



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 2

**SISTEMA PARA LA GESTIÓN DE INFORMACIÓN DEL DEPARTAMENTO DE
REPRODUCCIÓN DE DOCUMENTOS DE LA UNIVERSIDAD DE LAS CIENCIAS
INFORMÁTICAS (DRDUCI)**

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor: Eblis Manuel Caro Rivero

Tutores: MSc. Madelis Pérez Gil

MSc. Olga Lidia Martínez Acosta

La Habana, 2015

“Vive como si fueras a morir mañana; aprende como si el mundo fuera a durar para siempre”.

Mahatma Gandhi

Dedicatoria

Dedico este trabajo especialmente a mis padres, para demostrarles que es posible tocar un sueño con las manos.

Agradecimientos

A mis padres Manolo y Evelín por darme la luz de la vida y hacer de mí un hombre de bien, jamás podré pagarle lo mucho que me han dado.

A Mary, por quererme y soportarme, por brindarme su amor y delicadeza en cada momento, por ayudar a cumplir mi sueño y comprenderme como nadie lo hace. Por ser mi más valioso logro y mantenerme enamorado, el orgullo que me causa tenerte es demasiado grande para poder expresarlo con palabras.

A mis hermanos Elbis y Yely, por hacer mi sueño suyo y brindarme su apoyo incondicional, los quiero, esto sólo es un paso en nuestros planes.

A Mimi, por ser la abuela más cariñosa y sacrificada del mundo, por hacer sangrar sus manos por los suyos y brindarme su amor incondicional.

A mis abuelos Marta y Oscar por quererme por encima de todo y estar a mi lado en momentos decisivos.

A Magaly y a Nelson, por ser mis segundos padres, gracias por el apoyo y la comprensión.

A Yeyo, mi padrino sin bautizo, gracias por la confianza y por ser algo más que un padrino para mí.

A Made, por las peleas, los consejos, la exigencia y la profesionalidad, por ser mi madrina y amiga, por demostrarme que el diccionario es insuficiente cuando se trata de expresar gratitud verdadera.

A mi tutora y consejera Olga Lidia, por lo mucho que ha enseñado en tan corto tiempo, por hacerme llegar cada conocimiento con sutileza y cariño, gracias por tu profesionalidad.

A mis hermanos del alma Joel, Arianna y Tito, gracias por compartir conmigo estos cinco años, no hubiese sido lo mismo sin ustedes tres.

A mis compañeros de apartamento, a mis amigos del 2202 y del 2501, mejores amigos para mi vida de universitario no los quiero.

A todo aquel que haya tenido que ver con la realización de este Trabajo de Diploma que culmina cinco años de arduo estudio y dedicación.

Declaración de autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales sobre esta, con carácter exclusivo.

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

Eblis Manuel Caro Rivero
Autor

MSc. Madelis Pérez Gil
Tutora

MSc. Olga Lidia Martínez Acosta
Co-tutora

Las dificultades que se generan a partir de la gestión de la información en las empresas representan un problema al que se enfrenta la sociedad moderna. La solución a este fenómeno más generalizada es la implantación de un sistema informático para la gestión de la información. El Departamento de Reproducción de Documentos de la Universidad de las Ciencias Informáticas es una de las instituciones que afrontan hoy día este problema. El presente Trabajo de Diploma tiene como objetivo desarrollar un sistema informático que informatice el proceso de gestión de información inherente al Departamento de Reproducción de Documentos del Centro de Información Científico Técnico de la Universidad de las Ciencias Informáticas. Para el desarrollo de este sistema, se empleó XP como metodología de desarrollo de software, Django 1.7.1 como marco de trabajo, PostgreSQL 9.0 como sistema gestor de bases de datos y Python 2.7 como lenguaje de programación.

Palabras clave: sistema informático, gestión de información, reproducción de documentos.

Lista de tareas pendientes	xvi
Introducción	1
1 Fundamentación Teórica	6
1.1 Introducción	6
1.2 Gestión de información	6
1.3 Sistemas para la gestión de información	7
1.4 SGI y proceso de reproducción de documentos	9
1.5 Estudio del estado del arte	9
1.5.1 SGI en el proceso de reproducción de documentos	10
1.5.2 SGI en el proceso de reproducción de documentos en Cuba	11
1.5.3 SGI en el proceso de reproducción de documentos en la UCI	11
1.5.4 Conclusiones del estudio del estado del arte	12
1.6 Metodología de desarrollo del software	12
1.6.1 Programación Extrema	12
1.7 Herramientas y lenguajes informáticos	14
1.7.1 Python 2.7	15
1.7.2 Framework de desarrollo web Django 1.7.7	15
1.7.3 HTML5	15
1.7.4 CSS3	16
1.7.5 JavaScript	16

1.7.6	BPMN 2.0	16
1.7.7	Entorno de Desarrollo Integrado PyCharm <i>Community</i> 4.0.4	17
1.7.8	Gestor de Base de Datos PostgreSQL 9.0	17
1.7.9	PgAdmin III 1.12.2	17
1.7.10	Herramienta CASE para el modelado del sistema Visual Paradigm for UML 8.0	18
1.8	Conclusiones del capítulo	18
2	Propuesta de Solución	19
2.1	Introducción	19
2.2	Proceso de reproducción de documentos en el DRD de la UCI	19
2.2.1	Servicios generales	20
2.2.2	Impresiones a color	20
2.2.3	Control directivo	22
2.3	Características de la propuesta	25
2.3.1	Descripción del Módulo: Administración	26
2.3.2	Descripción del Módulo: Solicitudes	26
2.3.3	Descripción del Módulo: Tareas	26
2.3.4	Descripción del Módulo: Notificaciones	26
2.3.5	Descripción del Módulo: Recursos Materiales	26
2.3.6	Descripción del Módulo: Control Directivo	26
2.3.7	Roles y fucionalidades	27
2.4	Conclusiones del capítulo	27
3	Exploración, Planificación y Diseño	28
3.1	Introducción	28
3.2	Exploración	28
3.2.1	Historias de usuario	28
3.3	Planificación	36
3.3.1	Estimación del esfuerzo	36

3.3.2	Plan de iteraciones	37
3.3.3	Plan de Entregas	39
3.4	Diseño	39
3.4.1	Patrón arquitectónico	39
3.4.2	Patrones de diseño	41
3.4.3	Tarjetas CRC	42
3.4.4	Modelo de datos	46
3.5	Conclusiones del capítulo	47
4	Implementación y Pruebas	48
4.1	Introducción	48
4.2	Estándares de codificación	48
4.3	Desarrollo por iteraciones	49
4.3.1	Iteración 1	49
4.3.2	Iteración 2	55
4.3.3	Iteración 3	56
4.3.4	Iteración 4	57
4.4	Conclusiones del capítulo	58
	Conclusiones	59
	Recomendaciones	60
	Acrónimos	61
	Referencias bibliográficas	63
	Bibliografía	66
	Apéndices	70
A	Tareas de Ingeniería	71

A.1	Iteración 2	71
A.2	Iteración 3	73
A.3	Iteración 4	75
B	Casos de prueba de aceptación	78
B.1	Iteración 1	78
B.2	Iteración 2	80
B.3	Iteración 3	82
B.4	Iteración 4	84

Índice de figuras

2.1	Modelado de proceso del negocio para el subproceso “Servicios generales”.	21
2.2	Modelado de proceso del negocio para la actividad “Atender Tarea” del operario.	22
2.3	Modelado de proceso del negocio para el subproceso “Impresiones a color”.	23
2.4	Modelado de proceso del negocio para el subproceso “Control directivo”.	24
2.5	Modelado de proceso del negocio para la actividad “Consultar documentos” del supervisor.	25
3.1	Diagrama Entidad-Relación usado para representar el modelo de datos.	47
4.1	Resultados de la ejecución de los test unitarios de la primera iteración.	54
4.2	Resultados de la ejecución de las pruebas de aceptación de la primera iteración.	54
4.3	Resultados de la ejecución de los test unitarios de la segunda iteración.	55
4.4	Resultados de la ejecución de las pruebas de aceptación de la segunda iteración.	55
4.5	Resultados de la ejecución de los test unitarios de la tercera iteración.	56
4.6	Resultados de la ejecución de las pruebas de aceptación de la tercera iteración.	57
4.7	Resultados de la ejecución de los test unitarios de la tercera iteración.	57
4.8	Resultados de la ejecución de las pruebas de aceptación de la tercera iteración.	58

Índice de tablas

3.1	Historia de usuario # 1	30
3.2	Historia de usuario # 2	30
3.3	Historia de usuario # 3	31
3.4	Historia de usuario # 4	31
3.5	Historia de usuario # 5	31
3.6	Historia de usuario # 6	32
3.7	Historia de usuario # 7	32
3.8	Historia de usuario # 8	33
3.9	Historia de usuario # 9	33
3.10	Historia de usuario # 10	33
3.11	Historia de usuario # 11	34
3.12	Historia de usuario # 12	34
3.13	Historia de usuario # 13	34
3.14	Historia de usuario # 14	35
3.15	Historia de usuario # 15	35
3.16	Historia de usuario # 16	35
3.17	Historia de usuario # 17	36
3.18	Estimación de esfuerzo por historia de usuario	37
3.19	Plan de duración de las iteraciones	38
3.20	Tarjeta CRC # 1	43
3.21	Tarjeta CRC # 2	43

3.22	Tarjeta CRC # 3	44
3.23	Tarjeta CRC # 4	44
3.24	Tarjeta CRC # 5	44
3.25	Tarjeta CRC # 6	44
3.26	Tarjeta CRC # 7	45
3.27	Tarjeta CRC # 8	45
3.28	Tarjeta CRC # 9	45
3.29	Tarjeta CRC # 10	45
3.30	Tarjeta CRC # 11	46
3.31	Tarjeta CRC # 12	46
4.1	Tarea de ingeniería # 1	49
4.2	Tarea de ingeniería # 2	50
4.3	Tarea de ingeniería # 3	50
4.4	Tarea de ingeniería # 4	50
4.5	Tarea de ingeniería # 5	51
4.6	Tarea de ingeniería # 6	51
4.7	Tarea de ingeniería # 7	51
4.8	Tarea de ingeniería # 8	52
4.9	Tarea de ingeniería # 9	52
4.10	Tarea de ingeniería # 10	53
4.11	Tarea de ingeniería # 11	53
A.1	Tarea de ingeniería # 12	71
A.2	Tarea de ingeniería # 13	71
A.3	Tarea de ingeniería # 14	72
A.4	Tarea de ingeniería # 15	72
A.5	Tarea de ingeniería # 16	73
A.6	Tarea de ingeniería # 17	73

A.7 Tarea de ingeniería # 18	73
A.8 Tarea de ingeniería # 19	74
A.9 Tarea de ingeniería # 20	74
A.10 Tarea de ingeniería # 21	75
A.11 Tarea de ingeniería # 22	75
A.12 Tarea de ingeniería # 23	75
A.13 Tarea de ingeniería # 24	76
A.14 Tarea de ingeniería # 25	76
A.15 Tarea de ingeniería # 26	76
A.16 Tarea de ingeniería # 27	76
A.17 Tarea de ingeniería # 28	77
A.18 Tarea de ingeniería # 29	77
B.1 Prueba de aceptación # 1	78
B.2 Prueba de aceptación # 2	78
B.3 Prueba de aceptación # 3	79
B.4 Prueba de aceptación # 4	79
B.5 Prueba de aceptación # 5	80
B.6 Prueba de aceptación # 6	81
B.7 Prueba de aceptación # 7	81
B.8 Prueba de aceptación # 8	81
B.9 Prueba de aceptación # 9	82
B.10 Prueba de aceptación # 10	83
B.11 Prueba de aceptación # 11	83
B.12 Prueba de aceptación # 12	83
B.13 Prueba de aceptación # 13	84
B.14 Prueba de aceptación # 14	84
B.15 Prueba de aceptación # 15	85
B.16 Prueba de aceptación # 16	85

B.17 Prueba de aceptación # 17 86

La información ha constituido un elemento clave para el hombre desde sus inicios. Ha devenido como recurso estratégico determinante de la competitividad y sustento a los procesos de decisión, siendo el único elemento capaz de crear conocimiento y satisfacer las necesidades de las personas y de las entidades. Se ha convertido en la materia prima indispensable para el desarrollo, equilibrio y adaptabilidad de las organizaciones, sectores y países. Es vista objetivamente, como la expresión material del conocimiento con fines de uso. Surgida como producto de la necesidad empírica de la comunicación, ha sido transformada por el ser humano para hacer uso intensivo de ella en pos de su desarrollo y la creación de nuevos conocimientos. Razón por la que se considera un componente esencial de la vida cotidiana y profesional de la sociedad moderna.

La necesidad de información actual va en incremento y su forma de representación está acorde con el medio en que es necesitada. La forma de representar la información más difundida y tradicional es a través de documentos impresos, que no son más que objetos que soportan la información y sirven para transmitirla con mayor perdurabilidad en el tiempo. Producto del alto grado de desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC), el medio digital ha tomado un mayor protagonismo. Siendo incluso imprescindible para el correcto funcionamiento en algunas organizaciones y mecanismos corporativos.

Generalmente el nivel de existencia de información en las instituciones u organizaciones es directamente proporcional al tamaño y grado de funcionamiento de las mismas. Por lo tanto, mientras mayor resulten ser estos aspectos, mayor es el cúmulo de información y la necesidad de tenerla representada en los medios duros¹ y digitales, así como transitar entre ellos. La información ha generado en la estructura de las organizaciones nuevas formas de gestión derivadas del uso inteligente de la misma (EMERY, 1990).

El desarrollo actual de las TIC también ha tenido un impacto positivo en la manipulación y gestión de información en el ámbito corporativo, y por tanto, en casi todos los sectores de la sociedad. Diversos sistemas o herramientas informáticas permiten gestionar los extensos volúmenes de información generados. Ante la necesidad de manejar gran cantidad de datos, procesarlos y obtener de ellos información pertinente como apoyo a la toma de decisiones, surgen los sistemas para la gestión de información. Los mismos están formados por un conjunto de elementos que interactúan entre sí, con el objetivo de apoyar las actividades de una

¹El término se refiere al medio material en que se representan los documentos, en este caso se trata de papel.

organización.

Cuba es un país en desarrollo que no está exento de las necesidades del mundo actual, por consiguiente, tampoco de la nueva revolución de la tecnología digital y de su utilidad en el ámbito empresarial; lo que ha provocado que la dirección del país se trace metas y estrategias para llevar a cabo el desarrollo de las TIC. Debido a esto, en el país se desarrolla en estos momentos una ardua lucha para lograr la informatización de los procesos en las distintas ramas de la sociedad e insertarse en la evolución del mundo actual. Motivo por el cual surgen nuevos proyectos que tienen como propósito convertir a la nación cubana en una sociedad desarrollada tecnológicamente.

La Universidad de las Ciencias Informáticas (UCI) es uno de estos proyectos creado por la Revolución Cubana, cuyo objetivo esencial es, formar profesionales altamente calificados en la rama de la informática (*Portal de la Universidad de las Ciencias Informáticas 2015*). La UCI está constituida por siete facultades, las cuales comprenden a su vez un conjunto de centros de desarrollo de software. Los objetivos de estos centros giran en torno a la misión de la universidad y cada uno realiza su aporte en la informatización del país especializándose en diferentes áreas.

En la UCI también existen varios centros destinados exclusivamente al manejo y gestión de información. Entre estos centros, se destaca el Centro de Información Científico Técnico (CICT), entidad que tiene como misión proporcionar servicios y productos de información de alto valor agregado que apoyen las funciones principales de la Universidad (docencia, investigación, extensión universitaria y producción), mediante la selección adecuada de las fuentes de información y la utilización de las tecnologías de información y comunicación. Entre sus áreas incluye la Biblioteca Universitaria de la UCI y el Departamento de Reproducción de Documentos (DRD).

El DRD tiene entre sus funciones dar respuesta a las necesidades de reproducción de documentos demandadas por diferentes áreas de la Universidad, fundamentalmente el área docente. El manejo de una gran cantidad de información de manera manual resulta un proceso lento y tedioso que reduce la productividad del departamento, especialmente en los meses de mayo a julio, correspondiente al proceso de defensa de tesis y exámenes finales. Los períodos de espera para ser atendidos, para el completamiento de las solicitudes y para la adquisición de los resultados, son algunos de los elementos que repercuten en la calidad de los servicios brindados por el DRD de la UCI de manera negativa.

Otro elemento que dificulta la ejecución de los procesos de reproducción de documentos, es la manera de almacenar la información relativa al consumo de los recursos físicos, la cual resulta arcaica y engorrosa, limitando el proceso de análisis de la información. Esto también influye directamente en el desempeño de los trabajadores, exigiendo un esfuerzo adicional para realizar informes estadísticos sobre el cumplimiento de las tareas.

El control de los recursos materiales y las actividades laborales de los trabajadores, son tareas ejecutadas por los directivos del CICT. Estos deben ir personalmente a supervisar las acciones realizadas, consultar exte-

nuantemente tanto los documentos asociados al consumo de recursos como los vinculados a las solicitudes y verificar que todo concuerde. El control, que en principio debería formar parte del correcto funcionamiento interno del departamento, constituye una labor que ocupa parte importante del trabajo de algunos directivos del CICT.

Consultar la información generada por el flujo de trabajo es un recurso cuyo aprovechamiento es de vital importancia para el proceso de toma de decisiones y no un obstáculo para el proceso laboral. Esta contradicción constituye una de las principales razones que entorpecen el correcto funcionamiento de los procesos de reproducción de documentos en el DRD. Este fenómeno ocurre específicamente, puesto que se pierde una cantidad razonable de tiempo por parte del supervisor y los directivos del CICT al consultar dicha información.

Dada la situación anterior se plantea el siguiente **problema a resolver**: ¿Cómo facilitar el proceso de gestión de información inherente al Departamento de Reproducción de Documentos del Centro de Información Científico Técnico de la Universidad de las Ciencias Informáticas?

Siendo el **objeto de estudio** de esta investigación los sistemas para la gestión de información.

Para resolver el problema anterior se establece como **objetivo general**, desarrollar un sistema informático que informatice el proceso de gestión de información inherente al Departamento de Reproducción de Documentos del Centro de Información Científico Técnico de la Universidad de las Ciencias Informáticas.

Para dar cumplimiento al objetivo general se derivan los siguientes **objetivos específicos**:

- Elaborar el marco teórico de la investigación.
- Realizar una propuesta de solución a partir del análisis de la problemática dada en el DRD de la UCI.
- Definir las funcionalidades y el diseño con que debe cumplir el sistema.
- Implementar un sistema informático para la gestión de información que cumpla con los requerimientos definidos.
- Realizar pruebas al software implementado para validar que las funcionalidades implementadas cumplan con los requisitos definidos.

Se define como **campo de acción** los sistemas para la gestión de información del proceso de reproducción de documentos.

Para guiar el desarrollo de la presente investigación surgen las siguientes **preguntas científicas**:

- ¿Cuáles son las posiciones teóricas en cuanto a los sistemas para la gestión de información y su empleo en el proceso de reproducción de documentos?
- ¿Cuál es el estado real de desarrollo alcanzado, en la informatización de las áreas involucradas en el flujo y manejo de información del DRD de la UCI?
- ¿Cómo diseñar un sistema para la gestión de información que se ajuste a las características del proceso de reproducción de documentos, en el DRD de la UCI?

- ¿Cómo implementar un sistema para la gestión de información que garantice agilidad y rapidez en la ejecución de los procesos de reproducción de documentos del DRD de la UCI?

Con el propósito de dar cumplimiento a los objetivos anteriormente establecidos y a partir de las preguntas científicas planteadas, se elaboraron las siguientes **tareas de investigación**:

- Revisión bibliográfica para generar el marco teórico conceptual en lo referente al desarrollo de sistemas para la gestión de información.
- Estudio de sistemas homólogos para conocer aspectos regulares en el diseño de sistemas para la gestión de información aplicados al proceso de reproducción de documentos.
- Análisis de las herramientas y metodologías existentes para la realización del sistema.
- Análisis de las principales deficiencias y necesidades presentes en la forma actual de desarrollarse el proceso de gestión de información en el DRD de la UCI, para identificar los requisitos del sistema.

En el desarrollo de la presente investigación se emplearán los siguientes métodos científicos:

Métodos teóricos:

Análítico - Sintético: para la realización del análisis de la información obtenida en el proceso de investigación, realizado en el DRD de la UCI, y empleada para la realización del sistema. Además para el análisis de la información existente dentro y fuera del país, acerca de los sistemas para la gestión de información existentes. Identificando así, conceptos, definiciones, avances y otros elementos concluyentes de utilidad para la investigación en cuestión.

Inductivo - Deductivo: para el arribo a conclusiones a partir del estudio de la información recopilada sobre el proceso de reproducción de documentos y de las características generales y particulares de los elementos, procesos o fenómenos de interés para la investigación.

Modelación: para reflejar la estructura, relaciones y características de la solución a través de diagramas, facilitando también el diseño y la comprensión de las clases necesarias para la implementación de la aplicación.

Métodos Empíricos:

Entrevista: aplicada a los trabajadores del DRD de la UCI y al especialista del CICT, para constatar las condiciones en las que se encuentra el proceso de reproducción de documentos y con qué nivel de informatización cuentan para realizar la gestión de información generada a partir de la actividad laboral.

Observación: para monitorizar de cerca el negocio, y analizar los resultados de la gestión de información en el área del DRD de la UCI.

El presente trabajo de diploma se ha estructurado en introducción, tres capítulos, conclusiones y recomendaciones. A continuación un resumen de las diferentes temáticas que se abordan en los capítulos:

Capítulo 1: “Fundamentación teórica”.

En este apartado se define el marco teórico conceptual y se realiza un estudio del estado del arte de los

sistemas para la gestión de información en el proceso de reproducción de documentos. También se describen la metodología de desarrollo de software y las herramientas a utilizar para el desarrollo del sistema.

Capítulo 2: “Propuesta de solución”.

En este conjunto de epígrafes se brinda una propuesta de solución basándose en el análisis del proceso de reproducción de documentos en el DRD de la UCI. Se realiza una descripción detallada de dicho proceso y se modela el mismo. Se plantea la estructura modular del sistema y los roles que se definen para su explotación.

Capítulo 3: “Exploración, Análisis y Diseño”.

Este capítulo corresponde a las etapas de exploración, planificación y diseño del software. Aquí se detallan elementos tales como la captación de los requerimientos del sistema, la planificación temporal y la estructura del proyecto.

Capítulo 4: “Implementación y pruebas”.

En este capítulo se describen las etapas de implementación y prueba del software. Se detallan elementos como las pruebas unitarias, las tareas de ingeniería y la aplicación de las pruebas de aceptación.

1.1. Introducción

Durante los últimos años se han multiplicado los estudios tendentes a analizar la información como factor clave para la toma de decisiones en la empresa. La información es un recurso que se encuentra al mismo nivel que los recursos financieros, materiales y humanos, que constituyen la razón de ser de la gestión empresarial.

Para comprender los elementos anteriormente planteados es necesario definir y analizar ciertos aspectos de relevancia, que a su vez dan soporte teórico-metodológico y conceptual a la presente investigación. Precisamente a esto están dedicado los siguientes epígrafes, junto a la realización de un estudio del estado del arte, la selección de la metodología de desarrollo de software y las herramientas y lenguajes informáticos para el desarrollo del proyecto.

1.2. Gestión de información

El tránsito de la humanidad a la sociedad de la información condiciona que muchas instituciones, en el aspecto informativo, presenten una excesiva centralización de la información y un flujo abundante de documentos impresos y digitales. Sucede además, que quienes necesitan la información no disponen de ella en el momento y espacio adecuados. Con frecuencia se observa que directivos y trabajadores se encuentran abrumados por documentos e informaciones innecesarias.

La Gestión de Información (GI) surge como concepto dentro del campo de la Ciencia de la información, orientado al manejo de la Inteligencia Corporativa de una organización. Permite organizar la estructura interna de las organizaciones y reaccionar ante los cambios de su entorno, apoyándose en el uso de la información y de los recursos de información disponibles (RODRÍGUEZ, 2002).

Otros autores definen la GI como el proceso de organizar, evaluar, presentar, comparar los datos en un determinado contexto, controlando su calidad, de manera que ésta sea veraz, oportuna, significativa, exacta y útil y que esta información esté disponible en el momento que se le necesite. Está orientada al manejo de la información, documentos, metodologías, informes, publicaciones, soportes y flujos en función de los objetivos estratégicos de una organización (VIDAL, 2006), (MARQUÉS, 2002).

El desarrollo acelerado de la tecnología, acompañado de la renovadora industria del software y la incorporación de coherentes sistemas para la GI, proponen novedosas soluciones para potenciar valores a los denominados recursos intangibles, mejorar estrategias de administración y elevar niveles de eficiencia y eficacia. Ante situaciones de esta naturaleza, la GI se convierte en un aspecto estratégico para las organizaciones que se insertan en el actual entorno y asumen las nuevas tecnologías de información y comunicación. Muchas organizaciones recurren a la implementación de sistemas de gestión de información, espacios e infraestructuras para disponer de su propia información, compartir sus recursos y poseer canales de comunicación rápidos y eficientes, que colaboren con el desarrollo del trabajo y la toma de decisiones.

La mayoría de estas soluciones están basadas teóricamente en conceptos tales como la propia GI y los sistemas de información, sobre los cuales se aborda a continuación, ya que las concepciones más recientes defendidas para la GI, orientan a categorizar a las organizaciones desde un punto de vista donde deben ser consideradas fundamentalmente como sistemas de información.

1.3. Sistemas para la gestión de información

Antes de definir un Sistema para la Gestión de Información (SGI) —o como se le conoce también indistintamente: sistema de gestión de información o sistemas de información transaccionales— es necesario conocer primero algunos aspectos que caracterizan a los Sistema de Información (SI).

La denominación de SI es muy ambigua en tanto que engloba gran cantidad de definiciones y enfoques, y todo ello dependiendo de la perspectiva desde donde se analice (TRAMULLAS, 1997).

Una de las definiciones que más popularidad ha alcanzado en Internet, es la que plantea que: *un sistema de información es un conjunto de componentes que interactúan entre sí para alcanzar un fin determinado, el cual es satisfacer las necesidades de información de dicha organización. Estos componentes pueden ser personas, datos, actividades o recursos materiales en general, los cuales procesan la información y la distribuyen de manera adecuada, buscando satisfacer las necesidades de la organización (Definición de sistema de información - Qué es, Significado y Concepto s.f.).*

Otra de las definiciones contempladas, diferencia tres modelos fundamentales (LÓPEZ, 1991):

- Modelo A. Que contempla desde una perspectiva general y cuyo estudio es utilizado para el desarrollo del resto de los modelos. Este tipo de sistemas es denominado sistema de información generalizada y es un modelo compuesto por variables tales como el entorno, la adquisición y transmisión de datos, el proceso en sí, el almacenamiento de los datos y su utilización y transferencia.

- Modelo B. Es un subsistema dentro de las organizaciones. Destaca los sistemas de información para la gestión (MIS) y los SGI.
- Modelo C. Es el resultado de la conjunción de redes y centros de información, enmarcado en las políticas nacionales y territoriales de información. En este sentido el sistema actúa bajo el principio de la centralización, y la red bajo el principio de coordinación de centros en que, por delegación, se invisten de determinada responsabilidad en la recolección y difusión de fuentes (LÓPEZ, 1991).

Senn por su parte, diferencia entre los siguientes tipos de sistemas de información (SENN, 1992):

- Sistemas para el procesamiento de transacciones (sistemas transaccionales).
- Sistemas de información administrativa.
- Sistemas de información para la toma de decisiones.

Según Bootello, visiones más actuales de un SI sugieren que están orientados a cumplir tres objetivos específicos en las organizaciones (BOOTELLO, 2007-2008):

- Automatizar los procesos operativos.
- Proporcionar información que sirva de apoyo a la toma de decisiones.
- Lograr ventajas competitivas a través de su implantación y uso.

Dentro de este preámbulo se encuentran entonces los SGI, ya que el término, automatización/informatización de los procesos claves de una organización, generalmente viene aparejado a la implantación de un SGI, siendo este el objetivo fundamental que persiguen los mismos.

Teniendo en cuenta los elementos anteriormente planteados, la definición de SGI para la presente investigación se ajusta al concepto dado por la Organización Internacional de Normalización (ISO). El cual comprende que un SGI es un “sistema compuesto de equipos y de personal pertinente que realiza funciones de entrada, procesamiento, almacenamiento, salida y control de información, con el fin de llevar a cabo una secuencia de operaciones con datos”.

El SGI opera sobre el propio funcionamiento de la organización en la que está instaurado, se nutre del flujo de trabajo de la misma. A cambio de entradas¹ que suelen verse como cotidianas y resultan simples para los trabajadores, ofrece múltiples salidas², cuyo análisis arroja conclusiones ventajosas desde un punto de vista interno e incluso competitivo.

La aplicación de un SGI en una organización trae consigo determinados beneficios o ventajas. Su justificación ante la dirección general de la empresa resulta fácil ya que sus resultados son visibles y palpables. Un ahorro significativo de la mano de obra y del esfuerzo humano empleado en los procesos ejecutivos de la entidad, puede lograrse de inmediato con la implantación de un SGI. Razón por la cual es el primer tipo de SI en implantarse conscientemente en las organizaciones.

¹Datos suministrados al SGI para su procesamiento.

²Información que devuelve el SGI una vez procesadas las entradas.

1.4. SGI y proceso de reproducción de documentos

Una vez analizados los elementos anteriores no debe quedar duda del por qué del uso de los SGI en un gran número de instituciones. Resulta recomendable el hecho de instaurarlos en organizaciones que se orienten al tema principal de este trabajo, el proceso de reproducción de documentos. No existe mucha información referida a la relación de estos dos elementos, más allá de que uno se beneficia del otro. Pero se hace realmente necesario un análisis de esta relación, ya que son los componentes esenciales del presente trabajo de diploma.

Las acciones de impresión, escaneo y fotocopia son realizadas por dispositivos electrónicos y generan datos que no siempre son difíciles de almacenar y manejar por parte de los trabajadores. Pero esto solo ocurre cuando se está en presencia de un negocio pequeño. A medida que el negocio crece y se multiplican los clientes y los factores que intervienen en la prestación del servicio, el volumen de la información generada aumenta considerablemente. Por lo que se hace más complicado, el realizar reportes de activos, partes del estado actual del negocio o cualquier otro tipo de análisis que apoye a la toma de decisiones, como por ejemplo puede ser, la identificación de los clientes potenciales de la empresa.

Orientado a resolver este tipo de problemas, una buena solución es la puesta en práctica de un SGI. Basta con incorporarlo al flujo de trabajo de la organización, para abstraerse del problema que significa almacenar y administrar la información, y dar prioridad a realizar acciones que maximicen el rendimiento empresarial a partir de los resultados que el SGI genere.

El desenvolvimiento normal de la reproducción de documentos está basado en la recepción de solicitudes, para su posterior realización, o la oferta directa del servicio al cliente. Por tanto, la información referente a esta etapa del proceso debe quedar resguardada de manera que facilite su consulta y garantice su seguridad; elementos que por las vías tradicionales, como papeles o documentos digitales, resultan de difícil acceso. No siendo así con un SGI, ya que al poseer la información incluida dentro de su propio funcionamiento, y contar con herramientas y tecnologías informáticas que cooperan entre sí, bajo principios de interoperabilidad, posibilitan la accesibilidad y la seguridad de la información (BOOTELLO, 2007-2008).

Teniendo en cuenta lo antes planteado, la implementación de un SGI en áreas dedicadas a procesos de reproducción de documentos resulta una solución que trae consigo numerosas ventajas.

1.5. Estudio del estado del arte

Si bien el uso de los SGI se ha generalizado enormemente, su aplicación en el proceso de reproducción de documentos no cumple las mismas expectativas. Casos aislados de elaboración de sistemas informáticos, constituyen los representantes más notorios de esta categoría de aplicación. A continuación se analizan los mismos, haciendo énfasis en determinados aspectos que puedan ser de utilidad para esta investigación.

1.5.1. SGI en el proceso de reproducción de documentos

Caso Fedex: El caso Fedex constituye el ejemplo más experimentado desde el punto de vista histórico. Se basa en la aplicación de un SGI para el control y servicio de impresiones en la empresa Fedex Office³ (Kinko's), USA.

La compañía empezó en el año 1970 con el nombre de Kinko's, con una sola copiadora, una imprenta, tratamiento de películas, papelería y artículos para la escuela. Con el paso del tiempo, el tipo de clientela a la que Kinko's se enfocaba fue aumentando, lo cual ocasionaba una mayor demanda del servicio (...). Esto hizo pensar a Kinko's en la posibilidad de implantar ciertos mecanismos informacionales, con los objetivos paralelos de satisfacer la demanda existente y expandir el mercado al cual se enfocaba junto a los servicios que hasta el momento brindaba. Entre los años 1995 y 1996 Kinko's llevó a cabo este plan de crecimiento. Para ello instalaron sistemas centralizados de presupuestos y planificación financiera, invirtieron en la infraestructura tecnológica que incluía la interconexión de sus sucursales e instalaron un SGI. Se elaboró e implantó uno de los SGI más completos de la historia en los procesos de reproducción de documentos, funcional incluso hoy día, que incluye la automatización de los procesos, desde su planificación e inicio a partir de solicitudes, hasta su propia ejecución. El SGI está orientado no solamente a los procesos de reproducción de documentos sino que incluye además otras áreas de la empresa como lo son la planificación financiera y la gestión de los recursos humanos (*Fedex Office*, 2014).

SIDIA, PUCESE.

Sistema de auto-servicio para el control y la gestión de impresión (Sistema de auto-servicio para el control y la gestión de impresión (SIDIA))⁴ de la Pontificia Universidad Católica del Ecuador Sede Esmeraldas (PUCESE)⁵. Este sistema, surge en base a la necesidad que tienen estudiantes y profesores de la PUCESE al momento de imprimir sus documentos, puesto que no se poseía un sistema que les permitiera realizar impresiones de manera rápida y segura.

Este sistema está orientado a la informatización total del servicio de impresión en la PUCESE y no a brindar facilidades para la ejecución del mismo. Su puesta en práctica eliminó totalmente el número de operarios presentes en el negocio. Por otra parte, el sistema presenta escasos y pobres módulos. Solamente cuenta con un módulo para la administración de los usuarios y otro para la impresión. No implementa funcionalidades de reportes especializados, limitando así sus posibilidades de aprovechamiento en cuanto a la mejora de la gestión empresarial (CONSTANTINI, 2014).

Sistema de control de impresiones para los laboratorios de informática de la Universidad Nacional Experimental de Guayana (UNEG), Venezuela.

Este sistema forma parte del portal de la UNEG, y resulta en una aplicación muy objetiva y puntual. Se encarga de llevar un control riguroso de la realización de las solicitudes de impresión, los usuarios que la

³<http://fedex.com/us/office/>

⁴<http://pucesa.edu.ec/servicios/kioscos>

⁵<http://pucesa.edu.ec/>

realizan y sobre todo el monto o costo de la acción.

Es un sistema pequeño que le da poca cobertura al proceso en general y brinda pocas (casi ninguna) opciones avanzadas de administración, búsqueda o reportes que puedan ser utilizadas para el aprovechamiento de la información pertinente en la toma de decisiones a alto nivel empresarial.

1.5.2. SGI en el proceso de reproducción de documentos en Cuba

Sistema Integrado de Gestión Bibliotecaria Oráculo

Se trata de una tesina que está orientada a diseñar e implementar los Módulos: “Selección y control de materiales bibliográficos” y “Digitalización y Reproducción de materiales bibliográficos”, como parte del Sistema Integrado de Gestión Bibliotecaria Oráculo, para las universidades cubanas, que permite garantizar un mejor registro, control, análisis y acceso a la información que se tiene en la Biblioteca por parte de sus usuarios (CUBA y TASE, 2005).

A pesar de que esta solución fue implementada en la propia UCI y su fase de prueba fue desplegada en la Biblioteca de la UCI, su alcance resulta de carácter nacional y por tanto está ubicada en este epígrafe.

La solución se centra en los dos procesos mencionados anteriormente y por su cobertura limitada a los procesos y su poca completitud, no se puede catalogar como un SGI. Sin embargo si persigue los mismos principios de un SGI y realiza un tratamiento adecuado de la secuencia de las acciones en el proceso de Reproducción/Digitalización. Carece de un control estricto e intrínseco de los usuarios que tienen permitido su uso, puesto que su autenticación está orientada al uso de servidores LDAP⁶ (ibíd.).

1.5.3. SGI en el proceso de reproducción de documentos en la UCI

Automatización del servicio de impresión de la Dirección de Información

En la UCI existe cierto antecedente de la presente investigación. Se trata de un Trabajo de Diploma titulado “Automatización del Servicio de Impresión de la Dirección de Información”, actual Centro de Información Científico Técnico. El mismo tiene como objetivo desarrollar una aplicación basada en software libre para facilitar el proceso de impresión de documentos de la Dirección de Información (CABRERA y LLANES, 2009).

El sistema fue diseñado para satisfacer las necesidades de dicho centro hace aproximadamente 5 años del momento actual de la realización de esta investigación. El servicio al que estaba dirigido, era satisfacer la demanda de impresiones de las brigadas de las facultades de la UCI, registrando las solicitudes de las mismas para consumir este servicio, el cual dejó de ofertarse sin dar lugar a desplegar el sistema.

La solución en cuestión, a diferencia del presente trabajo, se limitaba a registrar las solicitudes, dejando de lado el resto de las actividades que conforma el proceso de reproducción de documentos propiamente dicho.

⁶Lightweight Directory Access Protocol (en español Protocolo Ligero de Acceso a Directorios) que hacen referencia a un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.

1.5.4. Conclusiones del estudio del estado del arte

El análisis de los distintos casos realizado anteriormente permitió arribar a varias conclusiones sustanciales para la investigación. Los sistemas existentes carecen de completitud de los procesos de GI, cubriendo solo parte del proceso de reproducción de documentos. Son sistemas independientes, no están conectados a ningún sistema externo o superior. La solución más cercana y abarcadora al caso de la problemática planteada, es un software privativo desarrollado para una empresa privada. No existe ningún sistema capaz de ser adaptado como solución al DRD de la UCI.

1.6. Metodología de desarrollo del software

Una metodología de desarrollo de software se puede definir como un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda en la construcción de un software (PRESSMAN, 2001).

En la actualidad existen varias metodologías divididas en dos grandes grupos, las tradicionales y las ágiles. Las primeras se centran en elaborar una documentación exhaustiva de todo el desarrollo del software y en cumplir con la planificación realizada en la fase inicial del proyecto, sin dar margen a posibles cambios, puesto que al ocurrir uno, se ven afectados varios componentes del proceso de desarrollo del software. Otro elemento que las caracteriza es la gran cantidad de participantes con los que cuenta, pues requiere de un equipo de trabajo capaz de administrar un proceso complejo en varias etapas (ALMARALES y SENCIAL, 2013). Alternativamente, surgen las metodologías ágiles, caracterizadas por ser más orientadas al desarrollo del software, con bajos niveles de formalización en la documentación requerida y por ser, a diferencia de las tradicionales, más adaptables a los cambios, requerir de pequeños grupos de trabajo y por ser apropiadas para entornos volátiles (ibíd.).

Según las características del software solicitado, dada la premura de tiempo en que se debería construir, los requisitos de alto nivel suministrados y con el fin de aprovechar los beneficios que proporciona los procesos de construcción rápida de software, se tomó la decisión de utilizar un modelo de desarrollo de tipo ágil; dado que estos métodos permiten la implementación del software con el mínimo de documentación, en el menor tiempo posible y teniendo al cambio como una constante.

1.6.1. Programación Extrema

Programación Extrema (XP) (del inglés Extreme Programming) es una metodología para el desarrollo de software, utilizada mayormente en proyectos de corta duración, caracterizándose por ser ágil. Se basa en la simplicidad, la comunicación y la reutilización del código desarrollado. Pretende minimizar la complejidad de un proyecto y de enfocarse directamente hacia el objetivo, hace uso de las relaciones interpersonales y la rapidez de reacción (BECK, 1999).

XP⁷ está centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje del equipo de desarrollo

⁷www.extremeprogramming.org, www.xprogramming.com, c2.com/cgi/wiki?ExtremeProgramming

y propiciando un buen clima de trabajo. XP se basa en la retroalimentación (feedback) continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, donde existe un alto riesgo técnico (BECK, 1999).

La principal suposición que se realiza en XP es la posibilidad de disminuir la mítica curva exponencial del costo del cambio a lo largo del proyecto, lo suficiente para que el diseño evolutivo funcione. Esto se consigue gracias a las tecnologías disponibles para ayudar en el desarrollo de software y a la aplicación disciplinada de doce prácticas, de las cuales las que hacen de XP la metodología idónea para esta tesis son las siguientes (CANÓS; LETELIER y PENADÉS, 2011):

- **Entregas pequeñas:** producir rápidamente versiones del sistema que sean operativas, aunque no cuenten con toda la funcionalidad del sistema. Esta versión ya constituye un resultado de valor para el negocio. Una entrega no debería tardar más de tres meses.
- **Refactorización (Refactoring):** es una actividad constante de reestructuración del código con el objetivo de remover duplicación de código, mejorar su legibilidad, simplificarlo y hacerlo más flexible para facilitar los posteriores cambios. Se mejora la estructura interna del código sin alterar su comportamiento externo.
- **Diseño simple:** se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto.
- **Integración continua:** cada pieza de código es integrada en el sistema una vez que esté lista. Así, el sistema puede llegar a ser integrado y construido varias veces en un mismo día.
- **Cliente *in-situ*:** el cliente tiene que estar presente y disponible todo el tiempo para el equipo. Éste es uno de los principales factores de éxito del proyecto XP. El cliente conduce constantemente el trabajo hacia lo que aportará mayor valor de negocio y los programadores pueden resolver de manera inmediata cualquier duda asociada. La comunicación oral es más efectiva que la escrita.
- **Desarrollo dirigido por Test:** es una característica fundamental de XP que la implementación de las aplicaciones y el fin de las iteraciones este guiado por la ejecución de test unitarios y pruebas de aceptación respectivamente.

El ciclo de vida ideal de XP es uno de los elementos más polémicos y sujetos a debate de la metodología. Mientras que Kent Beck –su fundador– la concibe como un conjunto de prácticas agrupadas en seis faces específicas, diversos autores difieren de su criterio, orientados a ajustar XP a las características propias de su entorno. La flexibilidad de esta metodología permite adaptarla a las necesidades de la organización. De esta forma, se pueden incorporar las prácticas que den resultado y modificar aquellas cuya especificación original no funcione de forma adecuada.

Las etapas señaladas por Beck en su libro *Extreme Programming Explained* son: exploración, planificación de entregas, iteraciones, producción, mantenimiento y muerte del proyecto. Otros autores como Manuel Calero Solís (CALERO, 2003), Gerardo Fernández Escribano (FERNÁNDEZ, 2002) y Doug Wallace, Isobel

Raggett y Joel Aufgang (WALLACE; RAGGETT y AUFANG, 2002) agrupan las prácticas definidas por Beck en cuatro etapas: planificación, diseño, implementación y pruebas.

Para la selección de las fases de la metodología en la presente investigación se tuvieron en cuenta dos elementos fundamentales: el objetivo de minimizar el tiempo requerido para la elaboración de los artefactos pertenecientes a cada una de las fases y que solo una persona llevará a cabo todas las tareas relacionadas con la elaboración del sistema. Se definen así específicamente cuatro fases: exploración, planificación, diseño e implementación y pruebas.

XP como metodología al fin, presenta determinados artefactos para guiar el desarrollo del proyecto. Estos artefactos son las Historias de Usuario (HU), las Tareas de Ingeniería (TI) y las tarjetas Clase-Responsabilidad-Colaboradores (CRC).

Las HU son la técnica utilizada para especificar los requisitos del software. Se trata de tarjetas en las cuales el cliente describe brevemente las características que el sistema debe poseer. El tratamiento de las HU es muy dinámico y flexible. Cada HU es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en poco tiempo (BECK, 1999). En las HU deben estar presente elementos tales como el rol específico del usuario a realizar la funcionalidad, el valor o beneficio de la misma para el cliente, y los criterios de aceptación o condiciones de satisfacción de la propia HU.

Por su parte las TI constituyen la base de la realización concreta de las HU. Una HU puede generar una o más TI, dependiendo de la complejidad de la misma. En las TI deben estar presentes los datos necesarios para que el programador realice la implementación del código en el tiempo estimado.

Las tarjetas CRC constituyen un artefacto que define la estructura del proyecto y las relaciones entre las clases. Estas tarjetas se dividen en tres secciones que contienen la información del nombre de la clase, sus responsabilidades y sus colaboradores. Una clase es cualquier persona, evento, concepto, pantalla o reporte. Las responsabilidades de una clase son los elementos que conoce y las que realiza, sus atributos y métodos. Los colaboradores de una clase son las demás clases con las que trabaja en conjunto para llevar a cabo sus responsabilidades.

También existen artefactos de carácter intermedio. Estos pueden ser el plan de iteraciones, las pruebas de aceptación y los test unitarios.

1.7. Herramientas y lenguajes informáticos

En todo proyecto, la selección de las herramientas y tecnologías es un punto sumamente importante. Estas definen la naturaleza de la solución y son la base de la calidad de la misma. A continuación se describen las herramientas y lenguajes informáticos seleccionados para desarrollar la solución.

1.7.1. Python 2.7

Python es un lenguaje de programación interactivo, interpretado y orientado a objetos. Proporciona estructuras de datos de alto nivel, tales como listas y matrices asociativas (llamados diccionarios), tipado dinámico y direccionamiento dinámico, módulos, clases, excepciones, gestión automática de memoria, etc. Tiene una sintaxis muy simple y elegante, sin embargo es un potente lenguaje de programación de propósito general. Fue diseñado en 1990 por Guido van Rossum. Al igual que muchos otros lenguajes de scripting es gratuito, incluso para fines comerciales, y se puede ejecutar en prácticamente cualquier ordenador moderno (SANNER, 1999).

Es un lenguaje que puede ser empleado para realizar cualquier tipo de programa, desde aplicaciones a servidores de red o incluso, páginas web. Al ser un lenguaje interpretado significa que no se necesita compilar el código fuente para poder ejecutarlo. Un programa en Python puede tener de 3 a 5 líneas de código menos que su equivalente en Java o C. Tiene una sintaxis muy visual, gracias a una notación con márgenes de cumplimiento obligatorio. La separación de porciones de código en muchos lenguajes, utilizan elementos como llaves o palabras clave (begin y end); en Python solo se tabula hacia dentro, para colocar un margen al código perteneciente a una función o un ciclo. Lo cual ayuda a que los programadores adopten las mismas notaciones y que los programas tengan un aspecto muy similar (M. A. ÁLVAREZ, 2003).

1.7.2. Framework de desarrollo web Django 1.7.7

Un *framework* (marco de trabajo) es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, librerías y un lenguaje interpretado para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Django es un *framework* web de código abierto escrito en Python que permite construir aplicaciones web más rápido y con menos código. Se centra en automatizar todo lo posible y se adhiere al principio DRY (*Don't Repeat Yourself*, o no te repitas a ti mismo). Mantiene de forma rigurosa un diseño limpio en su propio código. Fomenta el bajo acoplamiento: filosofía de programación que dice que las distintas partes de la aplicación deben ser intercambiables.

1.7.3. HTML5

HTML es un lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia para la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, entre otros. Es un estándar a cargo de la World Wide Web Consortium (W3C)⁸, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación (HICKSON y HYATT, 2009).

⁸<http://dev.w3.org>

HTML5 es la quinta versión importante del estándar e introduce algunos elementos que constituyen ventajas de su empleo y hacen que se ajuste a las necesidades actuales de la web. Por otro lado, algunos aspectos de diseño también son incluidos en el lenguaje, así como también algunos detalles de navegación. Dentro de sus ventajas sobresalen la compatibilidad y facilidad que plantea su aprendizaje debido al reducido número de etiquetas en las que se apoya. Con el uso de HTML5, se puede reducir la dependencia de los plug-ins que se tienen que tener instalados para poder ver una determinada página o sitio web (HICKSON y HYATT, 2009).

1.7.4. CSS3

CSS (Cascading Style Sheets, u Hojas de Estilo en Cascada) es la tecnología desarrollada por el W3C con el fin de separar la estructura de la presentación. Es un lenguaje constituido a base de selectores, propiedades y valores que tiene como objetivo dar aspecto visual al código HTML (BARCIA, 2003).

La novedad más importante que aporta CSS3, como tercera versión importante de cara a los desarrolladores de webs, consiste en la incorporación de nuevos mecanismos para mantener un mayor control sobre el estilo con el que se muestran los elementos de las páginas, sin tener que recurrir a trucos o hacks, que a menudo complicaban el código de las web (M. A. ÁLVAREZ, 2008).

1.7.5. JavaScript

JavaScript es uno de los lenguajes de script u orientado a documento, más usados en Internet. Creado por la empresa Netscape⁹, para añadir interactividad a las páginas web. Es compacto y basado en objetos, diseñado para el desarrollo de aplicaciones cliente-servidor a través de Internet (GONZÁLEZ y GAUDIOSO, 2001).

Es un lenguaje interpretado e independiente de la plataforma que permite dar respuestas a eventos iniciados por el usuario, como la entrada de datos en un formulario o la elección de algún enlace del documento HTML. La principal ventaja de JavaScript es que este proceso sucede sin ningún tipo de transmisión de datos a través de Internet, de tal forma que cuando un usuario escribe algo en un formulario, no es necesario que sea verificado, devuelto y enviado al servidor (ibíd.).

1.7.6. BPMN 2.0

Business Process Modeling Notation (en español Notación para el Modelado de Procesos de Negocio) (BPMN) es una notación gráfica estandarizada que permite el modelado de procesos de negocio, en un formato de flujo de trabajo (*workflow*). BPMN fue inicialmente desarrollada por la organización *Business Process Management Initiative* (BPMI), y es actualmente mantenida por el *OMG (Object Management Group)*, después de la fusión de las dos organizaciones en el año 2005 (WHITE, 2011). La versión empleada es la 2.0, de abril de 2011.

El principal objetivo de BPMN es proporcionar una notación estándar que sea fácilmente legible y entendible por parte de todos los involucrados e interesados del negocio (*stakeholders*). Entre estos interesados

⁹<http://netscape.aol.com/>

están los analistas de negocio (quienes definen y redefinen los procesos), los desarrolladores técnicos (responsables de implementar los procesos) y los gerentes y administradores del negocio (quienes monitorizan y gestionan los procesos). En síntesis BPMN tiene la finalidad de servir como lenguaje común para cerrar la brecha de comunicación que frecuentemente se presenta entre el diseño de los procesos de negocio y su implementación (WHITE, 2011).

1.7.7. Entorno de Desarrollo Integrado PyCharm Community 4.0.4

PyCharm es un Entorno de Desarrollo Integrado (IDE) de código abierto creado por JetBrains¹⁰ para dar soporte a programadores de Python específicamente. Cuenta con características que hacen de él un IDE ideal para el desarrollo de esta tesis. Entre estas se destacan el soporte inteligente para Python con completamiento de código inteligente, inspecciones de código, resaltado de errores en tiempo de ejecución, auto-correcciones y refactorizaciones de código automáticas; soporte para JavaScript, HTML5/CSS3 e idiomas de plantillas. Da soporte a *framework* de desarrollo web como Django y brinda un sorprendente paquete de herramientas como un depurador integrado, un corredor de prueba gráfica, un terminal incorporado y herramientas de base de datos integrada con soporte para SQL. Además es personalizable y multiplataforma (JETBRAINS, 2000-2015).

1.7.8. Gestor de Base de Datos PostgreSQL 9.0

Un Sistema Gestor de Base de Datos (SGBD) es un conjunto de programas que administran y gestionan la información contenida en una base de datos. Ayuda a realizar la definición de los datos, el mantenimiento de la integridad de éstos dentro de la base de datos, el control de su seguridad y privacidad; así como la manipulación de los mismos (S. ÁLVAREZ, 2007).

PostgreSQL es un SGBD relacional, robusto y confiable. Soporta gran variedad de sistemas operativos como Linux y UNIX en la mayoría de sus variantes y Windows 32/64bit. Se caracteriza por ser un sistema de alto rendimiento y gran flexibilidad, lo que permite a los desarrolladores generar nuevas aplicaciones o mantener las existentes. Se desempeña mejor en ambientes con altas cargas de usuario y consultas complejas, donde la integridad de los datos es muy importante (*PostgreSQL* 1996-2015).

Teniendo en cuenta los aspectos mencionados anteriormente se selecciona dicha herramienta, en su versión 9.0, para gestionar la base de datos del sistema.

1.7.9. PgAdmin III 1.12.2

PgAdmin III está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. Su interfaz gráfica soporta todas las características de PostgreSQL y facilita enormemente la administración del mismo.

La aplicación también incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar scripts programados y mucho más. Fue desarrollado por una comunidad de

¹⁰<https://www.jetbrains.com/>

expertos en PostgreSQL alrededor de todo el mundo y está disponible en más de una docena de idiomas. Es multiplataforma y puede funcionar bajo GNU / Linux, FreeBSD, Mac, Windows y Solaris (*PgAdmin* 2007).

1.7.10. Herramienta CASE para el modelado del sistema Visual Paradigm for UML 8.0

Existen diversas herramientas CASE que soportan el lenguaje de modelado UML, dentro de las que se destaca Visual Paradigm for UML (VP-UML) por ser bastante completa, disponer de varios diagramas como los de modelado del negocio y modelado de la base de datos, por la facilidad de uso para el modificación y creación de estos y por generar código a partir de dichos diagramas. Visual Paradigm es gratuito aunque se encuentra bajo una licencia que no permite su modificación o venta, ajustándose esta última a las políticas de soberanía tecnológica trazadas por el país. Teniendo en cuenta la importancia del uso del software libre en Cuba, la existencia de una Licencia UCI para el uso de Visual Paradigm y las características de esta herramienta, se selecciona, en su versión 8.0, para realizar el modelado del sistema.

1.8. Conclusiones del capítulo

El análisis de los principales conceptos asociados al dominio del problema facilitó una mejor comprensión del presente trabajo, dándose cumplimiento al primer objetivo específico de la investigación. El estudio del estado del arte realizado, permitió obtener las tendencias de la forma de estructurar y organizar los sistemas para la gestión de información en el proceso de reproducción de documentos, proporcionando ideas para el diseño del sistema. Arrojó resultados tales como la generalización del empleo de las tecnologías web para la implementación de este tipo de sistemas y la independencia con la que deben cumplir, al no estar ligados a sistemas superiores. Este estudio demostró que no existen soluciones alternativas que aplicar a la problemática presente en el DRD de la UCI. Por otra parte, se realizó la selección de los lenguajes, herramientas y metodología para guiar el proceso de desarrollo de software, lo que permitirá lograr mayor productividad, mayores resultados y garantizará la obtención de un producto de mayor calidad que satisfaga plenamente las necesidades del cliente.

2.1. Introducción

Una vez estudiados todos los elementos que dan base al desarrollo de la investigación es necesario comenzar con la concepción del sistema, a partir del análisis primario de una serie de elementos. El presente capítulo se centra en brindar una propuesta de solución basándose en el análisis del proceso de reproducción de documentos en el DRD de la UCI. Se realiza una descripción detallada de dicho proceso y se modela el mismo haciendo uso de BPMN. Se plantea la estructura modular del sistema y los roles que se definen para su explotación.

2.2. Proceso de reproducción de documentos en el DRD de la UCI

Actualmente el proceso de reproducción de documentos en el DRD de la UCI puede ser dividido en tres subprocesos: la realización de las solicitudes de servicios generales, la realización de las solicitudes de servicios a color y el control directivo.

El equipo de trabajadores del DRD está compuesto por un supervisor del departamento, tres operarios y un técnico. Los servicios que brinda el DRD son impresión, escaneo, fotocopia y encuadernación de documentos. Los mismos cubren la reproducción de varios tipos de documentos que incluye tesis de pregrado, tesis de postgrado, exámenes, documentos de soporte para eventos, modelos, diplomas, manuales de usuario, entre otros.

El consumo de los servicios por los clientes comienza con la realización de una solicitud. La solicitud del servicio se realiza a partir de una autorización por parte de los jefes de cada área a consumirlo, facultades, vice-rectorías, etc., seguida por la realización de la solicitud de forma presencial al DRD. Actualmente, la solicitud es recepcionada por el supervisor quien designa a uno de los operarios para que realice la tarea de reproducción del documento en cuestión. Después de realizada la tarea se procede a notificar al cliente la culminación del trabajo.

La recepción de las solicitudes, la asignación del trabajo, la realización de las labores de impresión, las

notificaciones y el control de las tareas y de los recursos empleados, son actividades realizadas manualmente. Estas labores generan una gran cantidad de información perteneciente al flujo de trabajo realizado, la cual queda plasmada en forma de documentos digitales tales como tablas de Excel¹ y documentos de formato duro (modelos de solicitudes, tabla de registro de solicitudes, estiba de almacén, etc.).

A continuación se realiza la descripción detallada de cada subproceso con su respectivo modelado haciendo uso de BPMN.

2.2.1. Servicios generales

Este proceso está basado en la prestación de un conjunto de servicios enunciados a continuación:

- Impresión de exámenes.
- Impresión de tesis de pregrado.
- Impresión de otros tipos de documentos².
- Fotocopia de documentos.
- Escaneo de documentos.
- Encuadernación.

El consumo de todos estos servicios, comienza a partir de una solicitud que se realiza por parte del cliente con una autorización previa de los jefes de cada área a consumir el servicio. En la Figura 2.1 se puede apreciar este proceso con mayor detalle.

Esta solicitud es presentada por el cliente de forma presencial en el DRD. La solicitud es recepcionada y registrada por el supervisor, quien valora a través de una consulta la disponibilidad de los recursos materiales con que cuenta el departamento. Si dicha solicitud puede ser atendida o no, de no ser posible se pospone la realización de la solicitud hasta que se cuente con los recursos necesarios. En caso de que se pueda realizar, el supervisor le asigna la realización de la solicitud a un operario, quien se encarga de la ejecución de las tareas de reproducción (Figura 2.2). Una vez concluidas estas tareas, el supervisor notifica al cliente que la solicitud ha sido realizada.

2.2.2. Impresiones a color

El proceso de impresión de documentos a color (Figura 2.3), es un proceso exclusivo debido a la baja disponibilidad de cartuchos de tinta a color y su alto costo. Por lo que la comunidad universitaria solo tiene acceso al mismo a través de una serie de mecanismos organizativos que culminan en el director del CICT, quien es el encargado de realizar la solicitud al técnico del DRD. Posteriormente el técnico realiza las actividades para la ejecución del servicio y le informa al director que se completó la solicitud. De los procesos principales del DRD de la UCI, éste es el más sencillo de realizar y constituye la función principal del técnico dentro del departamento.

¹Microsoft Excel es una aplicación distribuida por Microsoft Office para hojas de cálculo. Este programa es desarrollado y distribuido por Microsoft, y es utilizado normalmente en tareas financieras y contables.

²Veáse la descripción de los mismos en el epígrafe dedicado a la introducción de la investigación

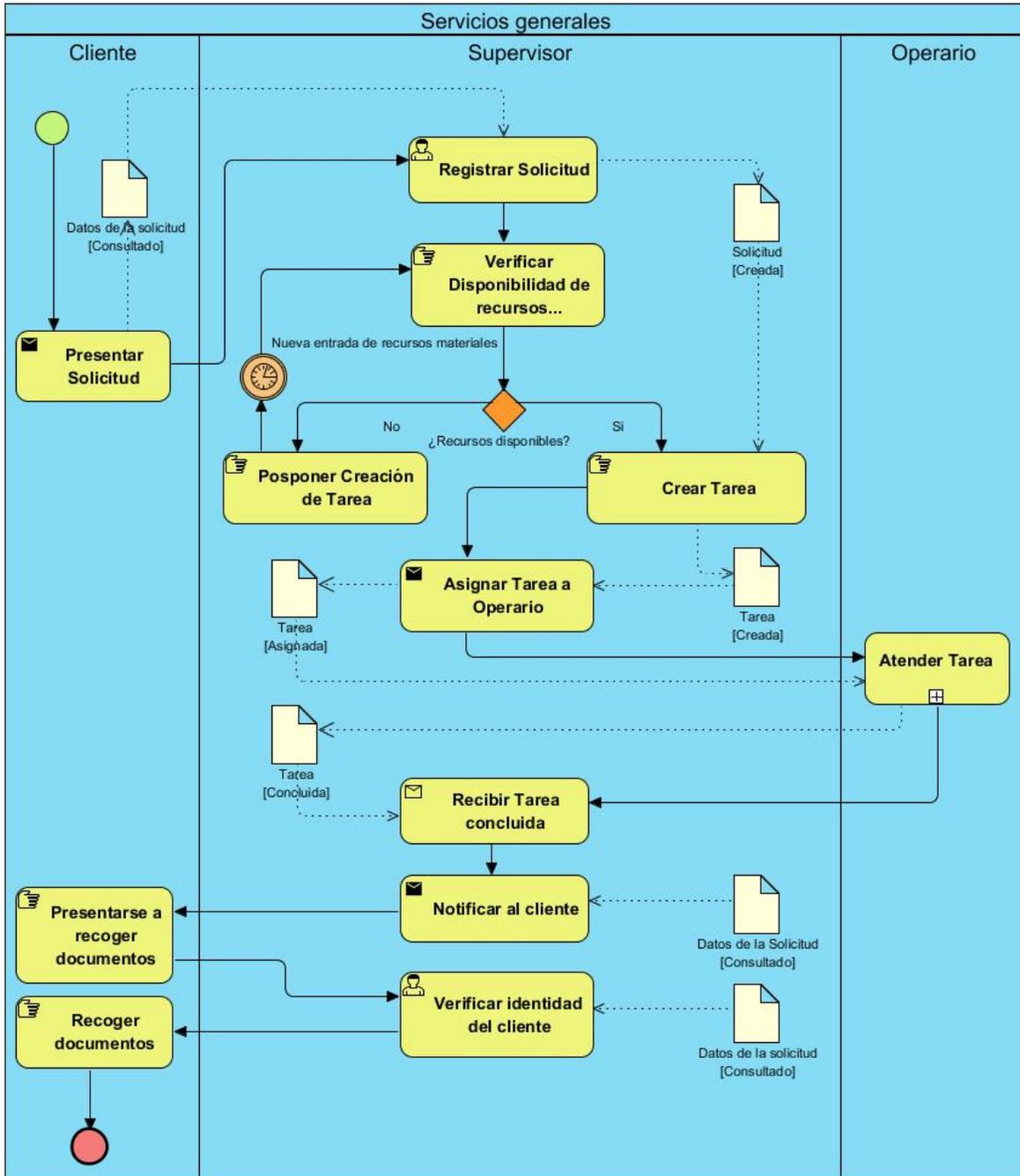


Figura 2.1. Modelado de proceso del negocio para el subproceso “Servicios generales”.

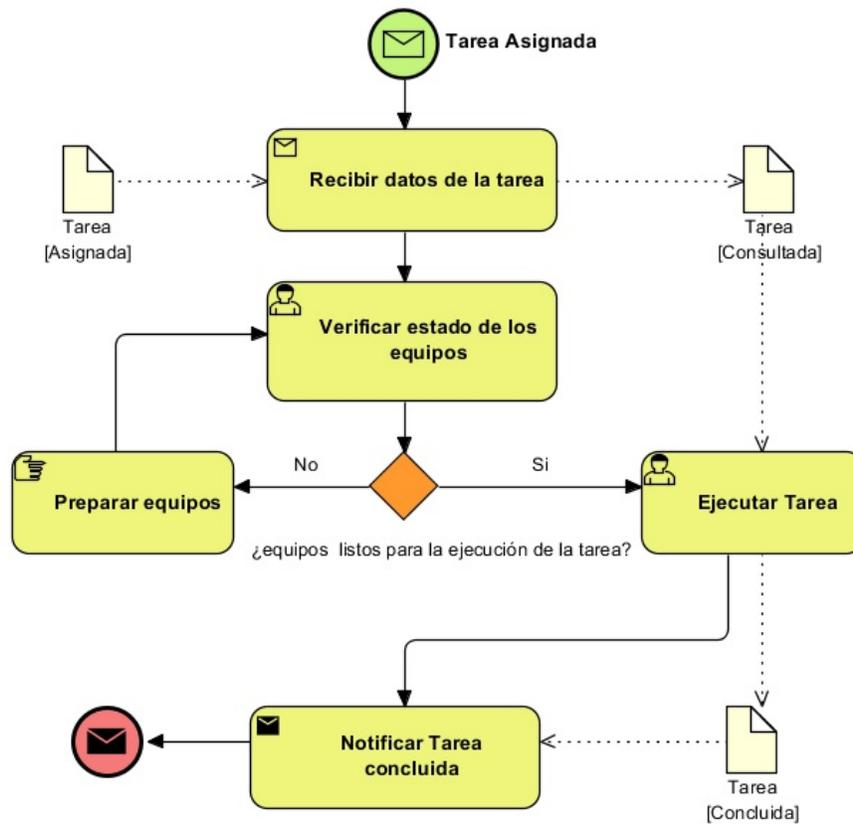


Figura 2.2. Modelado de proceso del negocio para la actividad “Atender Tarea” del operario.

2.2.3. Control directivo

El proceso de control directivo (Figura 2.4) consiste en el control que se establece por parte de los directivos del CICT sobre las labores realizadas por los trabajadores del DRD. El proceso es comenzado por un funcionario del CICT. El mismo verifica manualmente en conjunto con el supervisor, a través de partes y reportes estadísticos o históricos, que los recursos consumidos estén en correspondencia con los servicios prestados. Este proceso también arroja el estado de determinados indicadores de utilidad para la toma de decisiones.

Para la realización de dichos partes o reportes, al supervisor le resulta necesario consultar extenuantemente los documentos generados por el flujo de trabajo del departamento (Figura 2.5). Esta es una actividad de alta complejidad debido a que está sujeta a la observación detallada de los documentos. Para la elaboración de los reportes es necesario el cálculo de los indicadores definidos por el directivo. Por estas razones la aparición de errores humanos en este proceso resulta común, elemento que dificulta en gran medida la realización del control directivo.

Este proceso se aplica de manera similar a los servicios de reproducción de documentos a color, sólo con la diferencia de que el rol encargado de la consulta de los documentos y la elaboración de los partes y reportes,

es el técnico y no el supervisor.

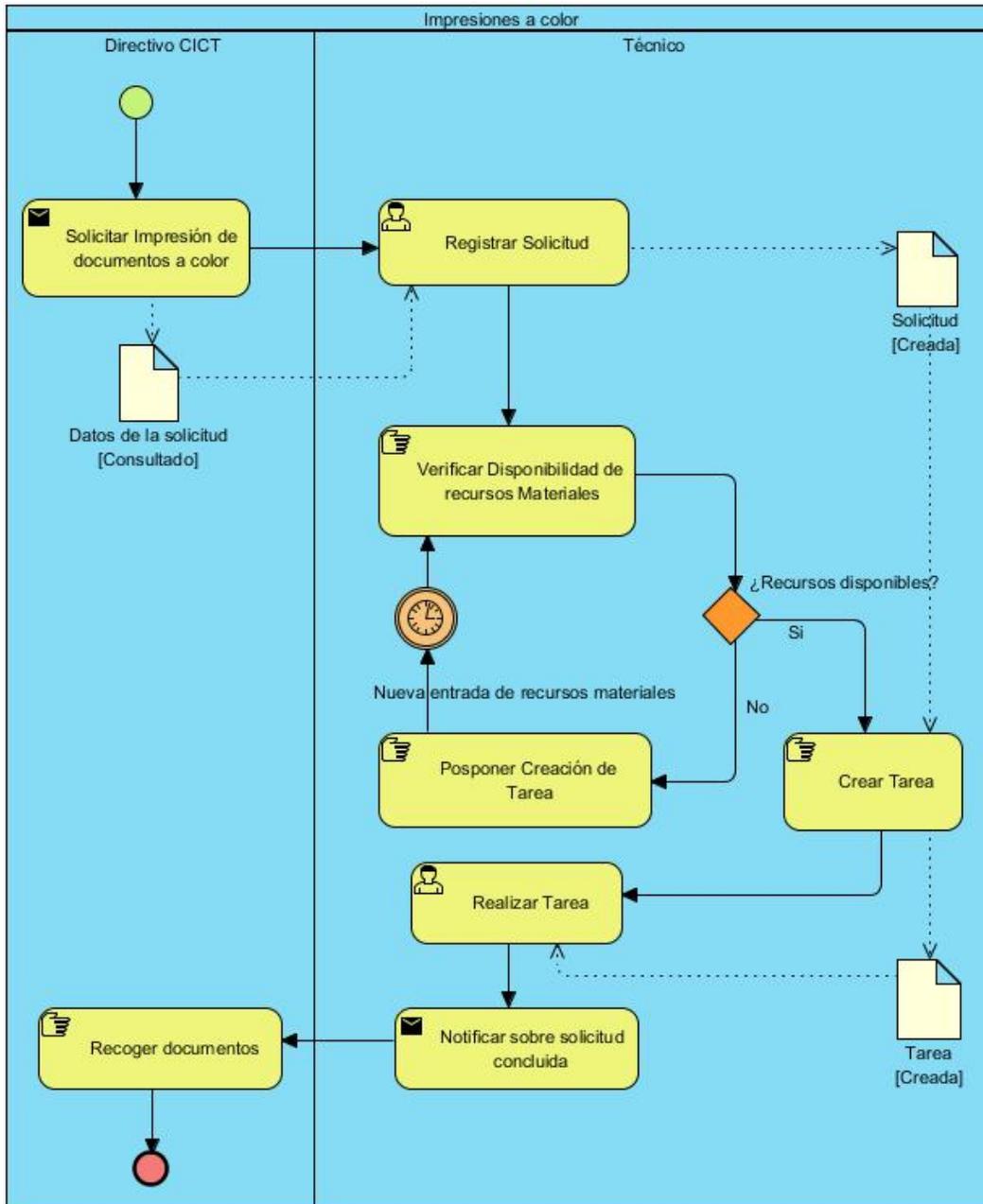


Figura 2.3. Modelado de proceso del negocio para el subproceso “Impresiones a color”.

Dentro del DRD también se realiza la gestión de los recursos materiales que se emplean en la reproducción de documentos. Tarea que brinda soporte al resto de las actividades antes mencionadas y permite mantener el control del ritmo de trabajo.

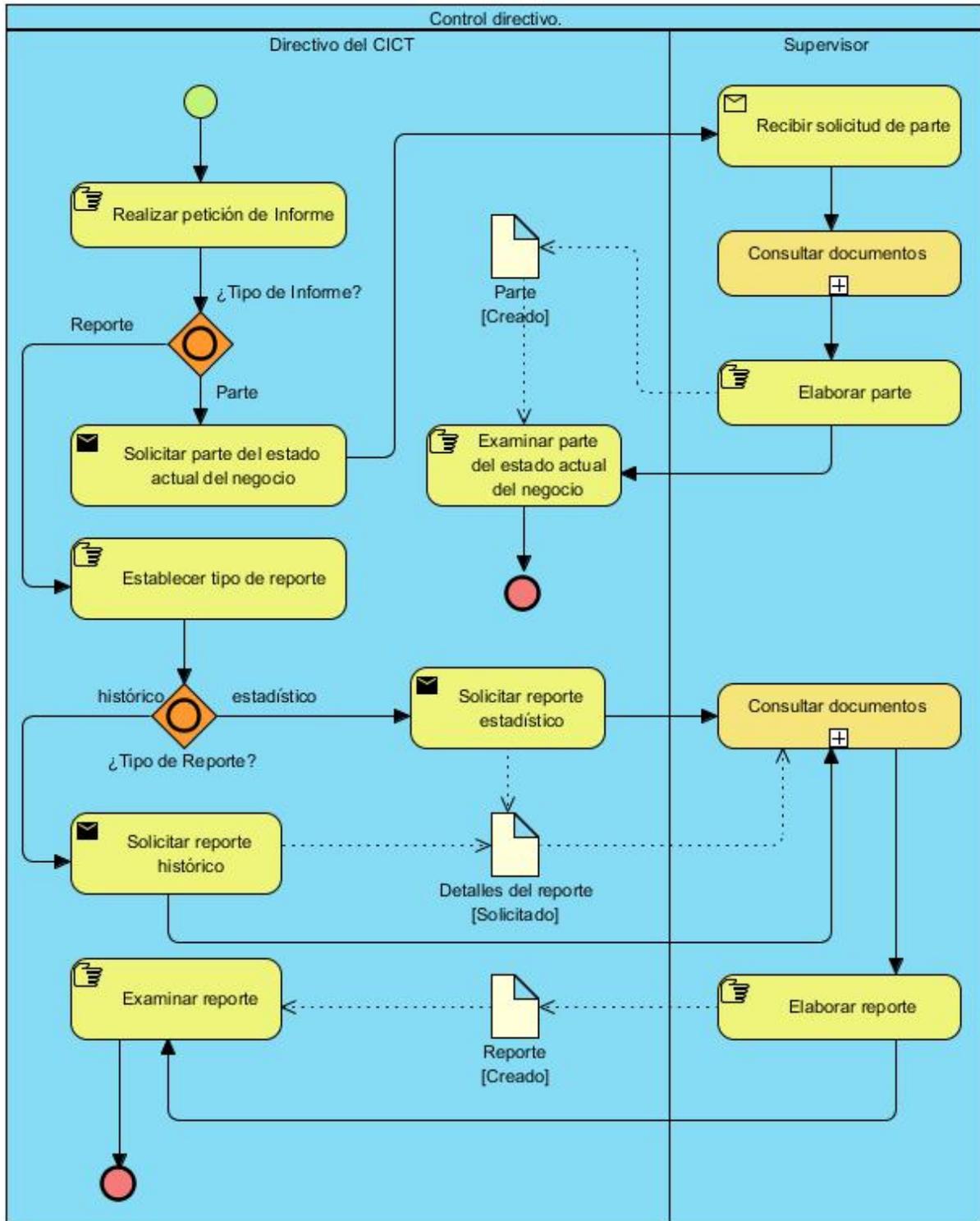


Figura 2.4. Modelado de proceso del negocio para el subproceso “Control directivo”.

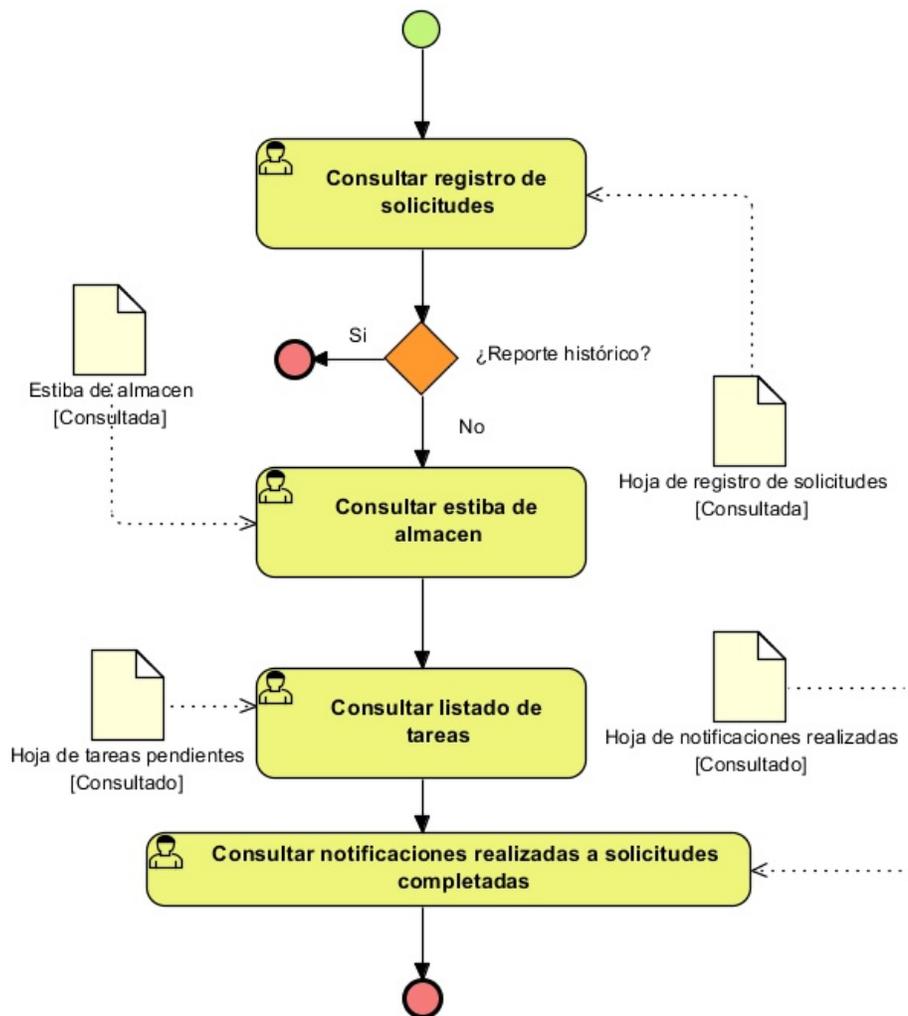


Figura 2.5. Modelado de proceso del negocio para la actividad “Consultar documentos” del supervisor.

2.3. Características de la propuesta

El presente trabajo se propone desarrollar una aplicación web que cuente con una interfaz sencilla, entendible y fácil de usar para todos los usuarios sin importar los niveles de conocimiento de las tecnologías que estos posean. En su desarrollo se utilizarán tecnologías y herramientas de código abierto, contribuyendo a reforzar así la batalla que está librando en estos momentos la Universidad, para migrar a software libre.

El sistema a desarrollar cuenta de seis módulos: Administración, Solicitudes, Tareas, Notificaciones, Recursos Materiales y Control Directivo. A continuación se muestra una descripción de dichos módulos.

2.3.1. Descripción del Módulo: Administración

Este módulo ofrecerá al supervisor la posibilidad de crear, modificar y eliminar los usuarios del sistema y establecer el correo de salida del DRD, el contenido por defecto del mensaje de las notificaciones enviadas a los clientes al culminar una solicitud y editar el perfil del usuario.

2.3.2. Descripción del Módulo: Solicitudes

Está formado por un conjunto de clases y funcionalidades que tienen como objetivo facilitar la gestión de la información asociada al registro, modificación y eliminación de solicitudes de los clientes para el consumo de los servicios ofrecidos por el Departamento. Permitirá la asignación de una solicitud a un trabajador para que la realice. Está orientado a automatizar las etapas de recepción y gestión de solicitudes de los procesos “Servicios generales” e “Impresiones a color” y eliminar así la manera en que se almacena la información referente a las solicitudes.

2.3.3. Descripción del Módulo: Tareas

Este módulo permitirá al técnico y a los operarios la realización de las tareas asociadas a las solicitudes de los clientes. Con su implementación se podrá dar seguimiento al trabajo individual de los operarios y se conocerá en todo momento en que equipo se realizó cada tarea, permitiendo así tener un mayor control sobre el trabajo realizado. Consiste en un módulo de transición entre el inicio y el final de las solicitudes de los clientes basado en brindar facilidades para la ejecución de las tareas.

2.3.4. Descripción del Módulo: Notificaciones

El módulo Notificaciones tiene como objetivo la gestión de las notificaciones generadas al culminar las tareas. Brinda la posibilidad al supervisor de enviar o cancelar las notificaciones y de cambiar el contenido del mensaje de las mismas. Es el módulo que se encarga de la comunicación a los clientes sobre la culminación del trabajo asociado a su solicitud.

2.3.5. Descripción del Módulo: Recursos Materiales

Recursos Materiales es un módulo que como su nombre lo indica tiene como objetivo la gestión de los recursos materiales y los equipos empleados en los procesos de reproducción. Tanto los consumidos por servicios generales como por servicios a color serán administrados a través de este módulo. Al igual que el módulo Solicitudes consiste en una funcionalidad segmentada, ya que los procesos antes mencionados son exclusivos y ejecutados por roles distintos.

2.3.6. Descripción del Módulo: Control Directivo

Contiene las clases y funcionalidades encargadas de facilitarle la elaboración de partes y reportes a los usuarios de tipo directivo y supervisor. Este módulo ofrecerá la posibilidad de conocer el comportamiento del proceso de reproducción de documentos en un período de tiempo definido y a partir de varios parámetros.

Será de suma utilidad para el proceso de toma de decisiones de los directivos. La representación de los reportes incluirá opciones de elaboración de gráficos para facilitar la comprensión de los resultados.

2.3.7. Roles y funcionalidades

Después del estudio del proceso de reproducción de documentos en el DRD se pueden resumir los roles identificados para el sistema y sus funciones.

- **Directivo:** encargado de realizar el control y las solicitudes de impresión a color.
- **Supervisor:** encargado de la gestión de las solicitudes de servicios generales, la asignación de tareas y la notificación a los clientes asociados a los servicios generales.
- **Técnico:** encargado de la ejecución de las tareas del servicio a color y la gestión de los recursos y equipos asociados a este servicio.
- **Operario:** encargado de la ejecución de las tareas de servicios generales.

2.4. Conclusiones del capítulo

En este capítulo se le realizó un análisis a la forma actual de desarrollarse el proceso de reproducción de documentos en el DRD de la UCI. Basándose en este análisis se propuso una solución, que consiste en una aplicación web que automatice el proceso mencionado anteriormente, dándole cumplimiento al segundo objetivo de esta investigación. Se especificó que el sistema debe contar con seis módulos y cuatro roles definidos. Se realizó una descripción de cada uno de los módulos y roles del sistema y se arribaron a otros elementos concluyentes, tales como que deben existir módulos con permisos mixtos a sus funcionalidades, debido al carácter excluyente de determinados subprocesos, y que el cliente, a pesar de jugar un papel importante en el negocio, no interactúa directamente con el sistema propuesto y por tanto no debe ser un rol incluido en el mismo.

3.1. Introducción

El desarrollo del presente capítulo está dedicado a la identificación de las principales funcionalidades de la solución propuesta. Durante los epígrafes que componen el mismo, se expondrán los artefactos generados por la puesta en práctica de la metodología de desarrollo de software XP en sus fases de Exploración, Planificación y Diseño.

3.2. Exploración

En esta fase, los clientes plantean a grandes rasgos las historias de usuario (HU) que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo de la familiaridad que tengan los programadores con la tecnología (BECK, 2000).

3.2.1. Historias de usuario

Una HU es una caracterización de una o más funcionalidades del sistema, descrita por el cliente o propietario del producto. Representa un incremento de valor al negocio, a implementar por el equipo de desarrollo. Una HU no es un documento de requisitos. No es una comunicación escrita entre los proveedores de requisitos y los implementadores de los mismos. Una HU es simplemente “algo que hacer”. Deben estar guiadas por la noción INVEST¹, por lo que deben ser diseñadas bajo los principios de independencia, negociabilidad, validez, estimabilidad, simplicidad y contrastabilidad (JEFFRIES, 2008).

Las HU deben ser escritas en conjunto por el propietario del software y los programadores. En caso de ser escritas por el equipo de desarrollo y no por los clientes, deben ser expresadas en un lenguaje natural. Las

¹Contracción de las siglas de las palabras del inglés independent, negotiable, valuable, estimatable, small, and testable.

HU deben proveer al cliente de algo tangible. Al final de cada iteración se le debe proporcionar una entrega del producto con el compendio de resultados de las HU.

Las HU deben ser confeccionadas de forma tal que para ser completadas el equipo de desarrollo dedicará de una a cinco semanas (BECK, 2000). Pero al ser este un proyecto que se basa en desarrollar una aplicación web, se siguió la recomendación dada por (WALLACE; RAGGETT y AUFANG, 2002), de hasta dos semanas como máximo.

Con el objetivo de determinar que HU son más vitales resolver, según las necesidades del cliente y poder realizar una correcta planificación de la implementación, cada HU es clasificada por el propietario del producto según su prioridad para el negocio en:

- **Alta:** HU que resultan fundamentales en el desarrollo del sistema.
- **Media:** HU que resultan funcionalidades necesarias pero no imprescindibles.
- **Baja:** HU que constituyen funcionalidades que sirven de ayuda al control de elementos asociados al equipo de desarrollo, a la estructura y no tienen nada que ver con el sistema en desarrollo.

También se tiene en cuenta la dificultad y posible existencia de errores durante la implementación de cada HU, por ello el equipo de desarrollo clasifica a cada una según el riesgo en su desarrollo:

- **Alto:** cuando en la implementación de la HU se considera la posible existencia de errores que lleven a la inoperatividad del código.
- **Medio:** cuando pueden aparecer errores en la implementación de la HU que puedan retrasar la entrega de la versión.
- **Bajo:** cuando pueden aparecer errores que serán tratados con relativa facilidad sin que traigan perjuicios para el desarrollo del proyecto.

Las HU son representadas mediante tablas divididas por las siguientes secciones:

- **Número:** esta sección representa el número, incremental en el tiempo, de la historia de usuario que se describe.
- **Nombre de Historia de Usuario:** identifica la HU que se describe entre los desarrolladores y el cliente.
- **Usuario:** rol del usuario que realiza la funcionalidad.
- **Prioridad en negocio:** se le otorga una prioridad (Alta, Media, Baja) a las HU de acuerdo a la necesidad de desarrollo.
- **Riesgo en Desarrollo:** se le otorga una medida de (Alto, Medio, Bajo), a la ocurrencia de errores en el proceso de desarrollo de la HU.
- **Iteración asignada:** número de la iteración donde va a desarrollarse la HU.
- **Puntos Estimados:** es el tiempo estimado en semanas ideales (40 horas) que se demorará el desarrollo de la HU.

- **Descripción:** breve descripción de la HU.
- **Observaciones:** señalamiento o advertencia del sistema.

Debido al análisis realizado al proceso de reproducción de documentos en el DRD, todos los aspectos relacionados con el mismo quedaron totalmente claros. Lo que permitió la elaboración de todas las historias de usuario, en la etapa de exploración. No siendo necesario realizar modificaciones ni adicionar nuevas HU a lo largo de las iteraciones.

A continuación se exponen las HU definidas por el cliente en conjunto con el equipo de desarrollo:

Tabla 3.1. Historia de usuario # 1

Historia de usuario	
Número: 1	Nombre: Autenticar usuario
Usuario: Todos los usuarios	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 0.4	Iteración asignada: 1
Programador responsable: Eblis M. Caro Rivero	
Descripción: La información en el sistema requiere estar protegida, se tiene que establecer una interfaz de autenticación para el acceso a la página principal, en la que podrán autenticarse los usuarios. Se debe solicitar el nombre de usuario y la contraseña para poder acceder. El sistema tiene que establecer niveles de seguridad por cada usuario, dándole los permisos estrictamente necesarios en cada caso solo a las funcionalidades que le corresponden.	
Observaciones: Los usuarios deben estar registrados para poder acceder al sistema. En caso de que alguno de los parámetros requeridos para la autenticación sean incorrectos se mostrará un mensaje que lo indique.	

Tabla 3.2. Historia de usuario # 2

Historia de usuario	
Número: 2	Nombre: Gestionar usuario
Usuario: Supervisor	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 0.6	Iteración asignada: 1
Programador responsable: Eblis M. Caro Rivero	
Descripción: El supervisor podrá gestionar los usuarios presentes en el sistema. La gestión incluye las acciones de crear, eliminar, modificar y ver la información de un usuario. Así como la asignación de un rol al mismo.	
Observaciones: No se le permite al supervisor ver ni modificar la contraseña de otro usuario. Los datos asociados a un usuario son: nombre de usuario, contraseña, primer nombre, segundo nombre, primer apellido, segundo apellido, estado y rol.	

Tabla 3.3. Historia de usuario # 3

Historia de usuario	
Número: 3	Nombre: Gestionar solicitud de servicios generales
Usuario: Supervisor	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Eblis M. Caro Rivero	
Descripción: El supervisor podrá gestionar la información de las solicitudes de los clientes para el consumo de servicios generales. La gestión incluye las acciones de crear, eliminar, modificar, buscar y ver la información de una solicitud pendiente. También podrá asignar la realización de la solicitud a un operario. Automáticamente, después de la asignación se generará una tarea asociada al operario. Al eliminar una solicitud se eliminará también la tarea generada por ella al ser asignada a un operario.	
Observaciones: Los datos necesarios para la creación de una solicitud de servicios generales son los siguientes: modelo a reproducir, tipo de documento, asignatura (en caso de que sea un examen), cantidad de tesis (en caso de que sea impresión de tesis de pregrado), área, fecha, observaciones, nombre del cliente, usuario del cliente, teléfono del cliente y documento a reproducir.	

Tabla 3.4. Historia de usuario # 4

Historia de usuario	
Número: 4	Nombre: Gestionar solicitud de servicios a color
Usuario: Directivo	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Eblis M. Caro Rivero	
Descripción: El directivo podrá gestionar la información de las solicitudes para el consumo de los servicios de impresión a color. La gestión incluye las acciones de crear, eliminar, modificar, buscar y ver la información de una solicitud pendiente. Automáticamente, después de la creación de la solicitud se generará una tarea asignada al técnico. Al eliminar una solicitud se eliminará también la tarea generada por ella.	
Observaciones: Los datos necesarios para la creación de una solicitud de impresión a color son los siguientes: nombre del cliente, nombre de usuario del cliente, teléfono del cliente, tipo de solicitud, fecha, área, estado, observaciones y documento a reproducir.	

Tabla 3.5. Historia de usuario # 5

Historia de usuario	
Número: 5	Nombre: Gestionar tarea de servicios generales
Usuario: Supervisor	
Prioridad en negocio: Alta	Riesgo en desarrollo: Bajo
Puntos estimados: 0.5	Iteración asignada: 2

Continúa en la próxima página

Tabla 3.5. Continuación de la página anterior

Programador responsable: Eblis M. Caro Rivero
Descripción: El supervisor podrá gestionar la información de las tareas generadas a partir de la asignación de solicitudes de los clientes para el consumo de servicios generales a operarios. La gestión incluye las acciones de reasignar, buscar y ver la información de una tarea.
Observaciones: Ninguna.

Tabla 3.6. Historia de usuario # 6

Historia de usuario	
Número: 6	Nombre: Atender tarea de servicios generales
Usuario: Operario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Eblis M. Caro Rivero	
Descripción: El operario podrá atender las tareas de servicios generales asignadas a él. Le será permitido acceder a una vista detallada de la tarea con los elementos necesarios para realizar la tarea, y una vez concluida podrá finalizar la misma en el sistema.	
Observaciones: Un operario solo podrá ver las tareas generadas por solicitudes asignadas a él. Exceptuando la impresión de exámenes, al concluir una tarea se generará automáticamente una notificación dirigida al cliente que solicitó el servicio vía correo electrónico y se actualizarán las cantidades de los recursos materiales empleados para su realización. Para concluir una tarea, es necesario asociarle el equipo en que fue realizada y las cantidades de recursos materiales empleados.	

Tabla 3.7. Historia de usuario # 7

Historia de usuario	
Número: 7	Nombre: Atender tarea de impresión a color
Usuario: Técnico	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Eblis M. Caro Rivero	
Descripción: El técnico podrá atender las tareas de servicios a color. Le será permitido acceder a una vista detallada de la tarea con los elementos necesarios para realizar la tarea, y una vez concluida podrá finalizar la misma en el sistema.	
Observaciones: Para concluir una tarea, es necesario asociarle el equipo en que fue realizada y las cantidades de recursos materiales empleados. Al concluir una tarea se actualizarán las cantidades de los recursos materiales empleados para su realización.	

Tabla 3.8. Historia de usuario # 8

Historia de usuario	
Número: 8	Nombre: Gestionar notificación
Usuario: Supervisor	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 0.5	Iteración asignada: 2
Programador responsable: Eblis M. Caro Rivero	
Descripción: El supervisor podrá editar el contenido de la notificación generada por la conclusión de una tarea originada por una solicitud de servicios generales. Le será posible también decidir si la notificación es enviada vía correo electrónico o es cancelada.	
Observaciones: Para que las notificaciones puedan ser enviadas, se debe colocar en la configuración del sistema, la dirección de correo electrónico designado para el DRD.	

Tabla 3.9. Historia de usuario # 9

Historia de usuario	
Número: 9	Nombre: Gestionar recurso material de servicios generales
Usuario: Supervisor	
Prioridad en negocio: Media	Riesgo en desarrollo: Bajo
Puntos estimados: 0.5	Iteración asignada: 3
Programador responsable: Eblis M. Caro Rivero	
Descripción: El supervisor tendrá la posibilidad de gestionar la información relativa a los recursos materiales empleados en la realización de las solicitudes de servicios generales. Dentro de las acciones de gestión se encuentran adicionar un nuevo recurso material y modificar o eliminar uno existente.	
Observaciones: Ninguna.	

Tabla 3.10. Historia de usuario # 10

Historia de usuario	
Número: 10	Nombre: Gestionar recurso material de impresión a color
Usuario: Técnico	
Prioridad en negocio: Media	Riesgo en desarrollo: Bajo
Puntos estimados: 0.5	Iteración asignada: 3
Programador responsable: Eblis M. Caro Rivero	
Descripción: El técnico podrá gestionar la información relativa a los recursos materiales empleados en la realización de las solicitudes de servicios a color. Dentro de las acciones de gestión se encuentran adicionar un nuevo recurso material y modificar o eliminar uno existente.	
Observaciones: Ninguna.	

Tabla 3.11. Historia de usuario # 11

Historia de usuario	
Número: 11	Nombre: Elaborar parte
Usuario: Directivo, Supervisor	
Prioridad en negocio: Media	Riesgo en desarrollo: Bajo
Puntos estimados: 0.2	Iteración asignada: 3
Programador responsable: Eblis M. Caro Rivero	
Descripción: El sistema le permitirá a los usuarios directivo y supervisor generar un parte del estado actual del proceso de reproducción de documentos en el DRD. En dicho parte estarán presentes todos los detalles del flujo de trabajo de dicho proceso.	
Observaciones: Los detalles a incluir en el parte son: cantidad de solicitudes realizadas, cantidad de tareas pendientes, cantidad de notificaciones realizadas y cantidad de recursos disponibles.	

Tabla 3.12. Historia de usuario # 12

Historia de usuario	
Número: 12	Nombre: Elaborar reporte
Usuario: Directivo, Supervisor	
Prioridad en negocio: Media	Riesgo en desarrollo: Alto
Puntos estimados: 2	Iteración asignada: 3
Programador responsable: Eblis M. Caro Rivero	
Descripción: El sistema le permitirá al directivo generar reportes a partir de la información estadística e histórica almacenada en el mismo sobre el proceso de reproducción de documentos en el DRD.	
Observaciones: Los reportes tendrán una tabla con los datos y un gráfico que permita un mayor nivel de interpretación. Los intervalos temporales para cada reporte son semanal, mensual y anual. Los tipos de reporte son servicios prestados, recursos consumidos y tipos de documentos reproducidos.	

Tabla 3.13. Historia de usuario # 13

Historia de usuario	
Número: 13	Nombre: Editar perfil de usuario
Usuario: todos los usuarios (supervisor, directivo, técnico, operario)	
Prioridad en negocio: Baja	Riesgo en desarrollo: Medio
Puntos estimados: 0.2	Iteración asignada: 4
Programador responsable: Eblis M. Caro Rivero	
Descripción: Cada usuario podrá editar la información asociada a su perfil. Los cambios incluyen la modificación de la contraseña.	
Observaciones: La vía de acceso a esta funcionalidad debe estar dentro de las opciones de usuario, que deben aparecer agrupadas en un ícono emergente dentro de el menú de acciones del banner.	

Tabla 3.14. Historia de usuario # 14

Historia de usuario	
Número: 14	Nombre: Alertas del sistema
Usuario: todos los usuarios (supervisor, directivo, técnico, operario)	
Prioridad en negocio: Baja	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 4
Programador responsable: Eblis M. Caro Rivero	
Descripción: El sistema contará con un mecanismo que le permita notificar a cada usuario informaciones de su interés. Todo el flujo de elementos dentro del proceso de reproducción de documentos será notificado a los involucrados.	
Observaciones: Las alertas del sistema deben aparecer en la página de inicio del sistema.	

Tabla 3.15. Historia de usuario # 15

Historia de usuario	
Número: 15	Nombre: Configurar opciones avanzadas
Usuario: Supervisor	
Prioridad en negocio: Baja	Riesgo en desarrollo: Medio
Puntos estimados: 0.4	Iteración asignada: 4
Programador responsable: Eblis M. Caro Rivero	
Descripción: El supervisor podrá configurar parámetros del sistema como el correo de salida y el mensaje por defecto de las notificaciones.	
Observaciones: Ninguna.	

Tabla 3.16. Historia de usuario # 16

Historia de usuario	
Número: 16	Nombre: Gestionar equipo de servicios generales
Usuario: Supervisor	
Prioridad en negocio: Baja	Riesgo en desarrollo: Medio
Puntos estimados: 0.2	Iteración asignada: 4
Programador responsable: Eblis M. Caro Rivero	
Descripción: El supervisor podrá gestionar la información relativa a los equipos empleados en la realización de las solicitudes de servicios generales. Dentro de las acciones de gestión se encuentran adicionar un nuevo recurso material y modificar o eliminar uno existente.	
Observaciones: Ninguna.	

Tabla 3.17. Historia de usuario # 17

Historia de usuario	
Número: 17	Nombre: Gestionar equipo de servicios a color
Usuario: Técnico	
Prioridad en negocio: Baja	Riesgo en desarrollo: Medio
Puntos estimados: 0.6	Iteración asignada: 4
Programador responsable: Eblis M. Caro Rivero	
Descripción: El técnico podrá gestionar la información relativa a los equipos empleados en la realización de las solicitudes de servicios a color. Dentro de las acciones de gestión se encuentran adicionar un nuevo recurso material y modificar o eliminar uno existente.	
Observaciones: Ninguna.	

3.3. Planificación

En esta fase el cliente establece la prioridad de cada historia de usuario y los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente para la realización del resto de las versiones. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días (CANÓS; LETELIER y PENADÉS, 2011).

La planificación se puede realizar basándose en el tiempo o el alcance. La velocidad del proyecto es utilizada para establecer cuántas historias se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de historias. Al planificar por tiempo, se multiplica el número de iteraciones por la velocidad del proyecto, determinándose cuántos puntos se pueden completar. Al planificar según el alcance del sistema, se divide la suma de puntos de las historias de usuario seleccionadas entre la velocidad del proyecto, obteniendo el número de iteraciones necesarias para su implementación (ibíd.). La planificación en este caso fue realizada basándose en el alcance del sistema.

3.3.1. Estimación del esfuerzo

Con el fin de conocer cuánto le tomará al equipo de desarrollo elaborar todas las historias de usuario y conocer cuántos puntos debe desarrollar, se realiza la estimación del esfuerzo por HU. Teniendo en cuenta que la velocidad del proyecto no es un elemento que queda claro hasta transcurrida varias iteraciones, se estimó de manera tal que inicialmente consta de aproximadamente 3 puntos por iteración, se agruparon las 17 HU en cuatro iteraciones. La fijación de la cantidad de iteraciones se estableció a partir de dos indicadores básicos, la experiencia del equipo de desarrollo (analogía) y el ajuste permitido por la planificación guiada por el alcance del sistema (cantidad total de puntos/velocidad del proyecto). En la Tabla 3.18 queda reflejada la estimación del esfuerzo realizada.

Tabla 3.18. Estimación de esfuerzo por historia de usuario

Iteración	Historias de usuario		Puntos estimados (semanas)
1	1	Autenticar usuario	0.4
	2	Gestionar usuario	0.6
	3	Gestionar solicitud de servicios generales	1
	4	Gestionar solicitud de impresión a color	1
2	5	Gestionar tareas de servicios generales	0.5
	6	Atender tarea de servicios generales	1
	7	Atender tarea de servicios a color	1
	8	Gestionar notificaciones	0.5
3	9	Gestionar recurso material de servicios generales	0.5
	10	Gestionar recurso material de impresión a color	0.5
	11	Elaborar partes	0.2
	12	Elaborar reportes	2
4	13	Editar perfil de usuario	0.2
	14	Alertas del sistema	1
	15	Configurar opciones avanzadas	0.4
	16	Gestionar equipo de servicios generales	0.2
	17	Gestionar equipo de servicios a color	0.6
Total			11.6

3.3.2. Plan de iteraciones

Después de identificadas y descritas las HU y estimar el esfuerzo dedicado a la realización de cada una de ellas, en XP se procede a generar el artefacto plan de iteraciones. El mismo muestra cuáles son las HU que serán implementadas en cada iteración para mejorar el desempeño del equipo de desarrollo. Al final de la última iteración el sistema estará listo para entrar en producción.

Se procede entonces a planificar la etapa de implementación. En la cual las HU serán desarrolladas y probadas en un ciclo iterativo, transcribiéndose en TI. Para un desempeño organizado el equipo establece un total de cuatro iteraciones. El plan de duración de las iteraciones se ve reflejado en la Tabla 3.19.

Iteración 1.

En esta iteración se implementarán HU de nivel de prioridad alta que estén relacionadas con la primera etapa del proceso de reproducción de documentos, creando con ella la línea base de la arquitectura del sistema. Además, se obtendrá una primera versión del producto en la que el cliente podrá probar algunas de las funcionalidades solicitadas como Autenticar usuario, Gestionar usuario, Gestionar solicitud de servicios generales y Gestionar solicitud de impresión a color descritas en las HU 1, 2, 3 y 4.

Iteración 2.

En esta iteración se implementarán el resto de las HU con prioridad alta en el negocio y se corregirán errores o no conformidades de los usuarios con los componentes implementados en la iteración anterior. Como resultado se obtendrá una versión más completa sobre las funcionalidades del sistema. Al culminar esta iteración los usuarios tendrán acceso entonces a funcionalidades que cubren hasta la etapa de notificaciones.

Iteración 3.

Esta iteración es la dedicada a la implementación HU de nivel de prioridad media. En esta iteración se implementarán las HU 9, 10, 11 y 12. Se corregirán errores o no conformidades de los usuarios con las funcionalidades desarrolladas en la iteración anterior. Como resultado se obtendrá una versión del sistema que cubre hasta la etapa de control directivo.

Iteración 4.

En esta iteración se implementarán HU de baja prioridad, completando el desarrollo de la aplicación. Se obtendrá la versión 1.0 del sistema para la gestión de información del Departamento de Reproducción de Documentos de la Universidad de las Ciencias Informáticas “DRDuci” donde los usuarios podrán ejecutar todas las funcionalidades y explotar el sistema a plenitud.

Tabla 3.19. Plan de duración de las iteraciones

Iteración	Historias de usuario		Duración (semanas)
1	1	Autenticar usuario	3
	2	Gestionar usuario	
	3	Gestionar solicitud de servicios generales	
	4	Gestionar solicitud de impresión a color	
2	5	Gestionar tareas de servicios generales	3.0
	6	Atender tarea de servicios generales	
	7	Atender tarea de servicios a color	
	8	Gestionar notificaciones	
3	9	Gestionar recurso material de servicios generales	3.2
	10	Gestionar recurso material de impresión a color	
	11	Elaborar partes	
	12	Elaborar reportes	
4	13	Editar perfil de usuario	2.4
	14	Alertas del sistema	
	15	Configurar opciones avanzadas	
	16	Gestionar equipo de servicios generales	
	17	Gestionar equipo de servicios a color	
Total			11.6

3.3.3. Plan de Entregas

Con el objetivo de mantener al cliente enfocado en mejorar el producto, y hacer que sea él quien realice sus pruebas de campo, se realizan varias entregas (*release*) del sistema. A continuación una tabla que refleja lo comentado, especificando la fecha en la que se realiza el encuentro entre el grupo de desarrollo y los propietarios del software para la liberación de las versiones del producto.

Iteración	Fecha de entrega
Iteración 1	23 de febrero de 2015
Iteración 2	17 de marzo de 2015
Iteración 3	13 de abril de 2015
Iteración 4	1 de mayo de 2015

3.4. Diseño

Durante el diseño de la solución, la máxima simplicidad posible es la clave para el éxito de XP. Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto. La complejidad innecesaria y el código extra deben ser evitados en todo momento. El diseño adecuado para el software es aquel que supera con éxito todas las pruebas, no tiene lógica duplicada, refleja claramente la intención de implementación de los programadores y tiene el menor número posible de clases y métodos. Se debe tener en cuenta que un diseño complejo tarda más en desarrollarse que uno simple, y que siempre es más fácil añadir complejidad a un diseño simple que reducirla de uno complejo.

XP construye un proceso de diseño evolutivo que se basa en refactorizar un sistema simple en cada iteración. Todo el diseño se centra en la iteración actual y no se hace nada anticipadamente para necesidades futuras. El resultado es un proceso de diseño disciplinado, que combina la disciplina con la adaptabilidad de una manera que la hace una de las más desarrolladas entre todas las metodologías ágiles (CANÓS; LETELIER y PENADÉS, 2011).

3.4.1. Patrón arquitectónico

Un patrón arquitectónico define la estructura básica de una aplicación, provee un conjunto de subsistemas predefinidos, incluyendo reglas, lineamientos para conectarlos y pautas para su organización. Constituye una plantilla de construcción.

Los patrones arquitectónicos expresan una organización estructural para un sistema, permiten estructurar los componentes y subsistemas del mismo. La abstracción más alta en cuanto al uso de patrones, se obtiene a través de patrones arquitectónicos.

Modelo-Vista-Controlador

El patrón Modelo-Vista-Controlador (MVC) según su nivel de abstracción se puede clasificar como un patrón de arquitectura. Este tipo de patrón especifica una serie de subsistemas y sus responsabilidades res-

pectivas e incluye las reglas y criterios para organizar las relaciones existentes entre ellos (LLEONART; GARCÍA y ROVIRA, 2010).

EL patrón MVC está acorde con la concepción inicial del sistema, este ayudará a obtener una estructura bien definida y se logrará que el proyecto esté organizado.

Este patrón está formado por tres niveles (VELOSO, 2008):

- El Modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- La Vista transforma el modelo en una página web que permite al usuario interactuar con ella.
- El Controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

Como ya se había planteado, para el desarrollo de la aplicación se empleará el *framework* Django, el cual a pesar de que hace uso intensivo del patrón arquitectónico MVC, realiza algunas modificaciones a la forma de llamar sus elementos y respectivas funciones y establece su propia filosofía para interpretarlo (ALCHIN, 2012). Django toma al modelo tal y como es, pero la forma de ver la vista y el controlador varía de cierta manera. Los desarrolladores de Django identifican en su arquitectura los mismos beneficios que en el MVC, pero orientados a lograr un mayor énfasis hacia el desarrollo ágil, y después de tener un conjunto de herramientas informáticas que forman un *framework* coherente y dan sentido a su flujo de trabajo, culminaron sus esfuerzos con algo a lo que han denominado patrón Modelo-Plantilla-Vista o *Model-Template-View* (MTV), por sus siglas en inglés.

Una vista en MTV, comparte el nombre y algunos elementos con la definición original del MVC de vista, sin embargo tiene más en común con el controlador, puesto que se encarga de realizar las tareas de la lógica, combinando algunas responsabilidades de las vistas tradicionales con la integridad de las tareas del controlador. Una vista acepta las entradas del usuario (incluyendo una simple petición de información), las manipula en concordancia con la lógica de la aplicación y muestra un retorno accesible para que los usuarios puedan ver la información representada por modelos. Las vistas son definidas normalmente como funciones estándar de Python que son llamadas cuando un usuario requiere una url específica. En términos de la web, incluso la más pequeña petición de información es considerada una acción, por eso las vistas son manipuladas de forma tal que permitan modificaciones de los datos a enviar en torno a la petición. Las vistas tienen acceso a los modelos, consultando y modificando la información necesaria para cumplir la tarea requerida por el usuario (ibíd.).

Por otra parte, una plantilla, es la encargada de presentar los datos al usuario, la tarea de como se presentan los mismos es tarea de la plantilla. Las plantillas en Django tienen la importancia suficiente para considerarlas una parte esencial del *framework* y una capa independiente que interactúa con el resto de las capas del patrón. Django provee a las plantillas de un lenguaje simple por encima del HTML para cumplir su propósito de manejar los detalles de como es mostrada la información a los usuarios, pudiendo incluso aplicar lógica en la presentación desde el servidor (ibíd.).

3.4.2. Patrones de diseño

Un patrón de diseño puede ser caracterizado como una regla de tres partes, la cual expresa una relación entre cierto contexto, un problema y una solución. Para el diseño de software, el contexto permite a quien usa el patrón entender el entorno en el cual reside el problema y qué solución pudiera ser apropiada dentro de ese entorno. Un conjunto de requisitos, incluyendo limitaciones y restricciones, actúan como un sistema de fuerzas que influencia en cómo el problema puede ser interpretado dentro de su contexto y cómo la solución puede ser aplicada efectivamente (PRESSMAN, 2003).

Los patrones de diseño hacen que sea más fácil reutilizar buenos diseños y arquitecturas. Los patrones de diseño ayudan a elegir las alternativas de diseño que hacen que un sistema sea reutilizable, y evitan aquellas que dificultan dicha reutilización. Un patrón de diseño nombra, abstrae e identifica los aspectos clave de una estructura de diseño común, lo que los hace útiles para crear un diseño orientado a objetos reutilizable. El patrón de diseño identifica las clases e instancias participantes, sus roles y colaboraciones, y la distribución de responsabilidades. En definitiva, los patrones de diseño ayudan a un diseñador a lograr rápidamente un buen diseño (GAMMA; HELM; JOHNSON y VLISSIDES, 2002).

Patrones GRASP

Los patrones *General Responsibility Assignment Software Patterns* (en español Patrones Generales de Software para la Asignación de Responsabilidades) (GRASP) son parejas de problema solución con un nombre, que codifican buenos principios relacionados frecuentemente con la asignación de responsabilidades. Estos describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones, garantizando la máxima capacidad de reutilización del código (VICONTI, 2011).

A continuación se reflejan los patrones GRASP empleados en el diseño del sistema:

Patrón Experto: Se mantiene el encapsulamiento, los objetos utilizan su propia información para llevar a cabo sus tareas. Se distribuye el comportamiento entre las clases que contienen la información requerida. Son más fáciles de entender y mantener.

Patrón Creador: Las clases que tienen la responsabilidad de crear objetos contienen toda la información necesaria para construir los mismos. Se pone de manifiesto al utilizar las *class-based-views*² de Django.

Patrón Alta Cohesión: Se aplica en la mayoría las clases del diseño, ya que en cada una solo se implementan las funcionalidades que le corresponden. El patrón alta cohesión se puede evidenciar en las plantillas encargadas de mostrar los datos al usuario y en las clases manejadoras de los formularios.

Patrón Bajo Acoplamiento

Cada clase se comunica con un número mínimo de clases posible. El patrón bajo acoplamiento se pone de manifiesto en la relación presente entre las clases del sistema, al establecer un número mínimo de depen-

²Las *class-based-views* son vistas que permiten realizar de forma sencilla, las funcionalidades básicas en casi todos los sistemas (Crear, Actualizar, Eliminar, Listar)

dencias entre estas, un ejemplo de ellas pueden ser las de los formularios, vistas, modelos y plantillas donde para cambiar el contenido presente en la plantilla basta con editar el contenido del formulario sin necesidad de cambiar el contenido del resto de las otras clases.

Patrón Controlador: Está representado por una clase a la cual se le asigna la responsabilidad de las operaciones del sistema, ofrece también una guía para tomar decisiones apropiadas. Este patrón se refleja en la clase *views* que es la encargada de controlar las acciones que realiza el usuario con la interfaz de la aplicación y dar respuesta a las peticiones realizadas.

Patrones GoF

Los patrones *Gang of Four* (en español Banda de los cuatro) (GoF) surgen como una forma indispensable de enfrentar los problemas propios de la codificación. Fueron dados a conocer en el libro “*Design Patterns—Elements of Reusable Software*” de Erich Gamma, Richard Helm, Ralph Jonson y John Vlissides, a partir de entonces estos patrones son conocidos como los patrones de la pandilla de los cuatro (GoF, *gang of four*). El empleo de los mismos garantiza una solución efectiva y a prueba de fallos para cada problema en cuestión, que no puede ser solucionado por las vías normales (GAMMA; HELM; JOHNSON y VLISSIDES, 2002).

A continuación se reflejan los patrones GoF emplados en el diseño del sistema:

Patrón *Front Controller* (Controlador Frontal): Django posee una implementación de Controlador Frontal que despacha las peticiones hacia métodos o clases, que en la práctica son páginas controladoras. Antes del despacho, la petición es procesada por varios filtros (*middlewares*).

Patrón *Decorator* (Decorador): Añade responsabilidades adicionales a un objeto dinámicamente, proporcionando una alternativa flexible a la especialización mediante herencia, cuando se trata de añadir funcionalidades.

Patrón *Mediator* (Mediador): Encargado de manejar la interacción entre los diferentes subsistemas. Realiza el mapeo objeto–relacional (ORM) a cargo del motor de Django. Simplifica y facilita los resultados de las consultas realizadas a la base de datos, así como brinda varias funcionalidades adicionales, que consultas SQL tradicionales no proveen.

3.4.3. Tarjetas CRC

Para el diseño de las aplicaciones, la metodología XP no requiere la representación del sistema mediante diagramas de clases utilizando notación Lenguaje Unificado de Modelado (UML) por sus siglas del inglés *Unified Modeling Language*. En su lugar se usan otras técnicas como las tarjetas CRC como una extensión informal de UML.

La técnica de las tarjetas CRC se puede usar para estructurar el sistema a través de análisis guiados por la responsabilidad. Las clases se examinan, se filtran y se refinan sobre la base de sus responsabilidades con respecto al sistema, y las clases con las que necesitan colaborar para completar sus responsabilidades

(WALLACE; RAGGETT y AUFANG, 2002).

La elaboración de las tarjetas CRC es una técnica orientada a establecer un planteo inicial de los objetos que participan en el sistema. El producto de la aplicación de esta técnica, es un conjunto de tarjetas asociadas a las clases, en las que se especifican las responsabilidades y sus colaboraciones con otras clases.

Las tarjetas CRC identifican y organizan las clases bajo el paradigma orientado a objetos, lo que incluye asignación de responsabilidades. Cada tarjeta contiene el nombre de la clase (que representa una o más historias de usuario), una descripción de las responsabilidades o métodos asociados con la clase, así como la lista de las clases con que se relaciona o que colaboran con ella. Las tarjetas CRC son el único trabajo de diseño que se genera como parte del proceso de XP (HOLMES y T. JOYCE, 2000).

Las responsabilidades son sentencias de alto nivel acerca de las acciones que realiza un objeto como también del conocimiento que él mantiene y provee. Los objetos no están aislados sino que colaboran unos con otros. Una colaboración es una solicitud hecha por un objeto a otro. Para identificarlas, se debe examinar cada par clase-responsabilidad con el fin de determinar si es necesario que la clase interactúe con otras para llevar a cabo esa responsabilidad (FRESNO; MAUCO; RIDAO; DOORN y RIVERO, 2007).

Ya que el empleo del *framework* de desarrollo web Django establece una estructura propia de las clases del sistema, la técnica de las tarjetas CRC solo tuvo que ser empleada para la elaboración de las clases entidades del modelo de datos del sistema. Las tarjetas CRC elaboradas son las siguientes:

Tabla 3.20. Tarjeta CRC # 1

Tarjeta CRC	
Clase: Solicitud	
Responsabilidad	Colaboración
Almacenar la información referente a las solicitudes realizadas por los clientes.	django.db.models PolymorphicModel

Tabla 3.21. Tarjeta CRC # 2

Tarjeta CRC	
Clase: Impresion	
Responsabilidad	Colaboración
Almacenar la información referente a las solicitudes del servicio Impresión.	django.db.models Solicitud PolymorphicModel

Tabla 3.22. Tarjeta CRC # 3

Tarjeta CRC	
Clase: Pregrado	
Responsabilidad	Colaboración
Almacenar la información referente a las solicitudes de Impresión de tesis de pregrado realizadas por los clientes.	django.db.models Solicitud PolymorphicModel

Tabla 3.23. Tarjeta CRC # 4

Tarjeta CRC	
Clase: Examen	
Responsabilidad	Colaboración
Almacenar la información referente a las solicitudes de impresión de exámenes realizadas por los clientes.	django.db.models Solicitud PolymorphicModel

Tabla 3.24. Tarjeta CRC # 5

Tarjeta CRC	
Clase: Otros	
Responsabilidad	Colaboración
Almacenar la información referente a las solicitudes de impresión de otros tipos de documentos tales como documentos.	django.db.models Solicitud PolymorphicModel

Tabla 3.25. Tarjeta CRC # 6

Tarjeta CRC	
Clase: Escaneo	
Responsabilidad	Colaboración
Almacenar la información referente a las solicitudes del servicio escaneo.	django.db.models PolymorphicModel

Tabla 3.26. Tarjeta CRC # 7

Tarjeta CRC	
Clase: Fotocopia	
Responsabilidad	Colaboración
Almacenar la información referente a las solicitudes del servicio fotocopia.	django.db.models PolymorphicModel

Tabla 3.27. Tarjeta CRC # 8

Tarjeta CRC	
Clase: Equipo	
Responsabilidad	Colaboración
Almacenar la información referente a los equipos empleados en el DRD de la UCI.	django.db.models

Tabla 3.28. Tarjeta CRC # 9

Tarjeta CRC	
Clase: RecursoMaterial	
Responsabilidad	Colaboración
Almacenar la información referente a los recursos materiales empleados en el DRD de la UCI.	django.db.models

Tabla 3.29. Tarjeta CRC # 10

Tarjeta CRC	
Clase: Tarea	
Responsabilidad	Colaboración
Almacenar la información referente a las tareas de ejecución de las solicitudes registradas.	django.db.models Solicitud User Equipo

Tabla 3.30. Tarjeta CRC # 11

Tarjeta CRC	
Clase: Notificacion	
Responsabilidad	Colaboración
Almacenar la información referente a las notificaciones que se generan a partir de la culminación de una tarea.	django.db.models Solicitud Tarea

Tabla 3.31. Tarjeta CRC # 12

Tarjeta CRC	
Clase: forms.py	
Responsabilidad	Colaboración
Generar formularios para el registro y actualización de clases entidades.	User Solicitud Tarea Notificación Equipo RecursoMaterial

3.4.4. Modelo de datos

Una vez diseñadas las tarjetas CRC, y definidas las clases entidades, se diseña el modelo físico de la base de datos que empleará el sistema para almacenar la información.

El modelo de datos permite describir las principales clases que componen la capa del modelo del sistema, el tipo de dato de los atributos que contendrá la base de datos y la forma en que se relacionan las entidades. Permite describir los elementos de la realidad que intervienen en un problema dado y la forma en que estos se relacionan.

A continuación el diagrama entidad-relación que representa el modelo físico de datos diseñado para el sistema.

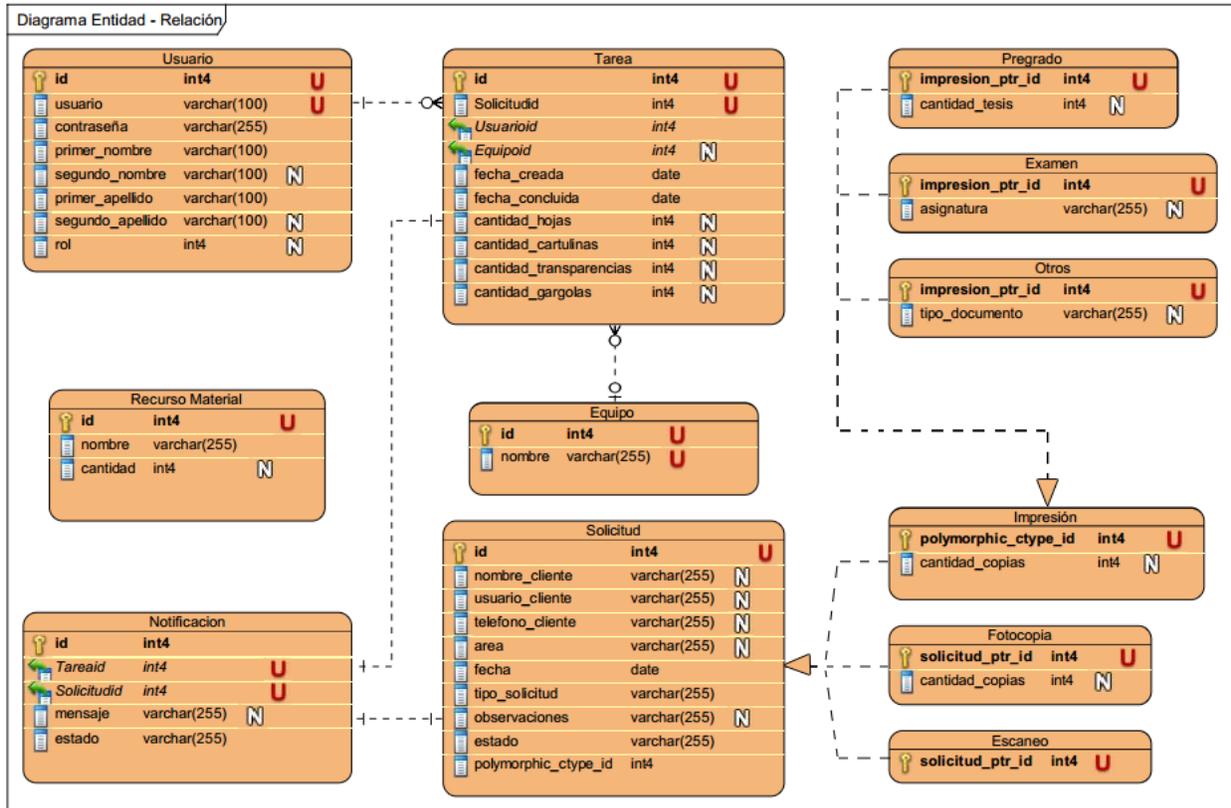


Figura 3.1. Diagrama Entidad-Relación usado para representar el modelo de datos.

3.5. Conclusiones del capítulo

En este capítulo se describieron las fases de exploración, planificación y diseño propuestas por la metodología XP. En la fase de exploración se generaron las HU que caracterizan al sistema. En la fase de planificación se realizó la estimación del esfuerzo necesario para la implementación de las HU. Se construyó además el plan de iteraciones, que permitió conocer las historias de usuario a implementar en cada una de las iteraciones, así como el orden de prioridad de las mismas durante su desarrollo. Se acordó con los propietarios del software un plan de entregas que establece la liberación de versiones del sistema después de cada iteración y fija la fecha en que serán entregadas. En la etapa de diseño se establecieron los elementos de la estructura del sistema. Se puntualizó sobre el empleo del *framework* Django y sobre su implementación del patrón arquitectónico MTV. El empleo de patrones generales de software para la asignación de responsabilidades garantizó una solución que tiene como premisa la reutilización de código y la solución a problemas dentro de contextos similares en el desarrollo de software. Se elaboraron las tarjetas CRC, las cuales permitieron definir las clases, las relaciones entre ellas a través de sus colaboradores y las responsabilidades de las mismas, lo que posibilita una reducción del acoplamiento y aumento de la reutilización en la herramienta. Este capítulo permitió dar cumplimiento al tercer objetivo específico de la investigación.

4.1. Introducción

Culminada la etapa de diseño, que sienta las bases estructurales del sistema, se inicia la fase de implementación y pruebas. En esta etapa se dividen las HU en TI para facilitar su comprensión e implementación, se codifican las pruebas unitarias, se implementan dichas TI y se ejecutan las pruebas de aceptación. La etapa de implementación y pruebas constituye el período con más resultados tangibles y sustanciales, pues la misma culmina con un sistema concluido y validado por el cliente.

En este capítulo se tratan aspectos tales como los estándares de codificación que rigen el código generado, las iteraciones realizadas para la implementación de las HU, las pruebas aplicadas y sus resultados.

4.2. Estándares de codificación

Los estándares de codificación comprenden todos los aspectos de generación de código en un proyecto y permiten entender de manera rápida, fácil y sencilla el código empleado en el desarrollo de un software. Garantizan un mantenimiento óptimo del código por parte de los programadores, independientemente de si son los creadores del mismo. Reflejan un estilo armonioso como si un único programador hubiera escrito todo el código de una sola vez. La confección de estos estándares debe ser definida al comienzo de la implementación para garantizar que todos los programadores trabajen de manera coordinada.

Para lograr este objetivo se utilizó la Guía de estilo para el código Python (PEP 8). Esta guía posee una gran cantidad de convenciones para escribir código legible, dentro de las cuales se destacan (VAN ROSSUM; WARSAW y COGHLAN, 2001):

- Usar cuatro espacios por indentación.
- Nunca mezclar tabulaciones y espacios.
- Limitar todas las líneas a un máximo de caracteres (120 en este proyecto).
- Separar funciones de alto nivel y definiciones de clase con dos líneas en blanco, mientras que las definiciones de métodos dentro de una clase son separadas por una línea en blanco.

- Codificación UTF-8 en todos los módulos.
- Las importaciones deben estar en líneas separadas.
- Evitar usar espacios en blanco innecesarios.
- Utilizar el estilo *CamelCase* para nombrar clases, y el *lower_case_with_underscores* para funciones y métodos.

4.3. Desarrollo por iteraciones

En la fase de implementación y pruebas se realizan las iteraciones planificadas. Durante el transcurso de las iteraciones se realiza la implementación de las historias de usuario seleccionadas para cada una de estas. Al inicio de las mismas, se lleva a cabo una revisión del plan de iteraciones y se modifica de ser necesario. Como parte de este plan, se descomponen las HU en TI o tareas de desarrollo. Estas tareas son para el uso de los programadores, pueden escribirse utilizando un lenguaje técnico y no necesariamente deben ser entendibles para el cliente (BECK, 2000).

Antes de implementar las tareas, se elaboran los test unitarios para cada una de ellas, los cuales prueban la funcionalidad y detallan los fallos de la misma. Una vez culminada la iteración se realiza la entrega de una versión del producto al propietario del software, y se ejecutan las pruebas de aceptación, validando la satisfacción del cliente con el sistema. Ajustándose a la planificación realizada, se llevaron a cabo cuatro iteraciones de desarrollo sobre el sistema, obteniéndose al finalizar un producto listo para su despliegue. A continuación se detallan cada una de las iteraciones.

4.3.1. Iteración 1

Durante esta iteración se abordaron las HU de mayor prioridad y se construyó la base de la arquitectura del sistema con el fin de obtener un producto con las funcionalidades críticas para ser mostrado al cliente y obtener una rápida retroalimentación de este. Se implementaron las HU: Autenticar usuario, Gestionar usuario, Gestionar solicitud de servicios generales y Gestionar solicitud de servicios a color. Para su implementación las mismas fueron divididas en TI.

HU1: Autenticar usuario

Tabla 4.1. Tarea de ingeniería # 1

Tarea	
Número de tarea: 1	Número de Historia de usuario: 1
Nombre de la tarea: Autenticar usuario	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha de inicio: 2 de febrero de 2015	Fecha de fin: 3 de febrero de 2015
Programador responsable: Eblis Manuel Caro Rivero	

Continúa en la próxima página

Tabla 4.1. Continuación de la página anterior

Descripción: Crear plantilla y vista para autenticar usuarios, establecer como función hash para las contraseñas el md5, manejar mensajes en caso satisfactorio o de autenticación fallida.
--

HU2: Gestionar usuario

Tabla 4.2. Tarea de ingeniería # 2

Tarea	
Número de tarea: 2	Número de Historia de usuario: 2
Nombre de la tarea: Listar usuarios	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha de inicio: 4 de febrero de 2015	Fecha de fin: 4 de febrero de 2015
Programador responsable: Eblis Manuel Caro Rivero	
Descripción: Crear plantilla, vista y url para listar usuarios, mostrar en un <i>datatable</i> que permita opciones de búsqueda sobre los campos mostrados, paginación y ordenamiento por las cabeceras de las columnas, manejar mensajes en caso de que no existan usuarios registrados.	

Tabla 4.3. Tarea de ingeniería # 3

Tarea	
Número de tarea: 3	Número de Historia de usuario: 2
Nombre de la tarea: Adicionar usuario	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha de inicio: 5 de febrero de 2015	Fecha de fin: 5 de febrero de 2015
Programador responsable: Eblis Manuel Caro Rivero	
Descripción: Crear plantilla, vista, formulario y url para adicionar un nuevo usuario, mostrar los campos del formulario de registro en forma de párrafo, manejar mensajes en caso de registro satisfactorio o de formulario no válido, redireccionar a la url listar usuarios en caso satisfactorio.	

Tabla 4.4. Tarea de ingeniería # 4

Tarea	
Número de tarea: 4	Número de Historia de usuario: 2
Nombre de la tarea: Editar usuario	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha de inicio: 6 de febrero de 2015	Fecha de fin: 6 de febrero de 2015
Programador responsable: Eblis Manuel Caro Rivero	

Continúa en la próxima página

Tabla 4.4. Continuación de la página anterior

Descripción: Crear plantilla, vista, formulario y url para editar un usuario registrado, mostrar los campos del formulario de actualización en forma de párrafo, manejar mensajes en caso de actualización de campos satisfactoria o de formulario no válido, redireccionar a la url listar usuarios en caso satisfactorio.
--

Tabla 4.5. Tarea de ingeniería # 5

Tarea	
Número de tarea: 5	Número de Historia de usuario: 2
Nombre de la tarea: Eliminar usuario	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha de inicio: 6 de febrero de 2015	Fecha de fin: 6 de febrero de 2015
Programador responsable: Eblis Manuel Caro Rivero	
Descripción: Crear plantilla, vista y url para eliminar un usuario registrado, mostrar confirmación en forma de <i>modal</i> , manejar mensajes en caso de eliminado satisfactorio, redireccionar a la url listar usuarios en caso satisfactorio.	

HU3: Gestionar solicitud de servicios generales

Tabla 4.6. Tarea de ingeniería # 6

Tarea	
Número de tarea: 6	Número de Historia de usuario: 3
Nombre de la tarea: Registrar solicitud de servicios generales	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 9 de febrero de 2015	Fecha de fin: 11 de febrero de 2015
Programador responsable: Eblis Manuel Caro Rivero	
Descripción: Crear plantillas, vistas, formularios y urls para registrar solicitudes de servicios generales de tipo impresión de tesis de pregrado, impresión de examen, impresión de otros tipos de documentos, escaneo y fotocopia, manejar mensajes en caso de registro satisfactorio o formulario no válido, redireccionar a la url listar solicitudes pendientes de servicios generales en caso satisfactorio.	

Tabla 4.7. Tarea de ingeniería # 7

Tarea	
Número de tarea: 7	Número de Historia de usuario: 3
Nombre de la tarea: Listar todas las solicitudes de servicios generales	
Tipo de tarea: Desarrollo	Puntos estimados: 0.3
Fecha de inicio: 11 de febrero de 2015	Fecha de fin: 12 de febrero de 2015

Continúa en la próxima página

Tabla 4.7. Continuación de la página anterior

Programador responsable: Eblis Manuel Caro Rivero
Descripción: Crear plantilla, vista y url para listar todas las solicitudes de servicios generales registradas en el sistema, mostrar en un <i>datatable</i> que permita opciones de búsqueda sobre los campos mostrados, paginación y ordenamiento por las cabeceras de las columnas, manejar mensajes en caso de que no existan solicitudes de este tipo registradas. Crear una columna en la tabla con las entradas a las acciones de ver detalles, editar, asignar y eliminar. Realizar cada una de estas acciones con sus respectivas vistas, plantillas, formularios y urls.

Tabla 4.8. Tarea de ingeniería # 8

Tarea	
Número de tarea: 8	Número de Historia de usuario: 3
Nombre de la tarea: Listar todas las solicitudes de servicios generales pendientes	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha de inicio: 12 de febrero de 2015	Fecha de fin: 13 de febrero de 2015
Programador responsable: Eblis Manuel Caro Rivero	
Descripción: Crear plantilla, vista y url para listar todas las solicitudes de servicios generales pendientes en el sistema, mostrar en un <i>datatable</i> que permita opciones de búsqueda sobre los campos mostrados, paginación y ordenamiento por las cabeceras de las columnas, manejar mensajes en caso de que no existan solicitudes de este tipo pendientes. Crear una columna en la tabla con las entradas a las acciones de ver detalles, editar, asignar y eliminar. Realizar cada una de estas acciones con sus respectivas vistas, plantillas, formularios y urls.	

HU4: Gestionar solicitud de servicios a color

Tabla 4.9. Tarea de ingeniería # 9

Tarea	
Número de tarea: 9	Número de Historia de usuario: 4
Nombre de la tarea: Registrar solicitud de servicios a color	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 16 de febrero de 2015	Fecha de fin: 18 de febrero de 2015
Programador responsable: Eblis Manuel Caro Rivero	
Descripción: Crear plantillas, vistas, formularios y urls para registrar solicitudes de servicios a color de tipo impresión, escaneo y fotocopia, manejar mensajes en caso de registro satisfactorio o formulario no válido, redireccionar a la url listar solicitudes pendientes de servicios generales en caso satisfactorio.	

Tabla 4.10. Tarea de ingeniería # 10

Tarea	
Número de tarea: 10	Número de Historia de usuario: 4
Nombre de la tarea: Listar todas las solicitudes de servicios a color	
Tipo de tarea: Desarrollo	Puntos estimados: 0.3
Fecha de inicio: 18 de febrero de 2015	Fecha de fin: 19 de febrero de 2015
Programador responsable: Eblis Manuel Caro Rivero	
Descripción: Crear plantilla, vista y url para listar todas las solicitudes de servicios a color registradas en el sistema, mostrar en un <i>datatable</i> que permita opciones de búsqueda sobre los campos mostrados, paginación y ordenamiento por las cabeceras de las columnas, manejar mensajes en caso de que no existan solicitudes de este tipo registradas. Crear una columna en la tabla con las entradas a las acciones de ver detalles, editar, asignar y eliminar. Realizar cada una de estas acciones con sus respectivas vistas, plantillas, formularios y urls.	

Tabla 4.11. Tarea de ingeniería # 11

Tarea	
Número de tarea: 11	Número de Historia de usuario: 4
Nombre de la tarea: Listar todas las solicitudes de servicios a color pendientes	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha de inicio: 20 de febrero de 2015	Fecha de fin: 20 de febrero de 2015
Programador responsable: Eblis Manuel Caro Rivero	
Descripción: Crear plantilla, vista y url para listar todas las solicitudes de servicios a color pendientes en el sistema, mostrar en un <i>datatable</i> que permita opciones de búsqueda sobre los campos mostrados, paginación y ordenamiento por las cabeceras de las columnas, manejar mensajes en caso de que no existan solicitudes de este tipo pendientes. Crear una columna en la tabla con las entradas a las acciones de ver detalles, editar, asignar y eliminar. Realizar cada una de estas acciones con sus respectivas vistas, plantillas, formularios y urls.	

Test Unitarios

En XP, el desarrollo dirigido por test es una de las prácticas que guían la etapa de implementación y pruebas. Los test unitarios o pruebas unitarias, como también se le conocen son pruebas automatizadas que se encargan de verificar la exactitud del código generado. Están escritos en el mismo lenguaje que el código que validan y crean bases de datos auxiliares para su ejecución. Se establece un test unitario por cada funcionalidad implementada. Una funcionalidad no esta completada hasta que haya pasado satisfactoriamente su test unitario correspondiente (CRISPIN y HOUSE, 2002).

El requisito mínimo que se establece para culminar una iteración es que todas las funcionalidades implementadas superen cada uno de sus test unitarios. En la Figura 4.1 se muestran los resultados de la ejecución de los test unitarios de la primera iteración.

```

Terminal
+ Microsoft Windows [Versión 6.3.9600]
X (c) 2013 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Mary\Desktop\DRDuci\drduci>python manage.py test app
Creating test database for alias 'default'...
.....
-----
Ran 11 tests in 4.742s

OK
Destroying test database for alias 'default'...
    
```

Figura 4.1. Resultados de la ejecución de los test unitarios de la primera iteración.

Pruebas de Aceptación

Las pruebas de aceptación son el principal mecanismo para validar el sistema con el cliente y medir su nivel de satisfacción. Estas pruebas marcan el nivel de calidad del software en relación a lo esperado por el cliente y generan las posibles no conformidades del mismo (CRISPIN y HOUSE, 2002).

En esta iteración se ejecutaron cuatro pruebas de aceptación por parte del cliente, una por cada HU. En la Figura 4.2 se muestran los resultados de la aplicación de las pruebas de aceptación de la primera iteración.

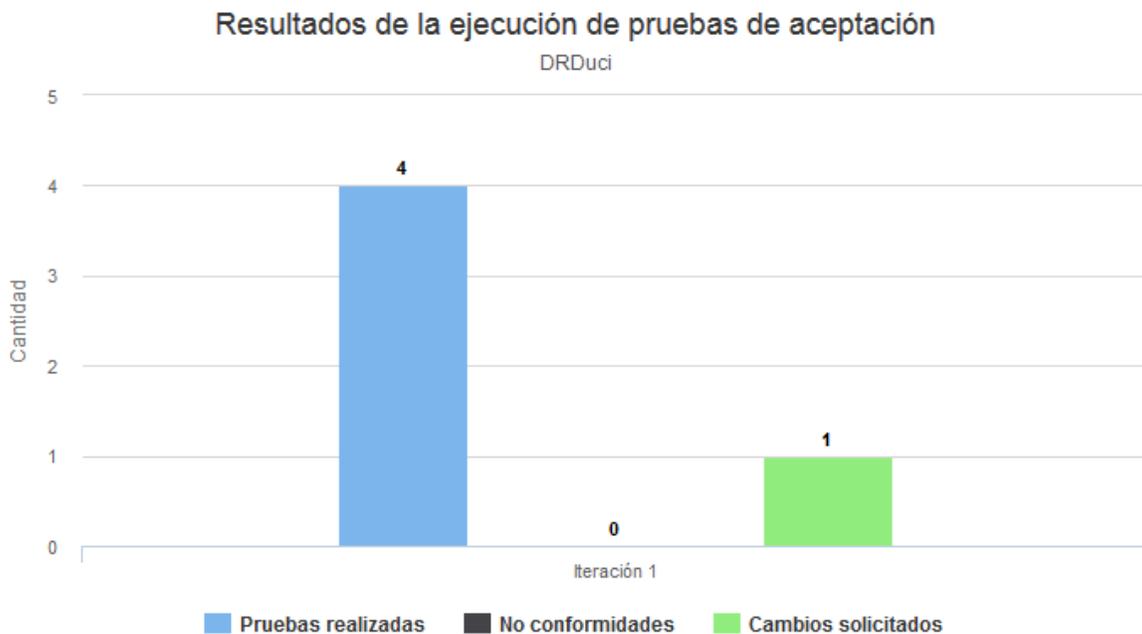


Figura 4.2. Resultados de la ejecución de las pruebas de aceptación de la primera iteración.

En la ejecución de las pruebas de aceptación se obtuvo como resultado que las cuatro HU implementadas

cumplen con los criterios de aceptación del cliente. Se realizó la petición de un cambio por parte del cliente en la HU Gestionar usuario, al editar un usuario se podrá cambiar el estado del mismo.

4.3.2. Iteración 2

Durante esta iteración se dio cumplimiento al cambio solicitado por el cliente en la entrega anterior y se implementaron las HU: Gestionar tareas de servicios generales, Atender tarea de servicios generales, Atender tarea de servicios a color y Gestionar notificaciones. Para ello se definieron seis TI, véase apéndice A.1.

Test Unitarios

En la Figura 4.3 se muestran los resultados de la ejecución de los test unitarios de la segunda iteración.

```

Terminal
C:\Users\Mary\Desktop\DRDuci\drduci>python manage.py test app
Creating test database for alias 'default'...
.....
-----
Ran 6 tests in 2.463s

OK
Destroying test database for alias 'default'...

C:\Users\Mary\Desktop\DRDuci\drduci>
    
```

Figura 4.3. Resultados de la ejecución de los test unitarios de la segunda iteración.

Pruebas de Aceptación

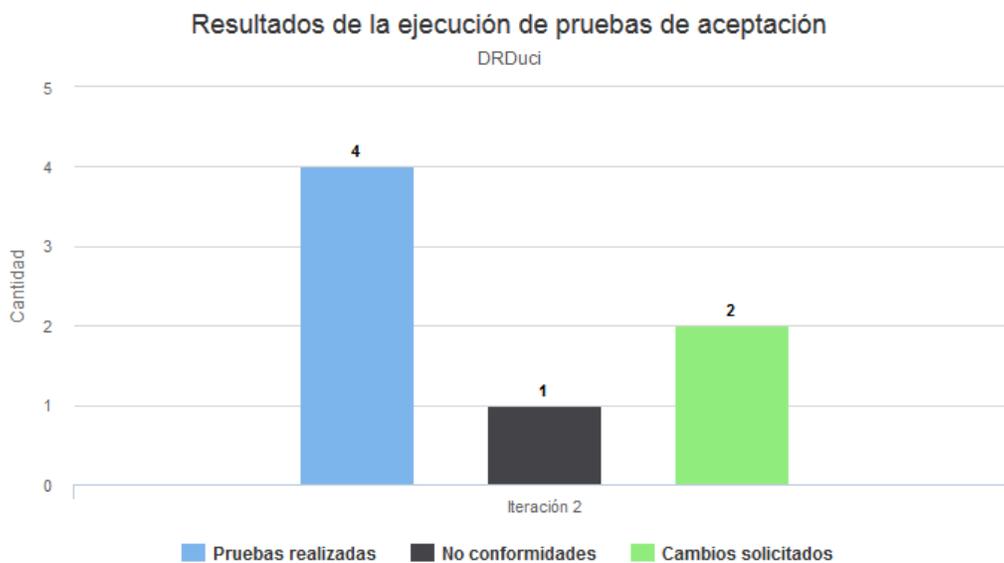


Figura 4.4. Resultados de la ejecución de las pruebas de aceptación de la segunda iteración.

En esta iteración se ejecutaron cuatro pruebas de aceptación por parte del cliente. En la Figura 4.4 se muestran los resultados de la aplicación de las pruebas de aceptación de la segunda iteración.

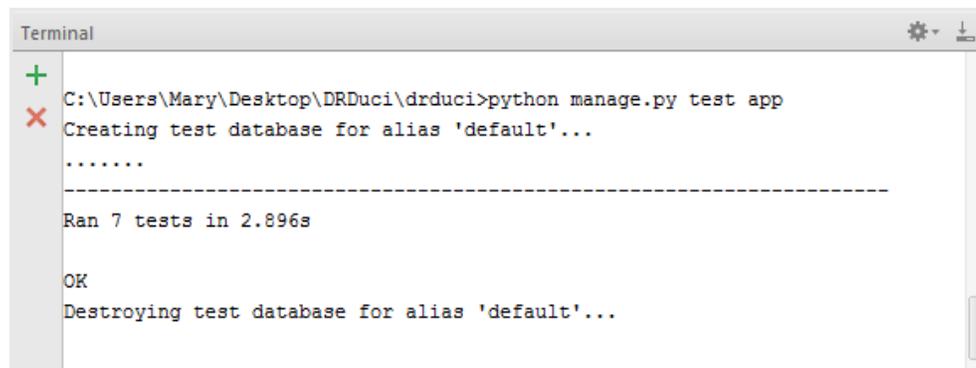
La ejecución de las pruebas de aceptación arrojó como resultado una no conformidad en cuanto a la HU Atender tarea de servicios generales, ya que el operario tuvo acceso a listar las solicitudes de servicios a color. Mientras, las otras tres HU implementadas cumplieron con los criterios de aceptación del cliente. Se realizaron dos peticiones de cambios por parte del cliente en la HU Gestionar notificaciones. Agregar la opción de reenviar una notificación enviada y enviar una cancelada.

4.3.3. Iteración 3

Durante esta iteración se dio solución a la no conformidad detectada en la entrega anterior así como se realizaron los cambios solicitados por el cliente en la entrega anterior. Después se implementaron las HU: Gestionar recurso material de servicios generales, Gestionar recurso material de servicios a color, Elaborar parte y Elaborar reportes. Para ello se definieron siete TI, véase apéndice A.2.

Test Unitarios

En la Figura 4.5 se muestran los resultados de la ejecución de los test unitarios de la tercera iteración.



```
Terminal
C:\Users\Mary\Desktop\DRDuci\drduci>python manage.py test app
Creating test database for alias 'default'...
.....
-----
Ran 7 tests in 2.896s

OK
Destroying test database for alias 'default'...
```

Figura 4.5. Resultados de la ejecución de los test unitarios de la tercera iteración.

Pruebas de Aceptación

En esta iteración se ejecutaron cuatro pruebas de aceptación por parte del cliente. En la Figura 4.6 se muestran los resultados de la aplicación de las pruebas de aceptación de la tercera iteración.

La ejecución de las pruebas de aceptación arrojó que las cuatro HU implementadas cumplen con los criterios de aceptación del cliente. Se realizó la petición de cambio por parte del cliente en la HU Generar reportes, agregar la opción de generar un reporte que refleje la cantidad de tesis reproducidas por facultad.

Durante esta entrega el cliente pudo realizar pruebas a un sistema más completo, que le permitió cubrir el proceso de gestión de información del DRD casi en su totalidad. El nivel de satisfacción expresado por el propietario del software en esta ocasión fue notorio y resaltó la calidad de los reportes realizados.

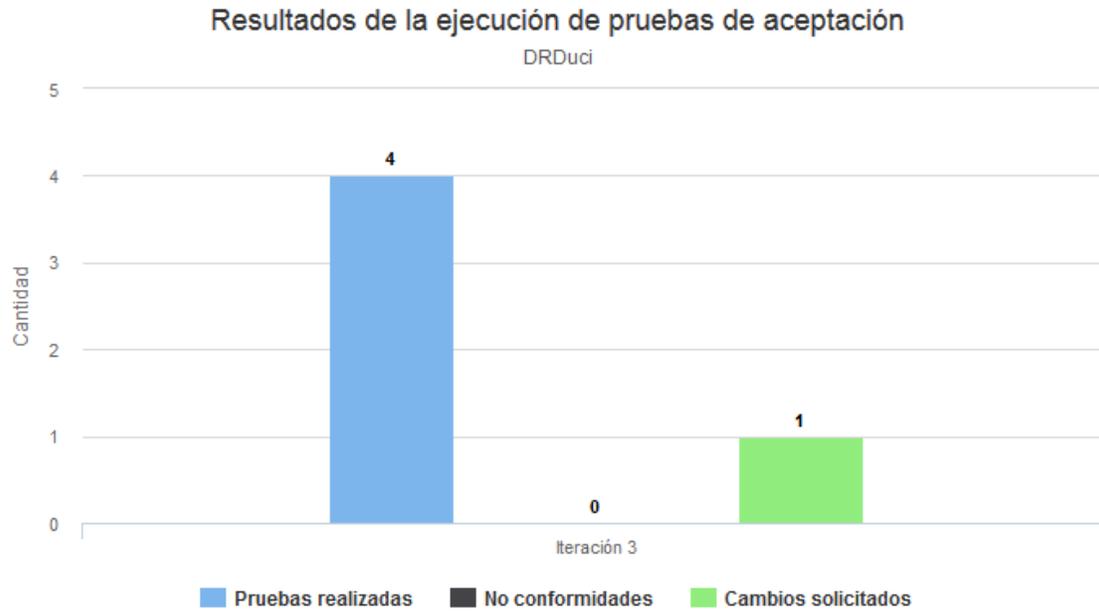


Figura 4.6. Resultados de la ejecución de las pruebas de aceptación de la tercera iteración.

4.3.4. Iteración 4

Durante el desarrollo de esta iteración se realizaron los cambios solicitados por el cliente en la entrega anterior. Después se implementaron las HU: Editar perfil de usuario, Alertas del sistema, Configurar opciones avanzadas, Gestionar equipo de servicios generales y Gestionar equipo de servicios a color. Para ello se definieron cinco TI, véase apéndice A.3.

Test Unitarios

En la Figura 4.7 se muestran los resultados de la ejecución de los test unitarios de la tercera iteración.

```

Terminal
C:\Users\Mary\Desktop\DRDuci\drduci>python manage.py test app
+
Creating test database for alias 'default'...
X
.....
-----
Ran 5 tests in 2.207s

OK
Destroying test database for alias 'default'...

C:\Users\Mary\Desktop\DRDuci\drduci>
  
```

Figura 4.7. Resultados de la ejecución de los test unitarios de la tercera iteración.

Pruebas de Aceptación

En esta iteración se ejecutaron cinco pruebas de aceptación por parte del cliente. En la Figura 4.8 se muestran los resultados de la aplicación de las pruebas de aceptación de la tercera iteración.

La ejecución de las pruebas de aceptación arrojó que las cinco HU implementadas cumplen con los criterios de aceptación del cliente. No se realizaron peticiones de cambio ni fueron identificadas no conformidades.

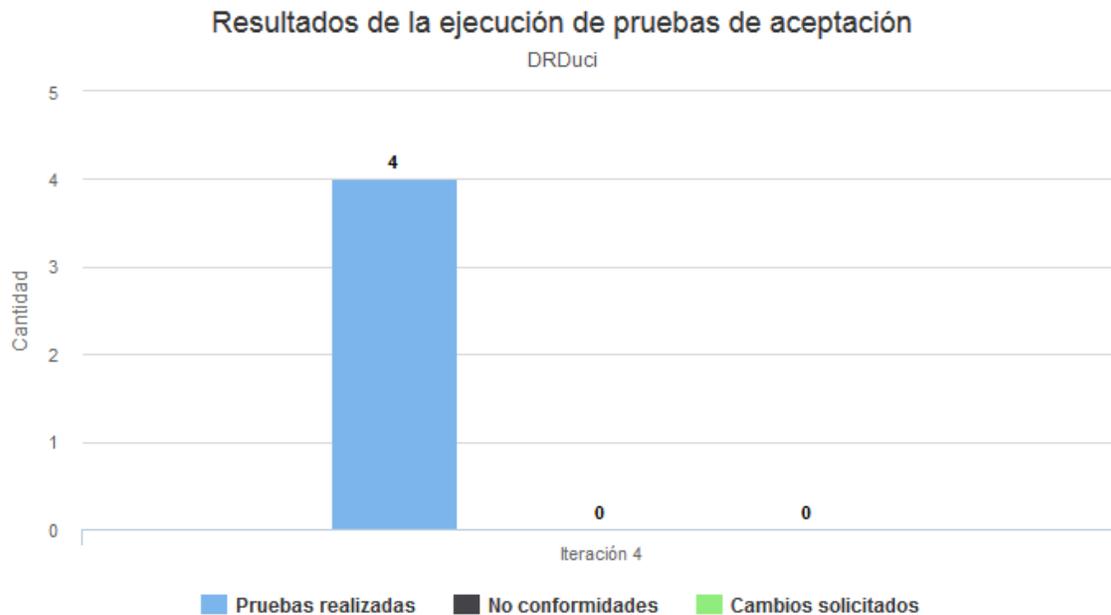


Figura 4.8. Resultados de la ejecución de las pruebas de aceptación de la tercera iteración.

4.4. Conclusiones del capítulo

En este capítulo se refleja la ejecución de las fase de implementación y pruebas dándoles cumplimiento al cuarto y quinto objetivo específico de la investigación. Para lo cual se realizaron las cuatro iteraciones planificadas, implementándose un total de 17 HU a través de 29 TI. El empleo de los estándares de codificación definidos facilitó la lectura, comprensión y mantenimiento del código para los desarrolladores.

Con el objetivo de validar el producto se aplicaron pruebas unitarias y pruebas de aceptación. Las pruebas unitarias sirvieron para comprobar internamente las funciones de los módulos, facilitando la detección de errores que pudieran conllevar a futuras no conformidades. Las pruebas de aceptación permitieron comprobar que las funciones son operativas a través de la interfaz del software, que la entrada se acepta de forma adecuada y se produce un resultado correcto. Se codificaron un total de 29 test unitarios y se realizaron pruebas de aceptación guiadas por un total de 17 casos de prueba de aceptación.

Conclusiones

Con la realización del presente Trabajo de Diploma se arribaron a las siguientes conclusiones:

1. La elaboración del marco teórico de la investigación permitió sentar las bases conceptuales del sistema informático a desarrollar y estableció la naturaleza del comportamiento de sus funcionalidades.
2. El análisis del proceso de reproducción de documentos en el DRD, permitió definir el flujo lógico de la información, identificar los roles que interactúan con el sistema y ayudó a estructurar una propuesta de solución para el sistema informático a desarrollar.
3. Se desarrolló un sistema informático para la gestión de información del Departamento de Reproducción de Documentos de la Universidad de las Ciencias Informáticas. El sistema desarrollado es actualmente la solución más ajustada para la gestión de información del proceso de reproducción de documentos. Informatiza tanto las etapas operativas de realización de solicitudes, ejecución de tareas, envío de notificaciones como el proceso de análisis de información y toma de decisiones. Facilitando así la ejecución del proceso de gestión de información. Lo cual posibilita que se cumpla satisfactoriamente el objetivo trazado para el presente Trabajo de Diploma, dando solución a la problemática planteada inicialmente.
4. La validación del sistema haciendo uso de las pruebas unitarias y de aceptación, permitió obtener una solución estable y correcta, que garantiza robustez y flexibilidad a cambios posteriores.

Recomendaciones

A la dirección del Centro de Información Científico-Técnica se le recomienda el despliegue de la solución en el DRD, de manera inmediata, posterior a que el sistema sea liberado por calidad de software, para que este comience a dar resultados tangibles para el departamento.

Para el mantenimiento del software se recomienda realizar las siguientes mejoras técnicas al sistema:

- Implementar mecanismos de seguridad elevados contra ataques de fuerza bruta en la autenticación de usuarios pues *django auth* no brinda esta posibilidad.
- Informatizar el proceso de realización de las solicitudes por parte de los clientes con el empleo del directorio activo instaurado en la Universidad con el fin de normalizar la vía en que este se realiza.

- BPMN** *Business Process Modeling Notation* (en español Notación para el Modelado de Procesos de Negocio). 16, 17, 19, 20
- CICT** Centro de Información Científico Técnico. 2, 3, 20, 22
- CRC** Clase-Responsabilidad-Colaboradores. 14, 42, 43, 46, 47
- DRD** Departamento de Reproducción de Documentos. 2–5, 12, 18–20, 22, 23, 26, 27, 30, 33, 34, 45, 56, 59, 60
- GI** Gestión de Información. 6, 7, 12
- GoF** *Gang of Four* (en español Banda de los cuatro). 42
- GRASP** *General Responsibility Assignment Software Patterns* (en español Patrones Generales de Software para la Asignación de Responsabilidades). 41
- HU** Historias de Usuario. 14, 28–30, 36–38, 47–49, 54–58
- IDE** Entorno de Desarrollo Integrado. 17
- ISO** Organización Internacional de Normalización. 8
- MTV** *Model-Template-View*. 40, 47
- MVC** Modelo-Vista-Controlador. 39, 40
- PUCESE** Pontificia Universidad Católica del Ecuador Sede Esmeraldas. 10
- SGI** Sistema para la Gestión de Información. 7–11
- SI** Sistema de Información. 7, 8
- SIDIA** Sistema de auto-servicio para el control y la gestión de impresión. 10
- TI** Tareas de Ingeniería. 14, 37, 48, 49, 55–58
- TIC** Tecnologías de la Información y las Comunicaciones. 1, 2
- UCI** Universidad de las Ciencias Informáticas. 2–5, 11, 12, 18–20, 27, 45
- UML** Lenguaje Unificado de Modelado. 42

UNEG Universidad Nacional Experimental de Guayana. 10

XP Programación Extrema. 12–14, 28, 37, 39, 42, 43, 47, 53

Referencias bibliográficas

- ALCHIN, M. 2012. *Pro Django*. Second Edition. www.it-ebooks.info : Apress, 2012.
- ALMARALES, Bárbara y SENCIAL, Robin. 2013. *Desarrollo de un portal web para la gestión de servicios en el Consejo Nacional de Patrimonio Cultural*. 2013.
- ÁLVAREZ, M. A. 2003. *Qué es Python*. 2003. Consultado el 5 de marzo de 2015. Dirección: <http://www.desarrolloweb.com/articulos/1325.php>.
- ÁLVAREZ, M. A. 2008. *Introducción a CSS3*. 2008. Consultado el 5 de marzo de 2015. Dirección: <http://www.desarrolloweb.com/articulos/introduccion-css3.html>.
- ÁLVAREZ, S. 2007. *Sistemas gestores de bases de datos. Introducción a este concepto y características especiales*. 2007. Consultado el 6 de marzo de 2015. Dirección: <http://www.desarrolloweb.com/articulos/sistemasgestores-bases-datos.html>.
- BARCIA, D. 2003. *Qué es CSS?* 2003. Consultado el 5 de marzo de 2015. Dirección: <http://www.maestrosdelweb.com/introcss/>.
- BECK, K. 1999. *Extreme Programming Explained. Embrace Change*. Pearson Education. 1999. Traducido al español como: "Una explicación de la programación extrema. Aceptar el cambio", Addison Wesley, 2000.
- BECK, K. 2000. *Extreme Programming Explained*. 2000. p. 179.
- BOOTELLO, R. 2007-2008. *Sistemas de Información*. MBA Edición, 2007-2008.
- CABRERA, Y. y LLANES, R. 2009. *Automatización del Servicio de Impresión de la Dirección de Información*. 2009.
- CALERO, M. 2003. Una explicación de la programación extrema (XP). *V Encuentro usuarios xBase 2003 MADRID*. 2003.
- CANÓS, J. H.; LETELIER, P. y PENADÉS, M. C. 2011. *Métodologías Ágiles en el Desarrollo de Software. DSIC -Universidad Politécnica de Valencia*. 2011.
- CONSTANTINI, P. E. 2014. *Implementación de un Sistema de Auto-Servicio para el Control y la Gestión de Impresión*. 2014.

- CRISPIN, L. y HOUSE, T. 2002. *Testing Extreme Programming*. 2002. 0-321-11355-1.
- CUBA, I. y TASE, J. 2005. *Sistema Integrado de Gestión Bibliotecaria. Módulos: Reproducción y Digitalización de materiales, Proceso de Selección y Adquisición de Materiales*. 2005.
- Definición de sistema de información - Qué es, Significado y Concepto*. [consultado 13 nov. 2014]. Dirección: <http://definicion.de/sistema-de-informacion/>.
- EMERY, J. C. 1990. *Sistemas de información para la dirección. El recurso estratégico crítico*. 1990. pág. 4.
- Fedex Office*, 2014. [consultado 13 nov. 2014]. Dirección: <http://fedex.com/us/office/>.
- FERNÁNDEZ, G. 2002. Introducción a Extreme Programming. En. *Ingeniería del Software II*. 2002.
- FRESNO, M.; MAUCO, V.; RIDAO, M.; DOORN, J. y RIVERO, L. 2007. *Derivación de objetos utilizando LEL y Escenarios en un caso real*. 2007.
- GAMMA, E.; HELM, R.; JOHNSON, R. y VLISSIDES, J. 2002. *Patrones de Diseño. Elementos de software orientado a objetos reutilizable*. Madrid. España. : Addison Wesley, 2002.
- GONZÁLEZ, J. y GAUDIOSO, E. 2001. *Aprender y formar en Internet*. 2001. Cap. 6. JavaScript.
- HICKSON, I. y HYATT, D. 2009. *HTML5*. 2009. Consultado el 5 de marzo de 2015. Dirección: <http://dev.w3.org/html5/spec/Overview.html#html-vs-xhtml>.
- HOLMES, B. y T. JOYCE, D. 2000. *Object-oriented programming with Java*. Sudbury. 2000.
- JEFFRIES, Ron. 2008. *How should user stories be written?* 2008. Consultado el 4 de marzo de 2015. Dirección: <http://xprogramming.com/blog/how-should-user-stories-be-written/>.
- JETBRAINS. 2000-2015. *PyCharm*. 2000-2015. Consultado el 6 de marzo de 2015. Dirección: <https://www.jetbrains.com/pycharm/>.
- LLEONART, E.; GARCÍA, A. y ROVIRA, M. 2010. *Patrones*. 2010.
- LÓPEZ, J. 1991. El desarrollo de los sistemas de información y documentación. *Cuadernos de la EUBD*. 1991, vol. 1, n.º 1, págs. 23-33.
- MARQUÉS, P. 2002. *La Información y el conocimiento*. 2002. [consultado 12 nov. 2014].
- PgAdmin*. 2007. Dirección: <http://www.purosoftware.com/programacion-bases-de-datos/11-pg-admin-3.html>.
- Portal de la Universidad de las Ciencias Informáticas*. 2015. [consultado 13 nov. 2014]. Dirección: <http://www.uci.cu/?q=mision>.
- PostgreSQL*. 1996-2015. Consultado el 6 de marzo de 2015. Dirección: <http://www.postgresql.org/about/>.
- PRESSMAN, R. S. 2001. *Ingeniería del Software: una tecnología estratificada. Ingeniería del Software. Un enfoque práctico*. Quinta Edición, 2001.

- PRESSMAN, R. S. 2003. *Software Engineer. A practitioner's Approach*. 7ma Edición. 2003.
- RODRÍGUEZ, K. 2002. Gestión de la información en las Organizaciones. 2002, vol. XX, n.º 1 y 2, págs. 19-34.
- SANNER, M. F. 1999. *Python: a programming language for software integration and development*. 1999.
- SENN, J.A. 1992. Análisis y Diseño de Sistemas de Información. *México: McGraw-Hill*. 1992, págs. 25-29.
- TRAMULLAS, J. 1997. Los sistemas de información: una reflexión sobre información, sistema y documentación. *Revista General de Información y Documentación*. 1997, vol. 7, n.º 1, págs. 207.
- VAN ROSSUM, G.; WARSAW, B. y COGHLAN, N. 2001. *Pep 8, Style guide for python code*. 2001.
- VELOSO, P. 2008. Uso de patrones de arquitectura. 2008.
- VIDAL, M. 2006. *Información y conocimiento para la dirección*. 2006.
- VISCONTI M. y Astudillo, H. 2011. Fundamentos de Ingeniería de Software. *Universidad Técnica Federico Santa María. Departamento de Informática*. 2011.
- WALLACE, D.; RAGGETT, I. y AUFANG, J. 2002. *Extreme Programming for Web Projects*. Boston : Addison Wesley, 2002.
- WHITE, S. A. 2011. *Introduction to BPMN*. 2011.

- ALCHIN, M. 2012. *Pro Django*. Second Edition. www.it-ebooks.info : Apress, 2012.
- ALMARALES, Bárbara y SENCIAL, Robin. 2013. *Desarrollo de un portal web para la gestión de servicios en el Consejo Nacional de Patrimonio Cultural*. 2013.
- ÁLVAREZ, M. A. 2003. *Qué es Python*. 2003. Consultado el 5 de marzo de 2015. Dirección: <http://www.desarrolloweb.com/articulos/1325.php>.
- ÁLVAREZ, M. A. 2008. *Introducción a CSS3*. 2008. Consultado el 5 de marzo de 2015. Dirección: <http://www.desarrolloweb.com/articulos/introduccion-css3.html>.
- ÁLVAREZ, S. 2007. *Sistemas gestores de bases de datos. Introducción a este concepto y características especiales*. 2007. Consultado el 6 de marzo de 2015. Dirección: <http://www.desarrolloweb.com/articulos/sistemasgestores-bases-datos.html>.
- BARCIA, D. 2003. *Qué es CSS?* 2003. Consultado el 5 de marzo de 2015. Dirección: <http://www.maestrosdelweb.com/introcss/>.
- BARTLE, P. 2009. *Información para la Gestión y gestión de la información*. 2009.
- BECK, K. 1999. *Extreme Programming Explained. Embrace Change*. Pearson Education. 1999. Traducido al español como: "Una explicación de la programación extrema. Aceptar el cambio", Addison Wesley, 2000.
- BECK, K. 2000. *Extreme Programming Explained*. 2000. p. 179.
- BECK, K. y FOWLER, M. 2000. *Planning Extreme Programming*. First Edition October 12, 2000. 2000.
- BOOTELLO, R. 2007-2008. *Sistemas de Información*. MBA Edición, 2007-2008.
- CABRERA, Y. y LLANES, R. 2009. *Automatización del Servicio de Impresión de la Dirección de Información*. 2009.
- CALERO, M. 2003. Una explicación de la programación extrema (XP). *V Encuentro usuarios xBase 2003 MADRID*. 2003.
- CANÓS, J. H.; LETELIER, P. y PENADÉS, M. C. 2011. *Métodologías Ágiles en el Desarrollo de Software. DSIC -Universidad Politécnica de Valencia*. 2011.

- CONSTANTINI, P. E. 2014. *Implementación de un Sistema de Auto-Servicio para el Control y la Gestión de Impresión*. 2014.
- CRISPIN, L. y HOUSE, T. 2002. *Testing Extreme Programming*. 2002. 0-321-11355-1.
- CUBA, I. y TASE, J. 2005. *Sistema Integrado de Gestión Bibliotecaria. Módulos: Reproducción y Digitalización de materiales, Proceso de Selección y Adquisición de Materiales*. 2005.
- D., E. y CASTILLO, T. 2012. *Sistema de Control de impresiones para los laboratorios de informática de la UNEG (SIMPALAB)*. 2012. c. 1, p. 1-20.
- Definición de sistema de información - Qué es, Significado y Concepto*. [consultado 13 nov. 2014]. Dirección: <http://definicion.de/sistema-de-informacion/>.
- El Libro de Django*. 2006. Consultado en marzo de 2015. Dirección: <http://django-book.mkauermann.com.ar/chapter01.html>.
- EMERY, J. C. 1990. *Sistemas de información para la dirección. El recurso estratégico crítico*. 1990. pág. 4.
- Fedex Office*, 2014. [consultado 13 nov. 2014]. Dirección: <http://fedex.com/us/office/>.
- FERNÁNDEZ, G. 2002. Introducción a Extreme Programming. En. *Ingeniería del Software II*. 2002.
- FRESNO, M.; MAUCO, V.; RIDAO, M.; DOORN, J. y RIVERO, L. 2007. *Derivación de objetos utilizando LEL y Escenarios en un caso real*. 2007.
- FUSTER, Francisco. 1999. *Archivística, archivo, documento de archivo... necesidad de clarificar los conceptos*. 1999.
- GAMMA, E.; HELM, R.; JOHNSON, R. y VLISSIDES, J. 2002. *Patrones de Diseño. Elementos de software orientado a objetos reutilizable*. Madrid. España. : Addison Wesley, 2002.
- GONZÁLEZ, J. y GAUDIOSO, E. 2001. *Aprender y formar en Internet*. 2001. Cap. 6. JavaScript.
- HICKSON, I. y HYATT, D. 2009. *HTML5*. 2009. Consultado el 5 de marzo de 2015. Dirección: <http://dev.w3.org/html5/spec/Overview.html#html-vs-xhtml>.
- HOLMES, B. y T. JOYCE, D. 2000. *Object-oriented programming with Java*. Sudbury. 2000.
- JEFFRIES, Ron. 2008. *How should user stories be written?* 2008. Consultado el 4 de marzo de 2015. Dirección: <http://xprogramming.com/blog/how-should-user-stories-be-written/>.
- JETBRAINS. 2000-2015. *PyCharm*. 2000-2015. Consultado el 6 de marzo de 2015. Dirección: <https://www.jetbrains.com/pycharm/>.
- LLEONART, E.; GARCÍA, A. y ROVIRA, M. 2010. *Patrones*. 2010.
- LÓPEZ, J. 1991. El desarrollo de los sistemas de información y documentación. *Cuadernos de la EUBD*. 1991, vol. 1, n.º 1, págs. 23-33.
- MARQUÉS, P. 2002. *La Información y el conocimiento*. 2002. [consultado 12 nov. 2014].

- MÉNDEZ, EM. 2010. *Metadatos y tesauros: aplicación de XML/RDF a los sistemas de organización del conocimiento en Intranets*. 2010.
- MUÑOZ, A. 2003. Sistemas de información en las empresas. *Hipertext.net*. 2003, n.º 1, págs. 1. Url: <http://www.hipertext.net>.
- NONAKA, I. y TAKEUCHI, H. 1995. *The Knowledge-Creating Company: How Japanese Company create the dynamic softinnovation*. 1995.
- PAVÓN, J. 2008-2009. *El patrón Modelo-Vista-Controlador (MVC)*. 2008-2009.
- PgAdmin*. 2007. Dirección: <http://www.purosoftware.com/programacion-bases-de-datos/11-pg-admin-3.html>.
- Portal de la Universidad de las Ciencias Informáticas*. 2015. [consultado 13 nov. 2014]. Dirección: <http://www.uci.cu/?q=mision>.
- PostgreSQL*. 1996-2015. Consultado el 6 de marzo de 2015. Dirección: <http://www.postgresql.org/about/>.
- PRESSMAN, R. S. 2001. *Ingeniería del Software: una tecnología estratificada. Ingeniería del Software. Un enfoque práctico*. Quinta Edición, 2001.
- PRESSMAN, R. S. 2003. *Software Engineer. A practitioner's Approach*. 7ma Edición. 2003.
- RODRÍGUEZ, K. 2002. Gestión de la información en las Organizaciones. 2002, vol. XX, n.º 1 y 2, págs. 19-34.
- ROJAS, Y. 2006. De la gestión de Información a la Gestión del Conocimiento. *ACIMED*. 2006, vol. 14, n.º 1.
- SAMUELSON, K. 1977. *Information Systems and networks*. Amsterdam: North Holland. 1977.
- SANNER, M. F. 1999. *Python: a programming language for software integration and developement*. 1999.
- SENN, J.A. 1992. Análisis y Diseño de Sistemas de Información. México: McGraw-Hill. 1992, págs. 25-29.
- TRAMULLAS, J. 1997. Los sistemas de información: una reflexión sobre información, sistema y documentación. *Revista General de Información y Documentación*. 1997, vol. 7, n.º 1, págs. 207.
- TRAMULLAS, J. 2000. *Organización y gestión del diseño de sistemas de información*. 2000.
- VAN ROSSUM, G.; WARSAW, B. y COGHLAN, N. 2001. *Pep 8, Style guide for python code*. 2001.
- VELOSO, P. 2008. *Uso de patrones de arquitectura*. 2008.
- VIDAL, M. 2006. *Información y conocimiento para la dirección*. 2006.
- VISCONTI M. y Astudillo, H. 2011. *Fundamentos de Ingeniería de Software*. Universidad Técnica Federico Santa María. Departamento de Informática. 2011.
- WALLACE, D.; RAGGETT, I. y AUFANG, J. 2002. *Extreme Programming for Web Projects*. Boston : Addison Wesley, 2002.

WHITE, S. A. 2011. *Introduction to BPMN*. 2011.

WIRFS-BROCK, R. 1990. *Designing Object-Oriented Software*. 1990.

WOODMAN, L. 1985. Information management in large organizations. *Information management from strategies to action*. 1985, págs. 95-114.

Apéndices

Tareas de Ingeniería

A.1. Iteración 2

Tabla A.1. Tarea de ingeniería # 12

Tarea	
Número de tarea: 12	Número de Historia de usuario: 5
Nombre de la tarea: Listar tareas de servicios generales	
Tipo de tarea: Desarrollo	Puntos estimados: 0.3
Fecha de inicio: 24 de febrero de 2015	Fecha de fin: 25 de febrero de 2015
Programador responsable: Eblis Manuel Caro Rivero	
Descripción: Crear plantilla, vista y url para listar todas las tareas de servicios generales registradas en el sistema, mostrar en un <i>datatable</i> que permita opciones de búsqueda sobre los campos mostrados, paginación y ordenamiento por las cabeceras de las columnas, manejar mensajes en caso de que no existan tareas de este tipo. Crear una columna en la tabla con las entradas a las acciones de ver detalles, reasignar y eliminar. Realizar cada una de estas acciones con sus respectivas vistas, plantillas, formularios y urls.	

Tabla A.2. Tarea de ingeniería # 13

Tarea	
Número de tarea: 13	Número de Historia de usuario: 5
Nombre de la tarea: Listar tareas pendientes de servicios generales	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha de inicio: 25 de febrero de 2015	Fecha de fin: 26 de febrero de 2015
Programador responsable: Eblis Manuel Caro Rivero	

Continúa en la próxima página

Tabla A.2. Continuación de la página anterior

<p>Descripción: Crear plantilla, vista y url para listar todas las tareas de servicios generales pendientes en el sistema, mostrar en un <i>datatable</i> que permita opciones de búsqueda sobre los campos mostrados, paginación y ordenamiento por las cabeceras de las columnas, manejar mensajes en caso de que no existan tareas de este tipo. Crear una columna en la tabla con las entradas a las acciones de ver detalles, reasignar y eliminar. Realizar cada una de estas acciones con sus respectivas vistas, plantillas, formularios y urls.</p>

Tabla A.3. Tarea de ingeniería # 14

Tarea	
Número de tarea: 14	Número de Historia de usuario: 6
Nombre de la tarea: Atender tarea de servicios generales	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 26 de febrero de 2015	Fecha de fin: 5 de marzo de 2015
Programador responsable: Eblis Manuel Caro Rivero	
<p>Descripción: Crear plantilla, vista y url para listar todas las tareas de servicios generales asignadas al operario autenticado en el sistema, mostrar en un <i>datatable</i> que permita opciones de búsqueda sobre los campos mostrados, paginación y ordenamiento por las cabeceras de las columnas, manejar mensajes en caso de que no existan tareas de este tipo. Crear una columna en la tabla con las entradas a las acciones de ver detalles y atender. Realizar cada una de estas acciones con sus respectivas vistas, plantillas, formularios y urls.</p>	

Tabla A.4. Tarea de ingeniería # 15

Tarea	
Número de tarea: 15	Número de Historia de usuario: 7
Nombre de la tarea: Atender tarea de servicios a color	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 5 de marzo de 2015	Fecha de fin: 12 de marzo de 2015
Programador responsable: Eblis Manuel Caro Rivero	
<p>Descripción: Crear plantilla, vista y url para listar todas las tareas de servicios a color asignadas al técnico autenticado en el sistema, mostrar en un <i>datatable</i> que permita opciones de búsqueda sobre los campos mostrados, paginación y ordenamiento por las cabeceras de las columnas, manejar mensajes en caso de que no existan tareas de este tipo. Crear una columna en la tabla con las entradas a las acciones de ver detalles y atender. Realizar cada una de estas acciones con sus respectivas vistas, plantillas, formularios y urls.</p>	

Tabla A.5. Tarea de ingeniería # 16

Tarea	
Número de tarea: 16	Número de Historia de usuario: 8
Nombre de la tarea: Listar notificaciones	
Tipo de tarea: Desarrollo	Puntos estimados: 0.3
Fecha de inicio: 12 de marzo de 2015	Fecha de fin: 13 de marzo de 2015
Programador responsable: Eblis Manuel Caro Rivero	
Descripción: Crear plantilla, vista y url para listar las notificaciones, mostrar en un <i>datatable</i> que permita opciones de búsqueda sobre los campos mostrados, paginación y ordenamiento por las cabeceras de las columnas, manejar mensajes en caso de que no existan notificaciones. Crear una columna en la tabla con las entradas a las acciones de disponer, enviar o reenviar. Realizar cada una de estas acciones con sus respectivas vistas, plantillas, formlarios y urls.	

Tabla A.6. Tarea de ingeniería # 17

Tarea	
Número de tarea: 17	Número de Historia de usuario: 8
Nombre de la tarea: Listar notificaciones pendientes	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha de inicio: 13 de marzo de 2015	Fecha de fin: 16 de marzo de 2015
Programador responsable: Eblis Manuel Caro Rivero	
Descripción: Crear plantilla, vista y url para listar las notificaciones pendientes, mostrar en un <i>datatable</i> que permita opciones de búsqueda sobre los campos mostrados, paginación y ordenamiento por las cabeceras de las columnas, manejar mensajes en caso de que no existan notificaciones pendientes. Crear una columna en la tabla con las entradas a las acciones de disponer, enviar o reenviar. Realizar cada una de estas acciones con sus respectivas vistas, plantillas, formlarios y urls.	

A.2. Iteración 3

Tabla A.7. Tarea de ingeniería # 18

Tarea	
Número de tarea: 18	Número de Historia de usuario: 9
Nombre de la tarea: Gestionar recurso material de servicios generales	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 18 de marzo de 2015	Fecha de fin: 20 de marzo de 2015
Programador responsable: Eblis Manuel Caro Rivero	

Continúa en la próxima página

Tabla A.7. Continuación de la página anterior

<p>Descripción: Crear plantilla, vista y url para listar todos los recursos materiales de servicios generales registrados en el sistema, mostrar en un <i>datatable</i> que permita opciones de búsqueda sobre los campos mostrados, paginación y ordenamiento por las cabeceras de las columnas, manejar mensajes en caso de que no existan recursos materiales de este tipo pendientes. Crear una columna en la tabla con las entradas a las acciones de ver adicionar cantidades, consumir cantidades y eliminar. Colocar un botón que enlace a la acción de adicionar un nuevo recurso material de servicios generales. Realizar cada una de estas acciones con sus respectivas vistas, plantillas, formularios y urls.</p>
--

Tabla A.8. Tarea de ingeniería # 19

Tarea	
Número de tarea: 19	Número de Historia de usuario: 10
Nombre de la tarea: Gestionar recurso material de servicios a color	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 23 de marzo de 2015	Fecha de fin: 25 de marzo de 2015
Programador responsable: Eblis Manuel Caro Rivero	
<p>Descripción: Crear plantilla, vista y url para listar todos los recursos materiales de servicios a color registrados en el sistema, mostrar en un <i>datatable</i> que permita opciones de búsqueda sobre los campos mostrados, paginación y ordenamiento por las cabeceras de las columnas, manejar mensajes en caso de que no existan recursos materiales de este tipo pendientes. Crear una columna en la tabla con las entradas a las acciones de ver adicionar cantidades, consumir cantidades y eliminar. Colocar un botón que enlace a la acción de adicionar un nuevo recurso material de servicios a color. Realizar cada una de estas acciones con sus respectivas vistas, plantillas, formularios y urls.</p>	

Tabla A.9. Tarea de ingeniería # 20

Tarea	
Número de tarea: 20	Número de Historia de usuario: 11
Nombre de la tarea: Elaborar parte	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha de inicio: 26 de marzo de 2015	Fecha de fin: 26 de marzo de 2015
Programador responsable: Eblis Manuel Caro Rivero	
<p>Descripción: Crear plantilla, vista y url para elaborar un parte del estado actual del sistema, mostrar en un conjunto de tablas que muestren la cantidad de solicitudes, tareas y notificaciones diferenciadas por el estado de cada una, y el tipo de servicio.</p>	

Tabla A.10. Tarea de ingeniería # 21

Tarea	
Número de tarea: 21	Número de Historia de usuario: 12
Nombre de la tarea: Elaborar reporte servicios prestados	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 27 de marzo de 2015	Fecha de fin: 31 de marzo de 2015
Programador responsable: Eblis Manuel Caro Rivero	
Descripción: Crear plantilla, vista y url para elaborar un gráfico y una tabla que reflejan la cantidad de servicios prestados, diferenciados por el tipo de servicio. Colocar un select para definir el intervalo de tiempo del reporte.	

Tabla A.11. Tarea de ingeniería # 22

Tarea	
Número de tarea: 22	Número de Historia de usuario: 12
Nombre de la tarea: Elaborar reporte recursos materiales consumidos	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 31 de marzo de 2015	Fecha de fin: 2 de abril de 2015
Programador responsable: Eblis Manuel Caro Rivero	
Descripción: Crear plantilla, vista y url para elaborar un gráfico y una tabla que reflejan la cantidad de recursos materiales consumidos, diferenciados por el tipo de recurso material. Colocar un select para definir el intervalo de tiempo del reporte.	

Tabla A.12. Tarea de ingeniería # 23

Tarea	
Número de tarea: 23	Número de Historia de usuario: 12
Nombre de la tarea: Elaborar reporte tipos de documento	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 31 de marzo de 2015	Fecha de fin: 10 de abril de 2015
Programador responsable: Eblis Manuel Caro Rivero	
Descripción: Crear plantilla, vista y url para elaborar un gráfico y una tabla que reflejan la cantidad de documentos reproducidos, diferenciados por el tipo de documento. Colocar un select para definir el intervalo de tiempo del reporte.	

A.3. Iteración 4

Tabla A.13. Tarea de ingeniería # 24

Tarea	
Número de tarea: 24	Número de Historia de usuario: 13
Nombre de la tarea: Editar perfil de usuario	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha de inicio: 14 de abril de 2015	Fecha de fin: 14 de abril de 2015
Programador responsable: Eblis Manuel Caro Rivero	
Descripción: Crear plantilla, vista y url para editar el perfil de usuario, dentro de esta plantilla crear plantilla, vista y url para cambiar la contraseña.	

Tabla A.14. Tarea de ingeniería # 25

Tarea	
Número de tarea: 25	Número de Historia de usuario: 14
Nombre de la tarea: Alertas del sistema	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 15 de abril de 2015	Fecha de fin: 22 de abril de 2015
Programador responsable: Eblis Manuel Caro Rivero	
Descripción: Colocar en cada funcionalidad de culminación de evento una entrada al historial. Listar en la página principal del sistema en forma de alertas los eventos concernientes a cada usuario autenticado	

Tabla A.15. Tarea de ingeniería # 26

Tarea	
Número de tarea: 26	Número de Historia de usuario: 15
Nombre de la tarea: Configurar opciones avanzadas	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha de inicio: 23 de abril de 2015	Fecha de fin: 24 de abril de 2015
Programador responsable: Eblis Manuel Caro Rivero	
Descripción: Crear plantilla, vista y url para editar las configuraciones del correo de salida del sistema y el contenido predeterminado de las notificaciones.	

Tabla A.16. Tarea de ingeniería # 27

Tarea	
Número de tarea: 27	Número de Historia de usuario: 16
Nombre de la tarea: Gestionar equipos de servicios generales	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha de inicio: 27 de abril de 2015	Fecha de fin: 27 de abril de 2015

Continúa en la próxima página

Tabla A.16. Continuación de la página anterior

Programador responsable: Eblis Manuel Caro Rivero
Descripción: Crear plantilla, vista y url para listar todas los equipos de servicios generales registrados en el sistema, mostrar en un <i>datatable</i> que permita opciones de búsqueda sobre los campos mostrados, paginación y ordenamiento por las cabeceras de las columnas, manejar mensajes en caso de que no existan equipos de este tipo registrados. Crear una columna en la tabla con las entradas a las acciones de editar y eliminar. Colocar un botón que enlace a la acción de adicionar un nuevo equipo de servicios generales. Realizar cada una de estas acciones con sus respectivas vistas, plantillas, formularios y urls.

Tabla A.17. Tarea de ingeniería # 28

Tarea	
Número de tarea: 28	Número de Historia de usuario: 17
Nombre de la tarea: Gestionar equipos de servicios a color	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha de inicio: 28 de abril de 2015	Fecha de fin: 29 de abril de 2015
Programador responsable: Eblis Manuel Caro Rivero	
Descripción: Crear plantilla, vista y url para listar todas los equipos de servicios a color registrados en el sistema, mostrar en un <i>datatable</i> que permita opciones de búsqueda sobre los campos mostrados, paginación y ordenamiento por las cabeceras de las columnas, manejar mensajes en caso de que no existan equipos de este tipo registrados. Crear una columna en la tabla con las entradas a las acciones de editar y eliminar. Colocar un botón que enlace a la acción de adicionar un nuevo equipo de servicios a color. Realizar cada una de estas acciones con sus respectivas vistas, plantillas, formularios y urls.	

Tabla A.18. Tarea de ingeniería # 29

Tarea	
Número de tarea: 29	Número de Historia de usuario: 12
Nombre de la tarea: Elaborar reporte tesis por facultades	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 29 de abril de 2015	Fecha de fin: 30 de abril de 2015
Programador responsable: Eblis Manuel Caro Rivero	
Descripción: Crear plantilla, vista y url para elaborar un gráfico y una tabla que reflejan la cantidad de tesis de pregrado reproducidos, diferenciados por el área. Colocar un select para definir el intervalo de tiempo del reporte.	

Casos de prueba de aceptación

B.1. Iteración 1

Tabla B.1. Prueba de aceptación # 1

Caso de prueba de aceptación	
Código: HU1_P1	Historia de usuario: 1
Nombre: Autenticar usuario en el sistema.	
Descripción: Prueba para la funcionalidad autenticar usuario.	
Condiciones de ejecución: El usuario debe estar previamente registrado. El usuario y contraseña deben ser válidos.	
Pasos de ejecución: Se intenta autenticar un usuario en el sistema con los datos válidos.	
Resultados esperados: El usuario se autentica correctamente en el sistema.	

Tabla B.2. Prueba de aceptación # 2

Caso de prueba de aceptación	
Código: HU2_P1	Historia de usuario: 2
Nombre: Gestionar usuario.	
Descripción: Prueba para la funcionalidad gestionar usuario.	
Condiciones de ejecución: El usuario debe estar autenticado. El usuario debe ser supervisor.	

Continúa en la próxima página

Tabla B.2. Continuación de la página anterior

<p>Pasos de ejecución: Se intenta registrar un usuario con los datos válidos. Se intenta registrar un usuario con datos inválidos. Se intenta registrar un usuario existente en el sistema. Se intenta modificar un usuario registrado en el sistema con datos válidos Se intenta modificar un usuario registrado en el sistema con datos inválidos Se intenta eliminar un usuario del sistema.</p>
<p>Resultados esperados: Si los datos son correctos, se registra, se modifica y se elimina en cada caso. Si los datos son incorrectos o el usuario ya existe, se redirecciona para la misma página mostrando los errores. En cada caso el sistema muestra mensajes que reflejan lo ocurrido.</p>

Tabla B.3. Prueba de aceptación # 3

Caso de prueba de aceptación	
Código: HU3_P1	Historia de usuario: 3
Nombre: Gestionar solicitud de servicios generales.	
Descripción: Prueba para la funcionalidad gestionar solicitud de servicios generales.	
<p>Condiciones de ejecución: El usuario debe estar autenticado. El usuario debe ser supervisor.</p>	
<p>Pasos de ejecución: Se intenta registrar una solicitud con los datos válidos. Se intenta registrar una solicitud con datos inválidos. Se intenta modificar una solicitud registrada en el sistema con datos válidos Se intenta modificar una solicitud registrada en el sistema con datos inválidos Se intenta asignar una solicitud pendiente registrada en el sistema Se intenta eliminar una solicitud del sistema.</p>	
<p>Resultados esperados: Si los datos son correctos, se registra, se modifica y se elimina en cada caso. Si los datos son incorrectos se redirecciona para la misma página mostrando los errores. En cada caso el sistema muestra mensajes que reflejan lo ocurrido.</p>	

Tabla B.4. Prueba de aceptación # 4

Caso de prueba de aceptación	
Código: HU4_P1	Historia de usuario: 4
Nombre: Gestionar solicitud de servicios a color.	
Descripción: Prueba para la funcionalidad gestionar solicitud de servicios a color.	

Continúa en la próxima página

Tabla B.4. Continuación de la página anterior

<p>Condiciones de ejecución: El usuario debe estar autenticado. El usuario debe ser directivo.</p>
<p>Pasos de ejecución: Se intenta registrar una solicitud con los datos válidos. Se intenta registrar una solicitud con datos inválidos. Se intenta modificar una solicitud registrada en el sistema con datos válidos Se intenta modificar una solicitud registrada en el sistema con datos inválidos Se intenta asignar una solicitud pendiente registrada en el sistema Se intenta eliminar una solicitud del sistema.</p>
<p>Resultados esperados: Si los datos son correctos, se registra, se modifica y se elimina en cada caso. Si los datos son incorrectos se redirecciona para la misma página mostrando los errores. Se asigna la solicitud, se cambia el estado de la solicitud y se crea una tarea con estado pendiente. Si se elimina una solicitud se eliminan también las tareas asociadas a la misma. En cada caso el sistema muestra mensajes que reflejan lo ocurrido.</p>

B.2. Iteración 2

Tabla B.5. Prueba de aceptación # 5

Caso de prueba de aceptación	
Código: HU5_P1	Historia de usuario: 5
Nombre: Gestionar tareas de servicios generales.	
Descripción: Prueba para la funcionalidad gestionar tareas de servicios generales.	
<p>Condiciones de ejecución: El usuario debe estar autenticado. El usuario debe ser supervisor.</p>	
<p>Pasos de ejecución: Se intenta reasignar una tarea pendiente Se intenta ver los detalles de una tarea. Se intenta eliminar una tarea del sistema.</p>	
<p>Resultados esperados: Se reasigna la tarea o se muestra una página con los detalles de la tarea según sea el caso. Se elimina la tarea, se cambia el estado de la solicitud asociada a la tarea. En cada caso el sistema muestra mensajes que reflejan lo ocurrido.</p>	

Tabla B.6. Prueba de aceptación # 6

Caso de prueba de aceptación	
Código: HU6_P1	Historia de usuario: 6
Nombre: Atender tarea de servicios generales.	
Descripción: Prueba para la funcionalidad atender tarea de servicios generales.	
Condiciones de ejecución: El usuario debe estar autenticado. El usuario debe ser operario.	
Pasos de ejecución: Se intenta ver los detalles de una tarea. Se intenta atender una tarea pendiente.	
Resultados esperados: Se muestra una página con los detalles de la tarea o se culmina la tarea la tarea según sea el caso. Al culminar una tarea se crea una notificación pendiente. En cada caso el sistema muestra mensajes que reflejan lo ocurrido.	

Tabla B.7. Prueba de aceptación # 7

Caso de prueba de aceptación	
Código: HU7_P1	Historia de usuario: 7
Nombre: Atender tarea de servicios a color.	
Descripción: Prueba para la funcionalidad atender tarea de servicios a color.	
Condiciones de ejecución: El usuario debe estar autenticado. El usuario debe ser técnico.	
Pasos de ejecución: Se intenta ver los detalles de una tarea. Se intenta atender una tarea pendiente.	
Resultados esperados: Se muestra una página con los detalles de la tarea o se culmina la tarea la tarea según sea el caso. En cada caso el sistema muestra mensajes que reflejan lo ocurrido.	

Tabla B.8. Prueba de aceptación # 8

Caso de prueba de aceptación	
Código: HU8_P1	Historia de usuario: 8
Nombre: Atender tarea de servicios a color.	
Descripción: Prueba para la funcionalidad gestionar notificaciones.	

Continúa en la próxima página

Tabla B.8. Continuación de la página anterior

<p>Condiciones de ejecución: El usuario debe estar autenticado. El usuario debe ser supervisor.</p>
<p>Pasos de ejecución: Se intenta enviar una notificación pendiente. Se intenta cancelar una notificación pendiente. Se intenta reenviar una notificación notificada. Se intenta enviar una notificación cancelada.</p>
<p>Resultados esperados: Se cambia el estado de la notificación a notificada y se envía un correo electrónico al usuario de la solicitud asociada a la notificación. Se cambia el estado de la notificación a cancelada. Se envía la notificación por correo electrónico al usuario de la solicitud asociada a la notificación. Se cambia el estado de la notificación a notificada y se envía un correo electrónico al usuario de la solicitud asociada a la notificación. En cada caso el sistema muestra mensajes que reflejan lo ocurrido.</p>

B.3. Iteración 3

Tabla B.9. Prueba de aceptación # 9

Caso de prueba de aceptación	
Código: HU9_P1	Historia de usuario: 9
Nombre: Gestionar recursos materiales de servicios generales.	
Descripción: Prueba para la funcionalidad gestionar recursos materiales de servicios generales.	
<p>Condiciones de ejecución: El usuario debe estar autenticado. El usuario debe ser supervisor.</p>	
<p>Pasos de ejecución: Se intenta adicionar un nuevo recurso material Se intenta adicionar determinada cantidad de unidades de un recurso material existente. Se intenta consumir determinada cantidad de unidades de un recurso material existente. Se intenta eliminar un recurso material existente.</p>	
<p>Resultados esperados: Se crea un nuevo recurso material de servicio general. Se adiciona la cantidad de unidades introducidas del recurso material en cuestión. Se consume la cantidad de unidades introducida del recurso material en cuestión. Se elimina el recurso material en cuestión. En cada caso el sistema muestra mensajes que reflejan lo ocurrido.</p>	

Tabla B.10. Prueba de aceptación # 10

Caso de prueba de aceptación	
Código: HU10_P1	Historia de usuario: 10
Nombre: Gestionar recursos materiales de servicios a color.	
Descripción: Prueba para la funcionalidad gestionar recursos materiales de servicios a color.	
Condiciones de ejecución: El usuario debe estar autenticado. El usuario debe ser técnico.	
Pasos de ejecución: Se intenta adicionar un nuevo recurso material Se intenta adicionar determinada cantidad de unidades de un recurso material existente. Se intenta consumir determinada cantidad de unidades de un recurso material existente. Se intenta eliminar un recurso material existente.	
Resultados esperados: Se crea un nuevo recurso material de servicio general. Se adiciona la cantidad de unidades introducidas del recurso material en cuestión. Se consume la cantidad de unidades introducida del recurso material en cuestión. Se elimina el recurso material en cuestión. En cada caso el sistema muestra mensajes que reflejan lo ocurrido.	

Tabla B.11. Prueba de aceptación # 11

Caso de prueba de aceptación	
Código: HU11_P1	Historia de usuario: 11
Nombre: Elaborar parte.	
Descripción: Prueba para la funcionalidad elaborar parte.	
Condiciones de ejecución: El usuario debe estar autenticado. El usuario debe ser supervisor o directivo.	
Pasos de ejecución: Se intenta acceder a la funcionalidad generar parte.	
Resultados esperados: Se muestra una página con las tres tablas que contienen los datos asociadas al estado actual de las solicitudes, tareas y notificaciones. En cada caso el sistema muestra mensajes que reflejan lo ocurrido.	

Tabla B.12. Prueba de aceptación # 12

Caso de prueba de aceptación	
Código: HU12_P1	Historia de usuario: 12
Nombre: Gestionar notificaciones.	
Descripción: Prueba para la funcionalidad elaborar reporte.	

Continúa en la próxima página

Tabla B.12. Continuación de la página anterior

<p>Condiciones de ejecución: El usuario debe estar autenticado. El usuario debe ser supervisor o directivo.</p>
<p>Pasos de ejecución: Se intenta generar el reporte. Se selecciona el intervalo temporal.</p>
<p>Resultados esperados: Se muestra una página con un select para seleccionar el intervalo temporal. Se muestra una página con una tabla que contienen los datos asociadas al reporte seleccionado y una gráfica que los expresa con un mayor nivel de claridad. En cada caso el sistema muestra mensajes que reflejan lo ocurrido.</p>

B.4. Iteración 4

Tabla B.13. Prueba de aceptación # 13

Caso de prueba de aceptación	
Código: HU13_P1	Historia de usuario: 13
Nombre: Editar perfil de usuario.	
Descripción: Prueba para la funcionalidad editar perfil de usuario.	
<p>Condiciones de ejecución: El usuario debe estar autenticado.</p>	
<p>Pasos de ejecución: Se intenta editar el perfil de usuario. Se intenta cambiar la contraseña.</p>	
<p>Resultados esperados: El sistema cambia los valores del usuarios editables. El sistema establece la nueva contraseña. En cada caso el sistema muestra mensajes que reflejan lo ocurrido.</p>	

Tabla B.14. Prueba de aceptación # 14

Caso de prueba de aceptación	
Código: HU14_P1	Historia de usuario: 14
Nombre: Alertas del sistema.	
Descripción: Prueba para la funcionalidad alertas del sistema.	
<p>Condiciones de ejecución: El usuario debe estar autenticado.</p>	
<p>Pasos de ejecución: Se realizan varias acciones.</p>	

Continúa en la próxima página

Tabla B.14. Continuación de la página anterior

Resultados esperados: El sistema cambia los valores del usuarios editables. El sistema debe listar en la página principal de los involucrados el evento realizado.

Tabla B.15. Prueba de aceptación # 15

Caso de prueba de aceptación	
Código: HU15_P1	Historia de usuario: 15
Nombre: Configurar opciones avanzadas.	
Descripción: Prueba para la funcionalidad configurar opciones avanzadas.	
Condiciones de ejecución: El usuario debe estar autenticado. El rol del usuario debe ser supervisor.	
Pasos de ejecución: Se intentan editar los parámetros de configuración del correo de salida del sistema y el contenido predeterminado de las notificaciones con valores válidos.	
Resultados esperados: El sistema cambia los valores del usuarios editables. Se cambian los parámetros de configuración del sistema. En cada caso el sistema muestra mensajes que reflejan lo ocurrido.	

Tabla B.16. Prueba de aceptación # 16

Caso de prueba de aceptación	
Código: HU16_P1	Historia de usuario: 16
Nombre: Gestionar equipos de servicios generales.	
Descripción: Prueba para la funcionalidad gestionar equipos de servicios generales.	
Condiciones de ejecución: El usuario debe estar autenticado. El rol del usuario debe ser supervisor.	
Pasos de ejecución: Se accede a la funcionalidad. Se intenta adicionar un nuevo equipo. Se intenta editar un equipo existente. Se intenta eliminar un equipo existente.	
Resultados esperados: Se muestra una página con el listado de los equipos registrados en el sistema de servicios generales. Se adiciona un nuevo equipo al sistema. Se edita el nombre del equipo seleccionado. Se elimina el equipo seleccionado. En cada caso el sistema muestra mensajes que reflejan lo ocurrido.	

Tabla B.17. Prueba de aceptación # 17

Caso de prueba de aceptación	
Código: HU17_P1	Historia de usuario: 17
Nombre: Gestionar equipos de servicios a color.	
Descripción: Prueba para la funcionalidad gestionar equipos de servicios a color.	
Condiciones de ejecución: El usuario debe estar autenticado. El rol del usuario debe ser supervisor.	
Pasos de ejecución: Se accede a la funcionalidad. Se intenta adicionar un nuevo equipo. Se intenta editar un equipo existente. Se intenta eliminar un equipo existente.	
Resultados esperados: Se muestra una página con el listado de los equipos registrados en el sistema de servicios a color. Se adiciona un nuevo equipo al sistema. Se edita el nombre del equipo seleccionado. Se elimina el equipo seleccionado. En cada caso el sistema muestra mensajes que reflejan lo ocurrido.	