de esta

Tercera Etapa: Traducción a características

Después de completar el registro de los componentes de la interfaz gráfica se procede a realizar la traducción de cada uno de los registros que componen la Matriz de Componentes en características del sistema, que representan requerimientos de la aplicación.

La entrada más importante de este proceso es el conjunto de matrices de componentes realizadas en la etapa anterior. Para cada una de estas matrices se debe obtener una pequeña lista de características, que describe las diferentes

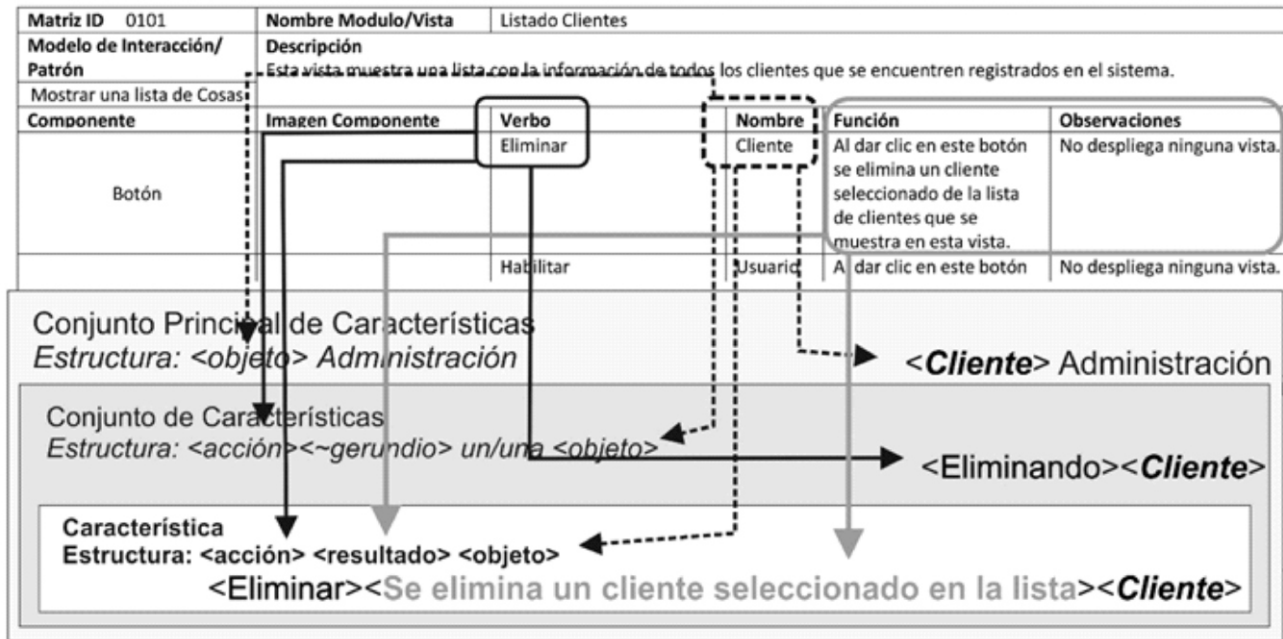
funciones que el sistema lleva a cabo a través de dicha interfaz y su interacción con el usuario. Esta etapa consiste en utilizar la plantilla existente para la construcción de características utilizando los elementos recolectados y organizados en las etapas anteriores de ReFree.

En la figura 9 se puede observar la plantilla de características para un caso genérico y cómo los distintos elementos registrados en la matriz de componentes son organizados para generar características, conjuntos de características y conjuntos mayores de características. .



Figura 9. Plantilla para la construcción de características, los formatos por (Palmer & Felsing, 2002).

1. Organización de los elementos de la MC en la plantilla de Características



2. Redacción característica a partir de la plantilla construida

Conjunto Principal de Características Candidato:

-Administración de Clientes

Conjunto de Características Candidato:

-Eliminando Clientes

Característica:

-"El sistema permite **eliminar** un **cliente** seleccionado de la lista de clientes"

Al someter cada uno de los registros de la matriz de componentes de la tabla 5 del sistema VTiger al esquema dado en la plantilla de la figura 8 se obtiene para esta la siguiente lista de características:

Matriz de Componentes 01

Nombre del Módulo o Vista: Contenedor Calendario

Conjunto Principal de Características Candidato:
 Administración Calendario

Conjunto de Características Candidato: No Definido

1. La vista cuenta con botones que permiten cargar dentro de ella diferentes vistas del calendario personal y calendario compartido.
2. El sistema muestra la vista de calendario.
3. El sistema muestra la vista de calendario compartido.
4. El sistema muestra el calendario en forma de lista.
5. El sistema cuenta con un acceso rápido para mostrar la información detallada de las últimas actividades registradas o modificadas.
6. El sistema permite seleccionar que actividades mostrar u ocultar en el calendario de acuerdo al tipo de actividad.

En el caso del sistema K2MedicalWEB las características encontradas para la matriz de componentes 0 de la vista Login mostrada en la etapa anterior se presentan a continuación:

Matriz de Componentes 0

Nombre del Módulo o Vista: Login

Conjunto Principal de Características Candidato:
 No definido

Conjunto de Características Candidato: No Definido

1. El sistema muestra los logos tanto del nombre de la aplicación como de la institución en la cual esta implementada. Esta información debe estar visible en todo momento durante el uso de la aplicación, en la parte superior de la interfaz.
2. El sistema muestra el tipo de información que requiere para realizar el ingreso a través de una etiqueta. En este caso marcada como "Información de la cuenta".
3. El sistema solicita ingresar en el espacio dado el



- nombre de usuario.
4. El sistema solicita ingresar en el espacio dado la contraseña de usuario.
 5. El sistema permite al usuario seleccionar la opción de recordar la información de usuario suministrada.
 6. El sistema evalúa los datos ingresados por el usuario para permitir o no el acceso del dicho usuario.

Resultados

Por medio de estos casos de aplicación se logra observar que ReFree es una técnica sencilla que permite extraer información de la interfaz gráfica de usuario de una aplicación heredada sin requerir acceso a su código fuente o bases de datos. Además, la técnica ReFree presenta la información recuperada en un formato sencillo y en lenguaje natural que permite que esta sea entendida por los diferentes interesados que participan en el proyecto, especialmente clientes y usuarios.

De igual forma, se reafirma la importancia de la información implícita dentro de las aplicaciones heredadas, cómo este tipo de aplicaciones es una fuente de información importante para aprender sobre los dominios de dichas aplicaciones, y para entender las necesidades de los usuarios ya satisfechas, evitando reinventar la rueda cada vez que se generen nuevos sistemas para un determinado dominio.

En cuanto a los casos de aplicación expuestos, es importante resaltar que para el sistema VTiger, el análisis de once vistas permite encontrar cincuenta y dos características relacionadas con el funcionamiento de ese segmento de la aplicación. En el K2MedicalWEB se toman un total de cincuenta y dos vistas para análisis y se obtiene un total de doscientas cincuenta y cinco características. Esto genera una idea del alto número de características con las que cuenta un sistema terminado que al realizar un rediseño o un nuevo sistema son pasadas por alto. A su vez, esta amplia cantidad de información recuperada de manera organizada puede ser utilizada en próximas actividades del ciclo de vida del software.

A través del análisis propuesto para el K2MedicalWEB no solamente se han identificado funciones que el sistema realiza, también se identifican errores y módulos que parecen no estar siendo utilizados dentro del sistema. Todos estos elementos son clave al realizar cualquier tipo de mantenimiento o rediseño de la aplicación. Y que, en este caso, son reportados para posibles decisiones organizacionales sobre el futuro de la aplicación.

Conclusiones

ReFree puede verse como una técnica aplicable a casos en que existe una aplicación heredada que esté siendo o haya sido utilizada por los usuarios, basada en la observación de la interfaz gráfica que obtiene una lista de descripciones en lenguaje natural del sistema analizado. Esta lista tiene como objetivo primordial el poder conocer el dominio de la aplicación y la aplicación heredada de manera más autónoma y el poder realizar tareas de documentación,

mantenimiento o rediseño.

Las etapas descritas por ReFree son de carácter genérico y pueden ser aplicadas a cualquier tipo de aplicación ejecutable que cuente con una interfaz gráfica de usuario, independientemente del dominio y tecnología de implementación. Sin embargo, en el caso en que no se cumplan las condiciones anteriormente mencionadas es imposible aplicar la técnica ya que esta se basa fundamentalmente en el análisis de los componentes gráficos de la interfaz.

En la ingeniería de requerimientos una de las actividades que puede requerir mayor tiempo es la de adquisición de conocimiento sobre el dominio de la aplicación. En ese sentido la técnica, tal como ha sido expuesta, ofrece la posibilidad de aliviar la curva de aprendizaje acerca del dominio de la aplicación, desde la perspectiva del desarrollador, es decir, desde su interacción con la aplicación heredada, lo que le permitirá, apropiarse del lenguaje del usuario, para inferir los elementos de información pertinentes para él y descubrir las características de la aplicación que dirigen los casos de uso.

En un futuro se puede considerar el uso de técnicas ReFree para mejorar la interacción con el usuario a la hora de iniciar un nuevo proyecto de software, optimizando la calidad del tiempo que se cuenta para interactuar con el usuario ya que el diseñador conoce previa y ampliamente elementos del dominio a abordar. El factor de calidad al que se apunta en este caso es el aumento de la eficiencia en el uso del tiempo de interacción desarrollador-usuario que se requiere para la realización del proceso de ingeniería de requerimientos; de esta manera, se espera que esta investigación junto a otras de carácter similar, como el Análisis Estructurado de Documentos, lleguen a desarrollar todo un cuerpo de conocimiento relacionado con este enfoque.

Se propone como siguiente paso a la aplicación de la técnica continuar con las tareas estándar del proceso de ingeniería de requerimientos: análisis, especificación y validación, ya que, como se propone en la temática, esta técnica se enfoca en ser una herramienta para la captura de requerimientos, es decir, se preocupa por obtener la mayor cantidad de información sobre el sistema desde una fuente como una aplicación heredada. No se pretende con esto desestimar ninguna de las etapas posteriores en la ingeniería de requerimientos, sino que como especifica (Bourque & Dupuis, 2004) en *Guide to the Software Engineering Body of Knowledge* estas son sub-áreas independientes del proceso de requerimientos.

Se propone para un trabajo futuro de investigación incluir la presente técnica dentro del proceso de software, tarea que no se presenta en el alcance de esta investigación debido a la extensión de tiempo que implica un desarrollo de un proyecto completo de software.

A su vez se enfatiza que la comparación con técnicas similares en ingeniería inversa no es posible ya que, hasta el momento del desarrollo del estado del arte para esta investigación, las técnicas de ingeniería inversa relacionadas con el análisis de interfaces gráficas se relacionan especialmente con el análisis estático y dinámico del código fuente o la estructura de los datos, haciendo imposible una comparación. No obstante, se propone como un trabajo futuro la posible comparación de ReFree con técnicas estándar para la captura de requerimientos, aunque es importante añadir que este tipo de experimentos en ingeniería de software son de muy difícil replicación y que los resultados arrojados pueden verse afectados por los tipos de proyectos, tiempos, equipos, e incluso relaciones entre los individuos que componen dichos equipos, lo cual puede arrojar resultados válidos solo para casos específicos.

Además, a través del estudio realizado en este trabajo, se encuentra la posibilidad de regresar al concepto de ingeniería inversa, alejando su uso de la exclusiva aplicación de complicados algoritmos y técnicas orientadas más a la observación y a la ejecución automática de instrucciones con resultados que, en algunos, casos se presentan en formatos de gran complejidad.

Agradecimientos

Agradecemos a la Universidad Industrial de Santander y a MTI Manejo de Tecnologías de la Información S.A.S por su apoyo en la realización de este proyecto de investigación.

Referencias

Abbott, R. J. (1983). Program design by informal English descriptions. *Communications of the ACM*, 26(11), 882894. doi:10.1145/182.358441.

ACSIS. (2012). X Encuesta de Gerencia de Proyectos. Bogotá: Asociación Colombiana de Ingenieros.

Bourque, P., & Dupuis, R. (2004). Guide to the Software Engineering Body of Knowledge 2004 Version. SWEBOK 2004 Guide to the Software Engineering Body of Knowledge. doi:10.1109/SESS.1999.767664.

Brett D. McLaughlin, G. P. & D. W. (2006). Head First Object-Oriented Analysis and Design. (O. Mary, Ed.) (First.). Sebastopol, CA: O'Reilly. Retrieved from http://pdf76.vmnbook.com/head-first-object-oriented-analysis-and-design_44236.pdf

Briski, K. A., Chitale, P., Hamilton, V., Pratt, A., Starr,

B., Veroulis, J., & Villard, B. (2008). Minimizing code defects to improve software quality and lower development costs. *Development Solutions*. IBM. Crawford, B., Soto, R., de la Barra, C. L.,

Crawford, K., & Olguín, E. (2014). The Influence of Emotions on Productivity in Software Engineering (pp. 307310). doi:10.1007/978-3-319-07857-1_54.

Dijkstra, E. W. (1972). The humble programmer. *Communications of the ACM*, 15(10), 859866. doi:10.1145/355604.361591.

Lamsweerde, A. van. (2009). *Requirements Engineering: From System Goals to UML Models to Software Specifications* (1st ed.). England: Wiley.

Palmer, R. S., & Felsing, M. J. (2002). *A Practical Guide to Feature-Driven Development*. Upper Saddle River: Prentice-Hall Inc.

Sommerville, I. (2011). *Ingeniería del Software* (9th ed.). México: Pearson Education.

The Standish Group International. (2013). CHAOS MANIFESTO 2013: Think Big, Act Small. The Standish Group International, 152. Retrieved from <http://www.standishgroup.com>

Tidwell, J. (2011). *Designing Interfaces*. (M. Treseler, Ed.) (2nd ed.). O'Reilly.

VTiger. (2015). VTiger CRM. Retrieved from <https://www.vtiger.com/open-source/>

Sistema VTiger (2015). [Gráfico]. Recuperado de <https://www.vtiger.com/>

Sistema K2MedicalWEB (2015). [Gráfico]. Recuperado de Servidores Cruz Roja Colombiana.

