



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 A. Núñez N.

Tutores:

Prof. Patricia Guerrero
Prof. Harold Vasquez

Naguanagua, Agosto de 2015.



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 A. Núñez N.

Tutores:

Prof. Patricia Guerrero
Prof. Harold Vasquez

Trabajo Especial de Grado presentado ante la ilustre Universidad de Carabobo,
como credencial para optar por el título de Licenciado en Computación.

Naguanagua, Agosto de 2015.

Dedicatoria

Agradecimientos



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.**

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 y los investigadores 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, para así garantizar un mejor aprovechamiento del instrumento en cuestión.

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

Autor:
Gabriel A. Núñez N.

Tutores:
Prof. Patricia Guerrero
Prof. Harold Vasquez

Índice general

Introducción	5
I. El Problema	6
1.1. Planteamiento del problema	6
1.2. Objetivos	9
1.2.1. Objetivo general	9
1.2.2. Objetivos específicos	9
1.3. Justificación de la investigación	10
II. Marco Teórico	12
2.1. Antecedentes de la investigación	12
2.1.1. Trabajos relacionados	12
2.1.2. Observación directa	13
2.2. Bases teóricas	13
III.Marco Metodológico	14
3.1. Metodología de investigación	14
3.1.1. Investigación-Acción	14
3.2. Metodología de desarrollo de software	16

3.2.1. SCRUM	17
3.2.2. Artefactos de RUP utilizados	20
IV.Resultados	22
4.1. Visión	22
4.1.1. Enunciado del problema	22
4.1.2. Descripción de los usuarios	23
4.1.3. Resumen del producto	23
4.1.4. Principales restricciones	24
4.2. Pila del producto (<i>product backlog</i>)	25
4.3. Requerimientos no funcionales	25
4.4. Casos de uso	26
4.5. Glosario	26
V. Conclusiones y recomendaciones	27
5.1. Conclusiones	27
5.2. Recomendaciones	27
Referencias	28

Índice de figuras

3.1. Carácter cíclico de la Investigación-Acción	15
3.2. Configuración de los artefactos a utilizar de SCRUM y RUP . .	21

Índice de tablas

1.1. Atributos esenciales de un software de calidad	11
3.1. Actividades del proyecto según la Investigación-Acción	16
3.2. Principios de los métodos ágiles	17
4.1. Actores del negocio	23
4.2. Usuarios del software	23
4.3. Beneficios y características principales del producto	24
4.4. Requerimientos funcionales del software	25
4.5. Requerimientos no funcionales del software	26

Introducción

Capítulo I

El Problema

En este capítulo se presenta la problemática que motiva el desarrollo de este trabajo de investigación, se describen los objetivos que se tienen pensado cumplir con el mismo, y se explican las razones que justifican la necesidad de su ejecución.

1.1. Planteamiento del problema

El color y la apariencia de la piel humana son importantes en el campo de la medicina. 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. Bersha (2010).

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 y los investigadores, razón por la cual dichos instrumentos han tomado suma importancia en el campo de la medicina 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 reflectancia difusa, denominado MiniScan XE Plus. Este dispositivo fue creado por la empresa HunterLab, y la misma lo describe como un instrumento utilizado para medir la 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 único software disponible para su utilización, designado HunterLab Universal Software. Este es un software comercial y privativo capaz de ejecutarse en sistemas operativos Windows, desde la versión 95 hasta la versión XP. Dicho software contiene funciones que abarcan la utilización del MiniScan XE Plus y de otros instrumentos ofrecidos por HunterLab; su interfaz gráfica de usuario está en inglés, además de que los resultados que genera no poseen el formato de gestión de información de pacientes con el que trabajan los dermatólogos y los investigadores del CIMBUC. Por último, este software fue discontinuado en el año 2008.

Tomando en cuenta lo mencionado anteriormente, se tiene que el HunterLab

Universal Software es privativo y está discontinuado, por lo tanto no existe la posibilidad de modificarlo ni de extenderlo. Dicho software está en inglés y ofrece funciones ajenas al uso exclusivo del MiniScan XE Plus, causando que su interfaz gráfica de usuario contenga más opciones de las necesarias para manejar tal instrumento, y que sea difícil de entender por los dermatólogos y los investigadores. Todo esto sumado al hecho de que los resultados generados por este software deben exportarse y almacenarse manualmente, porque no poseen el formato con el que trabajan los dermatólogos y los investigadores, produce la necesidad de contar con asistencia técnica especializada para su debida utilización y ralentiza las consultas con los pacientes.

De lo antedicho se desprende que, el HunterLab Universal Software posee una interfaz gráfica de usuario poco amigable, y el costo del tiempo de capacitación para su uso correcto podría ser alto. Dicho software no podrá modificarse ni extenderse, por lo tanto no se fomentará el uso del instrumento en cuestión, disminuyendo su potencial. El formato de los resultados de este software convertirá las consultas de los dermatólogos con los pacientes en una labor ineficiente en términos de tiempo. Por último, no se fomentará el desarrollo de nuevas funciones que utilicen sus resultados como insumo, sosegando así la posibilidad de que los investigadores realicen análisis más complejos, y de proveer resultados que les permitan a los dermatólogos establecer diagnósticos más completos de patologías dermatológicas en pacientes.

Motivado a lo anterior, se desarrolló un nuevo software modificable y extensible, con una interfaz gráfica de usuario amigable, con funciones para el uso exclusivo del MiniScan XE Plus, que genera resultados relevantes para los dermatólogos y los investigadores, empleando el uso del formato utilizado por ellos para registrar los resultados de las consultas, y utilizando los lineamientos de la ingeniería del software pertinentes.

Finalmente, con esta investigación se espera fomentar la utilización del MiniScan XE Plus por medio del nuevo software, mejorar la capacitación del

personal médico dermatológico y de investigación para su debido uso, reducir el tiempo de las consultas con los pacientes, aportar un software sobre el cual se puedan desarrollar nuevas investigaciones, que conlleven a la implementación de técnicas que empleen análisis más complejos, y, por último, realizar diagnósticos más completos y diversos sobre patologías dermatológicas presentes en pacientes.

1.2. Objetivos

1.2.1. 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.

1.2.2. Objetivos específicos

- Investigar el estado del arte relacionado a la investigación:
 1. Técnicas de óptica y colorimetría.
 2. MiniScan XE Plus.
 3. HunterLab Universal Software.
 4. Atributos de calidad del software.
- Seleccionar una metodología de investigación y una metodología de desarrollo para el nuevo software.
- Desarrollar el nuevo software, siguiendo las metodologías seleccionadas.
- Realizar las pruebas para el nuevo software.
- Elaborar el manual de usuario para el uso del nuevo software.

1.3. Justificación de la investigación

Empezando con la interfaz gráfica de usuario, Sommerville (2005) señala que el diseño cuidadoso de la misma 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.

Dicho lo anterior, el diseño de la interfaz gráfica de usuario del HunterLab Universal Software es la principal razón por la cual los dermatólogos y los investigadores 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 sus resultados no proporcionan el formato con el que trabajan. Por estas razones los dermatólogos y los investigadores 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 la correctitud del análisis de los resultados, la fiabilidad del diagnóstico, y, en consecuencia, la salud de los pacientes en consulta.

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 la organización del programa fuente, y en la documentación asociada.

El conjunto específico de atributos que se esperan de un software de calidad

Tabla 1.1: *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, la protección y la seguridad. El software confiable no debe causar daños físicos o económicos en caso de que ocurra 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 parte del usuario para quien está diseñado. Esto significa que debe tener una interfaz gráfica de usuario apropiada, y una documentación adecuada.

depende obviamente de su aplicación. En la tabla 1.1 se puede apreciar la generalización de estos atributos.

Debido a que el HunterLab Universal Software es privativo, el CIMBUC no dispone de su código fuente, de manera que este software no puede modificarse 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. Por último, la usabilidad de este software es baja, ya que la interfaz gráfica de usuario es poco amigable.

Las razones descritas anteriormente justifican la necesidad del desarrollo de un nuevo software para el uso del MiniScan XE Plus, que sea amigable, modificable, extensible, y que cumpla con los atributos esenciales de calidad; que emplee el uso del formato de historia médica con el que trabajan dermatólogos y los investigadores, que ofrezca las funciones que ellos necesitan para realizar análisis y establecer diagnósticos de patologías dermatológicas en pacientes. Por último, se ha tomado como caso de estudio el CIMBUC.

Capítulo II

Marco Teórico

Este capítulo presenta los antecedentes, la observación directa y las bases teóricas que sustentaron el desarrollo del presente trabajo de investigación, incluyendo la revisión bibliográfica realizada.

2.1. Antecedentes de la investigación

2.1.1. Trabajos relacionados

Primero se tiene el artículo científico titulado «*Comparing Quantitative Measures of Erythema, Pigmentation and Skin Response using Reflectometry*», realizado por Wagner et al., (2002), en la Universidad del Estado de Pensilvania, Estados Unidos, y publicado por Pigment Cell Res. En este artículo se obtiene el índice de eritema, que es utilizado para determinar el nivel inflamatorio de la epidermis en la piel de un paciente. El método utilizado en este artículo para su obtención fue implementado en el nuevo software.

Por último está el artículo científico titulado «*Recuperación del Coeficiente de Absorción de la Epidermis en la Piel Humana*», realizado por Narea et

al., (2015), en la Universidad de Carabobo, Venezuela, y publicado por la Sociedad Española de Óptica. En este artículo se determina el coeficiente de absorción, que es un parámetro óptico asociado a la piel, el cual indica el nivel de concentración de melanina presente en la epidermis de la piel de un paciente. La técnica empleada en dicho artículo para calcular este coeficiente fue implementada en el nuevo software.

2.1.2. Observación directa

El «*HunterLab Universal Software*», 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 98, Windows 2000 y Windows XP. Fue creado para la utilización del MiniScan XE Plus, además de otros instrumentos de la empresa HunterLab, y discontinuado en el año 2008. Este software dispone de algunas de las funciones que fueron desarrolladas en el nuevo software, razón por la cual fue una referencia importante de observación.

El archivo denominado «*MSXE + OCX*», es una hoja de cálculo habilitada para la ejecución de macroinstrucciones de Microsoft Excel, que fue proporcionada por el personal de soporte técnico de HunterLab como un ejemplo para utilizar el MiniScan XE Plus, empleando el uso de un kit de control denominado MiniScan XE Plus OCX Kit (*MSXE.ocx*). Este kit fue diseñado por la empresa HunterLab para dar acceso a las características comunmente utilizadas por dicho instrumento. El código contenido en la hoja de cálculo se empleó como referencia para observar el manejo del kit *MSXE.ocx*.

2.2. Bases teóricas

Capítulo III

Marco Metodológico

En este capítulo se describen la metodología de investigación y la metodología de desarrollo, que se emplearon para planificar, diseñar y ejecutar el desarrollo del nuevo software propuesto en el presente trabajo de investigación.

3.1. Metodología de investigación

3.1.1. Investigación-Acción

Baskerville (1999) define la Investigación-Acción como un método de investigación que a finales de la década de los 90 empezó a crecer en popularidad, para el uso en investigaciones académicas de sistemas de información. Este método produce resultados de investigación altamente relevantes, debido a que se fundamenta en la acción práctica, dirigida a resolver un problema mientras se informa cuidadosamente sobre la teoría.

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, es una forma de investigar de carácter colaborativo que

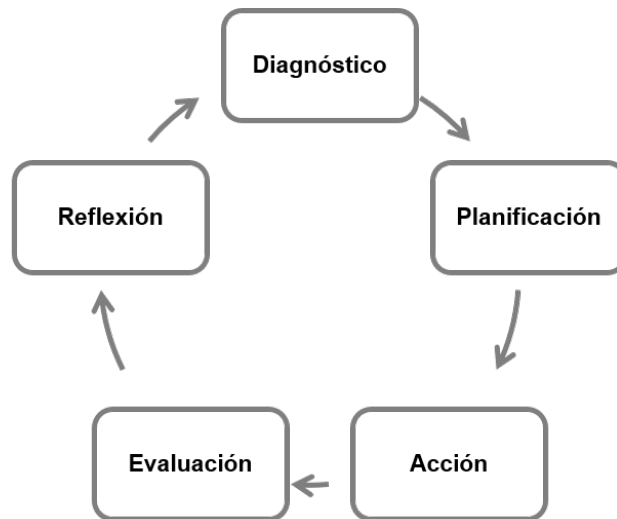


Figura 3.1: *Carácter cíclico de la Investigación-Acción* (Fuente: Baskerville, 1999).

busca unir teoría y la práctica entre investigadores y practicantes, mediante un proceso de naturaleza cíclica.

La representación más habitual de la Investigación-Acción es la descrita por Baskerville (1999), en forma de cinco fases que conforman un ciclo, las cuales se muestran en la figura 3.1 y se describen a continuación:

- **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.

Tabla 3.1: *Actividades del proyecto según la Investigación-Acción* (Fuente: Autor).

Fase	Actividades
Diagnóstico	Identificar los problemas y limitaciones que presenta el HunterLab Universal Software.
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 diseño y de calidad del software.
Evaluación	Realizar las pruebas de funcionalidad e interfaz gráfica de usuario del nuevo software.
Reflexión	Presentar los resultados y los análisis de las pruebas realizadas.

En la tabla 3.1 se muestran las actividades de la presente investigación, haciendo correspondencia a cada una de las fases de la Investigación-Acción descritas por Baskerville (1999).

3.2. Metodología de desarrollo de software

Para que el desarrollo del nuevo software cumpliera con los objetivos propuestos la presente investigación, y tomando en cuenta los lineamientos planteados por la ingeniería del software, se realizó una revisión del enfoque que debería tener la metodología de desarrollo a utilizar.

Según Sommerville (2005), en los años 80 y a 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 por medio de 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, que a menudo estaban dispersos geográficamente y trabajaban en el software durante largos periodos de tiempo.

Tabla 3.2: *Principios de los métodos ágiles* (Fuente: Sommerville, 2005).

Principio	Descripción
Participación del cliente	Los clientes deben estar fuertemente implicados en todo el proceso de desarrollo.
Entrega incremental	El software se desarrolla en incrementos, en los que el cliente especifica los requerimientos a incluir en cada incremento.
Personas, no procesos	Se deben reconocer y explotar las habilidades del equipo de desarrollo. A este se les debe dejar desarrollar su propia forma 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.

Ahora bien, debido a que no se dispuso de un equipo grande para el desarrollo del nuevo software, y a que no se iba a trabajar en este durante un largo periodo de tiempo, se eligió la utilización de 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 3.2.

3.2.1. SCRUM

De acuerdo con Schwaber y Sutherland (2013), esta metodología ágil es un marco de trabajo de procesos, que ha sido utilizado para gestionar el desarrollo de productos complejos desde principios de los años 90. SCRUM muestra la

eficacia relativa de las prácticas de gestión de productos y las prácticas de desarrollo.

La estructura de desarrollo de SCRUM se basa en ciclos de trabajo llamados *sprints*. Los *sprints* son iteraciones de una a cuatro semanas que suceden una detrás de la otra, con una duración fija y con fechas de culminación previamente establecidas. Se seleccionan los requerimientos que se van a desarrollar de una lista priorizada. Todos los días el equipo se reúne, y al final del *sprint* el equipo revisa el mismo con los *stakeholders*.

Hundermark (2015) explica de forma precisa los roles que conforman el equipo de desarrollo de SCRUM:

Los Roles

- **Dueño del producto (*Product Owner*):** su responsabilidad es optimizar el retorno de la inversión, asegurando que el equipo SCRUM este ocupado en entregar las características más valiosas del producto. Su trabajo principal es concentrarse en la efectividad, esto es construir el producto correcto para sus clientes.
- **Equipo de desarrollo:** es una colección de personas responsables por entregar incrementos de la funcionalidad del producto al final de cada *sprint*. El trabajo principal de este equipo es concentrarse en la eficiencia, esto es construir el producto correcto para su *Product Owner* y sus usuarios.
- **Maestro SCRUM (*SCRUM Master*):** gestiona todos los aspectos del proceso del equipo SCRUM. Su trabajo principal es concentrarse en el progreso continuo del equipo, acortando los ciclos de retroalimentación mediante los cuales aprende.

Las Reuniones

Como es sabido, el *sprint* marca cada una de las iteraciones dentro del ciclo de desarrollo de SCRUM. Por otra parte, la planificación, la continua revisión y la retrospectiva definen el inicio y el final del *sprint*. Las reuniones que ocurren en cada *sprint* son las siguientes:

- **Reunión de planificación del *sprint*:** esta reunión marca el inicio de cada *sprint*. Su propósito para el equipo SCRUM es planear el trabajo que van a realizar durante el *sprint* actual.
- **Reunión diaria del *sprint*:** el equipo de desarrollo se reúne para comunicar su trabajo, sincronizarlo, y crear un plan para las siguientes 24 horas. Esta colaboración es esencial para asegurar el progreso continuo y evadir cualquier obstrucción de trabajo.
- **Reunión de revisión del *sprint*:** su propósito primario es el de inspeccionar la iteración del producto que el equipo de desarrollo ha entregado, obtener una retroalimentación de los participantes en la reunión con respecto a la misma, y adaptar el plan para el *sprint* subsiguiente. Esta reunión está abierta para todo el personal dentro de la organización.
- **Reunión de retrospectiva:** es la reunión final del *sprint*, la cual nunca es omitida, sin importar lo que haya ocurrido en dicho *sprint*. Mientras que la reunión de revisión del *sprint* está enfocada en el producto, esta reunión está enfocada en el proceso, es decir, la forma en la que el equipo SCRUM está trabajando en conjunto, incluyendo sus habilidades técnicas, las prácticas de desarrollo del software y las herramientas que están usando. Esta reunión se limita a los miembros del equipo SCRUM.

Los Artefactos

- **Pila del producto (*product backlog*):** es una lista de ítems de trabajo descritos en un nivel funcional, que necesitan ser realizados a lo largo del tiempo. Los requerimientos son emergentes, lo que significa que no se puede saber por adelantado todos los detalles acerca de qué se quiere en el producto. Por esta razón este artefacto es un documento dinámico, que requiere un refinamiento constante para mantenerlo actual y útil.
- **Pila del *sprint* (*sprint backlog*):** esta pila es visualizada por el equipo de desarrollo en un *task board*, que es la representación física de la lista de trabajo que se ha resumido para realizar durante el *sprint* actual. Este artefacto le dice al equipo SCRUM y a todos los demás qué trabajo tienen planeado hacer en el *sprint*, y su estado actual.
- **Incremento:** es la suma de todos los ítems de la pila del producto que cumplen con la definición de terminado al final del *sprint*. El equipo de desarrollo presentará este en la revisión del *sprint*, y el *Product Owner* determinará cuándo liberar este incremento.

En esta metodología se pueden emplear varias técnicas y procesos. Dicho lo anterior, adicionalmente a la utilización de SCRUM, se incluyeron algunos artefactos de la metodología RUP (Rational Unified Process) descrita por Kroll y Kruchten (2003), para así generar suficiente documentación durante el diseño y el desarrollo del nuevo software.

3.2.2. Artefactos de RUP utilizados

- **Documento de visión:** describe la visión de los *stakeholders* con respecto al producto a desarrollarse, especificado en términos de las características y las necesidades clave de estos *stakeholders*.
- **Modelo de casos de uso:** describe los requerimientos funcionales del

software en términos de actores y casos de uso. Un actor representa el tipo de usuario del software, mientras que un caso de uso describe cómo va a interactuar cada actor con el software.

- **Requerimientos no funcionales:** representan los requerimientos que tienen un impacto significativo en la arquitectura y en la satisfacción del usuario.
- **Glosario:** este define la terminología empleada en los artefactos utilizados.

Para finalizar, en la figura 3.2 es ilustrada la configuración de la metodología SCRUM utilizada, en conjunto con los artefactos elegidos de la metodología RUP.

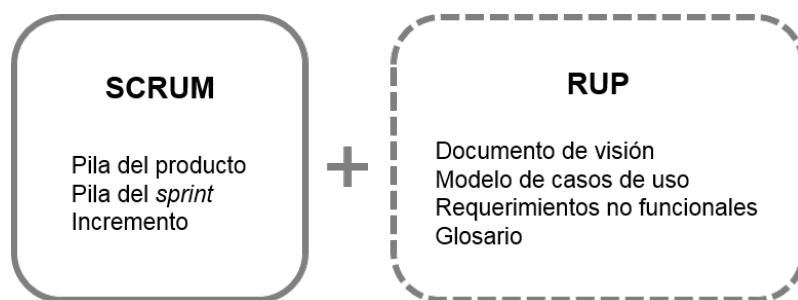


Figura 3.2: Configuración de los artefactos a utilizar de SCRUM y RUP (Fuente: Autor).

Capítulo IV

Resultados

En este capítulo se muestran los resultados que fueron alcanzados en el desarrollo del presente trabajo de investigación, siguiendo la metodología Investigación-Acción, y la configuración de la metodología SCRUM junto con los artefactos incluidos de la metodología RUP.

4.1. Visión

4.1.1. Enunciado del problema

El problema que se presenta es que se está utilizando el HunterLab Universal Software para el manejo del espectrofotómetro de reflexión difusa MiniScan XE Plus, que está en inglés, es comercial, privativo y fue descontinuado; esto afecta a los dermatólogos y los investigadores del Centro de Investigaciones Médicas y Biotecnológicas de la Universidad de Carabobo (CIMBUC).

El impacto causado por esto es que los dermatólogos y los investigadores encuentran el HunterLab Universal Software difícil de utilizar, e imposible de adaptar a sus necesidades, lo que ralentiza la actividad de consulta con los

pacientes, genera la necesidad de disponer de personal especializado para su debido uso, y disminuye el potencial de dicho instrumento.

Una solución satisfactoria sería disponer de un software para el uso del MiniScan XE Plus que esté en español, que sea amigable y mantenible, permitiendo que se adapte a las necesidades de los dermatólogos y los investigadores.

4.1.2. Descripción de los usuarios

Tabla 4.1: *Actores del negocio* (Fuente: Autor).

Actor	Descripción
Dermatólogo	Realiza mediciones a los pacientes. Determina diagnósticos a los pacientes. Gestiona los resultados de las mediciones.
Investigador	Realiza mediciones a los pacientes. Realiza actividades de investigación. Gestiona los resultados de las mediciones.

Tabla 4.2: *Usuarios del software* (Fuente: Autor).

Usuario	Responsabilidad	Experiencia	Uso
Dermatólogo	Calibrar el instrumento. Realizar mediciones. Gestionar los resultados.	Baja	Alto
Investigador	Calibrar el instrumento. Realizar mediciones. Gestionar los resultados.	Media	Alto

4.1.3. Resumen del producto

El software desarrollado, denominado a partir de ahora Spectrasoft, es una aplicación para el uso del MiniScan XE Plus, la recuperación de los datos de medición de dicho instrumento, la generación de resultados relevantes y la gestión de los mismos, el cuál está orientado a las actividades médicas dermatológicas y de investigación del Centro de Investigaciones Médicas y

Bi TECNOLÓGICAS de la Universidad de Carabobo (CIMBUC). La tabla 4.3 resume los beneficios y las características más importantes que provee el producto.

Tabla 4.3: *Beneficios y características principales del producto* (Fuente: Autor).

Beneficio al cliente	Característica que lo soporta
Los dermatólogos y los investigadores pueden conectarse con el MiniScan XE Plus, calibrarlo y realizar mediciones con este instrumento.	Comunicación con el MiniScan XE Plus y acceso a las características comunmente utilizadas por el mismo.
Se dispone de información relevante para el diagnóstico de patologías dermatológicas en la piel de los pacientes.	Muestra de los datos espectrales obtenidos de las mediciones, representación gráfica de los mismos, y cálculo coordenadas, índices y coeficientes asociados a dichos datos.
Se pueden gestionar los resultados de las mediciones empleando el uso del formato con el que trabajan los dermatólogos y los investigadores.	Guardado de los resultados en archivos portátiles que poseen el formato que utilizan los dermatólogos y los investigadores, y cargado de los mismos al Spectrasoft para su visualización.
Los dermatólogos y los investigadores pueden utilizar el Spectrasoft con facilidad.	Interfaz gráfica de usuario en español, la cual ofrece únicamente las funciones que necesitan los dermatólogos y los investigadores.
El Spectrasoft se puede adaptar a futuras necesidades de los dermatólogos y los investigadores.	Código abierto disponible para realizar cualquier modificación y/o extensión.

4.1.4. Principales restricciones

El software se desarrolla utilizando el lenguaje de programación C++ para favorecer su rendimiento, empleando únicamente tecnologías gratuitas, y, en la medida de lo posible, de código abierto. Este debe ser capaz de ejecutarse en sistemas operativos Windows actuales. Por último, la comunicación entre el software y el MiniScan XE Plus se logra tanto por medio de un puerto serial, como por medio de un adaptador USB.

4.2. Pila del producto (*product backlog*)

La lista de requerimientos funcionales que necesitan ser realizados a lo largo del desarrollo del Spectrasoft, son el producto de una reunión que se tuvo con los dermatólogos y los investigadores del CIMBUC, al igual que de la observación directa efectuada a las funciones que el HunterLab Universal Software ofrece para el manejo del MiniScan XE Plus. En la tabla 4.4 se muestran dichos requerimientos.

Tabla 4.4: *Requerimientos funcionales del software* (Fuente: Autor).

Código	Requerimiento	Prioridad
RF01	Estandarizar el MiniScan XE Plus.	Esencial
RF02	Recuperar los 31 puntos espectrales de la medición con el MiniScan XE Plus y mostrarlos en su forma numérica.	Esencial
RF03	Graficar una curva de reflectancia difusa a partir de los 31 puntos espectrales recuperados.	Esencial
RF04	Graficar una curva de absorbancia aparente a partir de los 31 puntos espectrales recuperados.	Esencial
RF05	Calcular y mostrar las coordenadas del sistema <i>CIE XYZ 1964</i> .	Esencial
RF06	Calcular y mostrar las coordenadas del sistema <i>CIE L*a*b* 1976</i> .	Esencial
RF07	Calcular y mostrar el coeficiente de absorción de la epidermis.	Esencial
RF08	Calcular y mostrar el coeficiente de esparcimiento de la epidermis.	Esencial
RF09	Calcular y mostrar el índice de eritema.	Esencial
RF010	Guardar los resultados en un archivo portátil, empleando el uso del formato de historia médica de los dermatólogos.	Esencial
RF11	Abrir y cargar los resultados guardados en el archivo portátil.	Esencial

4.3. Requerimientos no funcionales

De la misma forma que los requerimientos funcionales, la lista de los requerimientos no funcionales fue definida en la misma reunión con los dermatólogos y los investigadores, tomando en cuenta las restricciones del entorno en donde

se va a ejecutar el Spectrasoft. En la tabla 4.5 se describen estos requerimientos no funcionales.

Tabla 4.5: *Requerimientos no funcionales del software* (Fuente: Autor).

Código	Requerimiento	Prioridad
RNF01	El software debe ser fácil de utilizar, por lo que debe cumplir con el atributo de usabilidad de un software de calidad.	Esencial
RNF02	El software debe ser capaz de adaptarse a las necesidades de los dermatólogos, razón por la cual debe cumplir con el atributo de mantenibilidad de un software de calidad.	Esencial
RNF03	El software debe desarrollarse empleando únicamente tecnologías gratuitas, y, en la medida de lo posible, de código abierto.	Esencial
RNF04	El software debe ser capaz de ejecutarse en sistemas Windows actuales, con arquitecturas de 32 bits y 64 bits.	Esencial
RNF05	El software debe conectarse con el MiniScan XE Plus por medio de un puerto serial o de un adaptador USB.	Esencial
RNF06	El archivo portátil que maneja el software debe ser abierto por un visualizador/editor de hojas de cálculo.	Esencial
RNF07	El software debe desarrollarse utilizando el lenguaje de programación orientada a objetos C++.	Esencial

4.4. Casos de uso

4.5. Glosario

Capítulo V

Conclusiones y recomendaciones

5.1. Conclusiones

5.2. Recomendaciones

Referencias

- BASKERVILLE, RICHARD L. (1999). «*Investigating Information Systems with Action Research*». Association for Information Systems, Atlanta – Georgia, Estados Unidos.
- BERSHA, K. S. (2010). «*Spectral Imaging And Analysis Of Human Skin*». Universidad del Este de Finlandia, Finlandia.
- CIE. «*Commission Internationale de l'Eclairage, International Commission on Illumination*». Viena, Austria.
<http://www.cie.co.at/index.php>
- HUNDERMARK, P. (2015). «*Do Better SCRUM*». Agile42 - The Agile Coaching Company.
- HUNTERLAB. «*HunterLab, The World's true measure of color*». Hunter Associates Laboratory, Reston – Virginia, Estados Unidos.
<http://www.hunterlab.com/about-us.html>
- (2001). «*Universal Software Versions 4.10 and Above User's Manual*». Hunter Associates Laboratory, Reston – Virginia, Estados Unidos.
- (2006). «*MiniScan XE Plus User's Guide Version 2.4*». Hunter Associates Laboratory, Reston – Virginia, Estados Unidos.
- KROLL, P. y KRUCHTEN, P. (2003). «*The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP*». Addison-Wesley, Boston – Massachusetts, Estados Unidos.

- MICROSOFT. «*Visual Studio Community, a fully-featured, extensible IDE*».
<https://www.visualstudio.com/products/visual-studio-community-vs>
- NAREA, F.; VIVAS, S. y MUÑOZ, A. (2015). «*Recuperación del coeficiente de absorción de la epidermis en la piel humana*». Sociedad Española de Óptica, Universidad de Carabobo, Venezuela.
- PÉREZ, A. D. (2012). «*Estudio de la Reflexión Óptica Difusa en Tejido Biológico*». Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Zacatenco, México.
- QCUSTOMPLOT. «*A Qt C++ widget for plotting and data visualization*».
<http://www.qcustomplot.com/index.php/introduction>
- QT. «*Qt, a Cross-Platform Framework for Application Development*».
https://wiki.qt.io/About_Qt
- RESEARCH, MAGNETO TECH. «*USB to Serial adapters Wiki*».
<http://www.usb-serial-adapter.org/>
- SCHANDA, J. (2007). «*Colorimetry: understanding the CIE system*». John Wiley & Sons, Hoboken – Nueva Jersey, Estados Unidos.
- SCHWABER, K. y SUTHERLAND, J. (2013). «*The Definitive Guide to Scrum: The Rules of the Game*».
- SOMMERVILLE, I. (2005). «*Ingeniería del Software*». Pearson Education, Madrid, España.
- WAGNER, J.; JOVEL, C.; NORTON, H.; PARRA, E. y SHRIVER, M. (2002). «*Comparing Quantitative Measures of Erythema, Pigmentation and Skin Response using Reflectometry*». Pigment Cell Res, Universidad del Estado de Pensilvania, Estados Unidos.