



UNIVERSIDAD DE CARABOBO
FACULTAD EXPERIMENTAL DE CIENCIAS Y TECNOLOGÍA
DEPARTAMENTO DE COMPUTACIÓN

SOFTWARE PARA EL ESPECTROFOTÓMETRO MINISCAN XE PLUS USADO EN
EL DIAGNÓSTICO DE PATOLOGÍAS DERMATOLÓGICAS EN PACIENTES. CASO
DE ESTUDIO: CIMBUC.

AUTOR:

Gabriel Núñez

TUTORES:

Prof. Patricia Guerrero

Prof. Harold Vasquez

Naguanagua, 27 de julio de 2015

Resumen

El espectrofotómetro de reflexión difusa, denominado MiniScan XE Plus, es un instrumento de medición utilizado por el Centro de Investigaciones Médicas y Biotecnológicas de la Universidad de Carabobo (CIMBUC), que ayuda a los dermatólogos a establecer diagnósticos sobre patologías en la piel de pacientes, de manera precisa y sin necesidad de realizar biopsias. No obstante, el software comercial disponible para la utilización de tal instrumento es poco amigable, difícil de utilizar e imposible de modificar y extender. La presente investigación tiene como objetivo desarrollar un software amigable, modificable y extensible, que se ajuste a las necesidades de los dermatólogos y que garantice un mejor aprovechamiento del instrumento en cuestión.

Palabras claves: espectrofotómetro, análisis bioquímico de la piel, biopsia, software privativo, software libre.

Abstract

The diffuse reflectance spectrophotometer, called MiniScan XE Plus, is a measurement instrument used by the Medical Research and Biotechnology Center at the University of Carabobo (CIMBUC), which helps dermatologists to establish pathologies diagnoses in the skin of patients precisely, without need for biopsy. However, the available commercial software for the use of such an instrument is unfriendly, difficult to use and impossible to modify and extend. This research aims to develop a friendly, modifiable and expandable software that meets the needs of dermatologists and ensures a better use of the instrument itself.

Keywords: spectrophotometer, biochemical analysis of the skin, biopsy, privative software, open source software.

Capítulo 1

El Problema

«Software privativo significa que priva a los usuarios de su libertad.»

– Richard Stallman

1.1 Planteamiento del Problema

Bersha (2010) indica que durante el diagnóstico de enfermedades de la piel, la observación cuidadosa y la evaluación visual del área sospechada es siempre el primer paso, y el más importante. Esto es seguido generalmente por una escisión o biopsia por punción, en la que se extrae una muestra de tejido de la piel para un análisis microscópico. La observación visual suele ser subjetiva, y los pacientes a menudo se someten a cicatrices y dolor durante la biopsia. Por otro lado, las técnicas ópticas son por lo general no invasivas, y sus resultados son a menudo objetivos. Durante el diagnóstico no invasivo, no se crea ninguna ruptura en la piel, y los pacientes no se someten al dolor ni a cicatrices durante el tratamiento.

Los avances tecnológicos de la actualidad permiten emplear técnicas de óptica,

que tienen la capacidad de estudiar las propiedades estructurales y bioquímicas del tejido biológico, de manera precisa y no invasiva. Los instrumentos que emplean tales técnicas son de gran ayuda para los médicos dermatólogos, razón por la cual dichos instrumentos han tomado suma importancia en el área médica dermatológica.

Hoy en día existen diferentes tipos de estudios ópticos in-situ, in-vivo e invitro del tejido biológico, como la Espectroscopía de Reflectancia Difusa (ERD). Pérez (2012) asegura que con esta técnica es posible estudiar las propiedades bioquímicas y las condiciones estructurales de un tejido biológico, analizando la interacción luz-tejido de una manera no invasiva.

En este sentido, el Centro de Investigaciones Médicas y Biotecnológicas de la Universidad de Carabobo (CIMBUC) dispone de un espectrofotómetro de reflexión difusa, denominado MiniScan XE Plus, creado por la empresa HunterLab. Esta empresa lo describe como un instrumento utilizado para medir la transmisión y/o reflectancia de especímenes, como una función de longitud de onda, que aplica la técnica de ERD.

Ahora bien, el CIMBUC hace uso de este instrumento a través del software disponible para su utilización, designado HunterLab Universal Software (HLUS), que es un software comercial y privativo de 16 bits, diseñado para el sistema operativo Microsoft Windows versión 3.x, con la posibilidad de ejecutarse en Windows 95, Windows 2000 y Windows XP. Este software contiene funciones que abarcan la utilización del MiniScan XE Plus, y de otros instrumentos ofrecidos por HunterLab; además fue descontinuado en el año 2008. La interfaz gráfica de usuario de este software está en inglés. Por último, los resultados que genera este software no poseen el formato de gestión de información de pacientes con el que trabajan los dermatólogos del CIMBUC.

Tomando en cuenta lo mencionado anteriormente, se tiene que el HLUS es un software privativo y que está descontinuado, por lo tanto no existe la posibilidad de modificarlo ni extenderlo; ofrece funciones ajenas al uso exclusivo del MiniScan

XE Plus, causando que la interfaz gráfica de usuario contenga más opciones de las necesarias para manejar tal instrumento. Asimismo, como consecuencia de que la interfaz gráfica de usuario esté en inglés, esta es difícil de entender por los dermatólogos. Sumado al hecho de que los resultados generados por este software no poseen el formato con el que trabajan los dermatólogos, haciendo necesario su traspaso manual, lo que produce una ralentización en las consultas con pacientes. Todo esto conlleva a que los dermatólogos requieran de asistencia técnica especializada para la debida utilización de dicho software.

De lo antedicho se desprende que, el HLUUS posee una interfaz gráfica de usuario poco amigable, y el costo del tiempo de capacitación para su uso correcto podría ser alto. Este software no podrá modificarse ni extenderse, por lo tanto no se fomentará el uso del instrumento en cuestión, disminuyendo su potencial. Por último, tampoco se fomentará el desarrollo de nuevas aplicaciones que utilicen sus resultados como insumo, sosegando así la posibilidad de realizar análisis más complejos, y de proveer a los dermatólogos con resultados que les permitan establecer diagnósticos más completos.

1.2 Justificación

Con respecto a software de calidad, Sommerville (2005) explica lo siguiente: así como los servicios que proveen, los productos de software tienen un cierto número de atributos asociados que reflejan su calidad. Estos atributos no están directamente relacionados con lo que hace el software; más bien, reflejan su comportamiento durante su ejecución, en la estructura y organización del programa fuente, y en la documentación asociada.

El conjunto específico de atributos que se espera de un software de calidad depende obviamente de su aplicación. Esto se generaliza en el conjunto de atributos que se muestran en la Tabla 1, en la cual se pueden apreciar las características

esenciales de un software de calidad.

Tabla 1. *Atributos esenciales de un software de calidad* (Fuente: Sommerville, 2005).

Característica	Descripción
Mantenibilidad	El software debe describirse de tal forma que pueda evolucionar para cumplir las necesidades de cambio de los clientes. Este es un atributo crítico, debido a que el cambio en el software es una consecuencia inevitable de un cambio en el entorno de negocios.
Confiabilidad	Este atributo tiene un gran número de características, incluyendo la fiabilidad, protección y seguridad. El software confiable no debe causar daños físicos o económicos en el caso de una falla del sistema.
Eficiencia	El software no debe hacer que se malgasten los recursos del sistema, como la memoria y los ciclos de procesamiento. Por lo tanto, la eficiencia incluye tiempos de respuesta y de procesamiento, utilización de la memoria, etcétera.
Usabilidad	El software debe ser fácil de utilizar, sin esfuerzo adicional por el usuario para quien está diseñado. Esto significa que debe tener una interfaz gráfica de usuario apropiada, y una documentación adecuada.

Debido a que el HLUS es privativo, el CIMBUC no dispone de su código fuente, de manera que este software no puede ser modificado ni adaptarse a necesidades específicas, y, por lo tanto, no posee el primer atributo esencial para un software de calidad: la mantenibilidad. Por la misma razón, no se puede determinar con certidumbre el segundo atributo: la confiabilidad (madurez del software y tolerancia a fallas); además de que no se puede evaluar completamente el nivel de protección y seguridad del mismo. Por último, la usabilidad de este software es baja, ya que la interfaz gráfica de usuario es poco amigable. Por estas razones, se desarrolló un software que cumpliera con los atributos esenciales que debe poseer un software de calidad.

Sommerville (2005) señala que el diseño cuidadoso de la interfaz gráfica de usuario es una parte fundamental del proceso de diseño general del software. Si un software debe alcanzar su potencial máximo, es fundamental que su interfaz gráfica de usuario sea diseñada para ajustarse a las habilidades, experiencia y expectativas de sus usuarios previstos. Un buen diseño de la interfaz gráfica de usuario es crítico

para la confiabilidad del software. Muchos de los llamados errores de usuario son causados porque las interfaces gráficas de usuario no consideran las habilidades de los usuarios reales y su entorno de trabajo.

El diseño de la interfaz gráfica de usuario del HLUUS es la principal razón por la cual los dermatólogos requieren de personal técnico especializado, que los asista al momento de utilizarlo. Esto porque dicha interfaz está en inglés, ofrece funciones que no son necesarias para la utilización del MiniScan XE Plus, y no proporciona el formato con el que trabajan los dermatólogos. Por estas razones los dermatólogos perciben este software comercial como no intuitivo, ni auto descriptivo ni amigable, temiendo cometer errores al utilizarlo por su propia cuenta y generar resultados erróneos, poniendo en riesgo el diagnóstico, y en consecuencia, la salud de los pacientes en consulta.

En conclusión, siguiendo los lineamientos del diseño y la calidad del software que se consideraron pertinentes, se desarrolló un software amigable, modificable y extensible, que ofrece las funciones que necesitan los dermatólogos para establecer diagnósticos, emplea el formato de historia médica con el que trabajan, y permite guardar los resultados en un formato de archivo portátil. Por último, se creó una base sobre la cual se podrán trabajar proyectos futuros que necesiten utilizar los resultados de este nuevo software como insumo.

1.3 Objetivos de la Investigación

En la siguiente sección se especifican los objetivos del trabajo, distinguiendo entre el objetivo general y los objetivos específicos.

1.3.1 Objetivo General

Desarrollar un software para el espectrofotómetro MiniScan XE Plus, usado en el diagnóstico de patologías dermatológicas en pacientes, tomando como caso de estudio el CIMBUC.

1.3.2 Objetivos Específicos

- Investigar el estado del arte referente a las características de software para espectrofotómetros de reflexión difusa, el diseño y la calidad del software.
- Seleccionar una metodología de investigación y una metodología de desarrollo para el nuevo software.
- Diseñar y desarrollar el nuevo software, siguiendo las metodologías seleccionadas.
- Diseñar las pruebas para el nuevo software.
- Elaborar el manual de usuario del nuevo software.

Capítulo 2

Marco Teórico

2.1 Antecedentes

- En el artículo titulado “Recuperación del Coeficiente de Absorción de la Epidermis en la Piel Humana” de Narea y otros (2015), se obtiene el coeficiente de absorción de la epidermis en la piel humana a partir de datos espectrales. El procedimiento para obtener dicho coeficiente es implementado en el nuevo software para determinar el nivel de concentración de melanina en la epidermis de un paciente.
- En la tesis de maestría de Bersha (2010) titulada “Spectral Imaging and Analysis of Human Skin”, se calcula el índice de eritema partiendo de la coordenada a^* , correspondiente al espacio de color *CIELAB*. El método para la obtención del índice mencionado es implementado en el nuevo software, para determinar el nivel inflamatorio de la epidermis de un paciente.

2.2 Observación Directa

- **HunterLab Universal Software:** Es un software propietario de 16-bit diseñado para el Sistema Operativo Microsoft Windows Version 3.x, con la posibilidad de ejecutarse en Windows 95, Windows 2000, Windows NT y Windows XP, descontinuado en el año 2008. Este software dispone de algunas de funcionalidades desarrolladas en el nuevo software, razón por la cual es una importante referencia.
- **MiniScanXE Plus OCX Kit:** Es un archivo de control ActiveX diseñado por HunterLab para controlar y/o realizar mediciones con el “MiniScan XE Plus”, utilizando Visual Basic for Applications (VBA). Su principal objetivo es proveer a los desarrolladores con un componente reutilizable de software que da acceso a las características más comunmente utilizadas por el instrumento. La interfaz pública que expone este archivo es utilizada para realizar la comunicación entre el “MiniScan XE Plus” y el nuevo software.

Capítulo 3

Marco Metodológico

3.1 Metodología Investigación-Acción

La Investigación-Acción se orienta a la acción y al cambio, a la focalización de un problema y posee un modelo de proceso “orgánico” que engloba tanto etapas sistemáticas como iterativas, ayudando a resolver así problemas prácticos y a expandir el conocimiento científico.

Esta metodología tiene una doble finalidad: generar un beneficio al cliente de la investigación y al mismo tiempo, generar conocimiento de investigación relevante. Por lo tanto, esta metodología es una forma de investigar de carácter colaborativo que busca unir teoría y práctica entre investigadores y practicantes mediante un proceso naturaleza cíclica.

La representación más habitual de la Investigación-Acción es la descrita por Baskerville (1999), la cual se muestra a continuación en forma de cinco fases que conforman un ciclo (Ver Figura 2), que se describen a continuación.

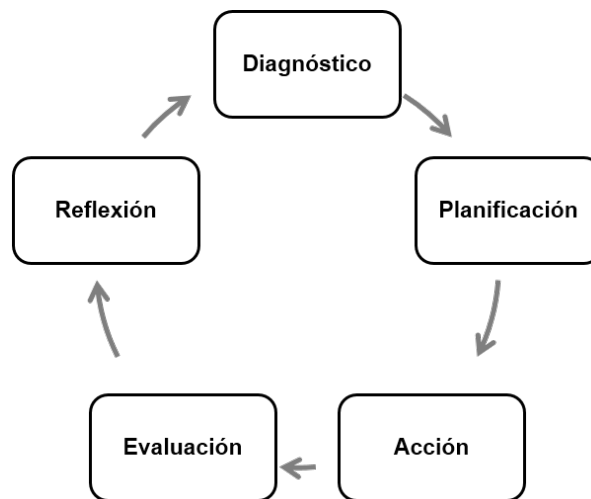


Figura 1. *Carácter cíclico de Investigación-Acción* (Fuente: Baskerville, 1999).

- **Fase de diagnóstico:** Se realiza el proceso de identificación de los problemas primarios de la investigación.
- **Fase de planificación:** Se especifican las acciones que se llevarán a cabo para solucionar los problemas primarios.
- **Fase de acción:** Se ejecutan las acciones planificadas en la fase anterior.
- **Fase de evaluación u observación:** Se efectúa una evaluación de los resultados obtenidos, para observar, conocer y documentar los efectos de las acciones que fueron realizadas.
- **Fase de reflexión:** Se toman los conocimientos adquiridos en la investigación-acción. Si las acciones ejecutadas no fueron exitosas, los conocimientos pueden proporcionar la base para el diagnóstico de un nuevo ciclo de investigación-acción.

En la Tabla 2 se muestran las actividades del presente proyecto, haciendo correspondencia a cada una de las fases de la Investigación-Acción.

Fase	Actividades
Diagnóstico	Identificar los problemas y limitaciones que presenta el software comercial del “MiniScan XE Plus”.
Planificación	Seleccionar la metodología de desarrollo, determinar los requisitos del software y realizar un plan de trabajo.
Acción	Desarrollar el software, tomando en cuenta los requisitos identificados previamente, los lineamientos de ingeniería del software, estándares de diseño y calidad de software.
Evaluación	Realizar las pruebas de funcionalidad del software en cuestión y de su interfaz gráfica de usuario.
Reflexión	Presentar los resultados y los análisis de las pruebas realizadas.

Tabla 2. *Actividades del proyecto según metodología Investigación-Acción* (Fuente: Elaboración propia).

3.2 Metodología de Desarrollo de Software

Para el desarrollo del software que cumpla con los objetivos planteados en esta investigación y tomando en cuenta los lineamientos planteados por la ingeniería del software, con el objetivo de obtener un software que sea fiable y que funcione eficientemente Pressman (2002), se ha realizado una revisión del enfoque que debería tener la metodología de desarrollo a utilizar.

Según Sommerville (2005), en los años 80 y principios de los 90, existía una opinión general de que la mejor forma de obtener un mejor software era a través de una planificación cuidadosa del proyecto, una garantía de calidad formalizada, la utilización de métodos de análisis y diseño soportados por herramientas CASE, y procesos de desarrollo de software controlados y rigurosos. El software que seguía lo mencionado previamente era desarrollado por grandes equipos que a veces trabajaban para compañías diferentes. A menudo estaban dispersos geográficamente y trabajaban en el software durante largos periodos de tiempo.

Ahora bien, debido a que no se dispone de un equipo grande para el desarrollo del software objetivo de la presente investigación, y a que no se trabajará en éste durante un largo periodo de tiempo, se utilizará una metodología de desarrollo de enfoque ágil. Acorde con Sommerville (2005), los métodos ágiles dependen de

un enfoque iterativo para la especificación, desarrollo y entrega del software, y están pensados para entregar software funcional de forma rápida a los clientes, quienes pueden entonces proponer que se incluyan en iteraciones posteriores del software nuevos requerimientos o cambios en los mismos. Si bien los métodos ágiles proponen procesos diferentes para el desarrollo y entrega incrementales de software, comparten unos principios en común, los cuales son ilustrados en la Tabla 3.

Principio	Descripción
Participación del cliente	Los clientes deben estar fuertemente implicados en todo el proceso de desarrollo. Su papel es proporcionar y priorizar nuevos requerimientos del software y evaluar las iteraciones del sistema.
Entrega incremental	El software se desarrolla en incrementos, donde el cliente especifica los requerimientos a incluir en cada incremento.
Personas, no procesos	Se deben reconocer y explotar las habilidades del equipo de desarrollo. Se les debe dejar desarrollar sus propias formas de trabajar, sin procesos formales.
Aceptar el cambio	Se debe contar con que los requerimientos del software cambian, por lo que el software se diseña para dar cabida a estos cambios.
Mantener la simplicidad	Se debe centrar la simplicidad tanto en el software a desarrollar como en el proceso de desarrollo. Donde sea posible, se trabaja activamente para eliminar la complejidad del software.

Tabla 3. Principios de los métodos ágiles (Fuente: Sommerville, 2005).

3.2.1 Metodología SCRUM

De acuerdo con Schwaber y Sutherland (2013), esta metodología es un marco de trabajo por el cual las personas pueden acometer problemas complejos adaptativos, y a la vez entregar productos del máximo valor posible, productiva y creativamente. SCRUM no es un proceso o una técnica para construir productos; en lugar de eso, es un marco de trabajo dentro del cual se pueden emplear varias técnicas y procesos.

El marco de trabajo SCRUM consiste en los equipos SCRUM, roles, eventos, artefactos y reglas asociadas. Cada componente dentro del marco de trabajo sirve a un propósito específico y es esencial para el éxito de SCRUM y para su uso (Schwaber y Sutherland (2013)).

SCRUM se basa en la teoría de control de procesos empírica. El empirismo asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce. Esta metodología emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control de riesgo. La implementación de este control de procesos está soportada por tres pilares, los cuales se muestran en la Tabla 4.

Pilar	Descripción
Transparencia	Los aspectos significativos del proceso deben ser visibles para aquellos que son responsables del resultado.
Inspección	Los usuarios SCRUM deben inspeccionar frecuentemente los artefactos de SCRUM y el proceso hacia un objetivo, para detectar variaciones.
Adaptación	Si un inspector determina que uno o más aspectos de un proceso se desvían de límites aceptables, y que el producto resultante no será aceptable, el proceso o el material que está siendo procesado debe ser ajustado.

Tabla 4. *Pilares del control de procesos de SCRUM* (Fuente: Elaboración propia).

Adicionalmente a la utilización de la metodología SCRUM, se incluyeron algunos artefactos de la metodología RUP (Rational Unified Process), para así generar suficiente documentación durante el diseño y el desarrollo del nuevo software. La configuración de la metodología SCRUM utilizada en conjunto con los artefactos elegidos de la metodología RUP, es la ilustrada en la Tabla 5.

Artefactos SCRUM
Backlog de producto: Lista dinámica de las cosas que se deben hacer, sin especificar cómo se deben hacer.
Backlog de sprint: Recopilación sintética de los ítems del backlog del producto, en donde se quiebran los ítems en tareas pequeñas que no demanden una labor superior a una jornada de trabajo.
Incremento de funcionalidad: Es lo que el equipo SCRUM entrega la final de cada sprint. El mismo debe asemejarse a un software funcionando, permitiendo implementarse operativamente sin restricciones en un ambiente productivo.
Artefactos RUP
Documento de Visión: Documento que define el alcance de alto nivel y propósito del producto.
Glosario: Documento que define la terminología empleada en los artefactos.
Configuración de documentos de requerimientos: Solamente requerimientos no funcionales.
Configuración de documentos de arquitectura: Solamente diagrama de Casos de Uso.

Tabla 5. Configuración de los artefactos a utilizar de SCRUM y RUP (Fuente: Elaboración propia).

Bibliografía

(2001). *Universal Software Versions 4.10 and Above User's Manual*. Hunter Associates Laboratory.

(2006). *MiniScan XE Plus User's Guide Version 2.4*. Hunter Associates Laboratory.

BASKERVILLE, RICHARD L. (1999). «Investigating Information Systems with Action Research».

BERSHA, K. S. (2010). *Spectral Imaging And Analysis Of Human Skin*. Tesina o Proyecto, University of Eastern England.

NAREA, F.; VIVAS, S. y MUÑOZ, A. (2015). «Recuperación del coeficiente de absorción de la epidermis en la piel humana».

PÉREZ, A. D. (2012). *Estudio de la Reflexión Óptica Difusa en Tejido Biológico*. Tesina o Proyecto, Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Zacatenco.

PRESSMAN, R. (2002). *Ingeniería del Software, un enfoque práctico*. McGraw Hill.

SCHWABER, K. y SUTHERLAND, J. (2013). *Scrum guide*.

SOMMERVILLE, I. (2005). *Ingeniería del Software*. Pearson Education.