

Universidad de las Ciencias Informáticas

Facultad 1



Título:

**Paquete de aplicaciones para la generación y análisis de
secuencias de medición del lector LF02**

**Trabajo de Diploma para optar por el título de Ingeniero en Ciencias
Informáticas**

Autores:

Carlos Manuel Ferrás Hernández

Yanet Leonor Quesada Hernández

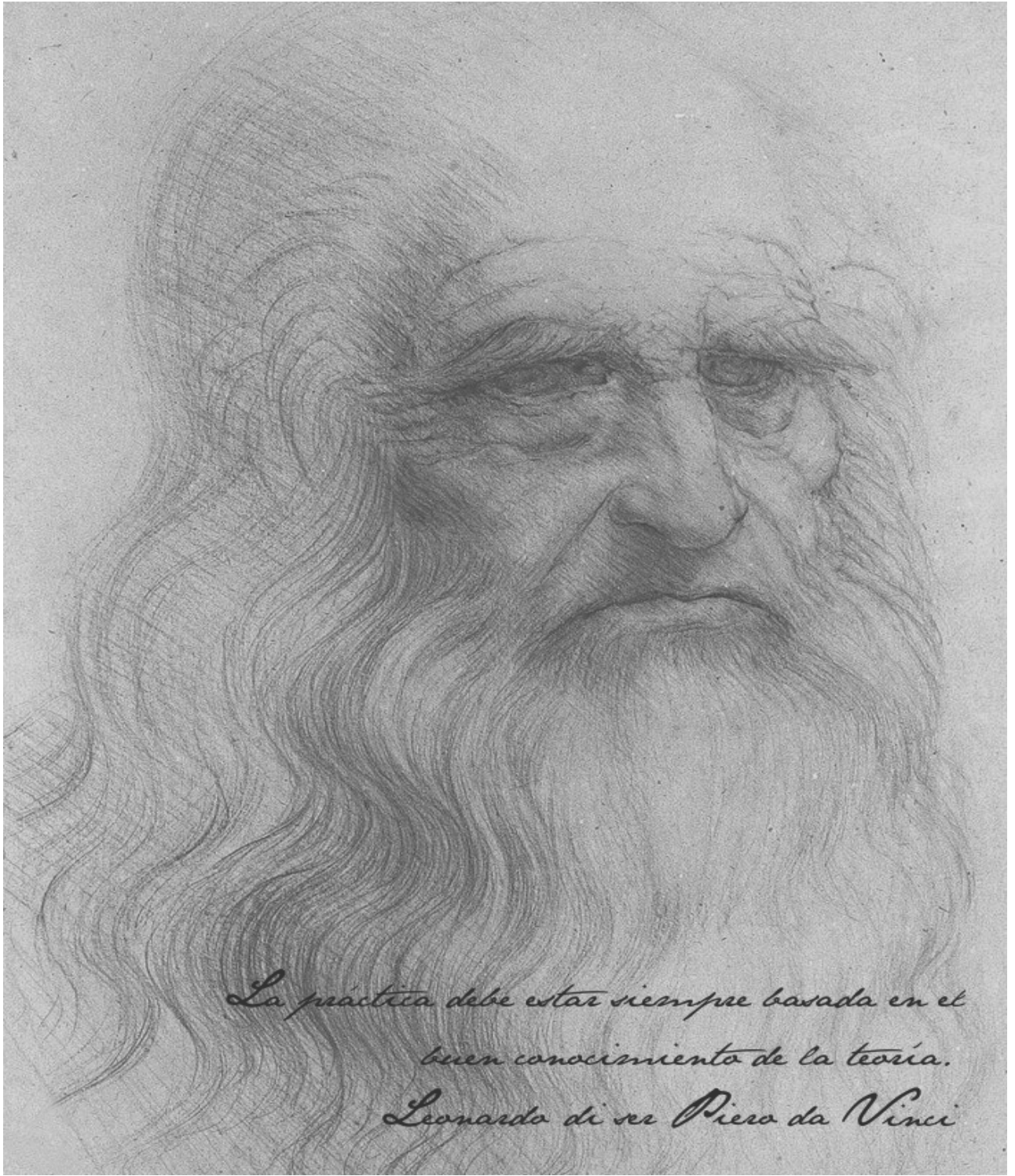
Tutores:

Msc. Luis Baly Gil

Ing. Nadia Porro Lugo

La Habana, Cuba, junio 2015

"Año 57 de la Revolución"



*La práctica debe estar siempre basada en el
buen conocimiento de la teoría.*

Leonardo di ser Piero da Vinci

Declaración de Autoría

Declaramos que somos los únicos autores de este trabajo y concedemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Carlos Manuel Ferrás
Hernández
Autor

Yanet Leonor Quesada
Hernández
Autor

Msc. Luis Baly Gil
Tutor

Ing. Nadia Porro Lugo
Tutor

Dedicatoria

De Carlos:

A mi mamá y mi papá por haber estado siempre, por darme la vida y guiarme por el mejor camino. Por hacer de mí quien soy hoy.

De Yanet:

A mi hermanita por ser lo mejor de mi vida y la fuerza que me impulsa a dar cada paso importante de mi vida. A mis padres por el esfuerzo dedicado para hacer de mi una buena persona, por el amor, la confianza depositada, por los valores inculcados y por ser los mejores padres del mundo. A mi novio por estar siempre ahí.

Agradecimientos

De Carlos:

A mi familia, a todos y cada uno de sus integrantes, a mi mamá, mi papá, mi hermano, mi abuela, mi tía, mis tíos, mis primos por darme su apoyo incondicional. A María por su paciencia, su dedicación y sobre todas las cosas, por confiar en mí. A mis tutores, especialmente a Baly por dedicarnos tanto de su tiempo, por su profesionalidad y calidad humana, por acogernos como parte de su equipo y confiarnos la realización de esta tarea con lo que representa para su proyecto. A la institución por prepararnos y poner a nuestra disposición los recursos necesarios durante estos cinco años de estudio. A mis amistades por soportarme, apoyarme y por hacer durante esos cinco años que la vida sea más amena. Y por último y no por eso menos importante, sino de manera especial a Yanet, por su amistad, su confianza, su sinceridad y por ser ella.

De Yanet:

A mis padres por estos años de sacrificio, dedicación y esfuerzo. A mi hermana por su confianza, por creer en mi y darme fuerzas para seguir cuando creía que no podía. A Dayron por su paciencia, su consideración, por seguir de cerca cada paso de este triunfo, por sus palabras de aliento, por llegar en el momento exacto y darle un nuevo rumbo a mi carrera, por su amor y su cariño. A mi familia por su apoyo durante estos cinco años, en especial a mis tíos Ismael, Tamara y Magalis. A mis abuelos Genoveva, Lourdes y Rubén. A Carlitos, por haberme escogido como compañera de tesis, por confiar en mí, por su paciencia y su amistad. A mis tutores, en especial a Baly por su apoyo, por creer en nosotros, por su ayuda, su atención y su compromiso. A los profesores que ayudaron en mi formación, en especial a Rosa Ortiz por confiar en mi. A mis amistades por estos años compartidos, por los momentos de alegrías y de lágrimas, en especial a David por su amistad incondicional, por dejar sus cosas a un lado para ayudarme con las mías, por su confianza y sus consejos. A Maria Leysi por ayudarnos con el documento cuando más lo necesitábamos. A la UCI por darnos esta oportunidad.

Resumen

El principal objetivo de este trabajo es la elaboración de un paquete de aplicaciones libres que permita que los procesos de generación y análisis de las secuencias de medición del lector de fotoluminiscencia automatizado LF02 se realicen en menos tiempo y con menos errores. Para ello se realizó un estudio de las herramientas para la generación y análisis de secuencias de medición para lectores de fotoluminiscencia existentes en la actualidad, donde ninguna de ellas satisface las necesidades actuales del Centro de Aplicaciones Tecnológicas y Desarrollo Nuclear, centro creador del lector LF02, pero permitió tomar algunas características como base para la solución desarrollada.

Para cumplir el objetivo trazado se desarrolló el paquete de aplicaciones Sequence–Toolkit, el cual contiene dos aplicaciones GenSec que permite crear las secuencias de medición para el lector LF02 en formato xml y GenRep para analizar los resultados generados, permitiendo crear un reporte en los formatos xml, txt tabulado y pdf para impresión, además de mostrar los resultados gráficamente.

A partir de las pruebas realizadas se demostró que el producto está listo para ser desplegado en un ambiente real ya que logró reducir el tiempo de generación de las secuencias de un día de trabajo a aproximadamente cinco minutos y el tiempo de análisis de los resultados de tres días a aproximadamente 10 minutos. Además disminuyeron los errores de estructura en la secuencia, en los datos de los procesos y en los cálculos de 33,3% a 3,3%, de 60% a 1,17% y de 73,3% a 3,3% respectivamente.

Palabras clave: análisis, datación, fotoluminiscencia, generación, secuencias

Índice de Contenido

Introducción.....	1
Capítulo 1: Fundamentación Teórica.....	6
1.1 Marco Conceptual.....	6
1.1.1 Datación.....	6
1.1.2 Datación absoluta.....	6
1.1.3 Datación por Fotoluminiscencia.....	7
1.2 Estudio de sistemas para los procesos de generación y análisis de secuencias de medición de fotoluminiscencia.....	8
1.3 Software dedicados a los procesos de generación y análisis de secuencias de medición de fotoluminiscencia.....	8
1.3.1 Gen5™.....	8
1.3.2 FirstLight.....	9
1.3.3 LDAS.....	10
1.3.4 Risø Sequence Editor.....	10
1.3.5 Valoración del estudio realizado.....	11
1.4 Metodología de desarrollo de software.....	13
1.4.1 SXP.....	13
1.5 Lenguajes utilizados.....	14
1.5.1 UML 2.1.....	14
1.5.2 Python 2.7.....	15
1.6 Marco de trabajo.....	16
1.6.1 Qt 4.....	16
1.7 Plataforma de desarrollo.....	16
1.7.1 Eclipse 3.7.2.....	17
1.8 Herramienta CASE.....	17
1.9 Herramienta para el Control de versiones.....	18
1.10 Tecnologías utilizadas.....	19
1.10.1 Formato XML.....	19

Índice

1.11 Conclusiones Parciales.....	20
Capítulo 2: Características del sistema.....	21
2.1 Propuesta de solución.....	21
2.1.1 Consideraciones de la solución propuesta.....	23
2.2 Conceptos asociados al dominio del sistema.....	23
2.3 Requisitos del sistema.....	25
2.4 Historias de Usuarios (HU).....	28
2.4.1 Historia de Usuario 1: Gestionar secuencias de medición.....	29
2.4.2 Historia de Usuario 2: Gestionar muestra.....	30
2.4.3 Historia de Usuario 3: Gestionar proceso.....	30
2.4.4 Historia de Usuario 4: Gestionar proceso TL.....	31
2.4.5 Historia de Usuario 5: Gestionar proceso OSL.....	32
2.4.6 Historia de Usuario 6: Gestionar proceso POSL.....	33
2.4.7 Historia de Usuario 7: Gestionar proceso LMOSL.....	34
2.4.8 Historia de Usuario 8: Gestionar proceso Pre Heating.....	35
2.4.9 Historia de Usuario 9: Gestionar proceso Illumination.....	36
2.4.10 Historia de Usuario 10: Gestionar proceso Irradiation.....	37
2.4.11 Historia de Usuario 11: Generar reporte.....	38
2.4.12 Historia de Usuario 12: Graficar los datos generados por el lector.....	39
2.5 Descripción de la Arquitectura de Software.....	40
2.6 Modelo de diseño.....	42
2.6.1 Patrones de diseño.....	42
2.6.2 Diagrama de Paquete.....	44
2.7 Conclusiones Parciales.....	46
Capítulo 3: Implementación y prueba.....	47
3.1 Plan de Release.....	47
3.2 Tareas de Ingeniería.....	48
3.3 Estándar de código.....	50
3.5 Descripción de los principales elementos del sistema.....	50

Índice

3.6 Pruebas.....	52
3.6.1 Casos de prueba de Aceptación.....	53
3.6.2 Resultados de las pruebas.....	57
3.7 Análisis de los resultados del sistema desarrollado.....	58
3.8 Aporte social de la solución.....	59
3.9 Conclusiones parciales.....	61
Conclusiones Generales.....	62
Recomendaciones.....	63
Referencias Bibliográficas.....	64
Glosario de términos.....	68

Introducción

En el transcurso del tiempo se han empleado diferentes métodos para estudiar el proceso de evolución del hombre y el universo. La determinación de edades de objetos de interés geológico y arqueológico surge desde la centuria de 1800. A pesar de los múltiples hallazgos arqueológicos hasta la década del '50 sólo podían hacerse dataciones relativas. Actualmente este método se sigue utilizando, pero desde la revolucionaria revelación del radio-carbono en 1948 se han descubierto nuevos procedimientos o métodos para medir el paso del tiempo, como es el caso de la datación absoluta (Salazar et al. 2006).

Estos métodos han cambiado la historia de la arqueología y ofrecen una importante ayuda para los distintos investigadores que se ven en la necesidad de fechar un determinado acontecimiento. La aplicación de una técnica u otra depende del material con que se esté trabajando, por ejemplo, la fotoluminiscencia es empleada para fechar material inorgánico de hasta 500.000 años de antigüedad.

Problemas como el cambio climático y estudios de estabilidad tectónica requieren de métodos absolutos para la determinación de edades, a partir de los cuales se pueden desenvolver un sin número de investigaciones y proyectos. Para tal fin se han creado laboratorios especializados en dichos métodos en diferentes países, y en ese sentido se han obtenido resultados relevantes en todas las esferas de la sociedad.

En Cuba se encuentra el Centro de Aplicaciones Tecnológicas y Desarrollo Nuclear (CEADEN), dedicado a actividades de ciencia, tecnología y medio ambiente. Este es un complejo científico técnico a ciclo completo que vincula la investigación y la innovación tecnológica, así como la producción y la prestación de servicios científico técnicos especializados. Este centro participa en proyectos de investigación, colaboración internacional, asistencia técnica y otros auspiciados por la Organización Internacional de Energía Atómica (OIEA), está adscrito al Ministerio de Ciencia, Tecnología y Medio Ambiente (CITMA) y a la Agencia de Energía Nuclear y Tecnologías de Avanzada de Cuba (AENTA) y pertenece al Polo Científico del Oeste de la Habana.

Introducción

Desde el 2001 el centro desarrolla un proyecto orientado a la evolución de facilidades experimentales en las técnicas de fotoluminiscencia y espectrometría de masa de ionización láser asistida por matriz (*MALDI*)¹ que permitan el desarrollo de estudios básicos y/o servicios integrados en campos como el fechado, la dosimetría retrospectiva, biotecnología, nanotecnología y ciencia de los materiales. En específico se prevé la creación de un laboratorio de fechado por fotoluminiscencia con capacidad de realizar estudios de fechado o de dosimetría retrospectiva en base a la fotoluminiscencia del cuarzo y la mejora de un espectrómetro de masa *MALDI*, incluyendo facilidades de introducción, posicionamiento y medición sobre un gran número de muestras, a las cuales se les aplicará una serie de procesos. Esta serie de procesos puede sufrir variaciones en dependencia del número de muestras o del tipo de material que se esté analizando y la misma está determinada por un gran número de parámetros de medición. Una vez terminadas las mediciones, estas deben ser analizadas, y de las mismas se extraen las magnitudes de interés. Para ello se cuenta con el Lector de Fotoluminiscencia automatizado LF02 creado en el propio centro.

Con el avance de las nuevas Tecnologías de la Información y las Comunicaciones (TIC's), el CEADEN se ha visto inmerso en la necesidad de automatizar sus procesos para ganar en eficiencia, rapidez y comunicación. Actualmente, se está trabajando en base a mejorar el trabajo en el Laboratorio de Fechado por Fotoluminiscencia, ya que el mismo presenta insuficiencias en los procesos de generación y análisis de las muestras. Estos procesos se realizan de forma manual, lo que resulta engorroso cuando el número de muestras es considerable, además propicia la entrada incorrecta de datos y errores en la estructura de las secuencias. Generar la secuencia en la que será medida una sola muestra puede tardar hasta un día entero, por lo que se requiere de mucho tiempo para realizar las mediciones a varias muestras.

¹ **MALDI**: La denominación "MALDI" deriva de las siglas de las palabras en inglés *Matrix Assisted Laser Desorption/Ionization* (desorción/ionización láser asistida por matriz). Es una metodología de ionización suave utilizada en espectrometría de masas que es capaz de identificar cualitativamente y cuantitativamente cualquier tipo de mezclas de sustancias. Asimismo, esta técnica también permite determinar la masa molecular de un compuesto determinado (Junior et al. 2006).

Introducción

Existen *software* dedicados a este tipo de actividad, pero son privativos y creados específicamente para el equipo donde será utilizado, por tanto no es posible usarlo en ningún otro sistema, ya que tienen precios de adquisición elevados y no se ajustan a las necesidades del centro. Además no se cuenta con el personal capacitado para el mantenimiento de las mismas. Se necesita emitir una secuencia en un formato bien estructurado, de fácil interpretación y flexible, que facilite el trabajo.

Por ello, el CEADEN inicia una colaboración con el Centro de *Software* Libre (CESOL) perteneciente a la Universidad de las Ciencias Informáticas (UCI). Dicho centro fue creado específicamente para hacer cumplir la política de migración hacia sistemas y aplicaciones libres que se lleva a cabo en todo el país desde el año 2004, la cual permite a los usuarios la libertad de usar, estudiar, compartir, copiar y modificar el *software*. El centro cuenta con el departamento de Servicios Integrales en Migración, Asesoría y Soporte (SIMAYS), el cual está especializado en brindar servicios relacionados con los procesos de migración, tales como: asesoría, consultoría, capacitación y soporte técnico. Además de desarrollar soluciones libres para apoyar el proceso de migración.

Por lo anteriormente expuesto, se decide realizar la investigación a partir del siguiente **problema a resolver**: ¿Cómo garantizar que los procesos de generación y análisis de las secuencias de medición del lector de fotoluminiscencia automatizado LF02 se realicen en menos tiempo y con menos errores?

Se define como **objeto de estudio**: los procesos de generación y análisis de secuencias de medición para lectores de fotoluminiscencia enmarcado en el **campo de acción**: los procesos de generación y análisis de las secuencias de medición del lector de fotoluminiscencia automatizado LF02.

A raíz del problema de investigación se define como **objetivo general** desarrollar un paquete de aplicaciones libres que permita que los procesos de generación y análisis de las secuencias de medición del lector de fotoluminiscencia automatizado LF02 se realicen en menos tiempo y con menos errores. Para dar cumplimiento al objetivo general se plantean los **objetivos específicos**:

- ✓ Estudiar las herramientas que se utilizan para la generación y análisis de secuencias de medición de lectores de fotoluminiscencia.
- ✓ Analizar y diseñar la propuesta de solución para la generación y el análisis de las secuencias de medición del lector de fotoluminiscencia automatizado LF02.

Introducción

- ✓ Implementar la propuesta de solución para la generación y el análisis de las secuencias de medición del lector de fotoluminiscencia automatizado LF02.
- ✓ Realizar pruebas al sistema.

A partir de lo antes mencionado se determina como **idea a defender** que: un paquete de aplicaciones libres que permita la generación y análisis de las secuencias de medición del lector de fotoluminiscencia automatizado LF02 garantizará que estos procesos se realicen en menos tiempo y con menos errores.

Para cumplir con los objetivos trazados se definen un conjunto de **tareas** a desarrollar tales como:

- ✓ Revisión bibliográfica sobre la temática en cuestión.
- ✓ Análisis de las metodologías de desarrollo de *software*.
- ✓ Elaboración de los artefactos y modelos asociados a la metodología de desarrollo de *software* escogida.
- ✓ Descripción de los requisitos funcionales y no funcionales para facilitar la implementación de la solución propuesta.
- ✓ Implementación de las funcionalidades del sistema propuesto.
- ✓ Diseño de las pruebas al sistema para verificar su correcto funcionamiento.

Para llevar a buen término el objetivo propuesto se emplean diferentes métodos o procedimientos de la investigación científica, en la búsqueda y procesamiento de la información.

Métodos teóricos:

Histórico–lógico: en la revisión de la literatura especializada para consultar la información necesaria en el proceso de investigación, para identificar las tendencias actuales de los procesos de generación y análisis de secuencias de medición para lectores de fotoluminiscencia en el mundo y como ha ido evolucionando, tomando los resultados alcanzados como base de comparación.

Analítico–sintético: para realizar un análisis de la literatura consultada y extraer los elementos más importantes relacionados con los procesos de generación y análisis de las secuencias de medición para lectores de fotoluminiscencia e identificar los conceptos que sustentan la base teórica de la investigación. Además en la descomposición del problema de la investigación en elementos por separados y profundizar en el estudio de cada uno de ellos.

Método empírico:

La entrevista: para obtener información sobre los principales elementos que se definen en los procesos de generación y análisis de las secuencias de medición del lector de fotoluminiscencia automatizado LF02. Constituyendo un medio para el conocimiento cualitativo del fenómeno en el lugar donde se desarrolla. En el Anexo² 5 se muestra la entrevista diseñada para obtener la información que demuestra la necesidad de la institución, basado en experiencias expuestas por el entrevistado, el especialista Msc. Luis Baly Gil.

El presente documento cuenta con tres capítulos:

Capítulo I: Fundamentación Teórica.

En este capítulo se exponen los elementos teóricos que sustentan la investigación. Se hace un análisis de trabajos realizados referentes a la generación y análisis de secuencias de medición para lectores de fotoluminiscencia y de las aplicaciones que se utilizan para el desarrollo de estos procesos, así como el aporte que tienen para la investigación. Además, se define en base al estudio realizado la metodología y las herramientas adecuadas para el desarrollo de la solución.

Capítulo II: Análisis y diseño del Sistema.

En este capítulo se describen las características del paquete de aplicaciones propuesto para solucionar el problema de investigación. Se definen los requerimientos funcionales y no funcionales de las aplicaciones que integran la solución. Se describen las funcionalidades a través de las historias de usuarios y se elabora el modelo de diseño de la aplicación.

Capítulo III: Implementación y Prueba.

En este capítulo se definen los aspectos relacionados con la implementación y ejecución de las pruebas. Se describen el plan de liberaciones y las tareas de ingenierías. Se expone el estándar de código a utilizar en la implementación del paquete de aplicaciones. Finalmente se presentan los casos de pruebas a realizar en base a la metodología de desarrollo de *software* seleccionada con el fin de obtener un producto de calidad.

²Anexos: Los anexos referenciados en el documento se encuentran disponibles en la versión digital del mismo.

Capítulo 1: Fundamentación Teórica

Capítulo 1: Fundamentación Teórica

Cuando se comienza un trabajo investigativo aparecen muchos elementos desconocidos que complican de cierta manera el curso del mismo. Por ello es necesario analizar y comprender la relación que tienen entre ellos y qué aporta cada uno. En el desarrollo de este capítulo se analizan conceptos relacionados con los sistemas de datación y sus técnicas principales que exponen los elementos teóricos que sustentan la investigación. Se realiza un estudio a nivel mundial y nacional acerca de la generación y análisis de secuencias de medición para lectores de fotoluminiscencia. Además se selecciona en base al estudio realizado la metodología de desarrollo de *software* a usar, lenguajes, herramientas y tecnologías adecuadas para la implementación de la solución a proponer.

1.1 Marco Conceptual

1.1.1 Datación

La datación es un método de fechado que se usa para expresar la edad en años o millones de años de rocas, minerales, fósiles, objetos o restos arqueológicos o geológicos (García, 2012).

Este método se aplica de forma diferente en dependencia del material que se esté estudiando. Por ello el mismo se compone de dos técnicas principales: la datación relativa y la datación absoluta.

1.1.2 Datación absoluta

La datación absoluta comprende otro grupo de técnicas que componen al método de datación. Actualmente este tipo de fechado es el más usado por las características que posee. Permite obtener una fecha dentro de una escala temporal y se refiere a un momento preciso, procede de otras ciencias como la física y la química (Medina et al. 2013). Está integrado por tres tipos fundamentales: la aplicación de sistemas físico-químicos, los isótopos radioactivos y las cronologías climáticas dentro de las cuales se encuentran diferentes técnicas tales como radioisótopos, radiocarbono, potasio-argón, la técnica de cronología relativa, el paleomagnetismo y la fotoluminiscencia, que en este caso es la técnica en que se basa la investigación. Las primeras mediciones del método se basaban en la estimación a partir del

Capítulo 1: Fundamentación Teórica

espesor de los estratos. Actualmente se usa para argumentaciones arqueológicas y comparación del registro arqueológico de contextos diferentes.

1.1.3 Datación por Fotoluminiscencia

La fotoluminiscencia es una técnica de datación muy utilizada en estudios arqueológicos gracias a su precisión en las mediciones. Se basa en la capacidad que tienen algunos minerales como el cuarzo y los feldespatos para emitir luz cuando son calentados. Esta emisión es lograda gracias a la imperfección de la estructura cristalina de estos minerales, que provoca a su vez que algunos electrones libres aumenten sus niveles de energía superior a su nivel fundamental. La cantidad de luz emitida en el momento del calentamiento depende del tiempo que dicho material haya estado recibiendo radiación ambiental. Esta técnica se ha aplicado con éxito a vidrios, ladrillos, escorias de fundición, sílex, rocas de hogares y lava; siendo también una técnica habitual en la autenticación de piezas de cerámicas pertenecientes a colecciones de museos en diferentes países (Duller, 2008).

El límite práctico de utilización es de unos 500.000 años. Esta técnica tiene la ventaja de que no necesita calibración y permite datar contextos exactos, además es una técnica sofisticada aunque generalmente es cara (García, 2012).

El proceso de medición por fotoluminiscencia está basado en aplicar a una o a varias muestras, determinados procedimientos como irradiación, iluminación y calentamiento, que eventualmente causarán efectos sobre las mismas emitiendo algún tipo de resultado, los cuales posteriormente serán medidos y analizados según corresponda. Tanto la generación de las muestras así como el análisis que se les realiza a las mismas forman parte de un proceso complicado y muy riguroso. Una vez que se conforme una secuencia para aplicar determinada medición es importante que no sufra cambios y evitar la entrada de errores que puedan cambiar totalmente el resultado final o perder las muestras escogidas, lo cual resultaría muy desconcertante para los investigadores, debido a la cantidad de recursos que se disponen para esta actividad.

Capítulo 1: Fundamentación Teórica

1.2 Estudio de sistemas para los procesos de generación y análisis de secuencias de medición de fotoluminiscencia

Con el avance de las tecnologías algunos países para ganar en tiempo, esfuerzo y recurso se han dado la tarea de crear lectores de fotoluminiscencia, siendo estos, equipos especializados para realizar las mediciones de forma automática. El liderazgo de esta área le corresponde a Dinamarca, que es el país con más avance en esta esfera gracias a su potente laboratorio Risø, encargado de producir los lectores más potentes y usados del mundo.

A pesar de que la técnica de datación por fotoluminiscencia fue descubierta desde la década del '50 no siempre es aplicada en las mejores condiciones o con la mejor tecnología. Muchos países utilizan equipos automatizados para realizar esta actividad, aunque no sean los más sofisticados, incluso llegan a resultar rústicos en algunos casos, debido a que el proceso no es meramente automático, pues trabaja con la labor manual del hombre. De esta manera se debe indicar en cada procedimiento al equipo. En cambio otros investigadores arrendan equipos en laboratorios de otros países para realizar las mediciones.

Actualmente existen lectores de fotoluminiscencia que tienen integrado sistemas informáticos capaces de reducir el tiempo y el esfuerzo humano. Estos sistemas están diseñados para generar la secuencia a medir y posteriormente analizarla de forma automática.

1.3 Software dedicados a los procesos de generación y análisis de secuencias de medición de fotoluminiscencia

1.3.1 Gen5™

Es un poderoso *software* para el análisis de datos de mediciones por fotoluminiscencia. Ofrece una combinación única de potencia y facilidad de uso, que a su vez permite aumentar la productividad y ahorrar tiempo. Permite lecturas de punto final, cinéticas, análisis espectral, análisis de área de pozo y captura de imágenes por pozos disponibles según los sistemas de detección incorporados, junto con control de temperatura, inyección de reactivos y operaciones de ampliado de placa. La salida de datos es totalmente configurable gracias a una exportación rápida a Excel, incluso puede realizar reportes con calidad para publicaciones (IEEE, 2012). Está disponible en ediciones para inglés, chino, japonés y ruso.

Capítulo 1: Fundamentación Teórica

Proporciona todas las herramientas de análisis necesarias, para obtener de manera sencilla y rápida sus datos contrastados, así como transformados y analizados en formato Excel (BioTek 2007).

Desventajas para esta investigación:

- ✓ Está creado para Windows XP, 2000 y Vista.
- ✓ Es privativo.

1.3.2 FirstLight

Es un *software* utilizado para mediciones de fotoluminiscencia que se ajusta a los estándares de interfaz de usuario de Windows, es potente, provee facilidad de uso y flexibilidad. Utiliza secuencias de comandos para la toma de datos además de poseer su propia definición de archivo de secuencia de comandos, cubriendo de esta manera formatos binarios y *ASCII*. Es un medio rápido y fácil de entrar en un programa de control que informa al sistema qué hacer con cada disco de la muestra. Describe la curva de brillo y brinda la posibilidad de poner los datos a partir de una carga de discos de la muestra en varios archivos. La edición es simple y se pueden guardar los archivos del programa por lotes para la posterior modificación y reutilización. Este programa por lotes se puede configurar para realizar todas las mediciones y los datos a continuación, puede ser transferida a un programa de hoja de cálculo para cualquier cálculo especial. Brinda la facilidad de no tener que observarlo ni tocarlo durante un día o dos y continuará todo el proceso de manera automática. Los datos erróneos de la curva de brillo pueden ser marcados y descartarse (Azad, 2010).

Desventajas que presenta este *software*:

- ✓ El formato en que genera la secuencia es incomprensible para los usuarios y por las aplicaciones que le darán seguimiento al análisis de las muestras, lo cual trae como consecuencia que haya que convertirlo a un formato más práctico.
- ✓ Carece de una estructura de datos que permite el almacenamiento de la información, ya sea un archivo de datos de la curva, resultados, imágenes, documentos, bibliografías, u otro elemento definido por el usuario.
- ✓ La ayuda que posee es en línea, lo cual se convierte en inaccesible para las instituciones que no tengan conexión a Internet.

Capítulo 1: Fundamentación Teórica

- ✓ Los parámetros tales como la excitación por Luminiscencia Ópticamente Estimulada (OSL por sus siglas en inglés *Optically Stimulated Luminiscens*) y base de tiempo, la velocidad de la rampa, la temperatura de punto final, el tiempo de precalentamiento y la temperatura, se pueden alterar lo cual abre paso a la introducción de datos erróneos que pueden afectar el resultado final de la medición.

1.3.3 LDAS

El Analizador de Datación por Fotoluminiscencia (o LDAS por sus siglas en inglés) es un paquete de *software* compatible para Windows 95, 98, 98SE, NT 4.0 y 2000, desarrollado para mostrar, editar, procesar y analizar datos de Fotoluminiscencia (o TL por sus siglas en inglés) y OSL. LDAS usa una interfaz de texto *ASCII* en la cual se pueden mostrar los datos en gráficas (lineal o de histograma) y exportarlos como imagen así como en documentos de textos y hojas de cálculo (Marešová et al. 2010).

Desventajas de este *software*:

- ✓ Es privativo.
- ✓ Diseñado para funcionar solo en sistemas Windows en sus versiones más antiguas, que son muy poco usadas en la actualidad.
- ✓ No posee una interfaz gráfica intuitiva.

1.3.4 Risø Sequence Editor

Es un *software* privativo creado para Windows para la generación y análisis de secuencia de medición por fotoluminiscencia. Se asemeja a una hoja de cálculo, con filas, columnas y celdas. El *software* permite un gran número de operaciones para una misma secuencia. La secuencia puede ser almacenada y reutilizada en futuras rutinas. Muestra en pantalla como es que se va ejecutando el proceso. Viene incluido con otros *software* creados específicamente para el equipo al que está integrado. Facilita la creación y almacenamiento de secuencias, abrir, ejecutar o interrumpir una secuencia (Lapp et al. 2009). Consta de dos *software*, el Analyst para el análisis de los datos generados y el Viewer para mostrar esta información

Desventajas del Risø Sequence Editor:

Capítulo 1: Fundamentación Teórica

- ✓ Está diseñado solo para trabajar en Windows.
- ✓ Es privativo.
- ✓ La secuencia que genera es muy difícil de comprender por usuarios y por otras aplicaciones.
- ✓ Almacena los datos en un formato que solo es accesible por el visualizador y el programa de análisis que vienen conjuntamente con él.

1.3.5 Valoración del estudio realizado

Características requeridas	Gen5™	FirstLight	LDAS	Risø Sequence Editor
Generación de secuencias de medición por fotoluminiscencia	X	X	X	X
Demostración gráfica resultados de mediciones		X	X	X
Herramienta de análisis	X	X	X	X
Interfaz gráfica fácil de usar	X	X		X
Software Libre				
Multiplataforma				
Estructura de la secuencia en formato bien estructurado y fácil de entender				
Realización de <i>merging</i> ³				
Generación de reporte				

Tabla 1: Comparativa entre las aplicaciones para la generación y análisis de secuencias de fotoluminiscencia

³**Merging**: nombra a una funcionalidad del lector LF02 que permite fusionar varios procedimientos correspondientes a una muestra determinada sin tener que retirarla para aplicarle otro proceso.

Capítulo 1: Fundamentación Teórica

Una vez culminado el estudio de los sistemas usados para llevar a cabo los procesos de generación y análisis de secuencias de medición para lectores de fotoluminiscencia a nivel mundial se puede constatar que aunque permiten generar secuencias de muestras, que es lo fundamental, no todos lo hacen de la forma más completa, pues a pesar de proveer un aumento en la productividad y un ahorro de tiempo, no logran abarcar todas las funcionalidades requeridas para dicho proceso en el centro.

Algunos *software* como el Risø Sequence Editor están preparados para trabajar conjuntamente con otros *software* capaces de acceder a los datos generados, pero no todos tienen esta característica. Algunos desarrollan formas obsoletas o muy atrasadas para llevar a cabo los procesos de análisis. Además son privativos, lo cual los convierte en un problema financiero para muchas instituciones, trayendo consigo gastos en la adquisición de los mismos y en el pago de licencias.

Para el caso del CEADEN que trabaja con un equipo creado en el mismo centro, ninguno de los *software* analizados es factible pues están diseñados para trabajar en el equipo para el que fueron creados o para trabajar en un solo sistema, generalmente Windows. Esta característica es un impedimento tanto para este centro como para cualquier institución que trabaje con el sistema operativo Linux, ya que por lo general estas aplicaciones no suelen ser multiplataforma. La forma de generar la secuencia se basa en una estructura con formatos específicos, poco comprensible por los usuarios y por otras aplicaciones. A pesar de que la mayoría están creados para Windows, no todos poseen una interfaz gráfica fácil de usar como es el caso del LDAS, y en el caso de FirstLight que carece de una estructura de datos que permita el almacenamiento de la información, por ejemplo un archivo de datos de la curva, resultados, imágenes, documentos, o cualquier elemento definido por el usuario, además, ninguno de ellos permite hacer *merging*, funcionalidad muy importante requerida por el centro.

El estudio se realizó con el objetivo de obtener una propuesta para el lector de fotoluminiscencia automatizado (LF02) desarrollado por el CEADEN, sin embargo ninguna de las aplicaciones mencionadas fue tomada como solución debido a las desventajas que presentan para la investigación. Además carecen de funcionalidades requeridas por el centro, capaz de responder ante los procedimientos incluidos en el LF02, y que a su vez facilite las mediciones y brinde mayores prestaciones.

Capítulo 1: Fundamentación Teórica

La solución propuesta está basada en desarrollar una nueva herramienta, en lugar de seleccionar alguna de las estudiadas, tomando como bases algunas de las características de las aplicaciones estudiadas e incorporándole otras funcionalidades.

1.4 Metodología de desarrollo de *software*

Cuando se quiere desarrollar un *software*, lo primero que se hace es pensar en las características del mismo, en el tiempo que se dispone y en el personal que se tiene. A partir de estos elementos se define qué metodología utilizar. El desarrollo de un *software* es una tarea complicada, de ahí se debe la existencia de tantas propuestas metodológicas que actúan de diferentes maneras durante el proceso de desarrollo en general.

Para algunos casos se cuenta con las metodologías tradicionales que se utilizan cuando el proyecto es muy grande y con un gran número de personal. En este caso se exige una documentación muy exhaustiva que involucra diferentes actividades y artefactos que son imprescindibles. Por otro lado están las metodologías ágiles, que son requeridas cuando el período de tiempo que se estima para la realización del proyecto es corto y el equipo de desarrollo es pequeño (Canós, Letelier y Panadés, 2003).

Aunque siempre es recomendable una buena documentación independientemente del tipo de metodología utilizada, es más fácil realizar cambios cuando se trata de una metodología ágil, gracias a la flexibilidad que proveen estas. Por tal razón son muy utilizadas hoy en día, demostrando su efectividad.

1.4.1 SXP

Es un híbrido de Scrum y XP. La base de esta metodología se centra en tomar lo mejor de cada una, lo cual favorece el desarrollo del *software*. Esta metodología ayuda a fortalecer el trabajo en equipo, incluyendo además al usuario final como parte del equipo de desarrollo, lo cual ayuda a enfocarse en su base principal que es la programación extrema y lograr el éxito del proyecto como requisito fundamental. Esta característica favorece el mejoramiento de su producción aumentando el nivel de interés del equipo. Ofrece una estrategia tecnológica, a partir de la introducción de procedimientos ágiles que permitan actualizar los procesos de *software* para el mejoramiento de la actividad productiva fomentando el desarrollo de la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del

Capítulo 1: Fundamentación Teórica

equipo, ayudando al líder del proyecto a tener un mejor control del mismo. Como método de estimación se utiliza la opinión de expertos y constan con métricas o indicadores para lograr una eficiente calidad. Consta de 4 fases principales: Planificación–Definición, Desarrollo, Entrega y Mantenimiento. De cada una de ellas se desencadenan 7 flujos de trabajo: concepción inicial, captura de requisitos, diseño con metáforas, complementación, prueba, entrega de la documentación, soporte e investigación, este flujo puede moverse y utilizarse en cualquier parte del ciclo de vida del proyecto. Estos flujos incluyen actividades tales como el levantamiento de requisitos, la priorización de la Lista de Reserva del Producto, definición de las Historias de Usuario, diseño, implementación, planificación de las iteraciones y las actividades que se van a realizar para lograr el producto, pruebas, y las tareas necesarias para realizar las investigaciones para documentar todo el proceso (Peñalver, Meneses y García, 2010).

SXP es ideal para proyectos que cuentan con un pequeño equipo de trabajo, cuando los requisitos son imprecisos y cambiantes, donde se tiene un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad. Promueve el trabajo del equipo en forma conjunta y en una misma dirección con un objetivo claro, y permite además seguir de forma clara el avance del equipo de desarrollo por parte del cliente, de forma que los jefes pueden ver día a día cómo progresa el trabajo.

1.5 Lenguajes utilizados

1.5.1 UML 2.1

Como lenguaje de modelado se utiliza el Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, *Unified Modeling Language*). Es considerado un lenguaje estándar para el diseño de *software* y otros sistemas de cómputo. Actualmente es el más conocido y utilizado. Es un potente lenguaje que ayuda a visualizar, especificar, construir y documentar el producto que se está llevando a cabo (Jacobson et al. 2000).

UML favorece la creación de los diferentes modelos que ofrecen las vistas necesarias para la construcción de un *software* de calidad y permite la comprensión del sistema que se quiere realizar tanto por parte de los usuarios finales, como de los desarrolladores que implementarán la solución.

Ventajas de usar UML:

Capítulo 1: Fundamentación Teórica

- ✓ El uso de lenguajes visuales permite la asimilación y el entendimiento por parte del equipo de desarrollo.
- ✓ Minimiza el tiempo que se emplea en el desarrollo de la arquitectura.
- ✓ Ayuda a la detección y resolución de errores.
- ✓ Favorece a la organización de la documentación del proyecto (Sierra, 2005).

1.5.2 Python 2.7

Es un lenguaje de programación potente, muy fácil de entender y de aprender. Posee un enfoque simple pero efectivo a la programación orientada a objetos gracias a su estructura eficiente de datos y de su alto nivel. Es un lenguaje de tipado dinámico, con una sintaxis muy limpia y elegante, y que favorece un código legible. Su naturaleza interpretada provee que se ejecute gracias a un programa intermedio llamado intérprete, en lugar de compilar el código a lenguaje de máquina que pueda entender y ejecutar directamente una computadora, lo cual permite el desarrollo rápido de aplicaciones sobre la mayoría de las plataformas (Van Rossum, 2013).

¿Por qué usar Python como lenguaje de programación?

Además de ser el lenguaje establecido por el cliente, brinda grandes facilidades:

- ✓ La sencillez y simplicidad de su sintaxis favorece mucho al programador.
- ✓ El tipado dinámico, el gestor de memoria, la gran cantidad de librerías disponibles y la potencia del lenguaje, hacen que Python sea sencillo y muy rápido.
- ✓ Es orientado a objetos, pero soporta también los estilos de programación procedural y funcional.
- ✓ Corre en múltiples plataformas, incluyendo Windows, Mac OS y Linux.
- ✓ Es adecuado tanto para programar scripts como aplicaciones de gran tamaño.
- ✓ Permite separar un programa en módulos y luego usarse en otros programas.
- ✓ Cuenta con administración automática de memoria a través de recolección de basura.
- ✓ Incluye una poderosa y extensa biblioteca de clases.
- ✓ Cuenta con una gran comunidad que se dedica a promover su desarrollo y adopción.

Capítulo 1: Fundamentación Teórica

- ✓ Por su naturaleza interactiva, resulta ideal para llevar a cabo programación experimental y desarrollo rápido.
- ✓ Es un lenguaje interpretado lo cual ahorra tiempo durante el desarrollo (Campos y Vargas, 2003).

1.6 Marco de trabajo

1.6.1 Qt 4

Es un marco de trabajo de código abierto con licencia GPL elegido por el cliente para crear las interfaces. Contiene herramientas de desarrollo tales como **QtDesigner** para diseñar y crear interfaces gráficas de usuario usando los componentes de Qt, genera un archivo XML cuyo contenido se puede convertir con los programas pertinentes a múltiples lenguaje de programación y **QtLinguist** que permite traducir la aplicación a otros idiomas. Es una tecnología que proporciona un conjunto de elementos gráficos para la creación de interfaces y aplicaciones en diferentes plataformas. Actualmente cuenta con gran éxito y una gran implantación en diferentes ámbitos, que van desde las aplicaciones de escritorio hasta los sistemas electrónicos industriales. En la última década Qt ha pasado de ser un producto usado por unos pocos desarrolladores especializados, a un producto usado por miles desarrolladores de código abierto en todo el mundo (Gutiérrez, 2009).

1.7 Plataforma de desarrollo

Es el entorno en el cual se desenvuelve la programación, se encuentra relacionada directamente a un sistema operativo; sin embargo, también es posible encontrarla ligada a una familia de lenguajes de programación o a una Interfaz de programación de aplicaciones. Entre las plataformas libres más utilizadas se encuentran NetBeans y Eclipse. En el caso de NetBeans permite desarrollar aplicaciones a partir de un conjunto de componentes de *software* llamados módulos. Debido a que los mismos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de *software*. Tiene una interfaz clara e intuitiva. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación. Cuenta con un aceptable

Capítulo 1: Fundamentación Teórica

completamiento de código. Es el principal competidor en el campo de código abierto frente a Eclipse (Trejos y Zamora, 2012).

1.7.1 Eclipse 3.7.2

Teniendo en cuenta que ambos IDEs son potentes, se toma a Eclipse 3.7.2 como plataforma de desarrollo pues con este se tienen mayores posibilidades de aprender, es recomendable para programadores novatos, como en este caso, que no se tiene mucha experiencia en proyectos reales, además ha sido el más usado y conocido por los desarrolladores de dicha solución. Es un entorno de desarrollo integrado (IDE), de Código Abierto y Multiplataforma sobre el que se pueden montar herramientas de desarrollo para cualquier lenguaje, pues el mismo no está orientado específicamente hacia ninguno en concreto. Para ello depende de la existencia de un *plugin* que le de soporte, en el caso de python, el plugin PyDev el cual permite al programador una interacción transparente entre el ambiente de desarrollo y el ambiente de depuración (Retamar, 2004).

La arquitectura de Eclipse permite, además de integrar diversos lenguajes sobre un mismo IDE, introducir otras aplicaciones accesorias que pueden resultar útiles durante el proceso de desarrollo como herramientas UML, editores visuales de interfaces y ayuda en línea para librerías.

1.8 Herramienta CASE

Visual Paradigm 8.0 es la herramienta CASE⁴ para modelado UML elegida por la UCI para el desarrollo de productos de gran alcance. Soporta el ciclo de vida completo del desarrollo de *software*, análisis, diseño, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Además, la herramienta es colaborativa, es decir, soporta múltiples usuarios trabajando sobre el mismo proyecto y permite control de versiones. También genera informes para una posterior documentación, exporta diagramas a (JPG, PNG, SVG, EMF y PDF) y posee una alta capacidad de integración con el lenguaje UML que será utilizado durante el modelado de la solución propuesta (Larramendi, 2011).

⁴CASE: Ingeniería de *Software* Asistida por Computadoras.

Capítulo 1: Fundamentación Teórica

Las herramientas CASE permiten reducir costo, recurso y dinero durante el proceso de desarrollo de *software* ya que permiten agilizar y mejorar el curso del mismo y su resultado final. Son utilizadas por los desarrolladores de la mejor forma posible, con el fin de obtener un producto final de alta calidad. Constituyen una potente arma en cuanto a la captura y especificación de requisitos. Además facilita el mantenimiento de sistemas, actualización de documentos y la aplicación de técnicas de una metodología.

1.9 Herramienta para el Control de versiones

Se elige **Git 1.9.7.5** con el fin de lograr una mayor interacción entre el equipo de desarrollo y el cliente, debido a que es muy conocida y utilizada por ambos. Permite revisar las actualizaciones efectuadas sobre la solución e incluye herramientas específicas para navegar y visualizar un historial de desarrollo no lineal. Presenta gestión distribuida, lo cual le provee a cada programador una copia local del historial del desarrollo entero, y los cambios se propagan entre los repositorios locales. Es muy eficiente dada la rapidez de gestión de diferencias entre archivos, entre otras mejoras de optimización en la velocidad de ejecución. Comprueba todo, lo cual impide que un archivo se corrompa o se pierda sin que se sepa (Nardi, 2012).

En la mayoría de los proyectos de la UCI, se utiliza Subversion como herramienta de control de versiones por las potencialidades que presenta, sin embargo es considerado por muchos como el abuelo de los sistemas de control de versiones; es centralizado, lento y más pesado que sus sucesores. Git por el contrario es uno de los jóvenes más potente de la generación de relevo; distribuido, rápido y ligero, acorde a las exigencias actuales.

Por su parte GitHub v 3 es una plataforma para alojar los proyectos, que utiliza Git como herramienta de control de versiones. Esta plataforma de desarrollo colaborativo de *software* además de alojar el repositorio de código, brinda herramientas que favorecen el trabajo en equipo dentro de un proyecto. Está enfocada al crecimiento de proyectos comunitarios y libres (Dabbish et al. 2012).

Capítulo 1: Fundamentación Teórica

1.10 Tecnologías utilizadas

1.10.1 Formato XML

El formato XML es el elegido por el cliente para la estructura de la secuencia de medición del lector de fotoluminiscencia LF02. Es muy recomendable cuando se requiere la creación de documentos de texto que contienen datos con un formato bien estructurado. Además de los datos puede incluirse un conjunto detallado de reglas que definen la estructura de dichos datos.

Los desarrolladores que utilicen XML pueden apoyarse en las características que ofrece el mismo para separar los datos de la forma en que se presentan. Un documento de este tipo puede leerse por medio de cualquier aplicación personalizada utilizando diferentes objetos de análisis gramatical, lo cual hace que XML sea muy flexible y que los datos del documento sean utilizados para diferentes fines. Por ejemplo, los datos con formato de un documento XML podrían analizarse y posteriormente presentarlos a un usuario, almacenarlos en una base de datos o ser utilizados por otra aplicación (Sturm, 2000).

La utilización de XML es muy recomendable ya que permiten ser leídos por personas que tengan un conocimiento muy básico del lenguaje, son razonablemente claros, pueden ser abiertos con cualquier editor de texto y determinar el contenido de dicho documento.

Ventajas de usar XML:

- ✓ Es un formato bien estructurado, lo que facilita validar fácilmente el contenido y la sintaxis.
- ✓ Los documentos XML se pueden construir como componente de otros documentos gracias a sus métodos de vinculación.
- ✓ XML puede ser usado como contenedor de datos, ya que puede representar casi cualquier tipo de dato, lo cual facilita su lectura por personas, analizadores sintácticos computarizados y aplicaciones.
- ✓ Proporciona flexibilidad lo cual permite definir la estructura exacta del documento y su validación fácilmente.
- ✓ Es sencillo de utilizar y reconocido internacionalmente (Sturm, 2000).

Capítulo 1: Fundamentación Teórica

1.11 Conclusiones Parciales

En el presente capítulo fueron estudiados diferentes sistemas encargados de la generación y análisis de secuencias de medición para lectores de fotoluminiscencia, donde se concluye que los mismos no satisfacen las necesidades actuales del desarrollo de estos procesos en el CEADEN. Por tanto se decide desarrollar un paquete de aplicaciones, capaz de suplantar estas necesidades, manteniendo algunas características de los sistemas analizados. La metodología de desarrollo de *software* que se decidió utilizar fue SXP. Como herramienta CASE se decidió utilizar Visual Paradigm de manera que facilite el entendimiento de los diagramas. Se define utilizar UML como lenguaje de modelado, ya que permite especificar, documentar y detallar los artefactos en el sistema, Python como lenguaje de programación y Eclipse como plataforma de desarrollo. Como marco de trabajo y librería gráfica se decidió utilizar Qt teniendo en cuenta que presenta un entorno gráfico ligero.

Capítulo 2: Características del sistema

Capítulo 2: Características del sistema

A partir de la metodología, los lenguajes y las herramientas definidas para el desarrollo de la propuesta de solución, se describen en este capítulo las características y principales funcionalidades de la misma. Se especifican los requisitos funcionales y no funcionales presentes en el sistema, además de la descripción de las historias de usuarios. Se expondrán los elementos fundamentales del diseño y de la arquitectura. Para comprender mejor el sistema se muestran los diagramas generados durante el flujo de trabajo.

2.1 Propuesta de solución

Debido a las insuficiencias en los procesos de generación y análisis de las secuencias de medición del lector LF02 en el CEADEN, provocadas por la forma en que actualmente se realizan dichos procesos se hace necesario realizar una propuesta de solución al problema planteado. Teniendo en cuenta que los procesos se realizan de forma manual, los mismos están propensos a entradas incorrectas de datos, errores en la estructura de las secuencias y que el tiempo necesario para generar la secuencia es considerable. Además el lector de fotoluminiscencia fue creado en el propio centro con funcionalidades que responden a necesidades surgidas en el desarrollo de los procesos mencionados.

Por lo anteriormente expuesto el laboratorio requiere de un *software* con funcionalidades específicas que responda a las necesidades del mismo y que se ajuste al lector LF02. Existen *software* dedicados a este tipo de actividad pero son *software* privativos creados específicamente para el equipo donde será utilizado y no se ajustan a las necesidades del centro, la presente investigación propone una solución a partir de un paquete de aplicaciones, tomando algunas características de las herramientas estudiadas e incorporándole otras funcionalidades.

- ✓ De Gen5™ fue tomada la idea de transferir los datos a un programa de hoja de cálculo para cualquier cálculo especial.
- ✓ De FirstLight fue tomada la idea de no tener que observarlo ni tocarlo durante todo el proceso, de manera que se ejecuta automáticamente.

Capítulo 2: Características del sistema

- ✓ De LDAS se tomó la posibilidad de mostrar los datos en gráficas y exportarlos como imagen así como en documentos de textos.
- ✓ Por su parte Sequence Editor fue el espejo gráfico, pues se pretende crear la interfaz gráfica semejante a la de dicha aplicación, teniendo en cuenta lo deseado por el cliente. Presentando semejanza a una hoja de cálculo, con filas, columnas y celdas. Capaz de realizar una o varias operaciones individualmente o en combinación y que muestre en pantalla cómo se va ejecutando el proceso. Además que la secuencia pueda ser almacenada y reutilizada en futuras rutinas.

El paquete de aplicaciones estará integrado de dos aplicaciones, una dedicada a la generación de las secuencias de medición del lector de fotoluminiscencia automatizado LF02 y la otra al análisis de dichas mediciones. La secuencia emitida por la primera aplicación estará estructurada en formato XML, lo que posibilitará que la interpretación de los datos por cualquier sistema y por las personas involucradas en esta actividad sea muy sencilla, gracias a las características de dicho formato. Esta secuencia será enviada al lector para la realización de las mediciones establecidas a partir de los procedimientos seleccionados en el *software*. Posteriormente la otra aplicación realizará el análisis y permitirá la visualización de los resultados obtenidos de la medición realizada por el lector mediante gráficas y finalmente generará un reporte en formato PDF, XML o texto tabulado para una posterior impresión si el usuario lo desea.

Esta solución tiene la ventaja de estar desplegada bajo la Licencia Pública General de GNU o (GNU GPL por sus siglas en inglés) en su versión 3, lo cual garantiza a los usuarios finales la libertad de usar, estudiar, compartir, copiar y modificar el *software* («GNU GENERAL PUBLIC LICENSE», 2007), convirtiéndola en *software* libre y código abierto, resolviendo el primer inconveniente relacionado con la adquisición del producto. Además implementa las funcionalidades acorde a las necesidades del centro con una rápida ejecución, de esta manera generará un ahorro de tiempo y recurso.

El hecho de usar Python como lenguaje de programación al unísono del marco de trabajo Qt, hace que la solución sea multiplataforma, lo que significa que puede ser desplegada en cualquier tipo de sistema operativo. Ambas aplicaciones serán utilizadas de manera rutinaria por el Laboratorio de Fechado por Fotoluminiscencia del CEADEN y servirá como paso de avance en estudio de presentes y futuros

Capítulo 2: Características del sistema

descubrimientos, además de nuevos proyectos vinculados a la aplicación de la ciencia, la técnica y la tecnología en el país.

2.1.1 Consideraciones de la solución propuesta

Un paquete de aplicaciones es una colección de aplicaciones, construidas en un mismo fichero para lograr un manejo de las mismas de manera más fácil («Ubuntu Community», 2012).

Permitiendo de esta manera que sean instaladas con el mismo instalador, generando más organización y que no haya que abrir cada una por separado cuando se necesite de ellas. Es realmente una ventaja para los usuarios ya que tributan al ahorro de tiempo y a facilidades de uso.

2.2 Conceptos asociados al dominio del sistema

Un modelo de dominio se utiliza como fuente de inspiración para el diseño de los objetos del *software*. Muestra a los modeladores clases conceptuales significativas en el dominio del problema. Se representa como un diagrama de clases o un conjunto de estos. Se utiliza para comprender los conceptos relacionados con el desarrollo y puesta en práctica de la aplicación (Larman, 2003).

Con el mismo se persigue plasmar todo lo relacionado con los procesos de generación y análisis de secuencias de medición del lector LF02, desarrollando de esta forma una amplia visión del sistema y del contexto en que será desplegado posteriormente.

Puede mostrar:

- ✓ Conceptos y objetos del dominio del problema: clases conceptuales.
- ✓ Asociaciones entre las clases conceptuales.
- ✓ Atributos de las clases conceptuales (Larman, 2003).

La decisión de cuanto detalle va a ser necesario o no incluir en cada modelo está completamente abierta a consideración del modelador.

A continuación se muestra el modelo de dominio del paquete de aplicaciones para la generación y análisis de secuencias de medición del lector de fotoluminiscencia LF02 (Sequence–ToolKit).

Capítulo 2: Características del sistema

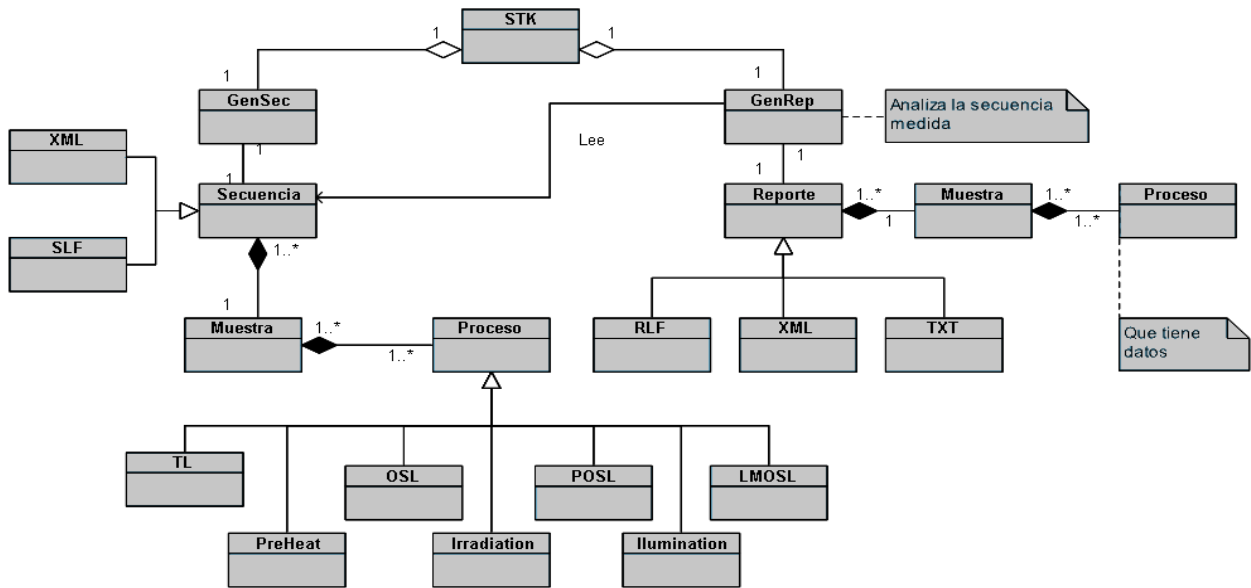


Figura 1: Modelo de Dominio del STK

Descripción de cada uno de los elementos que conforman el diagrama representado:

STK: representa el paquete que contiene las aplicaciones GenSec y GenRep.

GenSec: representa la aplicación encargada de generar la secuencia de medición.

Secuencia: representa la secuencia generada por GenSec, contiene una o varias muestras.

Muestra: conforman la secuencia que será medida por el lector en dependencia de los procesos que se definan para ser aplicados.

Proceso: pueden ser de tipo Fotoluminiscencia (TL por sus siglas en inglés), Luminiscencia Ópticamente Estimulada (OSL por sus siglas en inglés), Luminiscencia Ópticamente Estimulada Pulsada (POSL por sus siglas en inglés), Modulado Lineal de la Luminiscencia Ópticamente Estimulada (LMOSL por sus siglas en inglés), *Pre Heat*, *Irradiation*, *Ilumination* y se aplican a las muestras.

SLF: es la secuencia una vez modificada por las mediciones realizadas por el lector y es leída posteriormente por el GenRep.

GenRep: representa la aplicación encargada de analizar la secuencia medida por el lector y como parte del análisis genera un reporte que puede ser de tipo RLF, TXT o PDF y contiene un conjunto de muestras.

Capítulo 2: Características del sistema

Muestra: representa al conjunto de muestras que forman el reporte, aunque no son todas las muestras medidas, sino solamente las que tienen procesos que contengan datos.

Proceso: están asociados a las muestras, y representan solo los procesos que tienen datos.

2.3 Requisitos del sistema

Los requisitos del sistema son un conjunto de tareas, metas u objetivos que se persiguen con el desarrollo del *software*. Facilitan la comprensión de lo que desea el cliente, así como la forma en que los usuarios finales lograrán desempeñarse según el negocio.

Los requisitos funcionales son condiciones o capacidades que el sistema debe cumplir. Son usados para especificar los principales servicios que brinda el sistema. Los requisitos no funcionales responden a propiedades o cualidades que el *software* debe tener para que se pueda comportar de manera atractiva, confiable y segura. Están estrechamente vinculados con los requisitos funcionales y son esenciales en el éxito del producto final (Chaves, 2006).

Como resultado de dicha actividad se obtiene según la metodología, la Lista de Reserva del Producto (LRP). Esta lista priorizada define el trabajo que se va a realizar en el proyecto, logrando un entendimiento entre el cliente y el desarrollador del sistema. Garantiza la organización de los requisitos funcionales y no funcionales en dependencia de la prioridad que tengan para el negocio y el desarrollo del sistema. Esta lista puede crecer y modificarse a medida que se obtiene más conocimiento acerca del producto y del cliente. El objetivo es asegurar que el producto definido al terminar la lista es el más correcto, útil y competitivo posible y para esto la lista debe acompañar los cambios en el entorno y el producto (Peñalver, Meneses y García, 2010).

A continuación se muestra la relación de los requisitos funcionales y no funcionales asociados al paquete de aplicaciones, así como la prioridad que tienen, según la importancia que le atribuyó el cliente y su complejidad. Se realizó la captura de 41 requisitos funcionales y 10 no funcionales para un total de 51 requisitos del sistema en general.

Asignado a	Ítem *	Descripción	Estimación	Estimado por
------------	--------	-------------	------------	--------------

Capítulo 2: Características del sistema

Prioridad		Alta		
Carlos M. Ferrás	1	Crear secuencia de medición	0.5 semana	Yanet L. Quesada
Carlos M. Ferrás	2	Insertar muestra(s) a medir	0.4 semana	Yanet L. Quesada
Carlos M. Ferrás	3	Eliminar muestras	0.3 semana	Yanet L. Quesada
Carlos M. Ferrás	4	Definir proceso TL	0.4 semana	Yanet L. Quesada
Carlos M. Ferrás	5	Definir proceso OSL	0.4 semana	Yanet L. Quesada
Carlos M. Ferrás	6	Definir proceso POSL	0.4 semana	Yanet L. Quesada
Carlos M. Ferrás	7	Definir proceso LMOSL	0.4 semana	Yanet L. Quesada
Carlos M. Ferrás	8	Definir proceso Pre Heat	0.3 semana	Yanet L. Quesada
Carlos M. Ferrás	9	Definir proceso Iluminación	0.3 semana	Yanet L. Quesada
Carlos M. Ferrás	10	Definir proceso Irradiación	0.3 semana	Yanet L. Quesada
Carlos M. Ferrás	11	Editar proceso	0.4 semana	Yanet L. Quesada
Carlos M. Ferrás	12	Unir procesos	0.4 semana	Yanet L. Quesada
Carlos M. Ferrás	13	Separar procesos que hayan sido unidos	0.4 semana	Yanet L. Quesada
Carlos M. Ferrás	14	Mostrar estado del proceso	0.4 semana	Yanet L. Quesada
Carlos M. Ferrás	15	Eliminar proceso	0.5 semana	Yanet L. Quesada
Carlos M. Ferrás	16	Establecer protocolo	0.1 semana	Yanet L. Quesada
Carlos M. Ferrás	17	Ingresar propietario de la secuencia	0.1 semana	Yanet L. Quesada
Carlos M. Ferrás	18	Ingresar nombre de la secuencia	0.1 semana	Yanet L. Quesada
Carlos M. Ferrás	19	Guardar id del lector	0.1 semana	Yanet L. Quesada
Carlos M. Ferrás	20	Establecer uso de nitrógeno	0.1 semana	Yanet L. Quesada
Carlos M. Ferrás	21	Guardar tasa de dosis	0.1 semana	Yanet L. Quesada
Yanet L. Quesada	22	Guardar tasa de dosis externa	0.1 semana	Carlos M. Ferrás
Yanet L. Quesada	23	Ordenar la secuencia por muestras	0.2 semana	Carlos M. Ferrás
Yanet L. Quesada	24	Mostrar vista previa de la secuencia	0.2 semana	Carlos M. Ferrás

Capítulo 2: Características del sistema

Yanet L. Quesada	25	Guardar secuencia	0.2 semana	Carlos M. Ferrás
Yanet L. Quesada	26	Imprimir secuencia	0.1 semana	Carlos M. Ferrás
Yanet L. Quesada	27	Editar Secuencia	0.1 semana	Carlos M. Ferrás
Yanet L. Quesada	28	Detener la secuencia	0.1 semana	Carlos M. Ferrás
Prioridad		Media		
Yanet L. Quesada	29	Generar reporte	0.4 semana	Carlos M. Ferrás
Yanet L. Quesada	30	Definir parámetros para el reporte	0.4 semana	Carlos M. Ferrás
Yanet L. Quesada	31	Definir dependencia de procesos	0.4 semana	Carlos M. Ferrás
Yanet L. Quesada	32	Asociar procesos por un criterio determinado para el reporte	0.2 semana	Carlos M. Ferrás
Yanet L. Quesada	33	Desagrupar procesos unidos para realizar reporte	0.2 semana	Carlos M. Ferrás
Yanet L. Quesada	34	Guardar reporte	0.2 semana	Carlos M. Ferrás
Yanet L. Quesada	35	Imprimir reporte	0.3 semana	Carlos M. Ferrás
Prioridad		Baja		
Yanet L. Quesada	36	Graficar los datos generados por el lector	0.3 semana	Carlos M. Ferrás
Yanet L. Quesada	37	Definir la(s) curva(s) a graficar para un proceso determinado	0.3 semana	Carlos M. Ferrás
Yanet L. Quesada	38	Elegir escala para graficar	0.2 semana	Carlos M. Ferrás
Yanet L. Quesada	39	Permitir graficar respecto al tiempo o la temperatura para el proceso TL	0.2 semana	Carlos M. Ferrás
Yanet L. Quesada	40	Definir zona de señal	0.2 semana	Carlos M. Ferrás
Yanet L. Quesada	41	Definir zona de fondo	0.2 semana	Carlos M. Ferrás
RNF (Requisitos No Funcionales)				
Software				
1	Tener instalada la máquina virtual de Python 2.7<versión<3.0			
2	Tener instalada las librerías de 4			

Capítulo 2: Características del sistema

3	Tener instalada la librería Python-Dateutil
4	Tener instalada la librería Python-Matplotlib
Restricciones del diseño	
5	Lenguaje de programación Python
6	Usar Qt como marco de trabajo para la creación de interfaces
Usabilidad	
7	El sistema debe poseer una interfaz cómoda y ajustable que posibilite a los usuarios sin experiencia una rápida adaptación
8	El sistema debe ser multiplataforma
9	El sistema debe ser multilenguaje
Interfaz	
10	Presentar una Interfaz sencilla e intuitiva al usuario

Tabla 2: Lista de Reserva del Producto

2.4 Historias de Usuarios (HU)

El proceso de construcción de *software*, en una metodología ágil, inicia con el levantamiento de las historias de usuario, las cuales constituyen especie de unas tarjetas que presentan un fragmento de alguna funcionalidad del sistema esperado (Gutiérrez et al. 2005). Las HU constituyen la técnica utilizada en SXP para especificar los requisitos del *software*, lo cual equivaldría a los casos de uso en el proceso unificado. Luego de este proceso, los programadores realizan la estimación del esfuerzo requerido en la implementación de las historias de usuario. La estimación se inicia en la primera reunión de planeación y continúa a medida que se va construyendo el prototipo. Al igual que en la estimación del esfuerzo, el proceso de desarrollo depende de lo que entienden los programadores sobre las historias de usuario (Citón, 2006), de ahí la importancia de redactarlas en lenguaje natural y de proveer los detalles suficientes para estimar razonablemente el tiempo que llevará implementarlas. Es importante que se detallen con el cliente, pues constituyen una base para las pruebas funcionales del sistema (Peñalver, Meneses y García, 2010).

A continuación se exponen las Historias de Usuario para el sistema.

Capítulo 2: Características del sistema

2.4.1 Historia de Usuario 1: Gestionar secuencias de medición

Historia de Usuario	
Número: HU_1	Nombre Historia de Usuario: Gestionar secuencia de medición
Modificación de Historia de Usuario Número: ninguna	
Usuario: todos	Iteración Asignada: 1
Prioridad en Negocio: Muy Alta	Puntos Estimados: 3 semanas
Riesgo en Desarrollo: Muy Alta	Puntos Reales: 3 semanas
Descripción: Permite generar una secuencia de medición. Para generar una secuencia de medición nueva, en la aplicación GenSec , se selecciona dentro de archivo la opción nuevo . Luego en opciones , se ingresan las opciones de secuencias que quedarán registradas en la secuencia: Propietario: guarda el propietario de la secuencia. Nombre: guarda el nombre de la secuencia. Protocolo: guarda el protocolo a seguir para la medición. Uso de nitrógeno: registra si se usará o no nitrógeno. Tasa de dosis: guarda automáticamente la tasa de dosis que le será aplicada. Tasa de dosis externa: guarda automáticamente la tasa de dosis externa en caso de que se aplique alguna. Posteriormente se insertan las muestras y se definen los procesos que les serán aplicados. En caso de que se quiera organizar la estructura de la secuencia, se elige en la opción Editar, Ordenar por muestras . Para modificar una secuencia, se elige abrir en la opción archivo , y se abrirá la muestra guardada previamente a la cual se le podrán realizar los cambios deseados o en el botón resetear , el cual pone todos los estados en pendiente nuevamente y borra todos los datos adicionales permitiendo volver a usar esa secuencia pero con nuevos datos. Para guardar una secuencia basta con elegir guardar o guardar como en la opción archivo . El botón vista previa muestra una vista de cómo quedará la secuencia una vez generada. Para detener la secuencia se selecciona la muestra o el grupo de estas al cual se le aplicará Pause , en la columna comando se da doble clic y aparecerá un menú desplegable donde se elige la opción Pause . Aparecerá un cuadro de diálogo donde se introducirá el tiempo que se quiera pausar la secuencia.	

Capítulo 2: Características del sistema

<p>Observaciones: El formato para guardar una secuencia es slf o xml.</p>
<p>Prototipo de interfaz: Ver figura 8 y 9 del Anexo 3</p>

Tabla 3: HU Gestionar secuencias de medición

2.4.2 Historia de Usuario 2: Gestionar muestra

Historia de Usuario	
Número: HU_2	Nombre Historia de Usuario: Gestionar muestra
Modificación de Historia de Usuario Número: ninguna	
Usuario: todos	Iteración Asignada: 1
Prioridad en Negocio: Muy Alta	Puntos Estimados: 1 semana
Riesgo en Desarrollo: Muy Alta	Puntos Reales: 1 semana
<p>Descripción: Permite insertar o eliminar una o varias muestras. Para insertar una muestra se hace doble clic en la casilla debajo de la columna muestra y aparece un cuadro de diálogo para introducir el número de la muestra y luego aceptar.</p>	
<p>Observaciones: En caso de que se quiera introducir un grupo de muestras se puede introducir en el cuadro de diálogo de la siguiente manera: Separando los números por coma, o fijando con un guión el intervalo de muestras. Para eliminar una muestra se selecciona la misma y se presiona suprimir. El número máximo de muestras a insertar es 24.</p>	
<p>Prototipo de interfaz: Ver figura 10 del Anexo 3</p>	

Tabla 4: HU Gestionar muestra

2.4.3 Historia de Usuario 3: Gestionar proceso

Historia de Usuario

Capítulo 2: Características del sistema

Número: HU_3	Nombre Historia de Usuario: Gestionar proceso
Modificación de Historia de Usuario Número: ninguna	
Usuario: todos	Iteración Asignada: 1
Prioridad en Negocio: Muy Alta	Puntos Estimados: 3 semanas
Riesgo en Desarrollo: Muy Alta	Puntos Reales: 3 semanas
<p>Descripción:</p> <p>Permite insertar, eliminar o modificar un proceso. Para definir un proceso se hace doble clic en la columna comando, aparece un menú desplegable donde se elige el proceso a aplicar, rellenando en cada caso los parámetros de edición correspondientes.</p> <p>El sistema permite mostrar el estado de los procesos en (pendiente, en ejecución y terminado). Esta funcionalidad se va ejecutando automáticamente a medida que el proceso se vaya aplicando en el lector. Inicialmente el estado del proceso será pendiente hasta que comience a ejecutarse. El estado se muestra con un punto de color determinado según se encuentre, cuando está pendiente se muestra amarillo, cuando está en ejecución rojo y en verde para cuando haya terminado.</p> <p>Al seleccionar dos o más procesos, se activa el botón unir, el cual permite que el lector pueda aplicar merging. Cuando haya unión de procesos, al seleccionar alguno de ellos, se activará el botón separar, el cual permite separar dos procesos que hayan sido unidos previamente.</p> <p>Un proceso será eliminado luego de ser seleccionado y presionando suprimir.</p> <p>Para modificar el proceso basta con abrirlo y realizar los cambios en los parámetros.</p>	
<p>Observaciones:</p> <p>Para definir un proceso hay que definir previamente la(s) muestra(s) a medir.</p>	
Prototipo de interfaz: Ver figura 11 del Anexo 3	

Tabla 5: HU Gestionar proceso

2.4.4 Historia de Usuario 4: Gestionar proceso TL

Historia de Usuario	
Número: HU_4	Gestionar proceso TL

Capítulo 2: Características del sistema

Modificación de Historia de Usuario Número: ninguna	
Usuario: todos	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 0.4 semanas
Riesgo en Desarrollo: Alta	Puntos Reales: 0.4 semanas
<p>Descripción:</p> <p>Permite definir los parámetros de medición de TL. En la aplicación GenSec luego de seleccionar la muestra o el grupo de estas al cual se le aplicará Termoluminiscencia, en la columna comando se hace doble clic y aparece un menú desplegable donde se elige la opción TL. Aparecerá un cuadro de diálogo donde se introducen los parámetros de medición requeridos. Estos parámetros son:</p> <p>Final Temperature: este parámetro define la temperatura de medición.</p> <p>Time at final Temp: define el tiempo de calentamiento a la máxima temperatura.</p> <p>Channel: este parámetro se refiere al número de canales en los cuales será dividida la medición.</p> <p>Heating Rate: este parámetro se refiere a la velocidad de calentamiento de la muestra.</p> <p>Para modificar el proceso basta con abrir y realizar los cambios en los parámetros.</p> <p>El botón Información permite introducir el tipo de dato y cualquier comentario acerca del proceso.</p>	
Observaciones:	
Prototipo de interfaz: Ver figura 12 del Anexo 3	

Tabla 6: HU Gestionar proceso TL

2.4.5 Historia de Usuario 5: Gestionar proceso OSL

Historia de Usuario	
Número: HU_5	Gestionar proceso OSL
Modificación de Historia de Usuario Número: ninguna	
Usuario: todos	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 0.4 semanas
Riesgo en Desarrollo: Alta	Puntos Reales: 0.4 semanas

Capítulo 2: Características del sistema

<p>Descripción:</p> <p>Permite definir los parámetros de medición de OSL. En la aplicación GenSec luego de seleccionar la muestra o el grupo de estas al cual se le aplicará OSL, en la columna comando se hace doble clic y aparece un menú desplegable donde se elige la opción OSL. Aparece un cuadro de diálogo donde se introducen los parámetros de medición requeridos. Estos parámetros son:</p> <p>Light source: permite definir la fuente de luz a aplicar.</p> <p>Start Optical Power: permite definir la potencia de iluminación.</p> <p>Channel: este parámetro se refiere al número de canales en los cuales será dividida la medición.</p> <p>Time per Channel: calcula el tiempo asignado a cada canal.</p> <p>Time: define el tiempo de medición.</p> <p>Before Simulation: define el número de canales antes de la estimulación óptica.</p> <p>During Simulation: define el número de canales durante la estimulación óptica.</p> <p>After Simulation: define el número de canales después de la estimulación óptica.</p> <p>Final Temperature: este parámetro define la temperatura de medición.</p> <p>Heating Rate: este parámetro se refiere a la velocidad de calentamiento de la muestra.</p> <p>Stabilization: define el tiempo de estabilización de la temperatura final.</p> <p>Para modificar el proceso basta con abrir y realizar los cambios en los parámetros.</p> <p>El botón Información permite introducir el tipo de dato y cualquier comentario acerca del proceso.</p>
<p>Observaciones:</p>
<p>Prototipo de interfaz: Ver figura 13 del Anexo 3</p>

Tabla 7: HU Gestionar proceso OSL

2.4.6 Historia de Usuario 6: Gestionar proceso POSL

Historia de Usuario	
Número: HU_6	Gestionar proceso POSL
Modificación de Historia de Usuario Número: ninguna	
Usuario: todos	Iteración Asignada: 2

Capítulo 2: Características del sistema

Prioridad en Negocio: Alta	Puntos Estimados: 0.4 semanas
Riesgo en Desarrollo: Alta	Puntos Reales: 0.4 semanas
<p>Descripción:</p> <p>Permite definir los parámetros de medición de POSL. En la aplicación GenSec luego de seleccionar la muestra o el grupo de estas al cual se le aplicará POSL, en la columna comando se hace doble clic y aparece un menú desplegable donde se elige la opción POSL. Aparece un cuadro de diálogo donde se introducirán los parámetros de medición requeridos. Estos parámetros son:</p> <p>Light source: permite definir la fuente de luz a aplicar.</p> <p>Optical Power: permite definir la potencia de iluminación.</p> <p>Channel: este parámetro se refiere al número de canales en los cuales será dividida la medición.</p> <p>Time per Chan: calcula el tiempo asignado a cada canal.</p> <p>Number of Scan: define el número de pulsos que se le dará a la muestra.</p> <p>Time: define el tiempo de medición.</p> <p>Before Simulation: define el número de canales antes de la estimulación óptica.</p> <p>During Simulation: define el número de canales durante la estimulación óptica.</p> <p>After Simulation: define el número de canales después de la estimulación óptica.</p> <p>Final Temperature: este parámetro define la temperatura de medición.</p> <p>Heating Rate: este parámetro se refiere a la velocidad de calentamiento de la muestra.</p> <p>Stabilization: define el tiempo de estabilización de la temperatura final.</p> <p>Para modificar el proceso basta con abrir y realizar los cambios en los parámetros.</p> <p>El botón Información permite introducir el tipo de dato y cualquier comentario acerca del proceso.</p>	
Observaciones:	
Prototipo de interfaz: Ver figura 14 del Anexo 3	

Tabla 8: HU Gestionar proceso POSL

2.4.7 Historia de Usuario 7: Gestionar proceso LMOSL

Historia de Usuario	
Número: HU_7	Gestionar proceso LMOSL

Capítulo 2: Características del sistema

Modificación de Historia de Usuario Número: ninguna	
Usuario: todos	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 0.4 semanas
Riesgo en Desarrollo: Alta	Puntos Reales: 0.4 semanas
<p>Descripción:</p> <p>Permite definir los parámetros de medición de LMOSL. En la aplicación GenSec luego de seleccionar la muestra o el grupo de estas al cual se le aplicará LMOSL, en la columna comando se hace doble clic y aparecerá un menú desplegable donde se elige la opción LMOSL. Aparece un cuadro de diálogo donde se introducen los parámetros de medición requeridos. Estos parámetros son:</p> <p>Light source: permite definir la fuente de luz a aplicar.</p> <p>Channel: este parámetro se refiere al número de canales en los cuales será dividida la medición.</p> <p>Time: define el tiempo de medición.</p> <p>Time per Chan: calcula el tiempo asignado a cada canal.</p> <p>Start Optical Power: define la potencia inicial.</p> <p>End Optical Power: define la potencia final.</p> <p>Final Temperature: este parámetro define la temperatura de medición.</p> <p>Heating Rate: este parámetro se refiere a la velocidad de calentamiento de la muestra.</p> <p>Stabilization: define el tiempo de estabilización de la temperatura final.</p> <p>Para modificar el proceso basta con abrir y realizar los cambios en los parámetros.</p> <p>El botón Información permite introducir el tipo de dato y cualquier comentario acerca del proceso.</p>	
Observaciones:	
Prototipo de interfaz: Ver figura 15 del Anexo 3	

Tabla 9: HU Gestionar proceso LMOSL

2.4.8 Historia de Usuario 8: Gestionar proceso Pre Heating

Historia de Usuario	
Número: HU_8	Gestionar proceso Pre Heating
Modificación de Historia de Usuario Número: ninguna	

Capítulo 2: Características del sistema

Usuario: todos	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 0.3 semanas
Riesgo en Desarrollo: Alta	Puntos Reales: 0.3 semanas
<p>Descripción:</p> <p>Permite definir los parámetros de medición de Pre Heat. En la aplicación GenSec luego de seleccionar la muestra o el grupo de estas al cual se le aplicará Pre calentamiento, en la columna comando se hace doble clic y aparece un menú desplegable donde se elige la opción Pre Heat. Aparece un cuadro de diálogo donde se introducen los parámetros de medición requeridos. Estos parámetros son:</p> <p>Final Temperature: este parámetro define la temperatura de medición.</p> <p>Time at final Temp: define el tiempo de calentamiento a la máxima temperatura.</p> <p>Heating Rate: este parámetro se refiere a la velocidad de calentamiento de la muestra.</p> <p>Para modificar el proceso basta con abrir y realizar los cambios en los parámetros.</p> <p>El botón Información permite introducir el tipo de dato y cualquier comentario acerca del proceso.</p>	
Observaciones:	
Prototipo de interfaz: Ver figura 16 del Anexo 3	

Tabla 10: HU Gestionar proceso Pre Heating

2.4.9 Historia de Usuario 9: Gestionar proceso Illumination

Historia de Usuario	
Número: HU_9	Gestionar proceso de Illumination
Modificación de Historia de Usuario Número: ninguna	
Usuario: todos	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 0.3 semanas
Riesgo en Desarrollo: Alta	Puntos Reales: 0.3 semanas

Capítulo 2: Características del sistema

<p>Descripción:</p> <p>Permite definir los parámetros de medición de Iluminación. En la aplicación GenSec luego de seleccionar la muestra o el grupo de estas al cual se le aplicará Iluminación, en la columna comando se hace doble clic y aparece un menú desplegable donde se elige la opción Iluminación. Aparece un cuadro de diálogo donde se introducen los parámetros de medición requeridos. Estos parámetros son:</p> <p>Final Temperature: este parámetro define la temperatura de medición.</p> <p>Stabilization: define el tiempo de estabilización de la temperatura final.</p> <p>Optical Power: permite definir la potencia de iluminación.</p> <p>Heating Rate: este parámetro se refiere a la velocidad de calentamiento de la muestra.</p> <p>Light Source: define la fuente de luz.</p> <p>Time: define el tiempo de medición.</p> <p>Para modificar el proceso basta con abrir y realizar los cambios en los parámetros.</p> <p>El botón Información permite introducir el tipo de dato y cualquier comentario acerca del proceso.</p>
<p>Observaciones:</p>
<p>Prototipo de interfaz: Ver figura 17 del Anexo 3</p>

Tabla 11: HU Gestionar proceso Iluminación

2.4.10 Historia de Usuario 10: Gestionar proceso Irradiation

Historia de Usuario	
Número: HU_10	Gestionar proceso de Irradiation
Modificación de Historia de Usuario Número: ninguna	
Usuario: todos	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 0.3 semanas
Riesgo en Desarrollo: Alta	Puntos Reales: 0.3 semanas
<p>Descripción:</p> <p>Permite definir los parámetros de medición de Irradiation. En la aplicación GenSec luego de seleccionar la muestra o el grupo de estas al cual se le aplicará Irradiación, en la columna comando se hace doble clic y</p>	

Capítulo 2: Características del sistema

<p>aparece un menú desplegable donde se elige la opción Irradiation. Aparece un cuadro de diálogo donde se introducen los parámetros de medición requeridos. Estos parámetros son:</p> <p>Time: define el tiempo de medición.</p> <p>Dose: calcula la dosis en dependencia de la tasa de dosis de la fuente radiactiva.</p> <p>Irradiation Source: define el tipo de fuente de irradiación que será aplicada. En caso de que sea Externa, solo hay que definir los parámetros anteriores incluyendo este, pero si es Beta se activan otros parámetros como:</p> <p>Final Temperature: este parámetro define la temperatura de medición.</p> <p>Heating Rate: este parámetro se refiere a la velocidad de calentamiento de la muestra.</p> <p>Stabilization: define el tiempo de estabilización de la temperatura final.</p> <p>Para modificar el proceso basta con abrir y realizar los cambios en los parámetros.</p> <p>El botón Información permite introducir el tipo de dato y cualquier comentario acerca del proceso.</p>
Observaciones:
Prototipo de interfaz: Ver figura 18 del Anexo 3

Tabla 12: HU Gestionar proceso Irradiation

2.4.11 Historia de Usuario 11: Generar reporte

Historia de Usuario	
Número: HU_11	Generar reporte
Modificación de Historia de Usuario Número: ninguna	
Usuario: todos	Iteración Asignada: 3
Prioridad en Negocio: Media	Puntos Estimados: 3 semanas
Riesgo en Desarrollo: Media	Puntos Reales: 3 semanas

Capítulo 2: Características del sistema

<p>Descripción:</p> <p>Permite crear y editar un reporte acerca de las mediciones realizadas por el lector de fotoluminiscencia automatizado LF02. En la aplicación GenRep se abren los datos obtenidos de las mediciones realizadas por el lector a partir de una secuencia generada por la aplicación GenSec. Se selecciona la opción perfil de la barra de herramientas y se definen los parámetros adicionales que serán incluidos en el reporte.</p> <p>Luego se realiza la asociación de procesos lo cual se puede realizar directamente mediante la selección de los procesos con el cursor y haciendo clic en el ícono agrupar; o mediante la utilización del comando asociación por criterio (Ctrl+K).</p> <p>Para separar procesos seleccionados con el cursor se selecciona el ícono desagrupar; para romper todas las asociaciones se utiliza el comando desagrupar todo (Ctrl+U).</p> <p>Para guardar un reporte se selecciona la opción guardar. Los reportes pueden ser guardados en dos formatos diferentes, el primero en un fichero tipo punto rlf que es de tipo XML, o en un fichero tipo texto con tabulación que permita ser abierto por Excel u otro editor de texto o datos.</p> <p>Adicionalmente el reporte puede imprimirse en formato PDF.</p>
<p>Observaciones:</p> <p>Debe estar creada la secuencia por el GenSec y medida por el lector LF02 para poder trabajar con el GenRep.</p>
<p>Prototipo de interfaz: Ver figura 19 del Anexo 3</p>

Tabla 13: HU Generar reporte

2.4.12 Historia de Usuario 12: Graficar los datos generados por el lector

Historia de Usuario	
Número: HU_12	Graficar los datos generados por el lector
Modificación de Historia de Usuario Número: ninguna	
Usuario: todos	Iteración Asignada: 4
Prioridad en Negocio: Baja	Puntos Estimados: 2 semanas
Riesgo en Desarrollo: Baja	Puntos Reales: 2 semanas
<p>Descripción:</p> <p>Permite definir los parámetros requeridos para graficar los datos que generados por el lector luego de medir una</p>	

Capítulo 2: Características del sistema

secuencia determinada. En la aplicación **GenRep** se elige la opción **configuración (Ctrl+T)** y en esta se seleccionan las curvas a graficar. En caso de que se requiera graficar el proceso **TL**, en el mismo cuadro de diálogo, se selecciona la pestaña referente a TL, y se define si se va a graficar respecto a la temperatura o respecto al tiempo, además las ventanas formato del eje horizontal y vertical permiten definir la escala a graficar, ya sea lineal o logarítmica.

Luego se define si se va a calcular la señal, y el fondo, además se definen los canales iniciales y finales utilizados por defecto para calcular la señal y el fondo.

Observaciones:

Prototipo de interfaz: Ver figura 20, 21 y 22 del Anexo 3

Tabla 14: HU Graficar los datos generados por el lector

2.5 Descripción de la Arquitectura de Software

La arquitectura de un *software* es la representación de la estructura de los componentes del *software*, sus propiedades e interacciones. Permite la comunicación entre todas las partes participantes en el desarrollo de un sistema de cómputo. Destaca las actividades relacionadas con el diseño y que a su vez tendrán un gran impacto durante todo el trabajo de ingeniería, y sobre todo en el resultado final. Modela cómo se estructura el sistema y la relación existente en sus componentes (Valencia y Ferro, 2011).

Todo *software* creado para sistemas de cómputo muestra uno o varios estilos arquitectónicos. Cada estilo describe una categoría de sistema que abarca un conjunto de componentes encargados de realizar alguna función requerida por el sistema, y muestra además cómo se realiza la integración o comunicación entre dichos componentes, permitiendo al diseñador las propiedades generales del sistema (Pressman, 2006).

Los estilos arquitectónicos se encuentran en el centro de la arquitectura y constituyen buena parte de su sustancia. Un estilo arquitectónico es una transformación impuesta al diseño de todo un sistema, estableciendo la estructura de sus componentes. El estilo al igual que un patrón arquitectónico establece una transformación, sin embargo difieren en diferentes elementos. El patrón se centra en un aspecto específico, en lugar de toda la arquitectura, dígame comportamiento o funcionalidad a nivel de

Capítulo 2: Características del sistema

infraestructura. Los patrones se usan junto a los estilos para especificar aspectos generales del sistema (Pressman, 2006).

El estilo arquitectónico de llamada y retorno permite obtener una estructura de programa fácil de modificar y cambiar de tamaño. Es uno de los estilos más generalizados en sistemas a gran escala. Miembros de la familia son las arquitecturas de programa principal y subrutina, los sistemas basados en llamadas a procedimientos remotos, los sistemas orientados a objeto y los sistemas jerárquicos en capas. Como parte de este estilo se encuentra el **Modelo–Vista–Controlador**, que soporta la arquitectura del sistema a desarrollar. Este patrón separa el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario en tres clases diferentes:

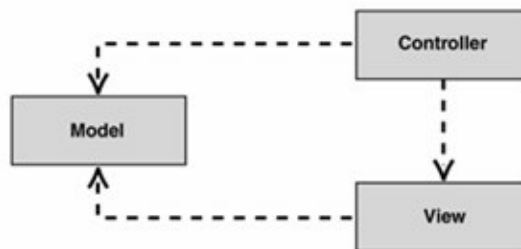


Figura 2: Representación del patrón arquitectónico Modelo–Vista–Controlador

Modelo: Administra el comportamiento y los datos del dominio de aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista) y responde a instrucciones de cambiar el estado (habitualmente desde el controlador).

Vista: Maneja la visualización de la información. Controla que información entra y sale del sistema.

Controlador: Interpreta las acciones del ratón y el teclado, informando al modelo y/o a la vista para que cambien según resulte apropiado.

Tanto la vista como el controlador dependen del modelo, el cual no depende de las otras clases. Esta separación permite construir y probar el modelo independientemente de la representación visual.

Entre las ventajas del estilo se encuentran las siguientes:

Capítulo 2: Características del sistema

- ✓ Soporte de vistas múltiples. Dado que la vista se halla separada del modelo y no hay dependencia directa del modelo con respecto a la vista, la interfaz de usuario puede mostrar múltiples vistas de los mismos datos simultáneamente.
- ✓ Adaptación al cambio. Los requerimientos de interfaz de usuario tienden a cambiar con mayor rapidez que las reglas de negocio. Los usuarios pueden preferir distintas opciones de representación, o requerir soporte para nuevos dispositivos. Dado que el modelo no depende de las vistas, agregar nuevas opciones de presentación generalmente no afecta al modelo (Reynoso y Kicillof, 2004).

2.6 Modelo de diseño

El modelo de diseño es un modelo de objetos que detalla las estructuras de datos, las arquitecturas y los componentes que son necesarios para implementar el sistema. Brinda control y organización sobre las actividades que se originan dentro del ciclo de vida del *software*. Se enfoca particularmente en el efecto que provocan en el sistema los requerimientos funcionales y no funcionales para describir la ejecución física de las HU, atendiendo además a otras restricciones que se puedan desatar. Se considera una base para la posterior actividad en el ciclo de vida del desarrollo de *software* de SXP (la implementación) (Sommerville, 2005).

2.6.1 Patrones de diseño

Los patrones de diseño se utilizan para describir la solución de un problema particular dentro de un contexto determinado mediante una estructura de diseño. Según Christopher Alexander "Cada patrón describe un problema que ocurre una y otra vez en nuestro entorno, y después describe la esencia de la solución a dicho problema, de tal forma que se pueda usar esta solución un millón de veces más, sin nunca hacerlo dos veces de la misma forma" (Pressman, 2006).

Patrones GRASP

GRASP es un acrónimo que significa *General Responsibility Assignment Software Patterns* (patrones generales de *software* para asignar responsabilidades). Este nombre surgió para indicar la trascendencia

Capítulo 2: Características del sistema

de **captar** estos principios, si se quiere diseñar eficazmente un *software*. Estos patrones son los patrones encargados de describir los principios fundamentales de la asignación de responsabilidades (Botero, 2011).

- ✓ **Experto:** este patrón resuelve el problema de asignar responsabilidades utilizando para esto, clases o métodos. Facilita la comprensión del sistema, su mantenimiento y adaptación a los cambios con reutilización de componentes (Pressman, 2006).
- ✓ **Creador:** el patrón Creador soluciona el problema de seleccionar qué clase debe ser la encargada de instanciar a otra. Propicia el principio para la creación de objetos. Es el que crea y guía la asignación de responsabilidades relacionadas con la creación de objetos, encontrando un creador para encadenar con el objeto originado en cualquier evento (Pressman, 2006).

En el diseño del paquete de aplicaciones se puede apreciar el uso de este patrón. Un ejemplo de ello se evidencia cuando la clase **uiGenRep** que es la contenedora de todas las funcionalidades de la aplicación **GenRep** instancia a la clase **REP**, encargada de crear el archivo rlf con el reporte generado:

```
self.myREP=createRLF.classRep(\atributos)
```

- ✓ **Bajo Acoplamiento:** Este patrón es una manifestación que asigna la responsabilidad de controlar el flujo de eventos del sistema a clases específicas. Permite que las clases no se vean afectadas por variaciones de otros componentes, haciendo que sean fáciles de entender y de reutilizar. Recomienda asignar responsabilidades de tal forma que se le dé soporte a una mayor reutilización y poca dependencia. Es una medida de la fuerza con que una clase se relaciona con otras, porque las conoce y recurre a ellas; una clase con bajo acoplamiento no depende de muchas otras (Larman, 2003).
- ✓ **Controlador:** Se implementa como parte del patrón arquitectónico utilizado, sirve de intermediario entre una determinada interfaz y el algoritmo que la implementa recibiendo los datos del usuario y enviándolos a las distintas clases según el método llamado.

Este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación, para aumentar la reutilización de código y a la vez tener un mayor control. Se recomienda dividir

Capítulo 2: Características del sistema

los eventos del sistema en el mayor número de controladores para poder aumentar la cohesión y disminuir el acoplamiento (Larman, 1999).

Patrones GOF

Los patrones GOF acrónimo de *Gang Of Four*, o Pandilla de los Cuatro por su traducción al español se definen como el conjunto de tareas o reglas para resolver problemas que surgen durante el desarrollo del *software* (Larman, 1999).

- ✓ **Decorador:** Es un patrón de tipo estructural encargado de asociar responsabilidades adicionales a un objeto dinámicamente, proporcionando una alternativa flexible a la especialización mediante herencia, cuando se trata de añadir funcionalidades. Brinda mayor flexibilidad que la herencia estática, permitiendo, entre otras cosas, añadir una funcionalidad dos o más veces. Propicia concentrar en lo alto de la jerarquía de clases guiadas por las responsabilidades. De esta forma las nuevas funcionalidades se componen de piezas simples que se crean y se combinan con facilidad, independientemente de los objetos cuyo comportamiento extienden («Patrones de diseño», 2006).

El uso de este patrón en la aplicación se evidencia en el uso del decorador **cursorAction** el cual es el encargado de poner el cursor del *mouse* en modo de Cargando y retornarlo al modo normal cuando se ejecutan métodos que requieren algún tiempo de procesamiento visible al usuario:

```
@cursorAction()
```

```
def saveAs(self, dir=False):
```

```
...
```

```
\\Cuerpo del método
```

```
...
```

2.6.2 Diagrama de Paquete

El diagrama de paquete permite describir como se agrupan los diferentes elementos del sistema, reflejando la dependencia entre los mismos. Además ayuda a representar la arquitectura en un lenguaje de modelado (Larman, 2003).

Capítulo 2: Características del sistema

Para un mejor entendimiento de la solución y conforme a la arquitectura se han organizado los diferentes elementos en paquetes. De ellos, tres paquetes principales, los cuales representan cada una de los elementos de la arquitectura, el paquete Vista que muestra dependencia del paquete Modelo, el paquete Controlador que muestra dependencia de los paquetes Vista y Modelo, y finalmente el paquete Modelo, el cual no depende de ninguno de los demás paquetes. Cada paquete está integrado por sub-paquetes, los cuales fueron divididos de acuerdo con las funcionalidades contenidas. A continuación se muestra cómo fue diseñado el sistema en términos de paquetes:

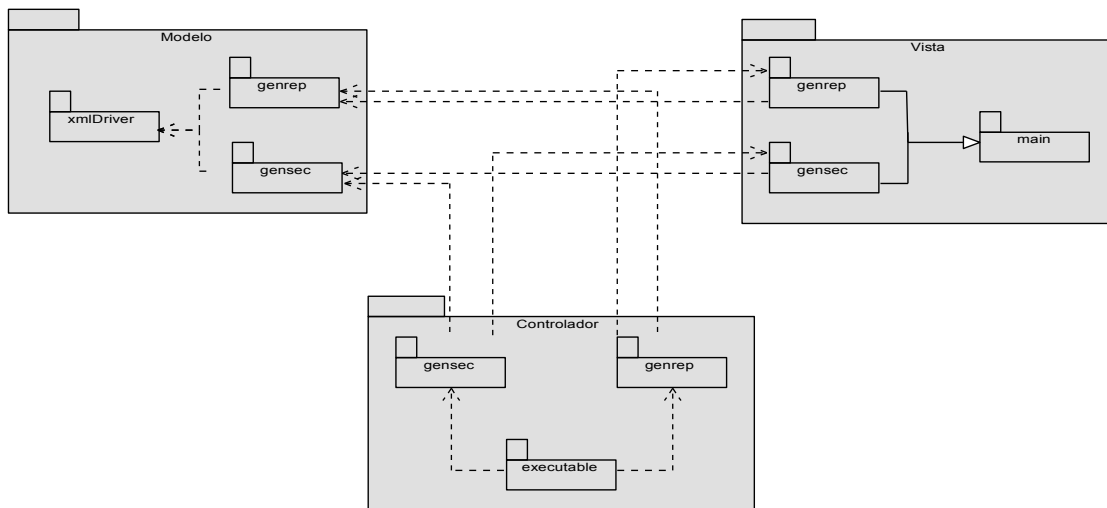


Figura 3: Diagrama de paquetes

Descripción de los paquetes:

- ✓ **Modelo:** Consta de tres sub-paquetes, gensec y genrep que dependen de xmlDriver. Contiene todas las clases principales del sistema que definen cada uno de los elementos que intervienen en la lógica del negocio, con sus respectivos métodos. Los cuales representan las funcionalidades que debe realizar el sistema. Contiene las clases principales encargadas de controlar y darle tratamiento a todo el flujo de información.

Capítulo 2: Características del sistema

- ✓ **Vista:** Contiene los sub-paquetes gensec y genrep que heredan del sub-paquete main contenido en el paquete. Sus elementos son los encargados de manejar el flujo de información que entra y sale del sistema. Es la conexión entre los usuarios y el sistema.
- ✓ **Controlador:** Contiene el sub-paquete ejecutables que presenta dependencia de otros dos sub-paquetes incluidos también en el paquete Controlador, los sub-paquetes genrep y gensec. Es el encargado de controlar las acciones del ratón y el teclado, informando a los paquetes modelo y vista que cambien según sea necesario.

2.7 Conclusiones Parciales

El desarrollo del capítulo permitió describir las características generales del sistema propuesto. Se realizó un buen uso de la metodología de desarrollo lo cual permitió generar cada uno de los artefactos descritos. Se realizó la captura de los requisitos del sistema, lo cual arrojó 51 requisitos, 41 funcionales y 10 no funcionales agrupados en 12 Historias de Usuarios, las cuales servirán de apoyo a la implementación de cada una de las funcionalidades, propiciando un entendimiento entre los programadores y el cliente. Además, la correcta definición de la arquitectura y los patrones de diseño, facilitaron la obtención de un sistema capaz de soportar cambios y mejoras posteriores en sus requerimientos.

Capítulo 3: Implementación y prueba

Capítulo 3: Implementación y prueba

En el presente capítulo se exponen los elementos fundamentales de la implementación y las pruebas del paquete de aplicaciones, haciendo énfasis en la planificación de las iteraciones del ciclo de desarrollo. Se muestra el estándar de código utilizado en el desarrollo del sistema a partir de los requerimientos. Se describen las principales clases con alguno de los métodos más significativos y las pruebas efectuadas a las diferentes funcionalidades del sistema con el fin de lograr un producto con calidad. Además del impacto social que trae consigo la solución a desarrollar.

3.1 Plan de Release

Teniendo en cuenta las funcionalidades descritas en las HU y el nivel de prioridad asignado para su realización por parte del cliente, son estimados el tiempo y esfuerzo para desarrollar cada una de ellas. Posteriormente se procede a la planificación de su implementación, definiendo cuál desarrollar en cada etapa por iteraciones. Resultando 4 iteraciones que abarcan el desarrollo del paquete de aplicaciones y que se muestran en la siguiente tabla.

Releases	Descripción de la iteración	Orden de la HU a implementar.	Duración total.
Iteración 1	Implementación de las HU de Muy Alta prioridad.	HU_1, HU_2, HU_3	7 semanas
Iteración 2	Implementación de las HU de Alta prioridad.	HU_4, HU_5, HU_6, HU_7, HU_8, HU_9, HU_10	3.4 semanas
Iteración 3	Implementación de las HU de prioridad Media.	HU_11	3 semanas
Iteración 4	Implementación de las HU de Baja prioridad.	HU_12	2 semanas

Capítulo 3: Implementación y prueba

Tabla 15: Plan de Release

3.2 Tareas de Ingeniería

Las Tareas de ingeniería se generan a partir de las actividades asociadas a las HU, estas permiten su posterior implementación. También posibilitan conocer qué programador está asignado a cada tarea, así como el tiempo que se necesita para su realización, lo que facilita la estimación del tiempo que se necesita para implementar cada HU, de acuerdo a su complejidad. Proporciona ventajas tales como:

- ✓ Permite organizar el proceso de implementación, pues las tareas se van implementando de acuerdo a su prioridad.
- ✓ Posibilita conocer el grado de complejidad de cada HU, teniendo en cuenta la cantidad de tareas asociadas (Peñalver, Meneses y García, 2010).

A continuación se muestran las cinco tareas de ingeniería creadas para el desarrollo de la HU_1, el resto de las tareas se encuentran en los anexos para un total de 35.

Tareas de Ingeniería para la HU_1

Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: HU_1
Nombre Tarea: Estudio de la librería PyQt4	
Tipo de Tarea: Investigación	Puntos Estimados: 0.4
Fecha Inicio: 1-12-2014	Fecha Fin: 5-12-2014
Programador Responsable: Carlos Manuel Ferrás Hernández	
Descripción: La librería PyQ4 permite integrar el marco de trabajo Qt con el lenguaje Python.	

Tabla 16: Tarea de ingeniería 1 para la HU_1

Tarea de Ingeniería	
Número Tarea: 2	Número Historia de Usuario: HU_1
Nombre Tarea: Creación de la estructura de la secuencia en formato xml	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 6-12-2014	Fecha Fin: 13-12-2014
Programador Responsable: Carlos Manuel Ferrás Hernández	

Capítulo 3: Implementación y prueba

Descripción: La estructura del fichero xml estará compuesta por muestras y procesos. La misma permite describir cualquier operación a través de sus parámetros.

Tabla 17: Tarea de ingeniería 2 para la HU_1

Tarea de Ingeniería	
Número Tarea: 3	Número Historia de Usuario: HU_1
Nombre Tarea: Diseño de la interfaz para la introducción de las muestras y la selección de los procesos a aplicar sobre las muestras seleccionadas.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.4
Fecha Inicio: 14-12-2014	Fecha Fin: 18-12-2014
Programador Responsable: Carlos Manuel Ferrás Hernández	
Descripción: El usuario podrá insertar una o varias muestras y seleccionar los procesos a realizarles.	

Tabla 18: Tarea de ingeniería 3 para la HU_1

Tarea de Ingeniería	
Número Tarea: 10	Número Historia de Usuario: HU_1
Nombre Tarea: Desarrollo de la funcionalidad guardar	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.3
Fecha Inicio: 19-12-2014	Fecha Fin: 22-12-2014
Programador Responsable: Yanet Leonor Quesada Hernández	
Descripción: Esta funcionalidad permite guardar una secuencia de medición y ser medida posteriormente.	

Tabla 19: Tarea de ingeniería 10 para la HU_1

Tarea de Ingeniería	
Número Tarea: 11	Número Historia de Usuario: HU_1
Nombre Tarea: Desarrollo de la funcionalidad imprimir.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.3
Fecha Inicio: 23-12-2014	Fecha Fin: 27-12-2014
Programador Responsable: Yanet Leonor Quesada Hernández	

Capítulo 3: Implementación y prueba

Descripción: Esta funcionalidad permite imprimir una secuencia de medición en formato pdf y ser guardada posteriormente en formato duro.

Tabla 20: Tarea de ingeniería 11 para la HU_1

3.3 Estándar de código

Como parte de las buenas prácticas de la metodología SXP se encuentran los Estándares de Codificación, los cuales aseguran que el código exprese claramente su propósito y agilice el proceso de refactorización, además que sea legible, entendible y refleje un estilo armonioso. Permitiendo de esta manera que todo el código se refleje de la misma forma como si fuera desarrollado por el mismo desarrollador. Para la implementación de la solución se hace uso del estándar **CamelCase**.

CamelCase es un término que se refiere a la práctica de escribir palabras compuestas en la primera letra del identificador en minúscula y la primera letra de cada palabra concatenada subsiguiente se escribe con mayúscula. Es un estilo de escritura que se aplica a frases o palabras compuestas. El nombre se debe a que las mayúsculas a lo largo de una palabra en *CamelCase* se asemejan a las jorobas de un camello. El nombre *CamelCase* se podría traducir como Mayúsculas/Minúsculas Camello. El término *case* se traduce como "caja tipográfica", que a su vez implica si una letra es mayúscula o minúscula y tiene su origen en la disposición de los tipos móviles en casilleros o cajas (IEEE, 2013).

Los programadores a menudo necesitan escribir identificadores descriptivos, como "final de archivo" o "tabla de caracteres", con el fin de mejorar la legibilidad de su código. Sin embargo, los lenguajes de programación más populares prohíben el uso de espacios dentro de los identificadores, ya que se interpretan como delimitadores entre fichas.

3.5 Descripción de los principales elementos del sistema

A continuación se hace referencia a las principales clases con una breve descripción de los métodos fundamentales:

Capítulo 3: Implementación y prueba

Para evitar la repetición de código y simplificar el futuro desarrollo del paquete de aplicaciones se crea una clase interfaz, que sirva de base para las aplicaciones que conforman el paquete, donde se posicionan los elementos en común de cada una de estas.

classBase: esta clase es la encargada de controlar todos los eventos y funcionalidades de esta interfaz base.

En función de reutilizar código se implementa otra interfaz, que sirva de base a las aplicaciones del paquete que tengan la tabla con los datos de los procesos y trabajen con los ficheros slf.

classTableBase: esta clase controla los eventos y funcionalidades de esta interfaz. Dentro de sus métodos más importantes se encuentran:

open(): es el encargado de abrir un archivo slf y rellenar el contenido de la tabla.

setValue(): pasada una posición en la tabla y una cadena, introduce la cadena en dicha posición.

consecutives(): retorna verdadero si las casillas seleccionadas en la tabla son de la misma fila y son casillas consecutivas, falso de lo contrario.

Para cada una de las aplicaciones del paquete se define una clase controladora (**classGenSecMain** y **classGenRepMain**) las cuales tienen como objetivo controlar los eventos y funcionalidades de cada aplicación.

Entre los principales métodos de **classGenSecMain** se encuentra:

addGroup(): adiciona una nueva fila a la tabla.

addComand(): adiciona una nueva columna a la tabla.

new(): borra todo el contenido de la tabla y los parámetros generales para comenzar a realizar una nueva secuencia.

reset(): se encarga de inicializar un fichero slf ya utilizado por el equipo, es decir, borra la información copiada en este por el equipo para reutilizarlo en una nueva medición.

vallidSample(): retorna verdadero si una cadena entrada es válida para el campo de la muestra, falso de lo contrario.

merge(): es el método encargado de unir dos o más procesos consecutivos en la misma muestra dentro del mismo process_order.

Capítulo 3: Implementación y prueba

cancelMerge(): todo lo contrario del merge, luego de estar un grupo de procesos agrupados este método es el encargado de Desagruparlos.

save(): guarda la secuencia creada en formato slf o xml.

print(): imprime la información contenida en la tabla.

Entre los principales métodos de **classGenRepMain** se encuentran:

createGraphic(): reemplaza la gráfica actual por una nueva, basándose en los valores de X y de Y pasados por parámetro.

group(): agrupa varios procesos consecutivos dentro de la misma muestra para expresar que son dependientes.

ungroup(): desagrupa procesos agrupados previamente.

selectRow(): selecciona una fila de la tabla para visualizar sus procesos con datos.

fillLateralPanel(): rellena el panel lateral con los procesos que tienen datos de la muestra seleccionada.

save(): guarda el reporte en formato rlf, xml, o txt tabulado.

print(): imprime los datos del reporte.

Para el trabajo con los ficheros slf y rlf se definen dos clases especiales **classSEQ** y **classRep** respectivamente. Encargada cada una del manejo de estos ficheros xml y de su estructura.

3.6 Pruebas

La prueba es el proceso de ejecución de un programa con la intención de descubrir un error. Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces, la misma tiene éxito si descubre algún error.

A los usuarios de programas extensos generalmente no les interesa los detalles de su funcionamiento; lo único que desean es conseguir respuestas, es decir, desean tratar el programa como una caja negra a la cual le introducen datos de entrada y obtienen de ella los datos de salida que esperan. De ahí el nombre de uno de los métodos de pruebas más usado que existe, el método de Caja Negra. Este método se centra básicamente en las pruebas funcionales realizadas sobre la interfaz del *software*. Permiten demostrar:

Capítulo 3: Implementación y prueba

- ✓ Que las funciones del *software* son operativas.
- ✓ Entradas aceptadas de forma adecuada.
- ✓ Salidas correctas.

De manera análoga, los datos de prueba se escogerán atendiendo a las especificaciones del problema, sin importar los detalles internos del programa, a fin de verificar que éste se ejecute de forma correcta (Fernández, 2005).

Este método se centra en los requisitos fundamentales del *software* y permite obtener entradas que prueben todos los requisitos funcionales del programa. Con este tipo de pruebas se intenta encontrar:

- ✓ Funciones incorrectas o ausentes.
- ✓ Errores de interfaz.
- ✓ Errores de rendimiento.
- ✓ Errores de inicialización y terminación.

Dentro de éste método se encuentra la técnica de Pruebas de Aceptación. Este tipo de prueba reduce el número de casos de pruebas y revelan la presencia o ausencia de errores. Estas pruebas las realiza el cliente. Son básicamente pruebas funcionales, sobre el sistema completo, y buscan una cobertura de la especificación de requisitos y del manual de usuario (Ponce et al. 2010).

La experiencia muestra que aún después del más cuidadoso proceso de pruebas por parte del desarrollador, quedan una serie de errores que sólo aparecen cuando el cliente comienza a usarlo. Para efectuar las pruebas se realizan los casos de pruebas a partir de las HU. Posteriormente el equipo de calidad comienza a aplicar los mismos, en muchas ocasiones a medida que se realizan los casos de pruebas el *software* es probado casi en su totalidad y se recogen las no conformidades que aparecen, esto ahorra tiempo, proporciona una mayor cantidad de pruebas y permite una mayor detección de errores y defectos.

3.6.1 Casos de prueba de Aceptación

A continuación se muestran los cinco casos de pruebas de aceptación creados para probar la funcionalidad recogida en la HU Gestionar secuencia de medición, la cual responde a una serie de

Capítulo 3: Implementación y prueba

requisitos, los cuales fueron probados con dichos casos de pruebas, el resto de las funcionalidades fueron probadas con los demás casos de pruebas recogidos en el Anexo 6.

Caso de Prueba de Aceptación	
Código Caso de Prueba: 1	Nombre Historia de Usuario: Gestionar secuencia de medición
Nombre de la persona que realiza la prueba: Luis Baly Gil	
Descripción de la Prueba: Permite crear una secuencia de medición en formato rlf o xml.	
Condiciones de Ejecución: Debe haber accedido a la aplicación GenSec.	
Entrada / Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Seleccionar dentro de archivo la opción nuevo. 2. Seleccionar opciones 3. Elegir opciones de secuencia 4. Ingresar los siguientes datos: 5. Propietario: admite solo texto 6. Nombre: admite solo texto 7. Protocolo: debe ser una cadena de 16 caracteres 8. Uso de nitrógeno: no acepta letras, debe ser un número que sea o cero o uno 9. Tasa de dosis: no acepta texto, solo números de hasta nueve dígitos 10. Tasa de dosis externa: no acepta texto, solo números de hasta nueve dígitos 11. Insertar muestras 12. Definir procesos 13. Elegir guardar como en la opción archivo 	
Resultado Esperado: Se creará una secuencia con una estructura xml, en el propio formato o en formato rlf.	
Evaluación de la Prueba: Satisfactoria	

Tabla 21: Caso de Prueba de Aceptación_1

Caso de Prueba de Aceptación	
Código Caso de Prueba: 2	Nombre Historia de Usuario: Gestionar secuencia de medición
Nombre de la persona que realiza la prueba: Luis Baly Gil	
Descripción de la Prueba: Permite ordenar la secuencia de medición	

Capítulo 3: Implementación y prueba

Condiciones de Ejecución: Debe estar creada la secuencia
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Elegir la opción editar 2. Ordenar por muestras
Resultado Esperado: Se ordena la estructura de la secuencia según el orden de las muestras a medir.
Evaluación de la Prueba: Satisfactoria

Tabla 22: Caso de Prueba de Aceptación_2

Caso de Prueba de Aceptación	
Código Caso de Prueba: 3	Nombre Historia de Usuario: Gestionar secuencia de medición
Nombre de la persona que realiza la prueba: Luis Baly Gil	
Descripción de la Prueba: Permite modificar una secuencia de medición	
Condiciones de Ejecución: Debe estar creada la secuencia	
Entrada / Pasos de ejecución:	
Se puede modificar una secuencia a partir de dos opciones diferentes:	
Opción uno:	
<ol style="list-style-type: none"> 1. Ir a archivo 2. Elegir abrir 3. Realizar los cambios deseados 	
Opción número dos:	
<ol style="list-style-type: none"> 1. Clic en el botón resetear (se ponen todos los estados en pendiente nuevamente y borra todos los datos adicionados permitiendo volver a usar esa secuencia pero con nuevos datos) 2. Elegir guardar 	
Resultado Esperado: Se guardará la secuencia modificada	
Evaluación de la Prueba: Satisfactoria	

Tabla 23: Caso de Prueba de Aceptación_3

Capítulo 3: Implementación y prueba

Caso de Prueba de Aceptación	
Código Caso de Prueba: 4	Nombre Historia de Usuario: Gestionar secuencia de medición
Nombre de la persona que realiza la prueba: Luis Baly Gil	
Descripción de la Prueba: Permite mostrar una vista previa de la secuencia de medición	
Condiciones de Ejecución: Debe estar creada la secuencia	
Entrada / Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Dar clic en el botón vista previa 	
Resultado Esperado: Aparecerá una ventana donde se mostrará una vista de cómo quedará la secuencia una vez creada.	
Evaluación de la Prueba: Satisfactoria	

Tabla 24: Caso de Prueba de Aceptación_4

Caso de Prueba de Aceptación	
Código Caso de Prueba: 5	Nombre Historia de Usuario: Gestionar secuencia de medición
Nombre de la persona que realiza la prueba: Luis Baly Gil	
Descripción de la Prueba: Permite detener una secuencia en ejecución por el tiempo que se determine	
Condiciones de Ejecución: Deben haber muestras insertadas	
Entrada / Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Seleccionar una o varias muestras 2. Dar doble clic en la columna comando 3. Elegir pause en el menú desplegable 4. Introducir el tiempo que se quiera pausar 5. Aceptar 	
Resultado Esperado: Se detendrá la secuencia el tiempo establecido.	
Evaluación de la Prueba: Satisfactoria	

Tabla 25: Caso de Prueba de Aceptación_5

Capítulo 3: Implementación y prueba

3.6.2 Resultados de las pruebas

Las pruebas fueron efectuadas en tres iteraciones según los resultados arrojados por los casos de pruebas. En la primera iteración se aplicaron 24 casos de pruebas de los cuales, tres arrojaron no conformidad para las HU_2, HU_11 y HU_12. En la segunda iteración se aplicaron tres casos de pruebas, quedando resueltas las no conformidades encontradas anteriormente y detectando una nueva no conformidad para la HU_11 subsanada de manera inmediata en la tercera iteración. Demostrando que el paquete de aplicaciones para la generación y análisis de secuencias de medición del lector de fotoluminiscencia LF02 está listo para ser usado en mediciones reales. En la gráfica siguiente se ilustra el resultado de las pruebas detalladamente.

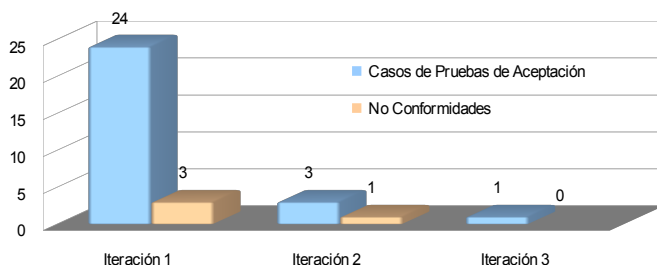


Figura 4: Resultados de las pruebas

No conformidades detectadas en la primera iteración:

- ✓ La funcionalidad definida en la HU_2 no emite una notificación de error al introducir números negativos.
- ✓ La funcionalidad definida para la HU_11 genera un fichero con formato rlf solamente.
- ✓ En la funcionalidad definida en la HU_12 si se modifican los límites de los cursores, la aplicación no funciona.

No conformidades detectadas en la segunda iteración:

Capítulo 3: Implementación y prueba

- ✓ La funcionalidad definida para la HU_11 genera el fichero en formato txt, de manera desorganizada.

3.7 Análisis de los resultados del sistema desarrollado

De la entrevista realizada al Msc. Luis Baly Gil, líder del laboratorio de fechado del CEADEN, se pudo conocer acerca de los problemas existentes con los procesos de generación y análisis de las secuencias de medición del lector LF02, fundamentalmente la existencia de errores de estructura en las mismas, en la entrada de datos y en los cálculos pertinentes para el análisis, además de los alargados tiempos que se requería para ello.

Con el fin de disminuir los errores en la estructura de la secuencia se crea un xml, el cual es completado automáticamente, a medida que se introduzcan las muestras y los procesos, impidiendo que se altere la misma. Para evitar la entrada errónea de datos, se desarrolla el *software* de manera que se introduzcan solo los datos que no afecten directamente la secuencia, validando en cada caso las entradas, por ejemplo el nombre de la secuencia o el propietario. En el resto de los casos dígase muestra, proceso o los parámetros de los procesos, que se necesita introducir información, se aplican distintos selectores, de forma que el usuario seleccione una entrada válida, en lugar de introducir una nueva, disminuyendo así este tipo de errores. Parte del análisis de los datos consiste en calcular diferentes elementos utilizando fórmulas matemáticas, las cuales forman parte de la implementación, por tanto estos cálculos se realizan automáticamente según el análisis deseado, permitiendo así disminuir los errores en los mismos.

Para comprobar que el sistema cumple con el objetivo planteado se toma un muestreo de 30 secuencias creadas manualmente, cada una de una muestra y 20 procesos asociados a cada muestra, para un total de 30 muestras y 600 procesos, a las cuales se les realiza un estudio por los expertos en fechado del laboratorio, arrojando los siguientes resultados:

De las 30 secuencias creadas manualmente se identifican 10 errores de estructura, 360 procesos con errores en los datos y en el análisis 22 muestras con errores de cálculos.

Posteriormente esas mismas secuencias se realizaron con el *software*, identificando de las 30 secuencias, un error en la estructura de la secuencia, dado por un tipo de medición donde se introdujo un proceso

Capítulo 3: Implementación y prueba

antes que otro indebidamente, 7 procesos con errores en los datos, dados por temperaturas asociadas a procesos que antecedian a otros con los mismos parámetros, períodos de irradiación elevados, fuente de luz equivocada y cantidad de canales incorrectos. Además hubo una muestra con error de cálculo durante el análisis. Este tipo de error puede ocurrir si el usuario que realiza las mediciones no posee un conocimiento previo o no tiene bien claro lo que está buscando, de la misma forma pueden afectar los cálculos en el análisis de los resultados. El sistema no es responsable de llevar este control, su objetivo se centra en crear la secuencia de medición a partir de lo indicado por el usuario, según la medición que requiera en ese momento y verificar que los datos introducidos sean válidos, pero el orden de los procesos y los datos referentes a estos puedan variar en dependencia del tipo de medición. En la siguiente tabla se muestra una comparación de los resultados obtenidos manualmente y con el *software*.

Aspectos	Total analizado	Sin el STK	Con el STK
Tiempo para crear una secuencia	30 secuencias	1 día	5 minutos
Tiempo para analizar una secuencia	30 secuencias	3 días	10 minutos
Errores en la estructura de la secuencia	30 secuencias	33,3%	3,3%
Errores en los datos introducidos	600 procesos	60%	1,17%
Errores de cálculos	30 muestras	73,3%	3,3%

Tabla 26: Comparación de resultados con el *Sequence-Toolkit*

3.8 Aporte social de la solución

El paquete de aplicaciones desarrollado como resultado de la presente investigación tiene un gran impacto en la sociedad cubana, pues es una solución creada para el único lector de fotoluminiscencia que tiene el país. Representa un ahorro para el país en concepto de tiempo y recurso. Además es una pieza fundamental para el proyecto iniciado desde el 2001 en el CEADEN vinculado al mega-proyecto: "Escenarios de peligro y vulnerabilidad de la zona costera cubana, asociados al ascenso del nivel medio del mar para los años 2050 y 2100". El cual es liderado por "Geología y geomorfología marino-costera del

Capítulo 3: Implementación y prueba

archipiélago cubano y su vínculo con los movimientos tectónicos recientes” del Instituto de Geología y Paleontología (IGP). Los datos de mayor importancia en este estudio son aquellos relacionados con la edad de los sedimentos, la cual será determinada mediante la técnica de Fotoluminiscencia del Cuarzo y otros minerales, a través del STK. Con este proyecto se pretende determinar en las zonas costeras, la dirección y velocidad de los movimientos tectónicos y de esta manera obtener una mejor comprensión acerca de la evolución de la tierra así como del impacto que tuvieron en el pasado los cambios climáticos. Sin embargo la técnica de datación mediante la fotoluminiscencia, producto de su complejidad y el alto costo de su instrumentación, estaba disponible solamente en países altamente desarrollados o emergentes como China y Brasil. El lector LF02 junto a este *software* convierte a Cuba en uno de los pocos países que aplica dicha técnica y en condiciones muy favorables. Además, los datos obtenidos permiten realizar pronósticos a partir de los cuales se puedan desarrollar un conjunto de medidas para mitigar los efectos del cambio climático en los próximos años. Para Cuba en su condición de isla, afrontar los riesgos que trae consigo el ascenso del nivel del mar es primordial, pues los principales efectos se verían reflejados en la pérdida de parte de su territorio, o de tierras cultivables por el incremento de la salinidad de los suelos.

En la figura 5 se señalan los países⁵ que presentaron ponencias en el último evento internacional de la temática (LED2014) celebrado en Montreal, Canadá en Julio del 2014, donde se puede apreciar que Cuba es uno de los pocos países junto a Dinamarca, Estados Unidos y Alemania que trabajan en el desarrollo de un sistema informático capaz de llevar los procesos de generación y análisis de secuencias de medición por fotoluminiscencia de manera automatizada.

⁵Datos tomados del CEADEN

Capítulo 3: Implementación y prueba

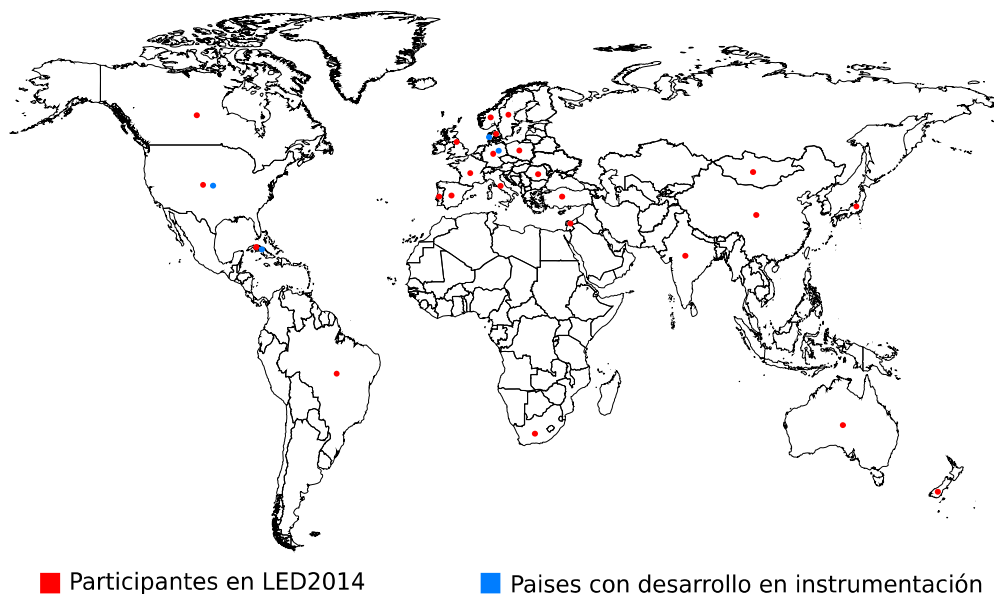


Figura 5: Países dedicados a estudios de datación

3.9 Conclusiones parciales

En el presente capítulo se trazaron 35 tareas de ingeniería que posibilitaron el desarrollo de las 13 HU, apoyándose en las iteraciones definidas en el plan de *Release*. Se profundizó en los componentes del sistema enfatizando en el estándar de código utilizado y la descripción de las principales clases y métodos del sistema. Gracias a las pruebas realizadas por el propio cliente, el producto está apto para ser utilizado, quedando demostrada su efectividad con un ahorro considerable de tiempo y un margen muy pequeño de error atribuido a la mala utilización por parte de los usuarios.

Conclusiones Generales

Al culminar la investigación se puede concluir que:

- ✓ Se redujo el tiempo de creación de una secuencia de un día a 5 minutos.
- ✓ Se disminuyó el tiempo de análisis de una secuencia de tres días a 10 minutos.
- ✓ Se disminuyó el margen de error en la estructura de la secuencia de 33,3% a 3,3%.
- ✓ Se disminuyó el margen de error en la entrada de los datos de 60% a 1,17%.
- ✓ Se disminuyeron los errores de cálculo de 73,3% a 3,3%.
- ✓ El estudio realizado acerca de las herramientas para la generación y análisis de secuencias de mediciones para lectores de fotoluminiscencia permitió tomar algunas características como base para la solución desarrollada.
- ✓ El desarrollo del paquete arrojó como resultados dos aplicaciones, una para generar la secuencia de medición (GenSec) en una estructura xml para el lector de fotoluminiscencia automatizado LF02 y otra para analizar los resultados obtenidos (GenRep).
- ✓ El desarrollo de una interfaz intuitiva y fácil de usar posibilitó la aceptación y el fácil manejo del paquete de aplicaciones por parte de los usuarios.
- ✓ El desarrollo del Sequence-Toolkit permitió obtener un producto libre, multiplataforma y multilenguaje, contribuyendo de esta forma a la soberanía tecnológica.
- ✓ Las pruebas realizadas al sistema demostró que el mismo está listo para ser usado en un entorno de trabajo real.

Recomendaciones

Como resultado del desarrollo de la presente investigación se obtuvo un paquete de aplicaciones. El cual está integrado por GenSec y GenRep, aplicaciones dedicadas a los procesos de generación y análisis de secuencias de mediciones por fotoluminiscencia. Para lograr un mejoramiento en dichos procesos se recomienda:

- ✓ Incorporar un gestor de pestañas a dichas aplicaciones con el fin de mejorar sus interfaces gráficas, logrando un sistema con mayores facilidades de uso.
- ✓ Adherir más traducciones al paquete de idioma en término de internacionalizar el paquete y que pueda ser usado fácilmente en otros países.
- ✓ Desarrollar la aplicación (GenVis) que contendrá el paquete con la idea de complementar el análisis de los datos, efectuando otros tipos de análisis a partir de lo obtenido de GenRep.
- ✓ Implementar el asistente de ayuda al paquete para una mayor aceptación de los usuarios que trabajarán con el paquete de aplicaciones.

Referencias Bibliográficas

- [1]. AZAD M, M. 2010. *Integrating humans with software and systems: Technical challenges and a research agenda*. 21-07-2009, vol. 13, no. 3, pp. 232-345. DOI 10.1002/sys.20145.
- [2]. BIOTEK, 2007. *Gen5™ & Gen5 Secure User's Guide* [en línea]. marzo 2007. S.I.: s.n. [Consulta: 25 septiembre 2014]. Disponible en: http://cores.imp.ac.at/fileadmin/additional_pages/core_facilities/genomics/Protocols/Genomics_Equipment_Protocols/Detection/Synergy2_FluorescencePlateReader.pdf.
- [3]. BOTERO, R. 2011. Patrones graps y anti-patrones: Un enfoque orientado a objetos desde la lógica de programación., vol. 8, pp. 22.
- [4]. CAMPOS, J.I. y VARGAS, S. 2003. Sunpy - Python para física solar: una implementación para seguimiento de correlaciones locales. [en línea], vol. 6, no. 1. [Consulta: 17 enero 2015]. ISSN 2256-3830. Disponible en: http://revistas.udenar.edu.co/index.php/rfacien/article/download/2003/pdf_21.
- [5]. CANÓS, J.H., LETELIER, P. y PANADÉS, M.C. 2003. Metodologías Ágiles en el desarrollo de *Software*. Metodologías Ágiles en el Desarrollo de *Software*. Universidad politécnica de valencia: alicante, España, ISBN 1666-1680.
- [6]. CHAVES, M.A., 2006. La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de *software*. 2006. S.I.: Universidad de Costa Rica.
- [7]. CITÓN, M. 2006. Método Ágil Scrum Aplicado Al Desarrollo De Un *Software* De Trazabilidad [en línea]. S.I.: Universidad De Mendoza, Facultad de Ingeniería en Informática. [Consulta: 15 enero 2015]. Disponible en: <http://www.um.edu.ar/catedras/claroline/backends/download.php?url=L01ldG9kb3NfQWdpbGVzL01ldG9kb19BZ2lsX1NjcnVtLnBkZg%3D%3D&cidReset=true&cidReq=II0162004>.
- [8]. DABBISH, L., COLLEEN, S., MELLON, C., TSAY, J. y HERBSLEB, J. 2012. *Social coding in GitHub: transparency and collaboration in an open software repository. Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pp. 1277-1286. ISSN 978-1-4503-1086-4. DOI 10.1145/2145204.2145396.
- [9]. DULLER, G.A.T., 2008. *Luminescence Dating: Guidelines on using luminescence dating in archaeology*. 2008. S.I.: Institute of Geography and Earth Sciences, Aberystwyth University, UK.
- [10]. FERNÁNDEZ, L. 2005. Un sondeo sobre la práctica actual de pruebas de *software* en España. Revista Española de Innovación, Calidad e Ingeniería del *software*, vol. 1, no. 2, pp. 43-54. ISSN 1885-4486.

Referencias Bibliográficas

- [11]. GARCÍA, G., 2012. Metodología Prehistórica. Tema 3. Los sistemas de datación en Prehistoria [en línea]. 13 noviembre 2012. S.l.: s.n. [Consulta: 26 noviembre 2014]. Disponible en: <http://hdl.handle.net/10045/25114>.
- [12]. GNU GENERAL PUBLIC LICENSE. *gnu.org* [en línea] 2007. [Consulta: 6 mayo 2015]. Disponible en: <http://www.gnu.org/licenses/gpl.txt>.
- [13]. GUTIERREZ, D.G., 2009. Tutorial de *QtDesigner* y *QtDevelop* [en línea]. 2009. S.l.: s.n. [Consulta: 15 enero 2015]. Disponible en: <Http://upcommons.upc.edu/pfc/bitstream/2099.1/7656/1/memoriafinalPFC.pdf>
- [14]. GUTIÉRREZ, J., ESCALONA, M., MEJÍAS, M. y TORRES, J. 2005. Pruebas del Sistema en programación Extrema. Departamento de Lenguajes y Sistemas Informáticos., no. 96.
- [15]. IEEE 2012. *Software* para Análisis de Datos Gen5. *Photovoltaic Specialists Conference (PVSC)* [en línea]. Austin, TX: s.n., pp. 000690 - 000693. [Consulta: 26 noviembre 2014]. ISBN 978-1-4673-0064-3. Disponible en: www.biotek.es/es/products/microplate_software/compare_software.html.
- [16]. IEEE 2013. User Evaluations for *Software Engineering Researchers (USER)*. *2nd International Workshop* [en línea]. San Francisco, CA: s.n., pp. 13 - 15. Disponible en: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6603079&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D6603079. 13778784
- [17]. JACOBSON, I., BOOCH, GRADY, RAMUNBAUGH y JAMES 2000. El Proceso Unificado de Desarrollo de *Software*. Madrid: Pearson Educación. ISBN 84-7829-036-2.
- [18]. JUNIOR, C., C.A. GOLINELLI, PASCHOALIN, L.P. y (primero) 2006. Desarrollo de la técnica de fraccionamiento de proteínas presentes en el suero del calostro por electroforesis bidimensional para su identificación por espectrometría de masa (Maldi-Tof). *Alimentaria*, vol. 6, pp. 120-121. ISSN 03005755.
- [19]. LAPP, T., JAIN, M., ANKJÆRGAARD, C. y PIRTZEL, L. 2009. *Development of pulsed stimulation and Photon Timer attachments to the Risø TL/OSL reader. Proceedings of the 12th International Conference on Luminescence and Electron Spin Resonance Dating (LED 2008)*, vol. 44, no. 5-6, pp. 571-575. ISSN 1350-4487. DOI 10.1016/j.radmeas.2009.01.012.
- [20]. LARMAN, C. 1999. *Parones del Gang Of Four*. S.l.: Patiense Hall.
- [21]. LARMAN, C. 2003. *UML y Patrones Introducción al análisis y diseño orientado a objetos*. Segunda. México: Pearson Educación. ISBN 970-17-0261-1.
- [22]. LARRAMENDI, C.T. 2011. *Plan de desarrollo de la Ingeniería de Requisitos del Proyecto de Informatización de los TPC*. La Habana: Universidad de las Ciencias Informáticas.

Referencias Bibliográficas

- [23]. MAREŠOVÁ, L., HOŠKOVÁ, B., URBÁNKOVÁ, E., CHALOUPKA, R. y SYCHROVÁ, H. 2010. *New applications of pHluorin—measuring intracellular pH of prototrophic yeasts and determining changes in the buffering capacity of strains with affected potassium homeostasis*. vol. 27, no. 6, pp. 317-325.
- [24]. MEDINA, J., REBELO, D., MORGADO, M., MCDADE, G., BONITO, J., MARTINS, L. y MARQUES, L. 2013. UNA CONTRIBUCIÓN PARA LA EDUCACIÓN Y LA CIUDADANÍA: EL TIEMPO GEOLÓGICO. Asociación Española para la Enseñanza de las Ciencias de la Tierra, vol. 21, no. 1, pp. 38-47. ISSN 132-9157.
- [25]. NARDI, A. 2012. «Desarrollo e implementación de la Colección Digital de la Universidad Nacional de Córdoba» «Repositorio Digital Ansenusa de Materiales Educativos» [en línea], vol. 4, no. 3. [Consulta: 1 enero 2015]. Disponible en: <http://hdl.handle.net/10760/17769>.
- [26]. *Patrones de diseño*, 2006. 2006. S.I.: Addison Wesley.
- [27]. PEÑALVER, G., MENESES, A. y GARCÍA, S. 2010. *SXP, METODOLOGÍA ÁGIL PARA EL DESARROLLO DE SOFTWARE*. Cuba: Universidad de las Ciencias Informáticas.
- [28]. PONCE, J., DOMÍNGUEZ, F., ESCALONA, M., MEJÍAS, M., PÉREZ, D., ARAGÓN, G. y RAMOS, I. 2010. Pruebas de aceptación en sistemas navegables. XII Jornada de Innovación y Calidad del *Software*. Universidad de Sevilla, España: s.n.
- [29]. PRESSMAN, R.S. 2006. *Ingeniería del Software*. Un enfoque práctico. Sexta. México: McGraw-Hill.
- [30]. RETAMAR, A., 2004. Mi primera hora con Eclipse. [en línea]. 2004. S.I.: s.n. [Consulta: 3 enero 2015]. Disponible en: [Uhttp://asturlinux.org/archivos/jornadas2004/ponencias/eclipse_ide.pdf](http://asturlinux.org/archivos/jornadas2004/ponencias/eclipse_ide.pdf)UH.
- [31]. REYNOSO, C. y KICILLOF, N., 2004. *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*. 2004. S.I.: s.n.
- [32]. SALAZAR, A., PARRA, A., VERGARAS, L. y MIRALLES, R. 2006. Evaluación No Destructiva por Ultrasonidos para la Catalogación de Cerámicas Arqueológicas. Universidad Politécnica de Valencia, Departamento de Comunicaciones, no. 38, pp. 22-24.
- [33]. SIERRA, M., 2005. Trabajando con *Visual Paradigm for UML* [en línea]. 2005. S.I.: s.n. [Consulta: 15 enero 2015]. Disponible en: <http://personales.unican.es/ruizfr/is1/doc/lab/01/is1-p01-trans.pdf>.
- [34]. SOMMERVILLE, I. 2005. *Ingeniería del Software*. Séptima. Madrid: Pearson Educación.
- [35]. STURM, J. 2000. *Desarrollo de Soluciones XML* [en línea]. España: McGraw-Hill Interamericana de España. ISBN 84-481-3136-3. Disponible en: dialnet.unirioja.es.

Referencias Bibliográficas

- [36]. TREJOS, M. y ZAMORA, D. 2012. Criterios de evaluación de plataformas de desarrollo de aplicaciones empresariales para ambiente web. Monografía para optar al título de Ingeniero de Sistemas y computación. Pereira: Universidad Tecnológica de Pereira.
- [37]. *Ubuntu Community*. [en línea], 2012. 25 abril 2012. S.l.: s.n. [Consulta: 21 noviembre 2014]. Disponible en: http://doc.ubuntu-es.org/Instalar_software#.C2.BFQu.C3.A9_es_un_paquete.3F.
- [38]. VALENCIA, A.M. y FERRO, M. 2011. Documentación y análisis crítico de algunas arquitecturas de *software* en aplicaciones empresariales [en línea]. S.l.: Universidad Tecnológica de Pereira. [Consulta: 3 febrero 2015]. Disponible en: <http://repositorio.utp.edu.co/dspace/handle/11059/2460>. T004.22 V152;6310000085610 F1602
- [39]. VAN ROSSUM, G., 2013. El tutorial de Python [en línea]. febrero 2013. S.l.: s.n. [Consulta: 21 octubre 2014]. Disponible en: <http://python.org.ar/pyar/Tutorial>.

Glosario de términos

Cronología climática: Parte de la historia que se encarga de la datación y ordenación de los hechos históricos a partir de estudios hacia los cambios climáticos.

Cronología relativa: Técnica de cronología utilizada para ubicar, en arqueología, un resto de material o de culturas en un período determinado, estableciendo relaciones del tipo «más moderno que» o «contemporáneo a», con este sistema no interesa el momento exacto sino el orden en que se dieron los acontecimientos.

Cuarzo: El cuarzo es un mineral compuesto de sílice (SiO_2). Tras el feldespato es el mineral más común de la corteza terrestre estando presente en una gran cantidad de rocas.

Datación relativa: La datación relativa es el conjunto de técnicas empleadas en arqueología para establecer relaciones cronológicas entre diversos elementos, ya sea por su ubicación en el espacio, por la similitud o disimilitud de rasgos o atributos.

Dosimetría retrospectiva: Es el cálculo de la dosis absorbida en tejidos y materia como resultado de la exposición a la radiación ionizante, tanto de manera directa como indirecta. Es una subespecialidad científica, en el campo de la física de la salud y la física médica, la cual se enfoca en el cálculo de las dosis internas y externas de la radiación ionizante.

Escorias de fundición: Es un material que contiene sustancias inútiles de un mineral, que se encuentran presentes en cualquier proceso metalúrgico que involucre fundiciones.

Espectrometría: La espectrometría es la medición de la cantidad de energía radiante que absorbe o transmite un sistema químico en función de la longitud de onda; es el método de análisis óptico más usado en las investigaciones químicas y bioquímicas.

Espectrómetro: Instrumento de medición que analiza el tipo de espectro que emite una fuente o que es absorbida por una sustancia que se encuentra en el camino de la luz que emite una fuente. Estos espectros de emisión o de absorción son como una huella digital de las sustancias que forman a nuestra

Glosario de términos

naturaleza. Permite comparar la radiación absorbida o transmitida por una solución que contiene una cantidad desconocida de soluto, y una que contiene una cantidad conocida de la misma sustancia.

Estratos: En Geología se llama estrato a cada una de las capas en que se presentan divididos los sedimentos, las rocas sedimentarias, las rocas piroclásticas y las rocas metamórficas cuando esas capas se deben al proceso de sedimentación.

Feldespatos: Son los minerales primarios más abundantes de la corteza terrestre y en los suelos son constituyentes importantes de la arena y el limo. Además, es una de las principales fuentes de potasio y calcio, los cuales pueden ser liberados de los feldespatos por procesos de meteorización.

Inorgánico: Se definen inorgánico a aquellos compuestos que están formados por distintos elementos, pero en los que su componente principal no siempre es el carbono, siendo el agua el más abundante. En los compuestos inorgánicos se podría decir que participan casi la totalidad de elementos conocidos. Se forman de manera ordinaria por la acción de varios fenómenos físicos y químicos.

Isótopos radioactivos: Los isótopos radiactivos, también llamados radioisótopos, son átomos con un número diferente de neutrones que un átomo de costumbre, con un núcleo inestable que se descompone, emitiendo rayos alfa, beta y gamma hasta que el isótopo queda estable. Una vez que esté estable, el isótopo se convierte en otro elemento completamente. La desintegración radiactiva es espontánea por lo que es difícil saber cuándo va a tener lugar o qué tipo de rayos que se emiten durante la descomposición.

LF02: Lector de Fotoluminiscencia en su versión número dos, creado por el Centro de Aplicaciones Tecnológicas y Desarrollo Nuclear.

Masa molecular: La masa molecular es la masa de una molécula de un compuesto. Se calcula sumando las masas atómicas relativas de todos los átomos que forman dicha molécula. Se mide en unidades de masa atómica, representadas como u, también llamadas unidades Dalton, representada como Da. Esta última unidad es la indicada en el Sistema Internacional de Magnitudes.

Metodología de ionización: Técnica analítica instrumental de alta sensibilidad capaz de identificar cualitativamente y cuantitativamente cualquier tipo de mezclas de sustancias.

Glosario de términos

Multiplataforma: Es un término usado para referirse a los programas, sistemas operativos, lenguajes de programación, u otra clase de *software*, que puedan funcionar en diversas plataformas. Por ejemplo, una aplicación multiplataforma podría ejecutarse en Windows, en GNU/Linux y en Mac OS.

Paleomagnetismo: El paleomagnetismo es la disciplina que estudia el magnetismo retenido por las rocas, con el objetivo de obtener un registro de las configuraciones del campo geomagnético en el pasado. Esto se basa en la propiedad que poseen los minerales magnéticos de retener la dirección del campo magnético respecto a algunos momentos dentro de la historia de la roca. Los orígenes de esas magnetizaciones son variados, pudiendo relacionarse a la formación y enfriamiento de una roca, así como a eventos posteriores a este que involucren reacciones químicas. También se puede generar debido a la exposición por largos períodos a campos magnéticos débiles o la caída de rayos.

PDF: El formato de documento portable o PDF por sus siglas en inglés (portable document format) es un formato de almacenamiento de documentos digitales. Puede ser presentado por Mac, Unix, Windows o GNU/Linux, lo cual lo convierte en multiplataforma. Es uno de los formatos más extendidos en Internet gracias a que puede contener cualquier tipo de texto, multimedia, video, sonido vínculos y miniaturas de páginas.

Potasio–argón: Se emplea para datar la edad de la roca en la que se encuentra depositado o incluido el resto fósil en yacimientos de cronología muy antigua.

Radiación ambiental: Es el tipo de radiación que emana la naturaleza del modo en el que el hombre no interviene, es un tipo de radiación natural que en la mayoría de los casos no constituye ningún problema para la salud humana. Normalmente este tipo de radiación es estudiada para comprender cierto tipo de complejos sucesos en la naturaleza.

Radiocarbono: Es una técnica de datación absoluta basada en carbono, específicamente en el Carbono₁₄ (C₁₄).

Relaciones cronológicas: Se establecen entre los distintos tipos de cronologías existentes, vinculando unos con otros.

Release: Se refiere a la versión funcional de un producto de *software*.

Glosario de términos

Sílex: Piedra formada por un material llamado sílice.

Sistemas físico–químicos: Representan sistemas que relacionan estudios y aplicaciones de la física vinculados a la química.

Software libre: Según la *Fundación del Software Libre* (FSF por sus siglas en inglés), se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, modificar el *software* y distribuirlo modificado.

Tectónica: Es la especialidad de la geología que estudia las estructuras geológicas producida por deformación de la corteza terrestre, por ejemplo las que adquieren las rocas después de haberse formado, así como los procesos que la originan.