

Facultad de Informática y Matemática

***SGIGEO: SISTEMA PARA LA GESTIÓN DE LOS
SERVICIOS DE IMPRESIÓN EN LA EMPRESA GEOCUBA
ORIENTE NORTE***

**Trabajo de diploma para optar por el Título de Ingeniero
Informático**

Autor: Daliuska García Marrero

Tutor: Ing. Roberto Antonio Becerra García

Consultante: Dr.C Jorge García Díaz

Holguín, Cuba

2012

Agradecimientos

A mi hijo, Andy Daniel, por esperarme y comprenderme, siendo aún tan pequeño.

A mi novio, Kiwer, por su amor y su atención. Gracias animarme tantas veces a continuar.

A todos mis amigos, sin mencionar nombres, ustedes saben bien quienes son, les quiero y les aprecio. Gracias por darme su apoyo.

A todos mis compañeros del aula, por todo el tiempo en que estudiamos, aprendimos y trabajamos juntos.

A Rey, por estar siempre que lo necesité.

A todos mis profes de estos años, por enseñarme y ayudarme. Gracias.

A Héctor y Camilo por su inmensa y oportuna ayuda sin estar graduados aún.

Y de forma muy especial a mi tutor, Becerra, porque sin él sencillamente no podría ser ingeniero. Mi agradecimiento eterno. Gracias.

Dedicatoria

Con infinito amor a mi hijo que es la motivación principal de mi buen comportamiento al intentar darle un buen ejemplo a seguir.

A mi abuela Ana por preocuparse, ocuparse y cuidar tanto de mi hijo, aún en su vejez.

A mis tíos, Daniel y Yoly, que siempre están cuando los necesito y confían en mí.

A Kiwer, gracias por tu inmenso amor, tu cariño, tu paciencia, tu apoyo incondicional. Y por estar siempre a mi lado cuando lo necesité.

A mi papá, por su apoyo incondicional y su amor.

Resumen

La Empresa GEOCUBA Oriente Norte tiene como política garantizar productos y servicios de calidad. Para mejorar los servicios que ofrece es necesario facilitar el control de las impresiones que se realizan en la misma con los diferentes medios, así como evaluar la factibilidad de las impresiones, además permita controlar con facilidad los materiales utilizados, volúmenes ejecutados según las características de impresión entre otros. Actualmente, el proceso de gestión de información de los servicios de impresión es realizado por un trabajador, resultando complejo. La información se almacena de forma manual y en formato digital, por lo que es complicado hacer búsquedas y actualizaciones. La forma en que se conserva, no garantiza protección y no asegura su disponibilidad para un acceso futuro. Esta investigación tiene como objetivo el desarrollo de un Producto Informático para optimizar el proceso de gestión de impresiones en la Empresa GEOCUBA Oriente Norte. Con el desarrollo del Sistema para la gestión de los servicios de impresión, se logró un medio que unificará los datos y garantizará la integridad y consistencia de los mismos, que posibilitará el acceso a la información fácilmente, y gestionarla de forma óptima. Para la implementación se utilizó el *framework* de desarrollo web de Python, Django, en conjunto con tecnologías de presentación como Twitter Bootstrap en la que se utilizan tecnologías modernas como HTML5 y CSS3. Como Gestor de Bases de Datos PostgreSQL y Apache 2 como servidor web. Para guiar el proceso ingenieril se usó la Metodología ágil de desarrollo de software XP.

Abstract

The Company GEOCUBA Oriente Norte she has as politics to guarantee products and services of quality. For it improved the services that she offers it is necessary to facilitate the control of the impressions that are carried out in the same one with the different means, as well as to evaluate the feasibility of the impressions, also allow to control with easiness the utilized materials, executed volumes according to the impression characteristics among others. At the moment, the process of administration of information of the impression services is carried out by a worker, being complex. The information is stored in a manual way and in digital format, for what is complicated to make searches and upgrades. The form in that is conserved, doesn't guarantee protection and it doesn't assure its readiness for a future access. This investigation has as objective the development of a Computer Product to optimize the process of administration of impressions in the Company GEOCUBA Oriente Norte. With the development of the System for the administration of the impression services, a means was achieved that it will unify the data and it will guarantee the integrity and consistency of the same ones that it will facilitate the access easily to the information, and to negotiate it in a good way. For the implementation the framework of development web of Python, Django, was used together with presentation technologies like Twitter Bootstrap in which modern technologies are used as HTML5 and CSS3. As Agent of Databases PostgreSQL and Apache 2 as servant web. To guide the engineering process the agile Methodology of software development XP it was used.

Índice

Introducción.....	1
Capítulo 1. Aspectos teóricos que fundamentan la investigación.....	6
1.1 Introducción	6
1.2 Sistemas de Gestión de Información	6
1.3 Sistemas existentes para la gestión de impresiones	7
1.4 Tecnologías y herramientas disponibles.....	8
1.4.1 Sistemas de Escritorio vs Web.....	8
1.4.2 Arquitectura Cliente/Servidor	9
1.4.3 Principales tecnologías y lenguajes de desarrollo web	10
1.4.3 Frameworks de desarrollo.....	13
1.4.4 Servidores de Aplicaciones Web.....	14
1.4.5 Entorno de Desarrollo Integrado	15
1.4.6 Herramientas CASE	17
1.4.7 Sistemas Gestores de Bases de Datos.....	18
1.5 Metodologías de desarrollo disponibles	19
1.5.1 Metodologías ágiles vs tradicionales.....	20
1.5.2 Principales características de las metodologías ágiles	23
1.5.3 Programación Extrema.....	25
1.6 Conclusiones del capítulo.....	32
Capítulo 2. Exploración y Planificación	33
2.1 Introducción.....	33
2.2 Exploración.....	33
2.2.1 Descripción de requisitos	33

2.2.1.1 Requerimientos Funcionales del Sistema	34
2.2.1.2 Requerimientos no Funcionales del Sistema	35
2.2.2 Personas relacionadas con la aplicación	36
2.2.3 Historias de Usuario	36
2.3 Planificación	39
2.3.1 Iteraciones.....	39
2.3.2 Plan de entrega.....	40
2.4 Conclusiones del capítulo.....	41
Capítulo 3. Implementación y pruebas	42
3.1 Introducción.....	42
3.2 Diseño del sistema	42
3.2.1 Modelo del dominio	42
3.2.2 Arquitectura.....	43
3.2.3 Diagrama de Despliegue.....	43
3.2.4 Diagrama de Paquetes.....	44
3.3 Implementación	45
3.3.1 Iteración 1	45
3.3.2 Iteración 2	45
3.3.3 Iteración 3	46
3.3.4 Iteración 4	47
3.4 Pruebas	48
3.4.1 Pruebas de aceptación.....	48
3.5 Valoración de la propuesta.....	51
3.6 Conclusiones del capítulo.....	54

Conclusiones Generales	55
Recomendaciones.....	56
Bibliografía	57
Anexos	59
Anexo 1: Tareas	59
Iteración 1	59
Anexo 2: Tareas	60
Iteración 2	60
Anexo 3: Tareas	62
Iteración 3	62
Anexo 4: Tareas	68
Iteración 4	68

Índice de tablas

Tabla 1: Comparación entre lenguajes de programación (Java, C#, PHP, Python, Ruby).....	12
Tabla 2: Comparación entre metodologías ágiles y tradicionales	23
Tabla 3: Prototipo de Historia de Usuario.....	29
Tabla 4: Prototipo de Tarea.....	29
Tabla 5: Prototipo de Prueba de Aceptación	30
Tabla 6: Historia de Usuario # 1 “Control de Acceso”	37
Tabla 7: Historia de Usuario #2 “Funcionalidad del Cliente”	37
Tabla 8: Historia de Usuario #3 “Gestión de Impresiones”	37
Tabla 9: Historia de Usuario #4 “Gestión de Tipos de Papel”	38
Tabla 10: Historia de Usuario #5 “Gestión de Proyectos”	38
Tabla 11: Historia de Usuario #6 “Gestión de Proyectos”	38
Tabla 12: Historia de Usuario #7 “Gestión de Usuarios”	39
Tabla 13: Historia de Usuario #8 “Generación de Reportes”	39
Tabla 15: Cronograma de entregas.....	41
Tabla 16: Prueba 1 de la HU “Control de Acceso”	49
Tabla 17: Prueba 2 de la HU “Funcionalidad del Cliente”	50
Tabla 18: Prueba 3 de la HU “Gestión de Impresiones”	51
Tabla 19: Prueba 4 de la HU “Generación de Reportes”	51
Tabla 20: Tarea # 1 “Entrar al Sistema”	59
Tabla 21: Tarea # 2 “Salir del Sistema”	59
Tabla 22: Tarea # 3 “Mostrar datos del usuario”	60
Tabla 23: Tarea # 4 “Cambiar contraseña del usuario”	60
Tabla 24: Tarea # 5 “Plantilla base de la interfaz de usuario”	60
Tabla 25: Tarea # 6 “Solicitar Impresión”	61
Tabla 26: Tarea # 7 “Mostrar impresiones pendientes”	61
Tabla 27: Tarea # 8 “Mostrar impresiones solicitadas”	61
Tabla 28: Tarea # 9 “Listar Impresiones”	62
Tabla 29: Tarea #10 “Adicionar Impresión”	62
Tabla 30: Tarea # 11 “Mostrar Impresión”	63

Tabla 31: Tarea # 12 “Actualizar Impresión”	63
Tabla 32: Tarea # 13 “Eliminar Impresión”	63
Tabla 33: Tarea # 14 “Listar Tipos de Papel”	64
Tabla 34: Tarea #15 “Adicionar Tipos de Papel”	64
Tabla 35: Tarea # 16 “Mostrar Tipos de Papel”	64
Tabla 36: Tarea # 17 “Actualizar Tipos de Papel”	65
Tabla 37: Tarea # 18 “Eliminar Tipos de Papel”	65
Tabla 38: Tarea # 19 “Listar Proyectos”	65
Tabla 39: Tarea # 20 “Adicionar Proyecto”	66
Tabla 40: Tarea # 21 “Mostrar Proyecto”	66
Tabla 41: Tarea # 22 “Actualizar Proyecto”	66
Tabla 42: Tarea # 23 “Eliminar Proyecto”	67
Tabla 43: Tarea # 24 “Listar Documentos”	67
Tabla 44: Tarea # 25 “Adicionar Documento”	67
Tabla 45: Tarea # 26 “Mostrar Documento”	68
Tabla 46: Tarea # 27 “Actualizar Documento”	68
Tabla 47: Tarea # 28 “Eliminar Documento”	68
Tabla 48: Tarea # 29 “Listar Usuarios”	69
Tabla 49: Tarea # 30 “Adicionar Usuario”	69
Tabla 50: Tarea # 31 “Mostrar Usuario”	69
Tabla 51: Tarea # 32 “Actualizar Usuario”	70
Tabla 52: Tarea # 33 “Eliminar Usuario”	70
Tabla 53: Tarea # 34 “Reporte Diario”	70
Tabla 54: Tarea # 35 “Reporte Semanal”	71
Tabla 55: Tarea # 36 “Reporte Mensual”	71
Tabla 56: Tarea # 37 “Reporte por Mes”	71
Tabla 57: Tarea # 38 “Reporte por Proyecto”	72
Tabla 58: Tarea # 39 “Reporte por Usuario”	72

Índice de Ilustraciones

Ilustración 1: Proceso de desarrollo de XP.....	25
Ilustración 2: Modelo del Dominio	43
Ilustración 3: Diagrama de Despliegue.....	44
Ilustración 4: Diagrama de Paquetes	44

Introducción

El uso de las Tecnologías de la Información y las Comunicaciones (TICs) se acrecienta y se extiende sin detenerse, sobre todo en los países desarrollados. Esto puede provocar no sólo un incremento considerable de la brecha digital, sino también social. Tal es su repercusión que la era actual es llamada la Era o Edad de la Información. De ahí la importancia de no negarse al desarrollo y uso de estas tecnologías, sobre todo en los países subdesarrollados que deben trazarse estrategias para no quedarse detrás en esta área y evitar así que se acentúe esta brecha.

En las últimas décadas, se ha generalizado la utilización de ambos medios: la computación y las comunicaciones, para lograr un progresivo avance en la gestión de la información y la posibilidad de administrarla desde cualquier lugar. De ahí la existencia de sitios y aplicaciones Web como expresión de un sistema de información, que pretende facilitar la organización y el acceso a los contenidos y servicios disponibles. Estas no solo muestran información estática o escrita con anterioridad a través de Internet o una Intranet, sino que son capaces de generar datos solicitados a través de consultas, debido a la dinámica sobre la que están basadas. Realizan determinadas tareas específicas según los requerimientos del usuario o cliente que solicite el servicio.

La informática se ha convertido en la actualidad en una de las ciencias que ha proporcionado una gran cantidad de avances en todos los procesos de cambio tecnológico. En el mundo, prácticamente nada es concebido sin la utilización de herramientas informáticas, por lo que cada vez más existen sitios y aplicaciones Web para facilitar la organización, el manejo y el acceso a los contenidos y datos, disponibles en las diferentes entidades y empresas de la sociedad. Dentro de este contexto la Empresa GEOCUBA Oriente Norte, perteneciente al esquema empresarial de las FAR en Cuba, demanda de estas herramientas informáticas en la actualidad.

La Empresa GEOCUBA Oriente Norte tiene como política garantizar productos y servicios de calidad. Brinda soluciones integradas a la sociedad y sus clientes

como soporte georeferencial en las esferas de la Geodesia, la Topografía, la Cartografía, el Catastro, la Hidrografía y los Estudios Marinos; y la Ayuda a la Navegación Marítima.

Provee soluciones geomáticas, Sistemas de Información Geográfica y servicios de aplicaciones en Infraestructura de Datos Espaciales.

Realiza Estudios del Medio Geográfico y de Soluciones Ambientales para su conservación, de Saneamiento Ambiental, de Estudios y Planes de Reducción de Desastres para prevenir o mitigar los peligros naturales o tecnológicos, la Vulnerabilidad y Riesgo ante catástrofes o eventos meteorológicos extremos. Desarrolla y presta servicios Científicos-Tecnológicos especializados, consultoría, asesoría técnica y las investigaciones afines con las geociencias. Además, comercializa servicios editoriales especializados y productos de Artes Gráficas.

Para garantizar las salidas de los productos que se refieren, se necesitan servicios internos que aseguren la calidad de lo que se produce. Uno de los diversos servicios que se ofrecen en la empresa es el de impresiones; el proceso para su gestión se realiza actualmente de forma manual (en papel), teniendo como principales riesgos la falta de centralización, integridad y seguridad en la información que se maneja; y gran dificultad a la hora de consultar información útil y de calidad requerida por la dirección de la empresa a la hora de tomar decisiones. Esta problemática conlleva al gasto innecesario de materiales (tanto de oficina como energéticos), afectación a la calidad de los servicios que se brindan, tanto a clientes internos como externos, y violaciones a la seguridad informática debido a que se dificulta la implementación de los protocolos de seguridad requeridos por el MINFAR.

Las consideraciones anteriores revelan el siguiente **problema científico**: ¿Cómo mejorar la gestión del proceso de impresiones en la Empresa GEOCUBA Oriente Norte?

El **objeto de estudio** en el cual se enmarca el problema planteado es el proceso de gestión de impresiones en un esquema empresarial.

Para dar solución al problema se propone como **objetivo** el desarrollo de un Sistema Informático para optimizar el proceso de gestión de impresiones en la Empresa GEOCUBA Oriente Norte.

A partir del objetivo se delimita como **campo de acción** la optimización del proceso de gestión de impresiones en la Empresa GEOCUBA Oriente Norte.

Para guiar la investigación se plantean las siguientes **preguntas científicas**:

- ¿Cuáles son los fundamentos teóricos que sustentan el proceso de gestión de impresiones en la Empresa GEOCUBA Oriente Norte?
- ¿Cómo desarrollar un sistema informático que informatice el proceso de gestión de impresiones en la Empresa GEOCUBA Oriente Norte?
- ¿Será sostenible la herramienta informática que se propone?

Para darle respuesta a las preguntas planteadas y cumplir el objetivo trazado, se realizaron las siguientes **tareas**:

1. Determinar los fundamentos teóricos y características sobre procesos de gestión de impresiones en la Empresa GEOCUBA Oriente Norte
2. Determinar el estado actual de los sistemas de información y tecnologías disponibles en la actualidad sobre el objeto de estudio planteado
3. Elaborar cronograma de actividades
4. Levantar requisitos y definir Historias de Usuario
5. Valorar la sostenibilidad de la propuesta
6. Ejecutar iteraciones.
7. Valorar la propuesta mediante pruebas de aceptación

Para darles solución a las tareas planteadas se usaron una combinación de métodos teóricos y empíricos.

Métodos teóricos:

Análisis y síntesis: Se emplea para procesar la información sobre los fundamentos teóricos, estado del arte y las conclusiones.

Histórico lógico: Se aplica para el estudio de la evolución del problema y la existencia de sistemas de gestión para el control de las impresiones similares al que se pretende desarrollar.

Enfoque sistémico: Es empleado para el análisis y determinación de las relaciones funcionales, dependencias y la modelación con un carácter sistémico.

Métodos empíricos:

Entrevistas: Se utiliza para conocer la interacción directa con el personal interesado en el sistema, recoger las informaciones importantes, las necesidades del Jefe de Agencia para ayudar en su labor y determinar los principales requerimientos del sistema.

Consulta de Documentos: Se emplea para determinar la información que será gestionada constante en el sistema y las fuentes bibliográficas consultadas para realizar la investigación.

Observación: Permite la comprensión de los procesos que se desean informatizar.

La tesis consta de introducción; 3 capítulos; conclusiones; recomendaciones; bibliografía y anexos. A continuación se aborda el contenido de los capítulos.

Capítulo1: Aspectos teóricos que fundamentan la investigación.

En este capítulo se tratan los aspectos teóricos que contienen los fundamentos necesarios para una eficiente comprensión de los temas tratados en el resto del documento. Además, sirve como soporte teórico a la construcción del sistema y se caracterizan las tecnologías y herramientas a utilizar en el desarrollo de la aplicación.

Capítulo 2: Exploración y Planificación

Se presenta una propuesta del sistema a desarrollar, se muestran los requerimientos de la propuesta de solución mediante las historias de usuario que proporcionaran los detalles sobre la estimación del riesgo y una planificación del tiempo que conllevará la implementación de dicha historia de usuario.

Capítulo 3: Implementación y Pruebas.

En el presente capítulo se muestra el diseño del sistema y se describen los artefactos relacionados con la implementación, se detallan las iteraciones para el desarrollo del sistema, exponiéndose las tareas generadas en cada una de ellas por cada historia de usuario y se describen las pruebas realizadas a la aplicación. Para ello se diseñan los casos de pruebas de aceptación a los que será sometida la aplicación en cada una de las iteraciones.

Capítulo 1. Aspectos teóricos que fundamentan la investigación.

1.1 Introducción

Este capítulo se dedica al estudio y fundamentación de las diferentes técnicas y herramientas de la ingeniería de software que se utilizarán para el desarrollo de la aplicación informática y se fundamentan los lenguajes de programación que se utilizarán para la implementación. De igual forma se describen las tecnologías y Framework necesarios para el desarrollo de la misma, así como el Sistema Gestor de Bases de Datos que se usará para lograr la persistencia de los datos.

1.2 Sistemas de Gestión de Información

Un tema recurrente en la actualidad es el estar en la llamada Era de la Información, donde este recurso juega un papel determinante. Para una organización en específico, la información constituye el principio para cumplir la satisfacción de sus objetivos y metas. Si bien información y conocimiento constituyen los dos pilares vitales que sustentan el trabajo de cualquier organización y que hacen realidad su producción, servicios o razón social, su gestión demanda un riguroso trabajo de diseño, organización, control y comunicación a fin de que sus diferentes miembros y usuarios puedan obtener el máximo de beneficios con el mínimo de costos.

Un sistema de información es un grupo de elementos utilizados para la administración de datos, los cuales se encuentran coordinados entre sí para su uso posterior. Estos elementos, además, fueron creados con el objeto de lograr un determinado fin. Desde el punto de vista empresarial, los sistemas de información tienen como propósito perfeccionar las actividades llevadas a cabo en una organización, y así alcanzar ventajas competitivas.

Los Sistemas de Información que logran la automatización de procesos operativos dentro de una organización, son llamados frecuentemente Sistemas Transaccionales, ya que su función primordial consiste en procesar transacciones tales como pagos, cobros, pólizas, entradas, salidas, entre otras. Por otra parte, los Sistemas de Información que apoyan el proceso de toma de decisiones son

los Sistemas de Soporte a la Toma de Decisiones, Sistemas para la Toma de Decisión de Grupo, Sistemas Expertos de Soporte a la Toma de Decisiones y Sistema de Información para Ejecutivos. El tercer tipo de sistema, de acuerdo con su uso u objetivos que cumplen, es el de los Sistemas Estratégicos, los cuales se desarrollan en las organizaciones con el fin de lograr ventajas competitivas, a través del uso de la tecnología de información [1].

1.3 Sistemas existentes para la gestión de impresiones

Hasta hace muy poco tiempo las empresas se limitaban a instalar hardware para resolver sus problemas de impresión y sólo se preocupaban por el tipo y la cantidad de equipos que adquirirían; lo que implicaba dar la espalda a las evoluciones que se producían en el mundo del software documental, que se dirigían hacia un control más sofisticado del uso de los recursos de impresión de documentos.

Afortunadamente, esta situación ha ido cambiando y cada vez son más las organizaciones que conocen los beneficios de controlar sus recursos de impresión mediante soluciones de software.

En la actualidad existen varias aplicaciones relacionadas con el control de servicios de impresión, por solo mencionar algunas se hallan **PrintManager** [2] y **PennCentral** [3].

PrintManager surge para permitir efectuar un control, es un sistema de control de impresiones en entornos Windows que permite monitorizar los recursos de impresión así como restringir y cobrar por su uso mediante monedero electrónico.

PennCentral es un software para el Print Output Management que permite realizar un control exhaustivo de todos los trabajos que se imprimen en una organización; este software hace posible centralizar el control de todas las impresiones desde un único servidor y permite además la impresión bajo demanda, de manera que una empresa podrá retener los trabajos hasta que el usuario propietario se identifique en la máquina.

1.4 Tecnologías y herramientas disponibles

Las herramientas, tecnologías y metodologías son componentes fundamentales para la realización de cualquier software, por lo que es importante el uso correcto de dichos componentes con el objetivo de obtener productos con la calidad requerida.

1.4.1 Sistemas de Escritorio vs Web

En la actualidad se manejan 3 tipos de aplicaciones, según el entorno en el que se despliega están: aplicaciones web, aplicaciones de escritorio y aplicaciones móviles (entorno que empieza a tomar fuerza en la actualidad). Las aplicaciones empresariales y sistemas de gestión poseen hoy mucha más fuerza en la Web y el escritorio respectivamente, que en el campo de los móviles. A continuación listamos las principales ventajas y desventajas de cada una de estas:

Aplicaciones Escritorio

Ventajas

- Interfaces de usuario ricas y eficientes.
- No necesaria infraestructura de red para múltiples clientes.

Desventajas

- Despliegue complejo de múltiples clientes (ya que se necesita instalar la aplicación en cada uno de los clientes).
- Tiempo de desarrollo de aplicaciones empresariales alto.

Aplicaciones Web

Ventajas

- Fácil despliegue de múltiples clientes (solo se necesita un navegador web instalado en la máquina cliente y un servidor).
- Gran cantidad de *frameworks* y tecnologías disponibles que facilitan el trabajo para el desarrollo de aplicaciones empresariales.
- Sistema de seguridad centralizado y manejado usualmente por los servidores

web.

- Tiempo de desarrollo para aplicaciones empresariales medio.

Desventajas

- Infraestructura de red necesaria para múltiples clientes.

Como se ha podido observar, las aplicaciones Web son más adecuadas para el desarrollo de aplicaciones empresariales o de gestión de información que su contraparte de escritorio, o al menos esa es la tendencia actual de este tipo de sistemas. Por lo tanto se propone el desarrollo de una aplicación Web para darle solución a la problemática planteada [4].

1.4.2 Arquitectura Cliente/Servidor

La arquitectura cliente/servidor es un modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes. Los principales componentes del esquema cliente/servidor son los Clientes, los Servidores y la infraestructura de comunicaciones. En este modelo, las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario.

Entre las principales características de la arquitectura cliente/servidor se pueden destacar las siguientes:

- El servidor presenta a todos los clientes una interface única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interface externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente [5].

1.4.3 Principales tecnologías y lenguajes de desarrollo web

Lenguajes de Programación

Los lenguajes de programación son el medio o idioma empleado por los programadores para transformar modelos o abstracciones a productos “tangibles”, por lo que su correcta selección puede tener un impacto alto en la productividad del proceso de desarrollo[6].

En el plano de las aplicaciones Web se habla de forma habitual de lenguajes del lado del cliente y lenguajes del lado del servidor.

Los **lenguajes del lado del cliente** son aquellos que pueden ser directamente interpretados por el navegador Web. Estos lenguajes no necesitan de un servidor. Dentro de los lenguajes al lado del cliente se encuentran: JavaScript, HTML, Visual Basic Script.

Las siglas HTML, en inglés (HyperText Markup Language), significan en español lenguaje de marcado de hipertexto, es un estándar utilizado en todo el mundo cuyas normas define un organismo llamado W3C (World Wide Web Consortium).

HTML es un lenguaje de composición de documentos que se encarga de definir una sintaxis e insertar instrucciones especiales indicándole al navegador como desplegar el contenido en el documento, incluyendo texto, imágenes y otros medios soportados. También brinda la posibilidad de lograr que un documento sea interactivo a través de ligas especiales de hipertexto, las cuales conectan distintos documentos, ya sea en su computadora o en otras, así como otros recursos de Internet, como FTP[7].

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas Web dinámicas. Una página Web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso usuario. JavaScript es un lenguaje de programación interpretado, por lo que no se debe compilar para ejecutarlos. En otras palabras, los programas escritos con

JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.

Twitter Bootstrap es un *framework* del lado del cliente que nos facilita la integración de tecnologías como HTML5 y CSS3 en aplicaciones web. En este se utiliza JQuery como API JavaScript para darle mayor dinamismo y presencia a este tipo de aplicaciones.

Para facilitar el trabajo y disminuir tiempos de desarrollo se decide utilizar Twitter Bootstrap como *framework* del lado del cliente.

Los lenguajes del lado del servidor son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato para que este muestre los resultados. Existen varios lenguajes que compiten por el dominio de este sector del mercado como son: Java, C#, PHP, Python, Ruby.

PHP es un acrónimo de (Personal Home Page, Procesador de Hipertexto), Pre-procesador de Hipertexto. Es un lenguaje de script incrustado dentro del HTML que se ejecuta del lado del servidor, el mismo actualmente es uno de los más usados en la creación de Web dinámicas.

Básicamente su interacción es la siguiente, el cliente hace una petición al servidor con el objetivo de que le envíe una página Web, el servidor ejecuta el intérprete de PHP, éste procesa el script solicitado que generará el contenido de manera dinámica. El resultado es enviado por el intérprete al servidor, y el servidor se lo envía al cliente [8].

Java es un lenguaje de programación orientado a objetos. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

Python es un lenguaje de programación de alto nivel cuya filosofía hace hincapié en una sintaxis muy limpia y que favorece un código legible.

Capítulo 1. Aspectos teóricos que fundamentan la investigación

A continuación se ofrece una comparativa de las características relevantes a la hora de seleccionar cuál de ellos utilizar para el desarrollo de la aplicación:

Característica	Java	C#	PHP	Python	Ruby
Software Libre	No	No	Sí	Sí	Sí
Multiplataforma	Sí	No	Sí	Sí	Sí
Ejecución	Bytecode ¹	CIL ² / JIT ³	Interpretado	Interpretado / Bytecode	Interpretado / Bytecode
Tipo de Datos	Estático	Ambos	Dinámico	Dinámico	Dinámico
Cantidad de bibliotecas existentes	Alta	Medio	Medio	Alta	Baja
Curva de Aprendizaje	Media	Media-Alta	Media	Baja	Alta
Conocido por el equipo de desarrollo	Sí	No	No	Sí	No

Tabla 1: Comparación entre lenguajes de programación (Java, C#, PHP, Python, Ruby)

Debido a las particularidades mostradas en la tabla anterior y el conocimiento previo de esta por el equipo de desarrollo se elige el lenguaje de programación Python como solución del lado del servidor.

¹ Código intermedio que permite interpretar el código escrito en varias plataformas de forma más eficiente que un intérprete puro.

² **Common Intermediate Language:** Lenguaje intermedio que permite que varios lenguajes compartan las mismas bibliotecas y entorno de ejecución.

³ **Just in Time Compilation:** Mecanismo que compila el código CIL a código nativo antes de ejecutar la aplicación.

1.4.3 Frameworks de desarrollo

Un Framework de desarrollo se encarga de definir y dar soporte a gran parte de los procesos que se llevan a cabo con frecuencia, es decir, implementa los principales patrones de la programación, tanto desde la visión de la arquitectura de un software hasta las implementaciones más usadas, brinda un conjunto de clases y librerías para ayudar a desarrollar y unir los diferentes componentes de un proyecto permitiendo que programadores y diseñadores se concentren en los requerimientos del software que se desea hacer y no tratando con los detalles de bajo nivel al proveer un sistema funcional. Facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas [9]

Para el lenguaje de programación Python existen varios frameworks, cada uno con sus ventajas y desventajas. A continuación se muestra un análisis de 3 de los más populares existentes para este lenguaje de programación: Django, Zope y TurboGear.

Django surge en el 2003, es un framework de desarrollo web de código abierto, escrito en Python, que cumple en cierta medida el patrón del Modelo Vista Controlador.

La meta fundamental de Django es facilitar la creación de sitios web complejos. Django pone énfasis en el re-uso, la conectividad, extensibilidad de componentes y del desarrollo rápido.

Zope es un framework para el desarrollo de páginas Web de código abierto, escrito en Python, que integra todo lo necesario para construir desde la web más sencilla hasta la aplicación más compleja. Lo más característico de Zope es su base de datos orientada a objetos, llamada ZODB o Zope Object Database. El enfoque más utilizado en el desarrollo de Webs con Zope es la creación de aplicaciones Web.

TurboGear fue creado en el 2005, es un framework para el desarrollo Web. Es un stack Web completo, que abarca desde el javascript del cliente hasta un mapper

relacional-objetos para la base de datos.

Teniendo en cuenta las consideraciones antes expuestas se selecciona Django como framework de desarrollo del lado del servidor.

1.4.4 Servidores de Aplicaciones Web

Un servidor Web es un programa que implementa el Protocolo de Transferencia de Hipertextos (HTTP, por sus siglas en inglés). Este protocolo está diseñado para transferir lo que se llama hipertextos, páginas Web o páginas de Lenguaje de Marcas Hipertextuales (HTML, por sus siglas en inglés): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos. Un servidor Web se encarga de mantenerse a la espera de peticiones HTTP llevada a cabo por un cliente HTTP que se suele conocer como navegador.

Las aplicaciones de servidor suelen ser la opción por la que se opta en la mayoría de las ocasiones para realizar aplicaciones Web. La razón es que, al ejecutarse ésta en el servidor y no en la máquina del cliente, éste no necesita ninguna capacidad adicional, como sí ocurre en el caso de querer ejecutar aplicaciones javascript o java. Así cualquier máquina cliente dotada de un navegador Web puede utilizar este tipo de aplicaciones. Algunos de los servidores Web más importantes son: Apache, Internet Information Server y el Cherokee [4].

Servidor Apache

Apache es un servidor Web HTTP de código abierto y multiplataforma. Este servidor Web está hecho con robustez y estabilidad por lo que tiene en estos momentos gran popularidad entre las empresas y desarrolladores Web, en lo que también ha influido la posibilidad de ser configurable.

El servidor Apache es un software que está estructurado en módulos. La configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo, los mismos se pueden clasificar en tres categorías:

Módulos Base: Módulo con las funciones básicas del Apache.

Capítulo 1. Aspectos teóricos que fundamentan la investigación

Módulos Multiproceso: Son los responsables de la unión con los puertos de la máquina, aceptando las peticiones y enviando a los hijos a atender a las peticiones.

Módulos Adicionales: Cualquier otro módulo que le añada una funcionalidad al servidor.

Por defecto se encuentran incluidos los módulos base y multiproceso:

- *Autenticación:* La identificación positiva de una entidad de red tal como un servidor, un cliente, o un usuario.
- *Control de Acceso:* La restricción en el acceso al entorno de una red. En el contexto de Apache significa normalmente la restricción en el acceso a ciertas URLs.
- *Algoritmo:* Un proceso definido sin ambigüedades o un conjunto de reglas para solucionar un problema en un número finito de pasos. Los algoritmos para encriptar se llaman normalmente algoritmos de cifrado.

Ventajas de Apache:

- Se ejecuta correctamente en la mayoría de los Sistemas Operativos.
- Es una tecnología gratuita de código fuente abierto.
- Este servidor es altamente configurable de diseño modular.
- Existe una gran variedad de módulos para Apache.
- Apache permite adecuar las respuestas ante los posibles errores que se puedan dar en el servidor.

Permite crear archivos los personalizados en correspondencia con las necesidades administrador [10].

1.4.5 Entorno de Desarrollo Integrado

Entorno de Desarrollo Integrado (IDE, por sus siglas en inglés), es un programa compuesto por un conjunto de herramientas para un programador. Un IDE es un entorno de programación que ha sido empaquetado como un programa de

aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de Interfaz de Usuario Gráfica (GUI, por sus siglas en inglés). Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes.

Los IDEs proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, Java, Delphi, Visual Basic, etc. En algunos lenguajes, un IDE puede funcionar como un sistema en tiempo de ejecución, donde se permite utilizar el lenguaje de programación en forma interactiva.

Puede dedicarse especialmente para un lenguaje de programación o bien para varios. Las opciones más interesantes para la elección de un IDE son Eclipse y Netbeans.

Eclipse

Eclipse es un IDE de código abierto independiente de una plataforma, emplea plug-ins para proporcionar toda su funcionalidad, a diferencia de otros entornos donde las funcionalidades están todas incluidas, las necesite el usuario o no. Eclipse es únicamente una armazón sobre la que se pueden montar herramientas de desarrollo. La arquitectura de plug-ins, permite además de integrar varios lenguajes, introducir otras aplicaciones accesorias que pueden resultar útiles durante el proceso de desarrollo, tales como: herramientas UML, editores visuales de interfaces, ayuda en línea para librerías, entre otros [11].

Netbeans

Netbeans es una herramienta de desarrollo para aplicaciones, escrita únicamente sobre la base de la tecnología Java, de forma que puede ejecutarse en cualquier ambiente que ejecute Java. Es un producto libre y gratuito, sin restricciones de uso de código abierto [12].

Después del análisis de los IDEs anteriormente mencionados se determina usar Eclipse por las facilidades del plug-in PyDev para el trabajo con Python.

1.4.6 Herramientas CASE

Una buena parte del tiempo de desarrollo de un sistema informático se dedica al diseño. Existen varias herramientas y lenguajes que se han creado para facilitar esa parte del trabajo. Ese es el caso de la Ingeniería de Software Asistida por Computadoras (CASE, por sus siglas en inglés).

Esto no es más que la aplicación científica de un conjunto de herramientas y métodos a un sistema de software, el cual se quiere que resulte en un producto de software de alta calidad, libre de defectos y mantenible [13].

También se refiere a los métodos de desarrollo de sistemas de información junto con herramientas automáticas que pueden usarse en el proceso de desarrollo de software [14].

El Lenguaje Unificado de Modelado (UML, por sus siglas en inglés) es un lenguaje gráfico para visualizar, especificar, construir y documentar artefactos de un sistema de software. Este ofrece una vía estándar para escribir los planos de sistemas informáticos, donde se incluyen aspectos conceptuales como procesos de negocios y funciones de sistemas, o aspectos concretos como sentencias en lenguajes de programación, esquemas de bases de datos y componentes de software reusables [15].

Cuando se complementan CASE con el lenguaje UML en una aplicación informática se crea lo que usualmente conocemos como herramienta CASE. Existen una gran variedad de estas en la actualidad, orientada a disímiles usos y metodologías de desarrollo.

Entre ellas destaca por su gran conjunto de funcionalidades e integración con varios IDE el Visual Paradigm. Esta es una poderosa herramienta CASE de diseño UML diseñada para una amplia gama de usuarios, donde se incluyen Ingenieros de Software, Analistas de Sistema, Analistas de Negocio y Arquitectos de Sistema, así como otros interesados en construir software mediante el uso del acercamiento orientado a objetos [16].

Entre sus aciertos se encuentra que implementa la última especificación del lenguaje UML (la versión 2.1), cuenta con soporte para las principales plataformas de escritorio y permite mantener sincronizado código fuente con diagramas.

Existen otras herramientas privativas como el Rational Rose (pionero en este tipo de herramientas) o el Enterprise Architect; y otras libres como el Umbrello UML, el ArgoUML, el BoUML, entre otras; pero las características del Visual Paradigm son más adecuadas para modelar este proyecto el cual propició su selección.

1.4.7 Sistemas Gestores de Bases de Datos

Es el conjunto de programas que tienen como objetivo servir de interfaz entre la base de datos, el usuario y las aplicaciones. Entre las funcionalidades que debe permitir se encuentran:

- Definir una base de datos: Especificar tipos, estructuras y restricciones de datos.
- Construir la base de datos: Guardar los datos en algún medio controlado por el mismo SGBD.
- Manipular la base de datos: Realizar consultas, actualizarla, generar informes [17].

Entre los SGBD comúnmente utilizados en el mundo se encuentran Oracle, MySQL, Microsoft SQL Server, PostgreSQL, InterBase, entre otros.

A continuación se muestran algunos que están dentro de los gestores de bases de mayor aceptación en la actualidad.

Microsoft SQL Server: Es un SGBD basada en el lenguaje SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea. Entre sus características figuran: soporte de transacciones, escalabilidad, estabilidad, seguridad, soporta procedimientos almacenados, incluye un potente entorno gráfico de administración, permite trabajar en modo cliente-servidor, y además permite administrar información de otros servidores de datos. Este sistema incluye una versión reducida, llamada MSDE con el mismo

motor de base de datos pero orientado a proyectos más pequeños. Microsoft SQL Server, al contrario de su más cercana competencia (Oracle), no es multiplataforma, ya que sólo está disponible en Sistemas Operativos de Microsoft [18].

MySQL: Desarrollado por la empresa MySQL AB, se disputa con PostgreSQL el puesto de SGBD más conocido y usado de código libre, el sistema se caracteriza por su amplio subconjunto del lenguaje SQL, disponibilidad en gran cantidad de plataformas y sistemas, diferentes opciones de almacenamiento, transacciones y claves foráneas, conectividad segura, replicación, búsqueda e indexación de campos de texto. Reduce el costo total de una base de datos propietaria, reduciendo el costo de las licencias por sobre el 90% y es un excelente motor de datos [19].

PostgreSQL: Es uno de los SGBD más veteranos y conocidos del mundo del código libre. Es un sistema de gestión de bases de datos relacional orientado a objetos y libre, publicado bajo la licencia BSD (es la licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution, pertenece al grupo de licencias de software Libre). PostgreSQL no es manejado por una sola empresa sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada PGDG (PostgreSQL Global Development Group, por sus siglas en inglés). PostgreSQL está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado del mundo [20].

Como SGBD para la creación de la aplicación Web se eligió PostgreSQL, debido a las características antes mencionadas y que el mismo satisface las necesidades del cliente y la aplicación que se propone.

1.5 Metodologías de desarrollo disponibles

Según la literatura, una metodología de desarrollo de software se refiere a los métodos, reglas, postulados, procedimientos y procesos que se usan para administrar un proyecto de ingeniería de software [21]. Estas se dividen en 2 grandes familias: metodologías tradicionales y metodologías ágiles. En los

siguientes epígrafes se hará una comparación entre las dos grandes familias de metodologías, para luego seleccionar la que se utilizará en la confección del sistema en que se desarrollará este trabajo.

1.5.1 Metodologías ágiles vs tradicionales

Las *metodologías ágiles* dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Este enfoque está mostrando su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad [22].

El nombre de metodología ágil viene de cuando en el 2001, 17 metodólogos de procesos tuvieron una reunión para discutir tendencias futuras del desarrollo de software. En esta reunión firmaron lo que se conoce por el *manifiesto ágil* [23], que no es más que los 12 principios básicos de una serie de metodologías que serían conocidas por ágiles para marcar el contraste con las tradicionales o pesadas existentes hasta el momento. Estos principios son los siguientes [24]:

1. Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
2. Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
3. Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al período de tiempo más corto posible.
4. Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
5. Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
6. El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.

7. El software funcionando es la medida principal de progreso.
8. Los procesos ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
9. La atención continua a la excelencia técnica y al buen diseño mejora la agilidad.
10. La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
11. Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
12. A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

En estos principios se puede observar que los mismos se centran en las personas y no tanto en los procesos, así como en entrega rápida de software y no en complejos diseños previendo el futuro. Un ejemplo de metodología ágil es la Programación Extrema (XP)

Las *metodologías pesadas* son consideradas como la vía tradicional de desarrollar software. Estas metodologías se basan en una secuencia de pasos, como definición de requerimientos, construcción de la solución, pruebas y despliegue. Estas metodologías requieren definir y documentar un conjunto estable de requerimientos al comenzar el proyecto [23]. Existen varias metodologías pesadas diferentes, donde las más conocidas son: Cascada (Waterfall), Modelo en Espiral (Spiral Model) y el Proceso Unificado de Desarrollo (Rational Unified Process).

Las metodologías pesadas han existido durante un largo tiempo. Ellas imponen un proceso disciplinado en el desarrollo de software con vistas a que este sea más predecible y eficiente. Estos no se han dado a conocer como muy eficientes y menos aún populares [23] siendo muy criticados por ser metodologías burocráticas, ya que habiendo tanto que seguir en la metodología el proceso de

software como un todo se ralentiza [25].

Existen 4 características comunes de las metodologías pesadas [23]:

- **Enfoque Predictivo:** Las metodologías pesadas tienen la tendencia a planificar primero grandes partes del proceso de software en gran detalle por un largo período de tiempo. Este acercamiento sigue una disciplina de ingeniería donde el desarrollo es predecible y repetitivo.
- **Documentación amplia:** El desarrollo de software tradicional mira los requerimientos como una pieza clave de documentación. Un proceso principal en las metodologías pesadas es el gran diseño por adelantado, en el cuál se cree que es posible recoger todos los requerimientos del cliente por adelantado, antes de escribir ningún código.
- **Orientación al Proceso:** El objetivo de las metodologías pesadas es definir un proceso que trabaje bien para cualquiera que lo utilice. El proceso consistirá en determinadas tareas que deben realizar los administradores, diseñadores, programadores, probadores, etc. Para cada una de estas tareas existe un procedimiento bien definido.
- **Orientación a las Herramientas:** Herramientas de gestión de proyectos, editores de código, compiladores, etc; deben usarse para completar y entregar cada una de las tareas.

Las *metodologías tradicionales* se enfocan más en pensar en el futuro que en entregar software a mediano y corto plazo, como es la necesidad de la mayoría de los usuarios. Además de estar orientados a grandes proyectos donde se tienen gran cantidad de programadores distribuidos en una serie de roles para lograr la organización del trabajo.

Hoy en día es muy importante tener el trabajo realizado en el plazo más corto posible de tiempo. En este aspecto las metodologías ágiles han ido ganando gran popularidad, incluso en grandes empresas como Microsoft, Nokia, Google, etc.

Metodología Ágil	Metodología Tradicional
------------------	-------------------------

Pocos artefactos	Más artefactos
Pocos roles	Más roles
No existe un contrato tradicional	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños trabajando en el mismo sitio	Grandes grupos
Menos énfasis en la arquitectura	La arquitectura es esencial

Tabla 2: Comparación entre metodologías ágiles y tradicionales

Dadas las características analizadas y las particularidades del desarrollador se determina que las metodologías ágiles dan una respuesta adecuada para desarrollar este proyecto.

1.5.2 Principales características de las metodologías ágiles

Existe un número de metodologías de desarrollo ágiles, como las que son apoyadas por La Alianza Ágil (The Agile Alliance). La mayoría de estas metodologías intentan minimizar el riesgo desarrollando software en ciclos de desarrollo cortos llamados iteraciones, los que típicamente duran de una a cuatro semanas.

Cada iteración es como un proyecto de software en miniatura, e incluye todas las tareas necesarias para liberar el mini-incremento de una nueva funcionalidad: planeación, análisis de requerimientos, diseño, codificación, pruebas y documentación. Mientras que una iteración puede no adicionar suficiente funcionalidad que garantice la liberación del producto, un proyecto de software ágil tiende a ser capaz de entregar nuevo software al final de cada iteración. Al final de cada iteración además el equipo puede reevaluar las prioridades del proyecto [26].

Las metodologías ágiles enfatizan en la necesidad de comunicación en tiempo real, preferentemente cara a cara en vez de documentos escritos. La mayoría de los equipos ágiles localizan a sus miembros en un mismo lugar e incluyen las personas necesarias para terminar el software. Como mínimo incluye a programadores y clientes (personas que definen el producto, ellos pueden ser gerentes del producto, analistas de sistema o realmente los propios clientes). También pueden incluirse probadores, diseñadores de iteración, escritores técnicos y gerentes [26].

Además se enfatiza que el software trabajando, es la principal medida de progreso que combinado con la preferencia de la comunicación cara a cara, producen una cantidad muy pequeña de documentación comparadas con otros métodos tradicionales [26].

Entre las metodologías ágiles más destacadas hasta el momento podemos nombrar: ICONIX y XP.

- **ICONIX:** es una metodología conducida por casos de uso, pero no incorpora tantos artefactos UML. Es una metodología relativamente pequeña al igual que XP, pero no descarta las etapas de análisis y diseño. Utiliza UML en sus etapas, de modo que se pueden seguir los requerimientos y adaptarse a nuevos cambios.
- **Programación Extrema (XP):** Es una evolución de los problemas causados por los largos ciclos de desarrollo del modelo tradicional [27]. Después de un número exitoso de ensayos en la práctica, la metodología XP fue teorizada en base a las prácticas y principios utilizados [28].

El ciclo de vida de XP consiste en 5 fases como se pueden observar en la Figura 1: Exploración, Planeación, Iteraciones con Lanzamientos, Producción, Mantenimiento y Muerte. Entre sus roles encontramos: Programador, Cliente, Probador, Rastreador, Instructor, Consultante y Gerente [29].

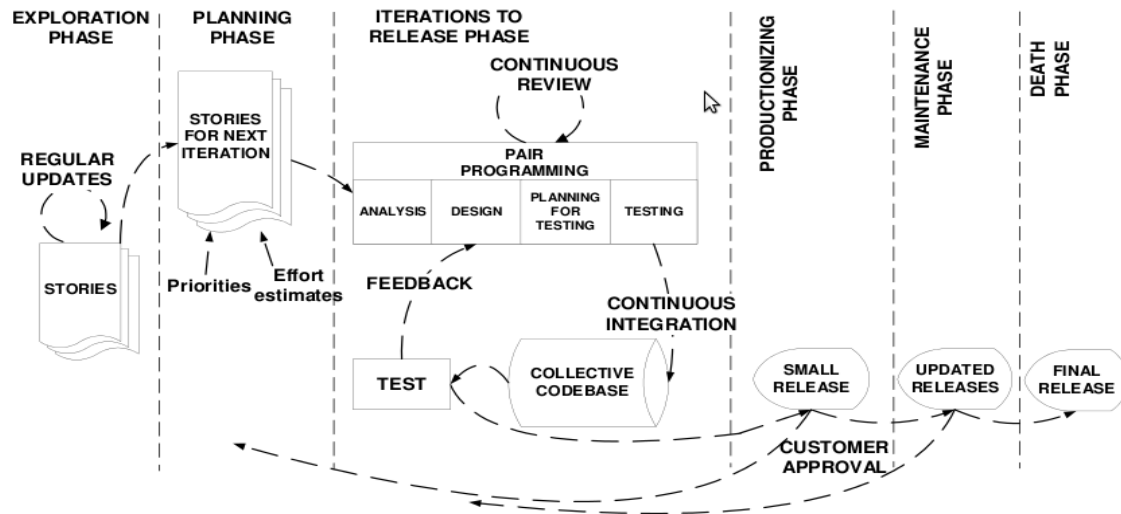


Ilustración 1: Proceso de desarrollo de XP

Entre sus prácticas encontramos el juego de planeación, pequeños y cortos lanzamientos, metáforas para definir el sistema como un todo, diseño simple, desarrollo dirigido por pruebas, refactorización, programación por parejas, pertenencia colectiva del código, integración continua, semanas de 40 horas, cliente en el sitio de desarrollo, estándares de código, espacio de trabajo abierto y reglas de desarrollo redactadas y aplicadas por el propio equipo [29].

En el siguiente epígrafe se hará una explicación de las prácticas de XP en mayor detalle.

1.5.3 Programación Extrema

Se puede considerar la Programación Extrema (XP, por sus siglas en inglés) como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software.

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo.

XP se basa en retroalimentación 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, y donde existe un alto riesgo técnico [28].

Algunos aspectos fundamentales de XP son[28]:

Distinción entre las decisiones a tomar por la parte del negocio y las decisiones que toman los miembros del equipo del proyecto. Esto quiere decir que las decisiones sobre las funcionalidades del sistema y sus prioridades las toma el cliente, mientras que las de la cantidad de trabajo a realizar por cada iteración, y la organización, el equipo. Es muy importante seguir esta práctica al pie de la letra ya que permite que cada parte del equipo se enfoque en su función y no sabotee el trabajo de su contrapartida.

Escribir pruebas unitarias antes de comenzar a escribir el código y mantenerlas corriendo todo el tiempo. Una de las prácticas básicas y más difíciles de seguir en XP es el desarrollo guiado por pruebas. El escribir la prueba antes del código, permite tener una medida de código terminado, lo cual ocurre cuando la prueba corre correctamente. Esto brinda las ventajas de tener software más estable y de mayor calidad, evitándose errores de regresión.

Integración y prueba de todo el sistema varias veces al día. Cada vez que se añada o cambie una pieza de código realizar la compilación e integración de todo el sistema. Permitiendo llevar el seguimiento del progreso del sistema y evitar el surgimiento de nuevos errores.

Producción de todo el software en parejas. En este, dos miembros del equipo se sientan en una sola estación de trabajo y crean código en conjunto. Usualmente se empareja un programador novato con uno avanzado logrando elevar tanto el nivel individual como colectivo de los miembros del equipo. También se recomienda que las parejas no sean fijas sino que los miembros de este roten con otros miembros del equipo, contribuyendo al principio ágil de la pertenencia colectiva de código.

Comenzar los proyectos con un diseño simple que evolucione constantemente para añadir la flexibilidad necesaria y elimine la complejidad innecesaria. Otra práctica difícil de seguir en XP, debido a que mayormente los programadores tienden a pensar primero en el *futuro*, lo que entra en contradicción con lo que XP basa sus cimientos: el cambio es inevitable e impredecible. La tendencia a crear diseños complejos inicialmente, resulta la mayoría de las veces en funcionalidades que nadie usa, incurriendo en gastos de recursos innecesarios. Por lo que una de las ideas básicas en XP consiste en que siempre el diseño más simple es el mejor diseño.

Poner un sistema mínimo en producción lo más rápido posible, creciendo en la dirección que pruebe ser la de mayor valor. Para los clientes no hay mejor medida de progreso que el software funcionando, sin importar cuan buen y flexible sea el diseño interno de este. Muchas veces los propios clientes no tienen claro que dirección quieren seguir con el proyecto. El entregar en el menor plazo posible le permite evaluar cuáles son las características de mayor importancia a implementar a continuación o detener el desarrollo de características innecesarias [28].

Todo equipo de desarrollo de software por mucho que se desee *auto-organizar* debe tener algún tipo de jerarquía para poder funcionar de forma correcta y cada cuál tenga una función determinada. En XP existen un grupo de roles que le dan solución a este problema. Es importante señalar que estos roles pueden o no ser exclusivos de una persona y hasta un miembro del equipo puede tener varios roles a la vez. Estos roles son:

Programador: Es la clase trabajadora, el programador escribe las pruebas unitarias y produce el código del sistema.

Cliente: Este rol cumple con el trabajo como contrapartida del programador. Este es el que decide qué trabajo va a realizar el programador, además de especificar prioridades. La persona que asuma este rol debe trabajar cara a cara con el equipo de desarrollo para que la comunicación entre el cliente y el equipo fluya de forma efectiva y natural.

Probador: Ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.

Rastreador: Proporciona realimentación al equipo. Verifica el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, para mejorar futuras estimaciones. Realiza el seguimiento del progreso de cada iteración.

Entrenador: Es responsable del proceso global. Debe proveer guías al equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente. También debe ser capaz de realizar adaptaciones al proceso que se adecuen mejor a las características del equipo de trabajo.

Consultante: Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto, en el que puedan surgir problemas.

Gerente: Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es la coordinación [28].

La *historia* es la unidad de funcionalidad en un proyecto XP. El progreso de un proyecto de XP se demuestra a través del código de una historia debidamente probada e integrada. Una historia debe ser entendible tanto por el cliente como por el programador, demostrable a través de algún tipo de prueba, de valor para el primero y lo suficientemente pequeña para que los segundos puedan construir alrededor de media docena de estas en cada iteración [30]. Las historias deben escribirse por el cliente y definir funcionalidades del sistema en el lenguaje del negocio [31]. En la Tabla 1 se puede ver los campos más comunes en una Historia de Usuario.

Historia de Usuario	
No.: [Id]	Nombre: [Nombre de la Historia]
Usuario: [Todos]	
Prioridad en el Negocio: [Alta, Media, Baja]	Nivel de Complejidad: [Alta, Media, Baja]

Estimación: [En semanas laborables]

Iteración Asignada: 1

Descripción: [Breve descripción de la historia]

Tabla 3: Prototipo de Historia de Usuario

Las historias a su vez se subdividen en tareas más concretas. Estas tareas de carácter técnico, permiten dividir una funcionalidad del negocio en unidades pequeñas estimables por los programadores. Al completar tareas podemos saber el por ciento del trabajo realizado en cada historia de usuario y llevar trazas de las mismas. Pueden existir además tareas independientes de las historias de usuario que den cumplimiento con una necesidad técnica en específico, como usualmente son las que tienen que ver con el soporte. En la Tabla 2 se muestra un prototipo de Tarea, las que son escritas por los propios programadores.

Tarea	
Número de Tarea: [#]	Número de HU: [#]
Nombre de la Tarea: [Nombre de la tarea]	
Tipo de Tarea: [Tipo de Tarea]	Estimación: [Días de duración de la tarea]
Fecha de Inicio: [Fecha de Inicio]	Fecha de Fin: [Fecha de Fin]
Programador Responsable: [Nombre del Programador Responsable]	
Descripción: [Descripción de la Tarea]	

Tabla 4: Prototipo de Tarea

El otro artefacto usado por XP para documentar el proceso son las Pruebas de Aceptación, las que describen escenarios donde se ponen a prueba funcionalidades del sistema desplegado. Este tipo de pruebas determina cuándo las historias de usuario han sido completadas [30]. En la Tabla 3 se muestra un prototipo de Prueba de Aceptación.

Caso de Prueba de Aceptación	
Código: [Código de la Prueba]	Historia de Usuario: [#]
Nombre: [Nombre de la Prueba]	

Descripción: [Descripción breve de la Prueba]

Condiciones de Ejecución: [Pre-condiciones para ejecutar la prueba]

Entrada/Pasos de Ejecución: [Descripción del caso de prueba paso a paso]

Resultado Esperado: [Resultado Esperado de la Prueba]

Evaluación de la Prueba: [Prueba Satisfactoria o No]

Tabla 5: Prototipo de Prueba de Aceptación

El proceso de XP cuenta con 6 fases (ver Figura 1). Cada una de vital importancia para todo el proceso. A continuación se describe en detalles las funciones de cada una de estas fases:

- **Exploración:** El cliente define las historias de usuario que desea incluir en la primera liberación del producto. Al mismo tiempo los miembros del equipo se familiarizan con las herramientas, tecnologías y prácticas que usarán durante el proyecto. Se probarán las tecnologías que se usarán y las posibilidades de arquitectura para el sistema, construyendo un prototipo del sistema. La fase de exploración puede tomar desde algunas semanas hasta meses, dependiendo de cuan familiar es la tecnología para los programadores [29].
- **Planeación:** En esta fase se establece el orden de prioridad para las historias de usuario y un acuerdo entre el cliente y el equipo de desarrollo sobre el contenido del primer lanzamiento del sistema. Los programadores estiman la cantidad de esfuerzo requerido por cada historia de usuario y se llega a un acuerdo en el cronograma del proyecto. Esta fase toma usualmente varios días [29].
- **Iteraciones a liberar:** Esta fase incluye varias iteraciones del sistema antes del primer lanzamiento. El cronograma establecido en la fase de planeación se separa en un número de iteraciones que tomarán de una a cuatro semanas de implementar. La primera iteración crea un sistema con la arquitectura de todo el sistema. Esto se logra seleccionando las historias que fuerzan a construir la estructura de todo el sistema. El cliente decide

las historias que serán seleccionadas para cada iteración. Las pruebas funcionales diseñadas por el cliente deben correr al final de cada iteración. Al final de la última iteración el sistema está listo para producción [29].

- **Producción:** La fase de producción requiere una serie extra de pruebas y el chequeo del funcionamiento del sistema antes de que pueda ser liberado al cliente. Además, en esta fase aún se pueden realizar nuevas decisiones y cambios si se incluyen en el lanzamiento actual. Durante esta fase las iteraciones se deben apresurar desde tres a una semana. Las ideas y sugerencias pospuestas son documentadas para ser implementadas durante la fase de mantenimiento [29].
- **Mantenimiento:** Después del primer lanzamiento de producción para el uso por el cliente, el proyecto XP debe mantener el sistema corriendo y además producir nuevas iteraciones. Para lograr esto, la fase de mantenimiento requiere un esfuerzo en las tareas de soporte del cliente. De esa manera, la velocidad de desarrollo se puede desacelerar después que el sistema entre en producción. Esta fase puede requerir incorporar nuevas personas al equipo de desarrollo y realizar cambios en la estructura actual del mismo [29].
- **Muerte:** La fase final o muerte del proceso se encuentra cercana cuando el cliente no tiene más historias para implementar. Esto requiere que el sistema satisfaga otras necesidades del cliente como son rendimiento y confiabilidad. Este es el momento en el que la documentación del sistema es finalmente escrita y no se realizan más cambios al diseño o la arquitectura del proyecto. La muerte del proyecto puede ocurrir también cuando el sistema no entrega el resultado esperado o si se hace muy caro seguir su desarrollo [29].

Como metodología se seleccionó XP para el desarrollo del sistema, debido a las características antes mencionadas y que la misma satisface las necesidades del cliente y la aplicación que se propone.

1.6 Conclusiones del capítulo

En este capítulo se realizó un estudio de la base teórica que sustentará el desarrollo del sistema. Se fundamentó además la selección de la tecnología y herramientas de la ingeniería del software a utilizar en el desarrollo del futuro sistema, profundizando en las características principales de cada una de ellas, teniendo en cuenta el hecho de que la mayoría de estas son libres y de código abierto, lo que reduce en gran medida el costo del proyecto, garantizando de esta forma poca dependencia hacia los productores de cada una de las tecnologías.

Se han seleccionado para la confección de la aplicación Web como principales tecnologías, el lenguaje de programación Python, el *framework* Django, el sistema gestor de base de datos PostgreSQL y como metodología XP.

Capítulo 2. Exploración y Planificación

2.1 Introducción

En el presente capítulo se procede con la elaboración de la propuesta del sistema a implementar, abordando en la misma una descripción de los módulos por los que estará compuesta, los diferentes roles establecido en ella y todo lo referente a ello.

También en el mismo se definirán las historias de usuario como elemento fundamental para conocer los requisitos del sistema y ser implementadas en cada una de las iteraciones previstas, mientras el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto, se pone a prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. Además de definir toda la planificación de entrega para la implementación de las mismas y el plan de iteraciones con su correspondiente descripción.

2.2 Exploración

En esta primera fase de XP se tiene como principal objetivo disponer de suficientes requisitos para ser implementados durante el tiempo fijado para una entrega en una determinada iteración. La misma permite definir el alcance del proyecto. En esta etapa los clientes definen sus necesidades a través de las historias de usuario, y el equipo de desarrollo se familiariza con las herramientas y tecnologías que se utilizarán en el proyecto.

2.2.1 Descripción de requisitos

Los requerimientos del sistema son las condiciones o capacidades que deben estar presentes en un sistema. Estos existen porque el tipo de producto en construcción exige cierta funcionalidad o calidad o porque el usuario quiere que el producto cumpla con ciertos requisitos. Los requerimientos pueden ser clasificados en requerimientos funcionales o requerimientos no funcionales.

2.2.1.1 Requerimientos Funcionales del Sistema

Los requerimientos funcionales especifican el comportamiento de entrada y salida del sistema y surgen de la razón fundamental de la existencia de la aplicación. Especifican acciones que el sistema debe ser capaz de realizar, sin tomar en consideración ningún tipo de restricción física [14].

A continuación se enumeran los requerimientos funcionales del sistema.

RF1: Módulo de Control de Acceso

RF1.1: Entrar al Sistema

RF1.2: Salir del Sistema

RF1.3: Mostrar Datos del Usuario

RF1.4: Cambiar Contraseña del Usuario

RF2: Módulo Cliente

RF2.1: Solicitar Impresión

RF2.2: Mostrar Impresiones Pendientes

RF2.3: Mostrar Impresiones Solicitadas

RF3: Módulo Administración

RF3.1: Gestionar Impresiones (CRUD)

RF3.2: Gestionar Tipos de Papel (CRUD)

RF3.3: Gestionar Proyectos (CRUD)

RF3.4: Gestionar Documentos (CRUD)

RF3.5: Gestionar Usuarios (CRUD)

RF3.6: Generar Reportes

- Reporte Diario
- Reporte Semanal
- Reporte Mensual
- Reporte por Mes
- Reporte por Proyecto
- Reporte por Usuario

2.2.1.2 Requerimientos no Funcionales del Sistema

Los requerimientos no funcionales son propiedades o cualidades que el sistema debe tener; estas características son las que hacen a la aplicación atractiva, usable, rápida o confiable. En muchos casos los requerimientos no funcionales son fundamentales en el éxito de la aplicación.

Apariencia o interfaz externa

- El diseño de la interfaz externa debe realizarse de manera sencilla, con un gran reconocimiento visual a través de elementos visibles que identifiquen cada una de sus acciones.
- Debe tener un modelo claro y fácil de utilizar por los usuarios, para que se sientan identificados, cómodos y confiados con el mismo.
- De no poderse ejecutar una acción por un usuario, se debe visualizar un mensaje de error que especifique por qué no se pudo ejecutar.

Usabilidad

- Debe haber una facilidad de uso por usuarios de cualquier nivel, presentando las funcionalidades visibles en todo momento que deben facilitar la navegación. Esto es necesario debido a que existen personas que interactuarán con la aplicación y tienen algunos conocimientos del trabajo con aplicaciones Web, pero existen otras que no los poseen.
- Debe brindar facilidades de selección de listas predeterminadas, así como búsquedas automatizadas.
- El diseño debe ser adaptable a diferentes resoluciones de pantalla.

Portabilidad

- La aplicación debe poder ser usada bajo los Sistemas Operativos Windows y Linux.

Ayuda y Documentación

- Contar con una Ayuda que especifique el funcionamiento de la aplicación y tenga ejemplos claros y fáciles de entender, y un Manual de Usuario, para guiar al usuario en la navegación de la aplicación.

2.2.2 Personas relacionadas con la aplicación

La aplicación va a ser operada fundamentalmente por 2 tipos de roles: Usuarios y Especialistas. A continuación se expone con mayor detalle las responsabilidades de cada rol:

Usuario: Es la persona que tiene permisos para ver la información referente al estado de las impresiones, crear sus peticiones de impresión para ser realizadas por él, y generar reportes de los datos de las impresiones realizadas.

Especialista: Es la persona facultada para la gestión del sistema en general. Podrá gestionar la administración de usuarios, tareas y del sistema.

2.2.3 Historias de Usuario

Como elemento indispensable en el desarrollo de software se encuentra la administración de requisitos, a la cual XP da cobertura mediante la confección de las historias de usuario, para luego ser desarrolladas en un proceso iterativo e incremental. Estas expresan las necesidades del sistema y tienen el mismo propósito que los casos de uso.

Las mismas son redactadas desde la perspectiva del cliente aunque los desarrolladores pueden brindar ayuda en su identificación. Por tanto serán descripciones cortas y escritas en el lenguaje del usuario, sin terminología técnica, siendo el contenido que ellas abarcan concreto y sencillo.

Las historias de usuario solamente proporcionarán los detalles sobre la estimación del riesgo y cuánto tiempo conllevará su implementación. El nivel de detalle de las historias de usuario debe ser el mínimo posible que permita hacerse una ligera idea de cuánto costará implementar el sistema.

Durante este proceso se identifican 8 historias de usuarios, las cuales se detallan a continuación.

Historia de Usuario	
No.: 1	Nombre: Control de Acceso
Usuario: Especialista	
Prioridad en el Negocio: Alta	Nivel de Complejidad: Media
Estimación: 10 días	Iteración Asignada: 1
Descripción: El sistema debe brindar la posibilidad de autenticarse y salir del sistema, asimismo como mostrar la información del usuario conectado.	

Tabla 6: Historia de Usuario # 1 “Control de Acceso”

Historia de Usuario	
No.: 2	Nombre: Funcionalidad del Cliente
Usuario: Usuario	
Prioridad en el Negocio: Alta	Nivel de Complejidad: Alta
Estimación: 10 días	Iteración Asignada: 2
Descripción: El sistema debe brindar la posibilidad de que los usuarios comunes soliciten impresiones y visualicen el estado de estas. Además de esto se debe crear la estructura de plantilla de la interfaz de usuario.	

Tabla 7: Historia de Usuario #2 “Funcionalidad del Cliente”

Historia de Usuario	
No.: 3	Nombre: Gestión de Impresiones
Usuario: Especialista	
Prioridad en el Negocio: Alta	Nivel de Complejidad: Media
Estimación: 2 días	Iteración Asignada: 3
Descripción: El sistema debe brindar la posibilidad de listar, adicionar, mostrar, actualizar y eliminar impresiones.	

Tabla 8: Historia de Usuario #3 “Gestión de Impresiones”

Historia de Usuario	
No.: 4	Nombre: Gestión de Tipos de Papel

Usuario: Especialista

Prioridad en el Negocio: Media

Nivel de Complejidad: Baja

Estimación: 2 días

Iteración Asignada: 4

Descripción: El sistema debe brindar la posibilidad de listar, adicionar, mostrar, actualizar y eliminar tipos de papel.

Tabla 9: Historia de Usuario #4 “Gestión de Tipos de Papel”

Historia de Usuario	
No.: 5	Nombre: Gestión de Proyectos
Usuario: Especialista	
Prioridad en el Negocio: Alta	Nivel de Complejidad: Baja
Estimación: 2 días	Iteración Asignada: 3
Descripción: El sistema debe brindar la posibilidad de listar, adicionar, mostrar, actualizar y eliminar proyectos.	

Tabla 10: Historia de Usuario #5 “Gestión de Proyectos”

Historia de Usuario	
No.: 6	Nombre: Gestión de Documentos
Usuario: Especialista	
Prioridad en el Negocio: Media	Nivel de Complejidad: Baja
Estimación: 2 días	Iteración Asignada: 3
Descripción: El sistema debe brindar la posibilidad de listar, adicionar, mostrar, actualizar y eliminar documentos.	

Tabla 11: Historia de Usuario #6 “Gestión de Proyectos”

Historia de Usuario	
No.: 7	Nombre: Gestión de Usuarios
Usuario: Especialista	
Prioridad en el Negocio: Alta	Nivel de Complejidad: Baja
Estimación: 3 días	Iteración Asignada: 3

Descripción: El sistema debe brindar la posibilidad de listar, adicionar, mostrar, actualizar y eliminar usuarios.

Tabla 12: Historia de Usuario #7 “Gestión de Usuarios”

Historia de Usuario	
No.: 8	Nombre: Generación de Reportes
Usuario: Especialista	
Prioridad en el Negocio: Alta	Nivel de Complejidad: Alta
Estimación: 8 días	Iteración Asignada: 4
Descripción: El sistema debe brindar la posibilidad de generar reportes en formato PDF de las impresiones solicitadas en el día, en la semana, en el mes, por mes y por proyecto.	

Tabla 13: Historia de Usuario #8 “Generación de Reportes”

2.3 Planificación

En esta etapa el cliente establece la prioridad de cada historia de usuario, y correspondientemente los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. También se toman acuerdos sobre el contenido de cada entrega.

2.3.1 Iteraciones

Todo proyecto que siga la metodología XP se ha de dividir en varias iteraciones antes de ser entregado. Luego de que las historias de usuarios fueron descritas e identificadas, así como estimado el esfuerzo que cada una de ellas conlleva se procede a definir las iteraciones en las que serán diseñadas, implementadas y probadas las mismas.

Para la selección del trabajo de cada iteración se tuvo en cuenta que este no tuviera más de 8 semanas de desarrollo, siguiendo las prácticas básicas de XP. En la Tabla 14 se muestran la distribución de las historias de usuario por cada iteración.

Iteraciones	Orden de las historias de usuario a implementar	Cantidad de tiempo de trabajo
1	<ul style="list-style-type: none">Control de Acceso	2 Semanas
2	<ul style="list-style-type: none">Funcionalidad del Cliente	2 Semanas
3	<ul style="list-style-type: none">Gestión de ImpresionesGestión de Tipos de PapelGestión de ProyectosGestión de Documentos	2 Semanas
4	<ul style="list-style-type: none">Gestión de UsuariosGeneración de Reportes	2 Semanas

Tabla 14: Distribución de las Historias de Usuario por cada iteración

2.3.2 Plan de entrega

Las historias de usuario servirán para crear el plan estimado de entregas. Cada una de ellas previamente evaluada en tiempo de desarrollo ideal será agrupada por el cliente en orden de importancia.

Para realizar una acertada estimación de la aplicación es necesario estimar de forma ideal el tiempo que les tomará la codificación de cada una de las historias de usuario. La estimación es uno de los temas más complicados del desarrollo de un proyecto que utiliza la metodología XP, y es por ello que resulta de vital importancia tener bien claros los requerimientos del cliente.

A continuación se expone el plan de entrega referente a la aplicación:




Entregable	Iteración	Versión	Fecha
 SGIGEO Sistema de Gestión para Impresiones	1	0.7	17 de Septiembre del 2012
 SGIGEO Sistema de Gestión para Impresiones	2	0.8	1 de Octubre del 2012
 SGIGEO Sistema de Gestión para Impresiones	3	0.9	15 de Octubre del 2012

Tabla 15: Cronograma de entregas

2.4 Conclusiones del capítulo

En este capítulo se hizo referencia a todo lo concerniente a la fase de exploración y planificación del sistema. Se realizó la descripción de las historias de usuarios. A partir de estas historias se realizó toda la planificación de entrega del software, la misma se efectuó conjuntamente con el cliente que es el actor principal en la etapa de planificación de la metodología seleccionada. Finalmente se cuenta con 8 historias de usuarios que serán desarrolladas en 4 iteraciones.

Capítulo 3. Implementación y pruebas

3.1 Introducción

En el presente capítulo se hace mención a la elaboración del diseño del sistema, los diagramas utilizados en el mismo, las tareas creadas por cada historia de usuario, el proceso de pruebas utilizado y finalmente la valoración de la propuesta.

3.2 Diseño del sistema

En la fase de diseño XP propone la idea, “Lo más simple que pueda funcionar”, aplicándose para ello las prácticas especializadas que la metodología propone, las cuales inciden directamente en la realización y elaboración del diseño de un software ya que, cuando se tiene un diseño simple se presta mayor atención a codificar.

3.2.1 Modelo del dominio

El modelo del dominio contiene palabras y conceptos relacionados con el problema que el sistema diseñado trata de resolver. Cuando se crea el modelo del dominio, el autor está creando una representación de los objetos y las acciones que debe realizar el sistema. Proporciona un vocabulario común que permite buena comunicación entre los usuarios y los desarrolladores.

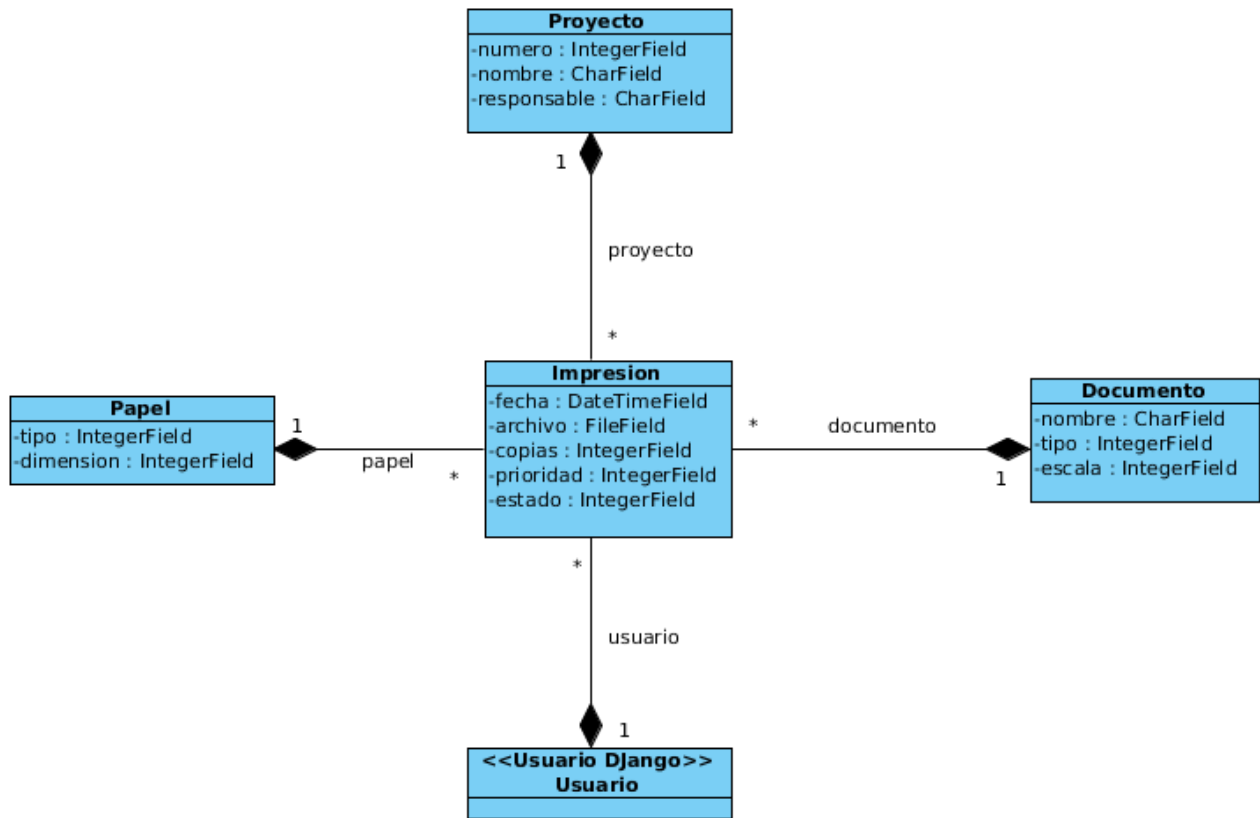


Ilustración 2: Modelo del Dominio

3.2.2 Arquitectura

La arquitectura propuesta para el desarrollo del sistema presenta tres tipos principales de componentes: las bibliotecas, los plugins y las aplicaciones.

3.2.3 Diagrama de Despliegue

Para modelar la infraestructura necesaria para la implantación del sistema y las relaciones entre sus componentes se utiliza el Diagrama de Despliegue, ya que este permite apreciar de forma visual cómo se encuentran relacionados físicamente los distintos dispositivos involucrados con el sistema.

En este caso la aplicación se encuentra hospedada en un servidor Web y esta se comunica con un sistema de gestión de base de datos (PostgreSQL).

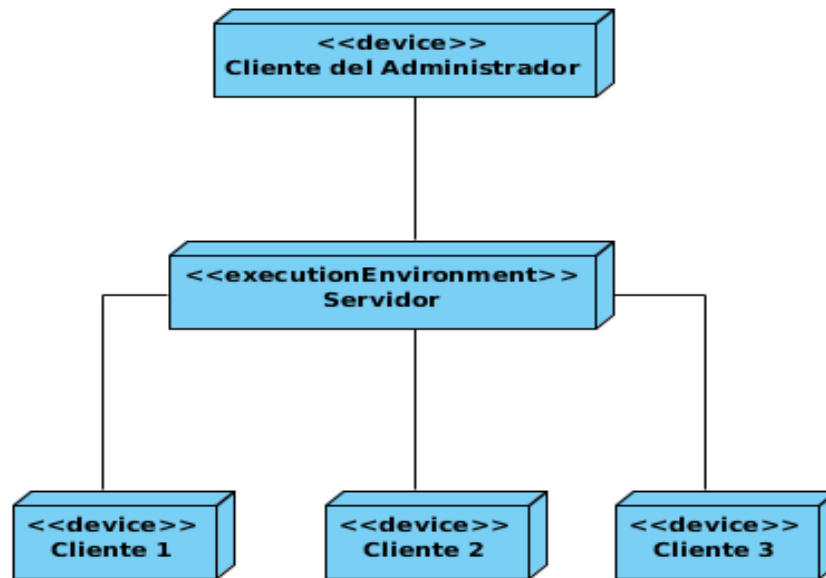


Ilustración 3: Diagrama de Despliegue

3.2.4 Diagrama de Paquetes

Para facilitar el desarrollo y la comprensión del sistema propuesto, este fue dividido en 4 paquetes, que agrupan las funcionalidades presentes en el mismo. Esta agrupación se hizo teniendo en cuenta las relaciones existentes entre los requerimientos funcionales del sistema anteriormente definido para tratar el tamaño del modelo.

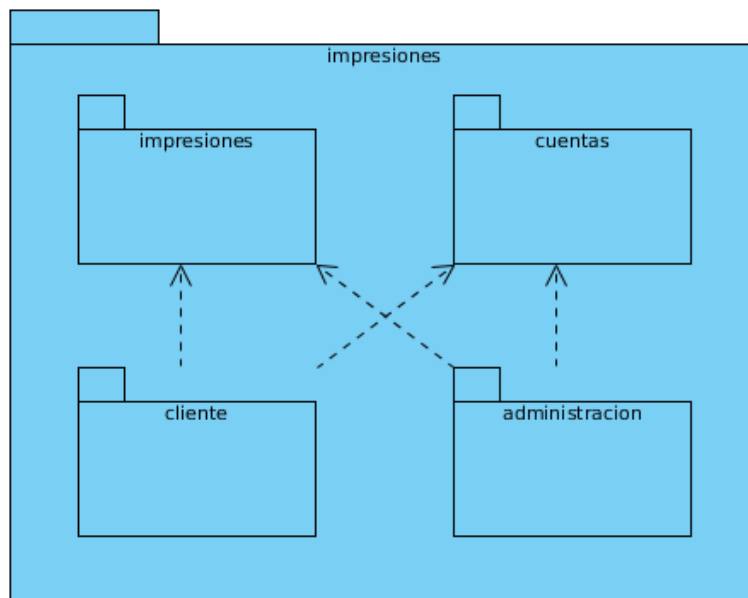


Ilustración 4: Diagrama de Paquetes

3.3 Implementación

En esta fase se plantea la implementación de las historias de usuario en su correspondiente iteración, obteniéndose en cada una de ellas una versión funcional del producto. Para ello se descomponen las historias de usuario en tareas de desarrollo para ser realizadas, describiendo cada una de estas.

3.3.1 Iteración 1

En esta iteración se pretende diseñar, implementar y probar una de las historias de usuario que más importancia tiene para el cliente, la cual es: implementar el control de acceso al sistema, obteniéndose las funcionalidades descritas en las mismas.

Una vez que se terminen de implementar y probar todas las funcionalidades de esta iteración, se procederá a desplegar la aplicación, la cual será mostrada al cliente con la finalidad de que fluya la retroalimentación de ideas y solventar posibles errores.

Para ello se trazaron las siguientes 4 tareas (ver Anexo 1 para mayor detalle):

1. Tarea # 1: Entrar al Sistema
2. Tarea # 2: Salir del Sistema
3. Tarea # 3: Mostrar datos del usuario
4. Tarea # 4: Cambiar contraseña del usuario

3.3.2 Iteración 2

En esta iteración se tiene como objetivo diseñar, implementar y probar una de las historias de usuario de gran prioridad, la cual es implementar la funcionalidad del cliente.

Una vez culminado el diseño, la implementación y la prueba de todas las funcionalidades de esta iteración, se procederá a desplegar la aplicación, el cual será mostrado al cliente con la finalidad de que fluya la retroalimentación de ideas y solventar posibles errores.

Para ello se trazaron las siguientes 4 tareas (ver Anexo 2 para mayor detalle):

1. Tarea # 5: Plantilla base de de la interfaz de usuario
2. Tarea # 6: Solicitar impresión
3. Tarea # 7: Mostrar impresiones pendientes
4. Tarea # 8: Mostrar impresiones solicitadas

3.3.3 Iteración 3

En esta iteración se tiene como objetivo diseñar, implementar y probar algunas de las historias de usuario del sistema, las cuales son: gestión de impresiones, gestión de tipos de papel, gestión de proyectos, y gestión de documentos.

Una vez culminado el diseño, la implementación y la prueba de todas las funcionalidades de esta iteración, se procederá a desplegar la aplicación, el cual será mostrado al cliente con la finalidad de que fluya la retroalimentación de ideas y solventar posibles errores.

Para ello se trazaron las siguientes 20 tareas (ver Anexo 3 para mayor detalle):

1. Tarea # 9: Listar impresiones
2. Tarea # 10: Adicionar impresión
3. Tarea # 11: Mostrar impresión
4. Tarea # 12: Actualizar impresión
5. Tarea # 13: Eliminar impresión
6. Tarea # 14: Listar tipos de papel
7. Tarea # 15: Adicionar tipos de papel
8. Tarea # 16: Mostrar tipos de papel
9. Tarea # 17: Actualizar tipos de papel
10. Tarea # 18: Eliminar tipos de papel
11. Tarea # 19: Listar proyectos

12. Tarea # 20: Adicionar proyecto
13. Tarea # 21: Mostrar proyecto
14. Tarea # 22: Actualizar proyecto
15. Tarea # 23: Eliminar proyecto
16. Tarea # 24: Listar documentos
17. Tarea # 25: Adicionar documento
18. Tarea # 26: Mostrar documento
19. Tarea # 27: Actualizar documento
20. Tarea # 28: Eliminar documento

3.3.4 Iteración 4

En esta iteración se tiene como objetivo diseñar, implementar y probar algunas de las historias de usuario de prioridad en el negocio alta, las cuales son: gestión de usuarios y generación de reportes.

Una vez culminado el diseño, la implementación y la prueba de todas las funcionalidades de esta iteración, se procederá a desplegar la aplicación, el cual será mostrado al cliente con la finalidad de que fluya la retroalimentación de ideas y solventar posibles errores.

Para ello se trazaron las siguientes 11 tareas (ver Anexo 4 para mayor detalle):

1. Tarea # 29: Listar usuarios
2. Tarea # 30: Adicionar usuario
3. Tarea # 31: Mostrar usuario
4. Tarea # 32: Actualizar usuario
5. Tarea # 33: Eliminar usuario
6. Tarea # 34: Reporte diario
7. Tarea # 35: Reporte semanal

8. Tarea # 36: Reporte mensual
9. Tarea # 37: Reporte por mes
10. Tarea # 38: Reporte por proyecto
11. Tarea # 39: Reporte por usuario

3.4 Pruebas

La fase de prueba de la metodología XP anima a frecuentemente estar probando tanto como sea posible. Para ello en cada una de las iteraciones se lleva a cabo los casos de pruebas de aceptación a los que se debe someter la aplicación. Estas pruebas deben ser verificadas en cada entrega que se realice del producto en la planificación establecida.

3.4.1 Pruebas de aceptación

Como técnica para garantizar que los requerimientos hayan sido cumplidos y que la aplicación es realmente lo que el cliente necesita, además de asegurar su correcto funcionamiento son realizadas las pruebas de aceptación. Las cuales son creadas a partir de las historias de usuario y desde la perspectiva del cliente. Una historia de usuario puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento.

Estas pruebas funcionan como una caja negra, pues cada una de ellas representa una salida esperada del sistema, donde es responsabilidad del cliente verificar la corrección de las pruebas y tomar decisiones acerca de las mismas.

El objetivo final de las mismas es lograr que los requerimientos sean cumplidos y que el sistema sea aceptable. Una vez que todas las historias de usuario hayan pasado sus pruebas de aceptación se considera entonces terminada la aplicación.

Caso de Prueba de Aceptación

Código: HU1_P1

Historia de Usuario: 1

Nombre: Control de Acceso

Descripción: Probar que se pueda autenticar un usuario en el sistema con datos válidos, que se muestran los

datos del usuario conectado, permitir que este cambie su contraseña y desautenticarse del sistema.

Condiciones de Ejecución: Deben existir un usuario de prueba sin privilegios administrativos y conocer la contraseña de este.

Entrada/Pasos de Ejecución:

1. Entrar a la página de inicio del sistema en el navegador.
2. Autenticarse en el sistema con el usuario y la contraseña de prueba presionando en el botón **Entrar** en la derecha de la barra superior.
3. Cerciorarse que en el botón derecho de la barra superior se muestre el nombre de usuario del usuario de prueba. Si esto no ocurre entonces la prueba falla y termina.
4. Presionar el botón de la derecha en la barra superior izquierda y accione el menú mi cuenta.
5. Cerciorarse que los datos del usuario de prueba mostrado son correctos, de lo contrario la prueba falla y termina.
6. En la pantalla de datos del usuario presionar el botón de cambiar contraseña.
7. Proveer los datos del formulario de cambio de contraseña, proveyendo a este con una nueva contraseña.
8. Desautenticarse del sistema utilizando en la opción **Salir** en el menú del botón de la derecha de la barra superior.
9. Cerciorarse que al intentar acceder a cualquier URL válida del sistema se muestra la pantalla de autenticación de este, de lo contrario la prueba falla y termina.
10. Autenticar el usuario de prueba con la nueva contraseña.
11. Cerciorarse que en el botón derecho de la barra superior se muestre el nombre de usuario del usuario de prueba. Si esto no ocurre entonces la prueba falla.

Resultado Esperado: Se muestra la página del cliente permitiendo mostrar los datos del usuario conectado y cambiar la contraseña de este.

Evaluación de la Prueba: Satisfactoria.

Tabla 16: Prueba 1 de la HU "Control de Acceso"

Caso de Prueba de Aceptación	
Código: HU2_P2	Historia de Usuario: 2
Nombre: Funcionalidad del cliente	
Descripción: Probar que se almacene de forma correctamente 2 solicitudes de impresión realizada por el un usuario común del sistema y luego que estas se muestren de forma correcta en las vistas de Impresiones Pendientes e Impresiones Solicitadas .	
Condiciones de Ejecución: Debe existir un usuario no administrador autenticado y deben existir 2 impresiones con estado de Finalizadas.	
Entrada/Pasos de Ejecución:	
1. En las opciones que se muestran en la pantalla del cliente presionar el vínculo de Solicitar Impresión.	

2. Entrar los datos de la impresión a adicionar y presionar el botón solicitar.
3. Realizar los pasos 1 y 2 con la segunda impresión.
4. Accionar el logo en la izquierda de la barra superior para volver al inicio del cliente.
5. Accionar el vínculo de **Impresiones Pendientes**.
6. Comprobar que se muestran los datos de las 2 impresiones recién solicitadas, de lo contrario la prueba falla y termina.
7. Ejecutar paso 4.
8. Accionar el vínculo de **Impresiones Solicitadas**.
9. Comprobar que se muestran los datos de 4 impresiones, las 2 cerradas previamente y las 2 añadidas recientemente, de lo contrario la prueba falla.

Resultado Esperado: Se añaden de forma correcta las 2 impresiones solicitadas y se muestran las 4 que quedan almacenadas en el sistema.

Evaluación de la Prueba: Satisfactoria.

Tabla 17: Prueba 2 de la HU “Funcionalidad del Cliente”

Caso de Prueba de Aceptación	
Código: HU3_P3	Historia de Usuario: 3
Nombre: Gestión de Impresiones	
Descripción: Probar que se realizan de forma satisfactoria las acciones de adicionar, listar, actualizar, mostrar en detalle y eliminar impresiones.	
Condiciones de Ejecución: Debe existir un usuario administrador autenticado.	
Entrada/Pasos de Ejecución: <ol style="list-style-type: none">1. En el menú Impresiones accionar Adicionar Impresión.2. Añadir los datos de una impresión válida y presionar el botón adicionar.3. En el menú Impresiones accionar Listar Impresiones.4. Comprobar que los datos de la recién adicionada impresión se muestran en la tabla, de lo contrario la prueba falla y termina.5. Presionar el ícono del Ojo para mostrar el detalle de la Impresión.6. Comprobar que los datos en detalle de la recién adicionada impresión coinciden con los añadidos, de lo contrario la prueba falla y termina.7. Presionar el botón Actualizar.8. Realizar varios cambios en la impresión y presionar el botón Actualizar.9. Realizar el paso 5.10. Comprobar que los datos en detalle coinciden con los modificados en el paso 8, de lo contrario la prueba falla y termina.11. Presionar el botón Eliminar y en el nuevo formulario presionar este de nuevo.12. Comprobar que los datos que la impresión modificada en el paso 8 no se muestra en la tabla, de lo contrario la prueba falla.	

Resultado Esperado: Se adiciona, lista, actualiza, muestra en detalle y elimina una impresión del sistema.

Evaluación de la Prueba: Satisfactoria.

Tabla 18: Prueba 3 de la HU “Gestión de Impresiones”

Caso de Prueba de Aceptación	
Código: HU8_P4	Historia de Usuario: 8
Nombre: Generación de Reportes	
Descripción: Probar que los reportes Por Mes, Por Proyecto y Por Usuario se generan de forma correcta en formato PDF.	
Condiciones de Ejecución: <ol style="list-style-type: none">1. Presionar la acción Por Mes del menú Reportes.2. Especificar un mes y un año que tenga varias impresiones en varios estados.3. Presionar el botón Generar.4. Comprobar que se genera un archivo PDF y que este cuenta con el formato y los datos correctos, de lo contrario la prueba falla y termina.5. Presionar la acción Por Proyecto del menú Reportes.6. Especificar un proyecto tenga varias impresiones en varios estados.7. Presionar el botón Generar.8. Comprobar que se genera un archivo PDF y que este cuenta con el formato y los datos correctos, de lo contrario la prueba falla y termina.9. Presionar la acción Por Usuario del menú Reportes.10. Especificar un usuario tenga varias impresiones en varios estados.11. Presionar el botón Generar.12. Comprobar que se genera un archivo PDF y que este cuenta con el formato y los datos correctos, de lo contrario la prueba falla.	
Entrada/Pasos de Ejecución: Se intenta generar un reporte de las impresiones indicándole una fecha.	
Resultado Esperado: Se genera un documento en PDF con los reportes solicitados en un formato adecuado para ello.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 19: Prueba 4 de la HU “Generación de Reportes”

3.5 Valoración de la propuesta

Para desarrollar un sistema es importante tener presente el impacto social, económico, tecnológico y ambiental que este tendrá, para poder determinar si el sistema informático es sostenible y así garantizar la calidad del mismo.

Todo sistema informático cuenta con las siguientes dimensiones de sostenibilidad: administrativa, socio-humanista, ambiental y tecnológica. Esta valoración nos da una idea de las ventajas que este facilita en cada uno de estos aspectos.

En la **dimensión administrativa** se valora si la solución planteada ahorra recursos, se tienen presente los gastos implicados para desarrollarla e implantarla, la calidad de la producción y los servicios, así como otros aspectos que garanticen la sostenibilidad administrativa de la solución.

No habrá gastos por mano de obra, ya que su realización será por un estudiante de la carrera de Ingeniería Informática como parte de un Trabajo de Diploma. Para su desarrollo se utilizará en su mayoría equipamiento existente en la empresa, por lo que no se incurrirá en gastos adicionales. Una vez implantado, se ahorrarán recursos de gran importancia como el tiempo, lapiceros y hojas, pues al emplearse medios computarizados se reducirán considerablemente el consumo de estos. No genera ingresos directamente a la entidad donde se realiza, ya que su finalidad es facilitar el trabajo de los trabajadores de la empresa.

El desarrollo del proyecto no incurre en gastos económicos, pues se hace uso de software libre, evitando el pago de licencias. Por lo tanto se puede concluir que en la dimensión administrativa el sistema informático es sostenible.

En la **dimensión socio-humanista** el sistema propuesto favorecerá grandemente los recursos humanos al facilitar una mayor rapidez en la gestión de la información, lo que implica mayores ganancias y por tanto una mejora en el nivel de vida. Por otra parte permitirá que el usuario se sienta cómodo, mediante una interfaz sencilla y amigable. Contribuye además la posibilidad de trabajar con un mayor bienestar personal, menos esfuerzo físico, disminuir así la posibilidad de errores y maximizar la eficiencia y confiabilidad en la información procesada. Se confeccionará un manual de usuario detallado donde se expliquen las características, funcionalidades y ventajas reales del sistema.

En la **dimensión tecnológica** se puede decir que cualquier usuario puede interactuar con el sistema, sin necesidad de contar con un alto nivel de conocimiento en el ámbito de la informática; el sistema informático comprende el

uso de la tecnología adecuada y asimilable con el usuario. La empresa cuenta con la infraestructura electrónica necesaria para un correcto funcionamiento del sistema, cuenta con una red interna que interconecta todas y cada una de las áreas de la empresa logrando un flujo continuo de las informaciones. Cuenta también con potentes servidores y estaciones de trabajo. Desde el inicio del diseño e implementación del sistema existió una estrecha comunicación entre los desarrolladores y los usuarios finales, para lograr la mayor satisfacción de estos últimos. El sistema informático permite adaptarse a cambios que no afecten su funcionalidad principal, debido a que ha sido desarrollado siguiendo las prácticas de la metodología XP en la ingeniería de software que permiten que el mantenimiento y los cambios ocurran de forma rápida y sencilla.

A partir de lo analizado anteriormente se arribó a la conclusión de que el sistema es sostenible desde la dimensión tecnológica.

En la **dimensión ambiental** el sistema contribuirá a la conservación del Medio Ambiente si tenemos en cuenta el uso de recursos que generan daños al mismo, se ve evidenciado en el ahorro de materiales de oficina como el papel que se obtiene a partir de la tala de árboles, lapiceros, lápices y energía eléctrica que se obtienen de la naturaleza y que se fabrican industrialmente. El sistema no usa colores agresivos y propone una interfaz sencilla, usa tonalidades claras que se encuentran en la gama de los colores pertenecientes al manual corporativo de la empresa con el objetivo de que los usuarios se identifiquen con el sistema, por lo que existirá una adecuada comunicación entre la aplicación y el usuario disminuyendo los riesgos de estrés psicológico; influye en gran manera sobre el usuario ya que anteriormente debía emplear una gran cantidad de tiempo para obtener diversos informes, pues los hacía a mano y ahora el sistema lo resuelve en unos segundos, por lo que la vista del usuario y la posición de sentarse no se ven afectadas más de lo normal.

Se puede plantear que este sistema informático no tiene incidencia sobre el deterioro del medio ambiente por lo que es sostenible según esta dimensión.

3.6 Conclusiones del capítulo

En este capítulo se hizo referencia a todo lo concerniente a la fase de Prueba de la metodología XP. Se realizó la descripción de las pruebas de aceptación a la que fue sometida la aplicación con la intención de verificar el funcionamiento de la misma, probando individualmente la implementación de cada historia de usuario en su correspondiente iteración y asignándole una evaluación.

Todas las pruebas realizadas fueron aceptadas por el cliente cumpliendo entonces el sistema con las historias de usuarios definidas inicialmente.

Conclusiones Generales

En esta investigación se pudo concluir que se dio cumplimiento al objetivo puesto que se informatizó el proceso de gestión de impresiones en la empresa Geocuba Oriente Norte en concordancia con las necesidades expuestas por el cliente. Además se puede concluir:

- En estudio realizado sobre la metodología de desarrollo de software se determinó utilizar XP la cual se modeló haciendo uso de los artefactos UML, esta metodología ayudó a guiar el proceso de diseño del sistema en cuestión, además de permitir que el trabajo se desarrollara con mayor eficacia.
- Se eligieron en el análisis de las tecnologías, Python como lenguaje de programación, el *framework* Django, PostgreSQL como Gestor de Base de Datos y el Servidor Web Apache. Las tecnologías utilizadas son software libre.
- Se constató que el sistema propuesto es sostenible en el tiempo, lo que pudo ser comprobado mediante la valoración de sostenibilidad en las dimensiones administrativa, socio – humanista, ambiental y tecnológica.
- El Sistema para la Gestión de los Servicios de Impresión tiene buena aceptación por parte de los usuarios finales pues soluciona los problemas que emitieron los mismos en el comienzo de la investigación lo que fue verificado por las pruebas de aceptación de la metodología XP.

Recomendaciones

Para dar continuidad de forma satisfactoria a este trabajo se recomienda:

- Implantar la aplicación en la entidad para mejorar el trabajo de la gestión de impresiones en esta.
- Realizar un curso de capacitación de los potenciales usuarios de esta para una explotación eficiente.
- Usar tecnologías AJAX para mejorar la interacción del usuario con los formularios de la aplicación y ahorrar ancho de banda.

Bibliografía

- [1] Agell, I.O. and Smithson S, «Information Systems Management: Opportunities and Risks». 1991.
- [2] «Print Manager». [Online]. Available: <http://www.acotec.es>. [Accessed: 11-ene-2012].
- [3] «PennCentral-document». [Online]. Available: <http://www.docpath.com>. [Accessed: 11-ene-2012].
- [4] Vegas, Jesús, «Introducción a las Aplicaciones Web». .
- [5] J. G. Valle, «Definición Arquitectura Cliente Servidor». 2005.
- [6] Gutiérrez. Jorge A Saavedra, «El Mundo Informático. Software Libre». 2007.
- [7] K., Bill Musciano, Chuck, «HTML La guía completa». 1999.
- [8] Rubén Marcos González, «Recursos para la programación en PHP». .
- [9] Potencier, Fabien, «Symfony 1.2, la guía definitiva», 2009. [Online]. Available: http://www.librosweb.es/symfony_1_2/. [Accessed: 26-ene-2012].
- [10] «Apache», 2011. [Online]. Available: <http://www.apache.com>.
- [11] S. Holzner, «Eclipse». O'Reilly, 2004.
- [12] «Netbeans». [Online]. Available: <http://www.netbeans.org/index/es.html>.
- [13] D. L. Kuhn, «Selecting and effectively using a computer aided software engineering tool», presented at the Annual Westinghouse computer symposium, Pittsburgh, PA (U.S.), 1989.
- [14] P. Loucopoulos y V. Karakostas, *System Requirements Engineering*. McGraw-Hill, 1995.
- [15] «Unified Modeling Language Specification. Version 1.4.2». .
- [16] «Visual Paradigm for UML User' s Guide». 2007.
- [17] S. Álvarez, «Sistemas Gestores de Bases de Datos». 2007.
- [18] González, Mabel, «SQL SERVER». .
- [19] S. Pozo Coronado, «MySQL con Clase». 2005.
- [20] Rafael Martínez, «Sobre PostgreSQL», 2006. [Online]. Available: http://www.postgresql-es.org/sobre_postgresql.
- [21] R. Burback, «Software Engineering Methodology: The Watersluice», PhD, 1998.
- [22] L. P. Canós, J y Penadés MC, *Metodologías Agiles en el Desarrollo de Software*. 2005.
- [23] M. A. Awad, «A Comparison between Agile and Traditional Software Development Methodologies». 2005.

- [24] «12 Principios del Manifiesto Ágil», 2001. [Online]. Available: <http://agilemanifesto.org/iso/es/principles.html>. [Accessed: 16-abr-2012].
- [25] M. Fowler, «The New Methodology», 16-abr-2012. [Online]. Available: <http://www.martinfowler.com/articles/newMethodology.html>. [Accessed: 16-abr-2012].
- [26] OTS Solutions, «Agile Methodology. Changing ways of software development...» 2010.
- [27] K. Beck, «Embracing Change With Extreme Programming», *IEEE Computer*, vol. 32, 1999.
- [28] K. Beck, *Extreme Programming Explained. Embrace Change*. 1999.
- [29] P. Abrahamsson, O. Salo, J. Ronkainen, y J. Warsta, «Agile software development methods. Review and Analysis», 2002.
- [30] K. Beck y M. Fowler, *Planning Extreme Programming*. Addison Wesley, 2000.
- [31] C. Patel y M. Ramachandran, «Story Card Based Agile Software Development», *International Journal of Hybrid Information Technology*, p. Abril 2009.

Anexos

Anexo 1: Tareas

Iteración 1

Tarea	
Número de Tarea: 1	Número de HU: 1
Nombre de la Tarea: Entrar al Sistema	
Tipo de Tarea: Implementación	Estimación: 24 horas
Fecha de Inicio: 3 de Septiembre de 2012	Fecha de Fin: 5 de Septiembre de 2012
Programador Responsable: Daliuska García Marrero	
Descripción: El sistema debe brindar la posibilidad de introducir el usuario y contraseña, con la finalidad de verificar y otorgar los permisos según el rol que se ocupe dentro de la aplicación.	

Tabla 20: Tarea # 1 “Entrar al Sistema”

Tarea	
Número de Tarea: 2	Número de HU: 1
Nombre de la Tarea: Salir del Sistema	
Tipo de Tarea: Implementación	Estimación: 16 horas
Fecha de Inicio: 6 de Septiembre de 2012	Fecha de Fin: 7 de Septiembre de 2012
Programador Responsable: Daliuska García Marrero	
Descripción: Facilita cerrar la aplicación.	

Tabla 21: Tarea # 2 “Salir del Sistema”

Tarea	
Número de Tarea: 3	Número de HU: 1
Nombre de la Tarea: Mostrar datos del usuario	
Tipo de Tarea: Implementación	Estimación: 24 horas
Fecha de Inicio: 10 de Septiembre de 2012	Fecha de Fin: 12 de Septiembre de 2012

Programador Responsable: Daliuska García Marrero

Descripción: Se implementan las funcionalidades que permiten la captura de los datos (nombre, área, cargo) y nivel de acceso (especialista, usuario) del usuario que se desea mostrar en la base de datos.

Tabla 22: Tarea # 3 “Mostrar datos del usuario”

Tarea	
Número de Tarea: 4	Número de HU: 1
Nombre de la Tarea: Cambiar contraseña del usuario	
Tipo de Tarea: Implementación	Estimación: 16 horas
Fecha de Inicio: 13 de Septiembre de 2012	Fecha de Fin: 14 de Septiembre de 2012
Programador Responsable: Daliuska García Marrero	
Descripción: Permiten modificar la contraseña de un usuario determinado.	

Tabla 23: Tarea # 4 “Cambiar contraseña del usuario”

Anexo 2: Tareas

Iteración 2

Tarea	
Número de Tarea: 5	Número de HU: 2
Nombre de la Tarea: Plantilla base de la interfaz de usuario	
Tipo de Tarea: Diseño	Estimación: 16 horas
Fecha de Inicio: 18 de Septiembre de 2012	Fecha de Fin: 19 de Septiembre de 2012
Programador Responsable: Daliuska García Marrero	
Descripción: Se implementan las funcionalidades que permiten crear todas las plantillas para la interfaz visual del sistema.	

Tabla 24: Tarea # 5 “Plantilla base de la interfaz de usuario”

Tarea	
Número de Tarea: 6	Número de HU: 2

Nombre de la Tarea: Solicitar Impresión

Tipo de Tarea: Implementación

Estimación: 16 horas

Fecha de Inicio: 20 de Septiembre de 2012

Fecha de Fin: 21 de Septiembre de 2012

Programador Responsable: Daliuska García Marrero

Descripción: Se implementan las funcionalidades que permiten insertar los datos necesarios para realizar la solicitud de una impresión, fecha, cantidad de copias, proyecto al que pertenece, tipo de papel, tipo de documento.

Tabla 25: Tarea # 6 “Solicitar Impresión”

Tarea	
Número de Tarea: 7	Número de HU: 2
Nombre de la Tarea: Mostrar impresiones pendientes	
Tipo de Tarea: Implementación	Estimación: 24 horas
Fecha de Inicio: 24 de Septiembre de 2012	Fecha de Fin: 26 de Septiembre de 2012
Programador Responsable: Daliuska García Marrero	
Descripción: Se implementan las funcionalidades que permiten mostrar las impresiones pendientes según criterios como fecha, proyecto al que pertenece y tipo de documento.	

Tabla 26: Tarea # 7 “Mostrar impresiones pendientes”

Tarea	
Número de Tarea: 8	Número de HU: 2
Nombre de la Tarea: Mostrar impresiones solicitadas	
Tipo de Tarea: Implementación	Estimación: 16 horas
Fecha de Inicio: 27 de Septiembre de 2012	Fecha de Fin: 28 de Septiembre de 2012
Programador Responsable: Daliuska García Marrero	
Descripción: Se implementan las funcionalidades que permiten mostrar las impresiones solicitadas según criterios como fecha, proyecto al que pertenece y tipo de documento.	

Tabla 27: Tarea # 8 “Mostrar impresiones solicitadas”

Anexo 3: Tareas

Iteración 3

Tarea	
Número de Tarea: 9	Número de HU: 3
Nombre de la Tarea: Listar Impresiones	
Tipo de Tarea: Implementación	Estimación: 3 horas
Fecha de Inicio: 2 de Octubre de 2012	Fecha de Fin: 2 de Octubre de 2012
Programador Responsable: Daliuska García Marrero	
Descripción: Se implementan las funcionalidades que permiten listar las impresiones, filtrarlas según criterios como fecha, proyecto al que pertenece y tipo de documento.	

Tabla 28: Tarea # 9 “Listar Impresiones”

Tarea	
Número de Tarea: 10	Número de HU: 3
Nombre de la Tarea: Adicionar Impresión	
Tipo de Tarea: Implementación	Estimación: 3 horas
Fecha de Inicio: 2 de Octubre de 2012	Fecha de Fin: 2 de Octubre de 2012
Programador Responsable: Daliuska García Marrero	
Descripción: Se implementan las funcionalidades que permiten insertar los datos necesarios para realizar una impresión, fecha, cantidad de copias, proyecto al que pertenece, tipo de papel, tipo de documento.	

Tabla 29: Tarea #10 “Adicionar Impresión”

Tarea	
Número de Tarea: 11	Número de HU: 3
Nombre de la Tarea: Mostrar Impresión	
Tipo de Tarea: Implementación	Estimación: 4 horas
Fecha de Inicio: 2 de Octubre de 2012	Fecha de Fin: 3 de Octubre de 2012
Programador Responsable: Daliuska García Marrero	

Descripción: Se implementan las funcionalidades que permiten mostrar las impresiones, filtrarlas según criterios como fecha, proyecto al que pertenece y tipo de documento.

Tabla 30: Tarea # 11 “Mostrar Impresión”

Tarea	
Número de Tarea: 12	Número de HU: 3
Nombre de la Tarea: Actualizar Impresión	
Tipo de Tarea: Implementación	Estimación: 3 horas
Fecha de Inicio: 3 de Octubre de 2012	Fecha de Fin: 3 de Octubre de 2012
Programador Responsable: Daliuska García Marrero	
Descripción: Se implementan las funcionalidades que permiten editar los datos de una impresión existente en el sistema.	

Tabla 31: Tarea # 12 “Actualizar Impresión”

Tarea	
Número de Tarea: 13	Número de HU: 3
Nombre de la Tarea: Eliminar Impresión	
Tipo de Tarea: Implementación	Estimación: 3 horas
Fecha de Inicio: 3 de Octubre de 2012	Fecha de Fin: 3 de Octubre de 2012
Programador Responsable: Daliuska García Marrero	
Descripción: Se implementan las funcionalidades que permiten eliminar una impresión determinada	

Tabla 32: Tarea # 13 “Eliminar Impresión”

Tarea	
Número de Tarea: 14	Número de HU: 4
Nombre de la Tarea: Listar Tipos de Papel	
Tipo de Tarea: Implementación	Estimación: 3 horas
Fecha de Inicio: 4 de Octubre de 2012	Fecha de Fin: 4 de Octubre de 2012
Programador Responsable: Daliuska García Marrero	

Descripción: Se implementan las funcionalidades que permiten listar un tipo de papel determinado.

Tabla 33: Tarea # 14 “Listar Tipos de Papel”

Tarea	
Número de Tarea: 15	Número de HU: 4
Nombre de la Tarea: Adicionar Tipos de Papel	
Tipo de Tarea: Implementación	Estimación: 3 horas
Fecha de Inicio: 4 de Octubre de 2012	Fecha de Fin: 4 de Octubre de 2012
Programador Responsable: Daliuska García Marrero	
Descripción: Se implementan las funcionalidades que permiten insertar un tipo de papel determinado.	

Tabla 34: Tarea #15 “Adicionar Tipos de Papel”

Tarea	
Número de Tarea: 16	Número de HU: 4
Nombre de la Tarea: Mostrar Tipos de Papel	
Tipo de Tarea: Implementación	Estimación: 4 horas
Fecha de Inicio: 4 de Octubre de 2012	Fecha de Fin: 5 de Octubre de 2012
Programador Responsable: Daliuska García Marrero	
Descripción: Se implementan las funcionalidades que permiten mostrar un tipo de papel determinado.	

Tabla 35: Tarea # 16 “Mostrar Tipos de Papel”

Tarea	
Número de Tarea: 17	Número de HU: 4
Nombre de la Tarea: Actualizar Tipos de Papel	
Tipo de Tarea: Implementación	Estimación: 3 horas
Fecha de Inicio: 5 de Octubre de 2012	Fecha de Fin: 5 de Octubre de 2012
Programador Responsable: Daliuska García Marrero	

Descripción: Se implementan las funcionalidades que permiten editar los datos de un papel determinado.

Tabla 36: Tarea # 17 “Actualizar Tipos de Papel”

Tarea	
Número de Tarea: 18	Número de HU: 4
Nombre de la Tarea: Eliminar Tipos de Papel	
Tipo de Tarea: Implementación	Estimación: 3 horas
Fecha de Inicio: 5 de Octubre de 2012	Fecha de Fin: 5 de Octubre de 2012
Programador Responsable: Daliuska García Marrero	
Descripción: Se implementan las funcionalidades que permiten eliminar un tipo de papel determinado.	

Tabla 37: Tarea # 18 “Eliminar Tipos de Papel”

Tarea	
Número de Tarea: 19	Número de HU: 5
Nombre de la Tarea: Listar Proyectos	
Tipo de Tarea: Implementación	Estimación: 3 horas
Fecha de Inicio: 8 de Octubre de 2012	Fecha de Fin: 8 de Octubre de 2012
Programador Responsable: Daliuska García Marrero	
Descripción: Se implementan las funcionalidades que permiten listar los proyectos, filtrarlos según criterios como número de proyecto, nombre del proyecto y responsable.	

Tabla 38: Tarea # 19 “Listar Proyectos”

Tarea	
Número de Tarea: 20	Número de HU: 5
Nombre de la Tarea: Adicionar Proyecto	
Tipo de Tarea: Implementación	Estimación: 3 horas
Fecha de Inicio: 8 de Octubre de 2012	Fecha de Fin: 8 de Octubre de 2012
Programador Responsable: Daliuska García Marrero	

Descripción: Se implementan las funcionalidades que permiten adicionar los datos de un proyecto, número de proyecto, nombre del proyecto y responsable.

Tabla 39: Tarea # 20 “Adicionar Proyecto”

Tarea	
Número de Tarea: 21	Número de HU: 5
Nombre de la Tarea: Mostrar Proyecto	
Tipo de Tarea: Implementación	Estimación: 4 horas
Fecha de Inicio: 8 de Octubre de 2012	Fecha de Fin: 9 de Octubre de 2012
Programador Responsable: Daliuska García Marrero	
Descripción: Se implementan las funcionalidades que permiten mostrar los proyectos, filtrarlos según criterios como número de proyecto, nombre del proyecto y responsable.	

Tabla 40: Tarea # 21 “Mostrar Proyecto”

Tarea	
Número de Tarea: 22	Número de HU: 5
Nombre de la Tarea: Actualizar Proyecto	
Tipo de Tarea: Implementación	Estimación: 3 horas
Fecha de Inicio: 9 de Octubre de 2012	Fecha de Fin: 9 de Octubre de 2012
Programador Responsable: Daliuska García Marrero	
Descripción: Se implementan las funcionalidades que permiten editar los datos de un proyecto determinado.	

Tabla 41: Tarea # 22 “Actualizar Proyecto”

Tarea	
Número de Tarea: 23	Número de HU: 5
Nombre de la Tarea: Eliminar Proyecto	
Tipo de Tarea: Implementación	Estimación: 3 horas
Fecha de Inicio: 9 de Octubre de 2012	Fecha de Fin: 9 de Octubre de 2012
Programador Responsable: Daliuska García Marrero	

Descripción: Se implementan las funcionalidades que permiten eliminar un proyecto determinado.

Tabla 42: Tarea # 23 “Eliminar Proyecto”

Tarea	
Número de Tarea: 24	Número de HU: 6
Nombre de la Tarea: Listar Documentos	
Tipo de Tarea: Implementación	Estimación: 3 horas
Fecha de Inicio: 10 de Octubre de 2012	Fecha de Fin: 10 de Octubre de 2012
Programador Responsable: Daliuska García Marrero	
Descripción: Se implementan las funcionalidades que permiten listar los documentos, filtrarlos según criterios como tipo de documento, nombre del documento y escala.	

Tabla 43: Tarea # 24 “Listar Documentos”

Tarea	
Número de Tarea: 25	Número de HU: 6
Nombre de la Tarea: Adicionar Documento	
Tipo de Tarea: Implementación	Estimación: 3 horas
Fecha de Inicio: 10 de Octubre de 2012	Fecha de Fin: 10 de Octubre de 2012
Programador Responsable: Daliuska García Marrero	
Descripción: Se implementan las funcionalidades que permiten adicionar los datos de un documento, tipo de documentos, nombre del documento y escala.	

Tabla 44: Tarea # 25 “Adicionar Documento”

Tarea	
Número de Tarea: 26	Número de HU: 6
Nombre de la Tarea: Mostrar Documento	
Tipo de Tarea: Implementación	Estimación: 4 horas
Fecha de Inicio: 10 de Octubre de 2012	Fecha de Fin: 11 de Octubre de 2012
Programador Responsable: Daliuska García Marrero	

Descripción: Se implementan las funcionalidades que permiten mostrar los documentos, filtrarlos según criterios como tipo de documento, nombre del documento y escala.

Tabla 45: Tarea # 26 “Mostrar Documento”

Tarea	
Número de Tarea: 27	Número de HU: 6
Nombre de la Tarea: Actualizar Documento	
Tipo de Tarea: Implementación	Estimación: 3 horas
Fecha de Inicio: 11 de Octubre de 2012	Fecha de Fin: 11 de Octubre de 2012
Programador Responsable: Daliuska García Marrero	
Descripción: Se implementan las funcionalidades que permiten editar los datos de un documento determinado.	

Tabla 46: Tarea # 27 “Actualizar Documento”

Tarea	
Número de Tarea: 28	Número de HU: 6
Nombre de la Tarea: Eliminar Proyecto	
Tipo de Tarea: Implementación	Estimación: 3 horas
Fecha de Inicio: 11 de Octubre de 2012	Fecha de Fin: 11 de Octubre de 2012
Programador Responsable: Daliuska García Marrero	
Descripción: Se implementan las funcionalidades que permiten eliminar un documento determinado.	

Tabla 47: Tarea # 28 “Eliminar Documento”

Anexo 4: Tareas

Iteración 4

Tarea	
Número de Tarea: 29	Número de HU: 7
Nombre de la Tarea: Listar Usuarios	

Tipo de Tarea: Implementación	Estimación: 5 horas
Fecha de Inicio: 12 de Octubre de 2012	Fecha de Fin: 12 de Octubre de 2012
Programador Responsable: Daliuska García Marrero	
Descripción: Se implementan las funcionalidades que permiten listar los usuarios, filtrarlos según criterios como área de trabajo o nivel de acceso que posee.	

Tabla 48: Tarea # 29 “Listar Usuarios”

Tarea	
Número de Tarea: 30	Número de HU: 7
Nombre de la Tarea: Adicionar Usuario	
Tipo de Tarea: Implementación	Estimación: 5 horas
Fecha de Inicio: 12 de Octubre de 2012	Fecha de Fin: 15 de Octubre de 2012
Programador Responsable: Daliuska García Marrero	
Descripción: Se implementan las funcionalidades que permiten la captura de los datos (nombre, área, cargo) y nivel de acceso (especialista, usuario) del usuario que se desea adicionar en la base de datos.	

Tabla 49: Tarea # 30 “Adicionar Usuario”

Tarea	
Número de Tarea: 31	Número de HU: 7
Nombre de la Tarea: Mostrar Usuario	
Tipo de Tarea: Implementación	Estimación: 4 horas
Fecha de Inicio: 15 de Octubre de 2012	Fecha de Fin: 15 de Octubre de 2012
Programador Responsable: Daliuska García Marrero	
Descripción: Se implementan las funcionalidades que permiten mostrar un usuario, filtrarlos según criterios como área de trabajo o nivel de acceso que posee.	

Tabla 50: Tarea # 31 “Mostrar Usuario”

Tarea	
Número de Tarea: 32	Número de HU: 7

Nombre de la Tarea: Actualizar Usuario

Tipo de Tarea: Implementación

Estimación: 5 horas

Fecha de Inicio: 15 de Octubre de 2012

Fecha de Fin: 16 de Octubre de 2012

Programador Responsable: Daliuska García Marrero

Descripción: Se implementan las funcionalidades que permiten editar los datos (nombre, área, cargo) y nivel de acceso (especialista, usuario) de un usuario determinado.

Tabla 51: Tarea # 32 “Actualizar Usuario”

Tarea	
Número de Tarea: 33	Número de HU: 7
Nombre de la Tarea: Eliminar Usuario	
Tipo de Tarea: Implementación	Estimación: 5 horas
Fecha de Inicio: 16 de Octubre de 2012	Fecha de Fin: 16 de Octubre de 2012
Programador Responsable: Daliuska García Marrero	
Descripción: Se implementan las funcionalidades que permiten eliminar un usuario determinado.	

Tabla 52: Tarea # 33 “Eliminar Usuario”

Tarea	
Número de Tarea: 34	Número de HU: 8
Nombre de la Tarea: Reporte Diario	
Tipo de Tarea: Implementación	Estimación: 12 horas
Fecha de Inicio: 17 de Octubre de 2012	Fecha de Fin: 18 de Octubre de 2012
Programador Responsable: Daliuska García Marrero	
Descripción: Se crea una vista a la que se le da como entrada una fecha y genera un reporte de todos los gastos realizados en esta en formato PDF.	

Tabla 53: Tarea # 34 “Reporte Diario”

Tarea	
Número de Tarea: 35	Número de HU: 8

Nombre de la Tarea: Reporte Semanal

Tipo de Tarea: Implementación

Estimación: 12 horas

Fecha de Inicio: 18 de Octubre de 2012

Fecha de Fin: 19 de Octubre de 2012

Programador Responsable: Daliuska García Marrero

Descripción: Se crea una vista a la que se le da como entrada una fecha y genera un reporte de todos los gastos realizados en esta en formato PDF.

Tabla 54: Tarea # 35 “Reporte Semanal”

Tarea	
Número de Tarea: 36	Número de HU: 8
Nombre de la Tarea: Reporte Mensual	
Tipo de Tarea: Implementación	Estimación: 12 horas
Fecha de Inicio: 22 de Octubre de 2012	Fecha de Fin: 23 de Octubre de 2012
Programador Responsable: Daliuska García Marrero	
Descripción: Se crea una vista a la que se le da como entrada una fecha y genera un reporte de todos los gastos realizados en esta en formato PDF.	

Tabla 55: Tarea # 36 “Reporte Mensual”

Tarea	
Número de Tarea: 37	Número de HU: 8
Nombre de la Tarea: Reporte por Mes	
Tipo de Tarea: Implementación	Estimación: 12 horas
Fecha de Inicio: 23 de Octubre de 2012	Fecha de Fin: 24 de Octubre de 2012
Programador Responsable: Daliuska García Marrero	
Descripción: Se crea una vista a la que se le da como entrada una fecha y genera un reporte de todos los gastos realizados en esta en formato PDF.	

Tabla 56: Tarea # 37 “Reporte por Mes”

Tarea

Número de Tarea: 38	Número de HU: 8
Nombre de la Tarea: Reporte por Proyecto	
Tipo de Tarea: Implementación	Estimación: 8 horas
Fecha de Inicio: 25 de Octubre de 2012	Fecha de Fin: 25 de Octubre de 2012
Programador Responsable: Daliuska García Marrero	
Descripción: Se crea una vista de todos los proyectos realizados, donde se le introducirá como entrada el proyecto al que se le desea generar el reporte en formato PDF.	

Tabla 57: Tarea # 38 “Reporte por Proyecto”

Tarea	
Número de Tarea: 39	Número de HU: 8
Nombre de la Tarea: Reporte por Usuario	
Tipo de Tarea: Implementación	Estimación: 8 horas
Fecha de Inicio: 26 de Octubre de 2012	Fecha de Fin: 26 de Octubre de 2012
Programador Responsable: Daliuska García Marrero	
Descripción: Se crea una vista donde se seleccionará un usuario y se generará un reporte de todas las impresiones solicitadas por este en formato PDF.	

Tabla 58: Tarea # 39 “Reporte por Usuario”