



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, 4 de agosto de 2015.



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

Trabajo Especial de Grado  
**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, 4 de 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 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, ingeniería biomédica, software privativo.

**Autor:**  
Gabriel A. Núñez N.

**Tutores:**  
Prof. Patricia Guerrero  
Prof. Harold Vasquez

Naguanagua, 4 de agosto de 2015.

# Introducción

# Capítulo I

## El Problema

En este capítulo se presenta la problemática que motiva al desarrollo de este trabajo de investigación, así como también la importancia que tiene para el ámbito médico dermatológico, y los objetivos a lograr con la investigación.

### 1.1. Planteamiento del problema

El color y la apariencia de la piel humana es importante 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, 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, 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 discontinuado 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 HunterLab



Universal Software es un software privativo y que está discontinuado, 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 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 a 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 realizar análisis más complejos, y de proveer a los dermatólogos con resultados que les permitan establecer diagnósticos más completos.

Motivado a lo anterior, se desarrolló un nuevo software modificable y extensible, con una interfaz gráfica de usuario amigable, que emplea el uso del formato utilizado por los dermatólogos para registrar los resultados de las consultas, y con funciones para el uso exclusivo del MiniScan XE Plus, 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 para su debido uso, reducir el tiempo de las

consultas con los pacientes, y, por último, aportar una base sólida sobre la cual se puedan desarrollar nuevos proyectos.

## 1.2. 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 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 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.

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
<b>Mantenibilidad</b>	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.
<b>Confiabilidad</b>	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.
<b>Eficiencia</b>	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.
<b>Usabilidad</b>	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 HunterLab Universal Software 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.

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

## **1.3. Objetivos**

### **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 y realizar las pruebas para el nuevo software.
- Elaborar el manual de usuario para el uso del nuevo software.

# Capítulo II

## Marco Teórico

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

### 2.1. Antecedentes

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 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 este artículo para calcular este coeficiente fue implementada en el nuevo software.

## 2.2. Observaciones directas

El «HunterLab Universal Software», es un software comercial y privativo de 16 bits diseñado para el sistema operativo Microsoft Windows version 3.x, con la posibilidad de ejecutarse en Windows 95, Windows 2000 y Windows XP. Fue creado para la utilización del MiniScan XE Plus, además de otros instrumentos de la empresa HunterLab, y descontinuado 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*.

# Capítulo III

## Marco Metodológico

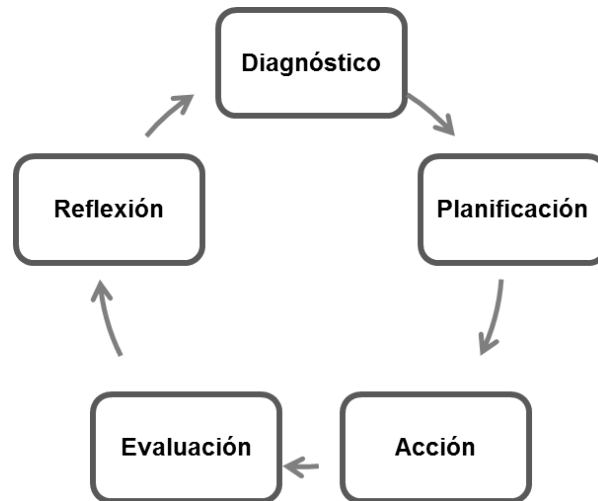
En este capítulo se describen las metodologías de investigación y de desarrollo que se emplearon para planificar y desarrollar el nuevo software propuesto en el presente trabajo de investigación.

### 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 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 describen en la Figura 1.

- **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 de la presente investigación, haciendo correspondencia a cada una de las fases de la Investigación-Acción descritas por Baskerville (1999).

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

Fase	Actividades
<b>Diagnóstico</b>	Identificar los problemas y limitaciones que presenta el HunterLab Universal Software.
<b>Planificación</b>	Seleccionar la metodología de desarrollo, determinar los requisitos del software y realizar un plan de trabajo.
<b>Acción</b>	Desarrollar el software, tomando en cuenta los requisitos identificados previamente, los lineamientos de diseño y de calidad del software.
<b>Evaluación</b>	Realizar las pruebas de funcionalidad e interfaz gráfica de usuario del nuevo software.
<b>Reflexión</b>	Presentar los resultados y los análisis de las pruebas realizadas.

## 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.

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.

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

Principio	Descripción
<b>Participación del cliente</b>	Los clientes deben estar fuertemente implicados en todo el proceso de desarrollo.
<b>Entrega incremental</b>	El software se desarrolla en incrementos, en los que el cliente especifica los requerimientos a incluir en cada incremento.
<b>Personas, no procesos</b>	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.
<b>Aceptar el cambio</b>	Se debe contar con que los requerimientos del software cambian, por lo que el software se diseña para dar cabida a estos cambios.
<b>Mantener la simplicidad</b>	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.

### 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*. Estos *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 y sincronizar su trabajo, para luego 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 lo que el equipo de desarrollo ha entregado y obtener una retroalimentación de los participantes en la reunión, para 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 cuando 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. La configuración de la metodología SCRUM utilizada, en conjunto con los artefactos elegidos de la metodología RUP, es la ilustrada en la Tabla 5.

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

Artefactos SCRUM
<b>Pila del producto:</b> lista dinámica de las cosas que se deben hacer, sin especificar cómo se deben hacer.
<b>Pila del <i>sprint</i>:</b> recopilación resumida de los ítems de la pila del producto, en donde se dividen los ítems en tareas pequeñas que no demanden una labor superior a una jornada de trabajo.
<b>Incremento:</b> el producto final de cada <i>sprint</i> . El mismo debe asemejarse a un software funcionando, permitiendo implementarse operativamente sin restricciones en un ambiente productivo.
Artefactos RUP
<b>Documento de visión:</b> define el alcance en alto nivel y propósito del producto.
<b>Glosario:</b> documento que define la terminología empleada en los artefactos.
<b>Documento de requerimientos no funcionales:</b> describe los requerimientos que tienen un impacto significativo en la arquitectura y en la satisfacción del usuario.
<b>Diagrama de casos de uso:</b> muestra los procesos del negocio que son proporcionados para los actores del negocio.

# Capítulo IV

## Resultados

En este capítulo se muestran los resultados que fueron alcanzados en el desarrollo del presente trabajo de investigación, siguiendo la configuración de la metodología SCRUM y los artefactos incluidos de la metodología RUP, la cual fue detallada en el capítulo anterior.

### 4.1. Pila del producto (*product backlog*)

**Tabla 6.** *Requerimientos funcionales del software* (Fuente: Elaboración propia).

Código	Requerimiento	Prioridad
<b>RF01</b>	Recuperar los 31 puntos espectrales de la medición con el MiniScan XE Plus y mostrarlos en su forma numérica.	Esencial
<b>RF03</b>	Representar los 31 puntos espectrales en forma de una curva de reflectancia difusa.	Esencial
<b>RF04</b>	Representar los 31 puntos espectrales en forma de una curva de absorbancia aparente.	Esencial
<b>RF05</b>	Calcular y mostrar las coordenadas de cromaticidad <i>CIE XYZ 1964</i> .	Esencial
<b>RF06</b>	Calcular y mostrar las coordenadas del espacio <i>CIE L*a*b* 1976</i> .	Esencial
<b>RF07</b>	Calcular y mostrar el coeficiente de absorción de la epidermis.	Esencial
<b>RF08</b>	Calcular y mostrar el coeficiente de esparcimiento de la epidermis.	Esencial
<b>RF09</b>	Calcular y mostrar el índice de eritema.	Esencial
<b>RF10</b>	Guardar los resultados en un archivo portátil, empleando el uso del formato de historia médica de los dermatólogos.	Esencial
<b>RF11</b>	Abrir y cargar los resultados guardados en el archivo portátil.	Esencial

## 4.2. Requerimientos no funcionales

**Tabla 7.** *Requerimientos no funcionales del software* (Fuente: Elaboración propia).

Código	Requerimiento	Prioridad
<b>RNF01</b>	El software debe ser capaz de ejecutarse en sistemas Windows actuales, con arquitecturas de 32 bits y 64 bits.	Esencial
<b>RNF02</b>	El software debe conectarse con el MiniScan XE Plus por medio de un adaptador USB.	Esencial
<b>RNF03</b>	El archivo portátil que maneja el software debe poder ser abierto por un visualizador/editor de hojas de cálculo.	Esencial
<b>RNF04</b>	El software debe desarrollarse utilizando el lenguaje de programación orientada a objetos C++.	Esencial



# Capítulo V

## Conclusiones

# Referencias

- BASKERVILLE, RICHARD L. (1999). «*Investigating Information Systems with Action Research*». Association for Information Systems, Atlanta, GA.
- BERSHA, K. S. (2010). «*Spectral Imaging And Analysis Of Human Skin*». University of Eastern Finland, Finland.
- CIE. «*Commission Internationale de l'Eclairage, International Commission on Illumination*». Vienna, 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.  
<http://www.hunterlab.com/about-us.html>
- (2001). «*Universal Software Versions 4.10 and Above User's Manual*». Hunter Associates Laboratory, Reston, Virginia.
- (2006). «*MiniScan XE Plus User's Guide Version 2.4*». Hunter Associates Laboratory, Reston, Virginia.
- KROLL, P. y KRUCHTEN, P. (2003). «*The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP*». Addison-Wesley, Boston, MA.

- 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.
- PÉREZ, A. D. (2012). «*Estudio de la Reflexión Óptica Difusa en Tejido Biológico*». Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Zacatenco, México, D.F..
- 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](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, New Jersey.
- 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.