

**Universidad de Matanzas Sede “Camilo Cienfuegos”**

**Facultad de Ciencias Técnicas.**

**Departamento Informática.**



**“Aplicación web para la gestión del proceso de información del estado de los servicios en la Dirección Provincial de Bufetes Colectivos de Matanzas(DPBC)”**

---

**Trabajo de Diploma en opción al Título de Ingeniería Informática**

**Autor:** Celia Fuentes Gómez

**Tutor:** MSc. Liz Pérez Martínez

Matanzas, Cuba

Junio de 2018

*Tu tiempo es limitado, de modo que no lo malgastes viviendo la vida de alguien distinto. No quedés atrapado en el dogma, que es vivir como otros piensan que deberías vivir. No dejes que los ruidos de las opiniones de los demás acallén tu propia voz interior. Y, lo que es más importante, ten el coraje para hacer lo que te dicen tu corazón y tu intuición. Ellos ya saben de algún modo en qué quieres convertirte realmente. Todo lo demás es secundario.*

*Si sacas todo lo que tienes y confías en ti, esfuérzate, ¡porque lo vas a lograr!, ya que nadie deja de llegar cuando se lo propone...*

*Steve Jobs*

## **Dedicatoria.**

*A mis padres por su amor y apoyo incondicional;*

*A mi hijo por ser el baluarte de mi vida;*

*A mi familia que me han apoyado en todo momento y  
circunstancia;*

*A mi esposo por su amor y comprensión;*

*A mis amigos y vecinos que siempre han estado a mi lado  
cuando los necesito.*

*A todos muchas gracias.*

# Agradecimientos

La culminación del presente trabajo constituye la cúspide de esta importante etapa de mi carrera académica, la cual no hubiese sido posible sin el apoyo y colaboración de un grupo importante de personas durante todo el trayecto de mi vida a las cuales les envío mis más sinceros agradecimientos.

A mi familia, a ustedes que me dieron su amor y sus tiempos libres. A ustedes debo todo lo que soy. El agradecimiento más grande que pueda existir es a mi Mamá que espero que este tan orgullosa de mí como yo de ti a pesar de todas las vicisitudes que hemos vivido juntas, a ti que eres todo amor hasta en los momentos que me vuelvo insoportable. Gracias por confiar en mí y apoyarme en cada momento de mi vida, por guiar mis pasos con sus sabios consejos y ejemplos que han hecho de mí una persona de bien gracias tías. Por enseñarme a alcanzar mis metas sin importar lo difícil que sean y estar ahí siempre regañándome como si fuera una niña pequeña a ti que te considero mi hermana gracias por siempre estar conmigo; gracias a todos por respetar las decisiones que tomo en la vida, por estar siempre dispuestos a ayudarme a lograr que este sueño se hiciera realidad, por el amor que me brindan cada día.

A mi hijo por ser la fuente de mi inspiración, porque haces que al despertar cada mañana con mamá ya salió el sol: levántate, mi vida cobre sentido, por tener siempre una caricia, un beso, una flor en tus manitas para mí. Te amo mi pequeño. A mi esposo por su constante preocupación por mis estudios y su consideración en los momentos más cruciales de mi vida.

Mi agradecimiento eterno a Liz, que no solo ha sido mi tutora, sino mi amiga, que ha estado a mí disposición a pesar de todos los contratiempos. Por su preocupación, por cederme parte de su tiempo, por no perder la paciencia conmigo y decir tu puedes, por todo esto estaré eternamente en deuda.

A mis vecinos que han sido mi familia, que me han cuidado y apoyado en todo momento.

A mis profesores, a quienes les agradezco mi formación profesional.

A los que fueron mis profesores y se convirtieron en amigos y a pesar de la distancia he contado con sus consejos.

A todos mis amigos.

Agradecer además a todos los que de una u otra manera se preocuparon por mí, a aquellos que en todo momento me preguntaron: ¿cómo va la tesis?

A todos, muchas gracias

Celia

## Declaración de Autoría

Yo, Celia Fuentes Gómez, declaro que soy la única autora de este trabajo y autorizo a la Universidad de Matanzas sede “Camilo Cienfuegos” y a la Dirección Provincial de Bufetes Colectivos de Matanzas, a que hagan el uso que estimen pertinente de él.

Para que así conste firmamos la presente a los 14 días del mes de junio de 2018.

---

Firma del Autor

Celia Fuentes Gómez

---

Firma del Tutor

MSc. Liz Pérez Martínez

---

## Opinión del Tutor

---

### DATOS PERSONALES DEL TUTOR

**Nombre y apellidos:** Liz Pérez Martínez

**Centro de trabajo:** Universidad de Matanzas.

**Organismo a que pertenece:** Ministerio de Educación Superior – MES.

**Cargo que ocupa:** Vicedecana Facultad de Ciencias Técnicas.

**Especialidad de la que es graduado:** Ingeniería Informática. Universidad de Matanzas, 2012.

**Categoría docente o investigativa:** Asistente.

**Grado científico:** Máster en Ciencias, Universidad de Matanzas, 2015.

### DATOS DE LA TESIS Y EL DIPLOMANTE

**Nombre y apellidos:** Celia Fuentes Gómez

**Centro de estudio:** Universidad de Matanzas sede “Camilo Cienfuegos”.

**Título de la Tesis:** Aplicación web para la gestión del proceso de información del estado de los servicios en la Dirección Provincial de Bufetes Colectivos de Matanzas.

### OPINIÓN SOBRE EL TRABAJO

La tesis presentada posee gran actualidad, pues intenta resolver un problema real presente en las empresas de nuestro país, y además contribuye a la informatización de nuestra sociedad.

El tutor de este trabajo de diploma considera que, durante su ejecución, la estudiante mostró las cualidades que a continuación se detallan:

Independencia y creatividad no solo en el campo de la investigación, el análisis y diseño del sistema, sino también investigando los *frameworks* existentes y

utilizando modernas técnicas de programación. Ha llegado a un resultado que lo crece profesionalmente por ser su primer gran aporte a la práctica social.

Los aspectos tanto metodológicos como de la investigación científica se evidencian con facilidad en todo el desarrollo de su trabajo. Desempeño su investigación con rigor y siguiendo las etapas consecuentemente. La documentación y las pruebas realizadas, tienen gran valor y están correctamente estructurados. Demostró dominio sobre la temática enfrentada y captó con rapidez los principales elementos de la materia abordada. En el trabajo se aprecia profesionalidad, manifestado desde el tratamiento de los conceptos estudiados y referenciados en la bibliografía, hasta las conclusiones a las que arribó, lo que ha contribuido en gran medida a la solución de los problemas enfrentados. Ha dejado planteados importantes elementos a tener en consideración en futuras investigaciones.

Por todo lo anteriormente señalado, considero que la estudiante Celia Fuentes Gómez reúne los requisitos para el título de Ingeniero Informático y esperamos le sea otorgada la mejor calificación de este Tribunal. Esperamos que su futuro como profesional colme todas sus expectativas.



---

MSc. Liz Pérez Martínez

Dpto. Informática

Universidad de Matanzas



Junio/2018

## Opinión del Cliente

---

Matanzas, 13 de junio del 2018  
"Año 60 de la Revolución".

A quien pueda interesar:

Ref: Desarrollo de la aplicación web para la gestión del proceso de información del estado de los servicios en la Dirección Provincial de Bufetes Colectivos de Matanzas(DPBC)"

Considero que el módulo de sistema de gestión y control de los servicios de la DPBC Matanzas deviene como una poderosa herramienta que satisface las necesidades implantadas por la organización. Esta herramienta simplifica las operaciones necesarias para la contratación, el seguimiento de los contratos (fase), el nivel de infracciones cometidas; así como sus causas, además de confeccionar reportes con los que no se contaban automáticamente y el nivel oportuno de la información en tiempo real, requisitos indispensables en los distintos bufetes de nuestra provincia, reduciendo considerablemente el esfuerzo y tiempo necesarios dedicados por nuestros trabajadores a esta tarea.

Atentamente;

MSc. Sergio Jorge Pagés Valdés  
Director Provincial de Bufetes Colectivos Matanzas.



## Resumen

---

El presente Trabajo de Diploma fue realizado en la Dirección Provincial de Bufetes colectivos de Matanzas, se fundamenta en diseñar un sistema de seguimiento y control de los servicios, que permita la flexibilidad y comercialización de las prestaciones ofrecidos a las personas jurídicas y a los clientes no nacionales posibilitando que de esta manera aumenten los ingresos de la organización. Entre las facilidades que brinda el sistema se encuentran la generación automática de reportes y estadísticas útiles para tomar decisiones posteriores sobre el sistema de contratación de los abogados; así como la gestión del estado real del proceso en un período de tiempo determinado. Se plantean los métodos teóricos y empíricos que se utilizaron en la investigación. Se desarrolló un sistema web de gestión que automatiza el proceso de información del estado de los contratos, para la Dirección Provincial de Bufetes Colectivos de Matanzas que posibilita agilizar muchos asuntos que se realizan actualmente de forma indirecta y casi manual. Se empleó la metodología de desarrollo de software Programación Extrema (XP), el lenguaje de programación C# y como gestor de base de datos Oracle 11g con Toad. Se obtuvo un sistema funcional, el cual se le realizaron una serie de pruebas que permitieron constatar su funcionalidad y verificar que cumpliera con las necesidades y exigencias del cliente.

---

## Abstract

---

Diploma's present work was accomplished in the Provincial Management of Buffets Collective of Matanzas, is based on designing a tracking system and control of the services, that it enables flexibility and commercialization of the social benefits offered to the juridical persons and to the customers not nationals making possible that proceeds this way gain organization. Between the ease that offers the system the automatic generation of reports and useful statistics to take later decisions on the hiring system of the lawyers meet; As well as the management of the real state of the process in given period of time. They propose the theoretic and empiric methods that were used in investigation. The web of management that automates the informing process of the state of the contracts that makes it possible to speed up a lot of matters that come true at present of indirect and almost manual way, for the Provincial Management of Buffets Collective of Matanzas unfolded an application itself. Used the methodology of development of software Extreme Programing (XP), the programming language C # and like database manager Oracle 11g with Toad. A functional system was obtained, which accomplished a series of proofs that they enabled verifying your functionality for him and verifying that he fulfilled the needs and the customer's requirements.

# Tabla de contenido

Índice de Tablas .....	9
Índice de Ilustraciones.....	11
Introducción .....	1
Capítulo I. Fundamentación teórica y tendencias tecnológicas.....	5
1.1 Antecedentes del trabajo.....	5
1.2 Objeto de estudio.....	8
1.3 Descripción de los procesos que serán objeto de automatización.....	10
1.4 Tecnologías consideradas para la implementación del sistema.....	10
1.4.1 Lenguaje del lado del cliente .....	10
HTML 5.....	11
CSS 3 .....	11
JavaScript .....	12
1.4.2 Lenguaje del lado del servidor.....	12
C#.....	12
Motor de vista Razor .....	13
1.4.3 Frameworks.....	13
Bootstrap 3 .....	13
jQuery.....	14
ASP.Net .....	15
1.4.4 Entity Framework y Object Relational Mapping.....	19
1.4.5 Técnicas de desarrollo web Ajax .....	20
1.4.6 Herramientas de desarrollo.....	21
Gestor de base de datos: Oracle 11g y Toad.....	21
Microsoft Visual Studio 2013 Express y ReSharper 8.0.....	22
Modelo Vista Controlador (MVC) .....	23

1.5 Metodología de desarrollo de software: <i>Extreme Programming (XP)</i> .....	25
1.6 Conclusiones del capítulo.....	28
Capítulo 2. Descripción de la solución propuesta. ....	29
2.1 Descripción de la propuesta.....	29
2.2 Equipo de trabajo y roles.....	30
2.3 Análisis de Requerimientos.....	32
Navegadores:.....	34
2.4 Artefactos de la metodología .....	35
2.4.1 Plan de Entregas .....	37
2.4.2 Descripción de las historias de usuario. ....	40
2.5 Etapa de Diseño.....	46
2.6 Estudio de Factibilidad.....	46
2.6.1 Estimación de Costo.....	46
2.6.2 Costo real.....	48
2.6.3 Análisis de costos y beneficios.....	48
2.7 Conclusiones del capítulo.....	49
Capítulo 3. Resultados obtenidos y pruebas realizadas al software.....	50
3.1 Introducción .....	50
3.2 Estándares de programación.....	50
3.3 Pruebas.....	51
3.3.1 Pruebas de Caja Blanca.....	52
Pruebas de Arquitectura.....	52
Realización de Pruebas Unitarias.....	52
3.3.2 pruebas de Caja Negra. ....	53
Pruebas de Aceptación.....	54
Prueba de estrés .....	60

3.4 Validación de la solución propuesta .....	61
3.5 Conclusiones del capítulo.....	63
<b>Conclusiones generales</b> .....	65
Recomendaciones.....	67
Bibliografía .....	68
Anexo 1. Comparación entre las metodologías ágiles y tradicionales.....	70
Anexo 2 Imágenes del sistema .....	71
Acrónimos.....	72

## Índice de Tablas

Tabla 1.1 comparativa de las herramientas investigadas.....	7
Tabla 2.1 Muestra el porciento de cumplimiento de las herramientas analizadas con los requisitos planteados por el cliente .....	8
Tabla 3.2 Equipo de trabajo y roles.....	30
Tabla 4.2 Usuarios y permisos.....	31
Tabla 5.2 Requisitos de hardware .....	34
Tabla 6.2 Requisitos de Software .....	34
Tabla 7.2 Historias de usuario, prioridad, riesgo y dependencia.....	36
Tabla 8.2: Plan de entregas.....	38
Tabla 9.2 HU17.....	40
Tabla 10.2 Crear Interfaz Principal.....	40
Tabla 11.2 HU2.....	41
Tabla 12.2 Autenticarse.....	41
Tabla 13.2 HU15.....	42
Tabla 14.2 Autenticación de usuarios .....	42
Tabla 15.2 Agregar usuarios. ....	43
Tabla 16.2 HU3.....	43
Tabla 17.2 Roles.....	43
Tabla 18.2 Permisos .....	44
Tabla 19.2 HU 4.....	44
Tabla 20.2 Adicionar contratos.....	45
Tabla 21.2 HU 5.....	45
Tabla 22.2 Estimación de costo. ....	47
Tabla 23.2 Gastos.....	48

Tabla 24.3 Resultados de la implementación de SFC .....	51
Tabla 25.3 Prueba de Aceptación.....	55
Tabla 26.3 Prueba de Aceptación 1 .....	55
Tabla 27.3 Prueba de Aceptación 2 .....	56
Tabla 28.3 Prueba de Aceptación 3 .....	57
Tabla 29.3 Prueba de Aceptación 4 .....	58
Tabla 30.3 Prueba de Aceptación 5 .....	59
Tabla 31.3 Tiempo de obtención y procesamiento de información.....	62
Tabla 32 Anexo 1 .....	70



## Índice de Ilustraciones

Ilustración 1. 1 capas de MVC.....	24
Ilustración 2.1 .....	25
Ilustración 3.1 Proceso de Programación Extrema(XP).....	27
Ilustración 4.3 Prueba de Arquitectura.....	52
Ilustración 5.3 Prueba Unitaria. ....	53
Ilustración 6.3 Prueba de Caja Negra .....	54
Ilustración 7 Autenticarse .....	71
Ilustración 8 Página principal de la administración .....	71

## Introducción

Actualmente las tendencias del mundo en la esfera jurídicas se encuentran revolucionadas por la expansión acelerada de las tecnologías informáticas y la continua demanda de las personas sobre el conocimiento del estado de sus trámites. La información continua y real de los servicios constituye un recurso vital para el mundo moderno, pues el buen manejo de esta permite obtener mayores niveles económicos. Para lograr la eficiencia de un sistema destinada a la gestión de la información, es importante el estudio a profundidad de sus fundamentos, de modo que sea posible acelerar el estado de los procesos.

Con el avance tecnológico e informático de la sociedad se ha logrado un mayor nivel de organización en todas las empresas, cada día se genera, recibe, envía y almacena grandes volúmenes de información, lo cual facilita la rapidez y eficacia de los procesos.

En el ámbito jurídico se gestiona diariamente grandes flujos de información, complejizándose cada día más, el trabajo de letrados, técnicos jurídicos y tribunales en general. Por lo que se hace necesario el desarrollo de herramientas tecnológicas que viabilicen este proceso.

En la actualidad el número de contratos aumentan cada vez más, de ahí la necesidad de ayudar a los trabajadores del sector jurídico con herramientas eficaces que permitan realizar en el menor tiempo posible todo el proceso de certificación y gestión del estado de los contratos. Con tal propósito, han sido desarrolladas las aplicaciones informáticas: InterCAS, ONBC, CLAIM.SA, Gesseb. La mayoría de las herramientas informáticas anteriormente mencionadas se encuentra disponibles solo para clientes que las puedan comprar, además de que son programas que responden a las leyes oriundas de los países donde fueron concebidas y no a las cubanas.

Además, en este sector es común el empleo de las herramientas de Microsoft Office para el control de los contratos. Esto provoca en ocasiones el envío postergado hacia otros organismos del contrato, así como el retraso en la

ejecución de las pruebas, la pérdida de la información ya que no se encuentra centralizada y disponible en una base de datos, entre otros múltiples problemas determinados por la inexistencia de un sistema informático para la gestión de este proceso.

Lo anteriormente expuesto permite resumir como **situación polémica**, las dificultades existentes en la elaboración, control y seguimiento del estado de los contratos. A partir de la problemática anterior emergió el siguiente **problema científico** a resolver ¿Cómo contribuir al perfeccionamiento y control de la gestión de los servicios de la Dirección Provincial de Bufetes Colectivos (DPBC) Matanzas? Como **objeto de estudio** la gestión y control de los servicios en la DPBC Matanzas y como **campo de acción** perfeccionamiento del sistema de gestión y control de los servicios de la DPBC Matanzas.

A manera de respuesta de este problema se plantea la siguiente **hipótesis**: si se desarrolla un sistema informático entonces se contribuirá al perfeccionamiento y control de la gestión de los servicios de la DPBC Matanzas.

Para dar cumplimiento a la hipótesis planteada se traza el siguiente **objetivo general**: desarrollar un sistema informático que perfeccione el sistema de gestión y control de los servicios de la DPBC Matanzas.

Los **objetivos específicos** de la investigación son:

- Analizar el estado del arte de la temática y las tendencias tecnológicas actuales necesarias para el desarrollo de la propuesta de solución.
- Diseñar la propuesta de solución a través de la metodología de software seleccionada.
- Validar la propuesta de solución mediante las pruebas funcionales.

Dentro de los métodos y técnicas de **metodología de la investigación** se utilizaron los métodos empíricos de entrevista e investigación con el objetivo de conocer todos los detalles sobre la organización, el tratamiento que se realiza a la información técnica, y los niveles de acceso a la misma que tienen los distintos trabajadores. Las entrevistas fueron realizadas a los letrados y a trabajadores que

están involucrados en los procesos de contratación. Las investigaciones se realizaron con el fin de analizar cómo se comportaba la problemática en su contexto natural y detallar el flujo de trabajo actual. Además, se realizó el análisis de documentos accesibles en sector jurídico relacionado con la investigación

El **resultado esperado** de este trabajo es contar con un sistema desarrollado en un ambiente web, la que permitirá rapidez, confiabilidad y seguridad en la interactividad entre los usuarios de la misma, posibilitando gestionar la información jurídica generada en el proceso de gestión de los contratos.

Atendiendo a lo planteado anteriormente, el trabajo de diploma queda estructurado en introducción, tres capítulos, conclusiones, recomendaciones, referencias bibliográficas y anexos, según sigue:

- Una Introducción, donde se caracteriza la situación problemática y se fundamenta el problema científico a resolver.
- Un primer capítulo donde se recoge el marco teórico referencial del tema y los principales conceptos que constituyen la base teórica de la investigación, así como el análisis de las principales tendencias tecnológicas y el estudio de los antecedentes que enmarcan la problemática planteada.
- Un segundo capítulo donde se caracteriza la organización objeto de estudio, se describe el desarrollo e implementación del software a través de la metodología XP.
- Un tercer capítulo donde se muestran las principales interfaces del prototipo inicial y se detallan los resultados de las pruebas funcionales aplicadas al software.
- Un apartado de conclusiones donde se verifica el cumplimiento de los objetivos trazados al inicio de la investigación.
- Las recomendaciones en la cual se plasman una serie de propuestas encaminadas a la continuidad de esta investigación.

- Las referencias de la bibliografía citada.
- Y los anexos correspondientes.

## Capítulo I. Fundamentación teórica y tendencias tecnológicas.

En este capítulo se describen los antecedentes del proceso que será objeto de estudio y se detallan aspectos generales de la organización, se refiere el análisis crítico que presenta el proceso actualmente y las causas que originaron la situación polémica. Se detallan además los sistemas existentes que automatizan el proceso a investigar, analizando sus características principales. Después de explicar el marco teórico referencial, se definen las tecnologías y la metodología de desarrollo de software utilizadas para llevar a cabo la elaboración del sistema.

### 1.1 Antecedentes del trabajo

Actualmente, existen en la red de redes(Internet) múltiples sitios web de abogados, que en su mayoría brindan información actualizada de los servicios que brindan, asesoramiento legal, contratos gratuitos y artículos referentes a las leyes oriunda del país que los ofertas.

Los distintos programas desarrollados para la organización de bufetes colectivos se centran en hacer cada día más satisfactorio el proceso de trámites de los clientes, razón de ser de la organización y que el trabajo de los letrados se desarrolle de forma fluida y sin interrupción debido a la burocracia que exige otras organizaciones y entidades legales que influyen en el proceso. Por esta razón se realizó una íntegra búsqueda bibliográfica para tener conocimiento de la mayor cantidad de trabajos interrelacionados con la materia en Cuba y en el mundo.

En la investigación desarrollada se realiza el estudio de siete herramientas que facilitan el proceso de organización, gestión, promoción, control y documentación de la información que a continuación relacionamos.

1.CLAIM S.A(1): Aplicación que promociona los servicios de consultores de propiedad intelectual, la cual constituye la única aplicación el sector jurídico integrada con pasarela de pago y que se genera tanto en plataforma Windows como en Linux. Está concebida para ser empleada en organizaciones cubanas,

*Fundamentación teórica y tendencias tecnológicas.*

aunque es utilizada a nivel mundial. No es una herramienta gratuita, su implementación está concebida en los 2436.00 €

2. InterCAS: Aplicación de escritorio que gestiona y controla los servicios de la organización nacional de bufetes colectivos, la cual constituye la única aplicación en este sector que gestiona los procesos de certificación.

3. Gesseb: Aplicación que gestiona los servicios en la organización específicamente en el departamento de Recursos Humanos, genera reportes generales de contratación, carece de herramientas que faciliten la operación de los usuarios.

4. Meci: Aplicación dinámica que trae consigo una biblioteca virtual donde se aglomera todas las bibliografías del sector jurídico en general, promociona los servicios de la organización.

5. ONBC: Aplicación que promociona los servicios de la organización, con énfasis en la gestión del conocimiento, carece de elementos para la recepción de envíos y notificaciones.

6. Sirec: Aplicación que gestiona los servicios en las notarías, además brinda prestaciones a la DPBC Matanzas, herramienta que no interactúa con otras aplicaciones por ser oriunda de ese sector, actualmente se encuentra en modificación por necesidades propias de la entidad

7. BES: Aplicación web que promociona los servicios de los bufetes especializados a nivel nacional, herramienta obsoleta que no ha sido actualizada desde su lanzamiento en 2004.

Para el análisis se tuvieron en cuenta los siguientes elementos que satisfacen los requisitos planteados por el cliente:

1. Aplicación Web.
2. Software gratuito.
3. Aplicación, software vigente (no más de siete años).

*Fundamentación teórica y tendencias tecnológicas.*

4. Codificadores de asuntos de la organización.
5. Control del proceso de certificación de los documentos.
6. Gestión del proceso de contratación.
7. Gestión de la ubicación y estado de los procesos.
8. Envío de infracciones
9. Recepción de Infracciones
10. Envío de notificación.
11. Recepción de notificaciones.
12. Reportes

*Tabla 1.1 comparativa de las herramientas investigadas*

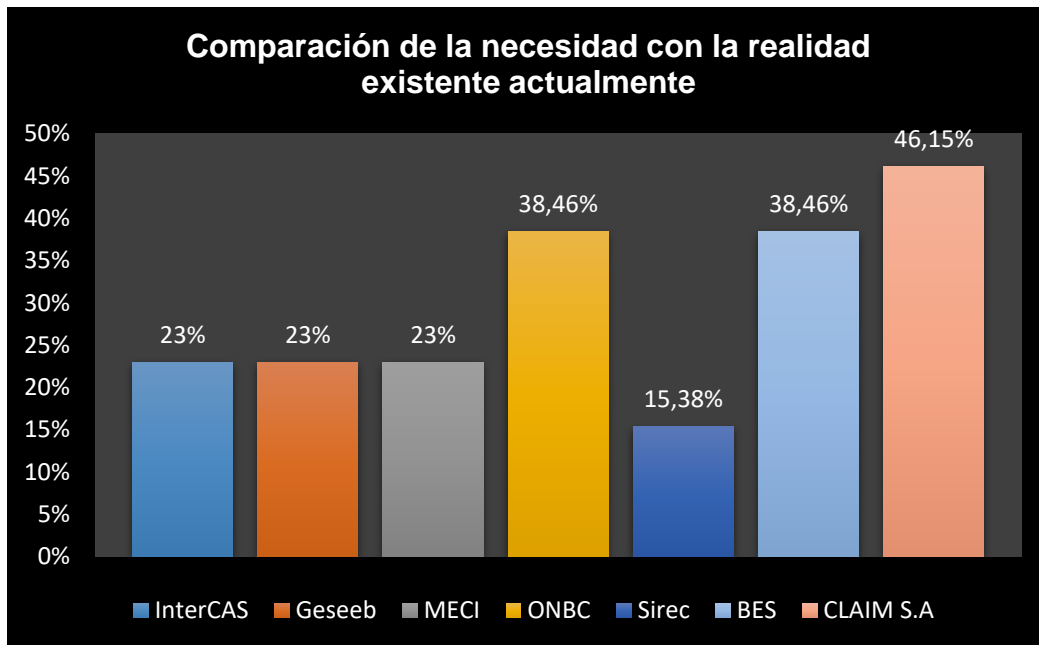
Nº	Herramientas	1	2	3	4	5	6	7	8	9	10	11	12
1	InterCAS		X	X			X						
2	Gesseb	X	X	X									
3	MECI	X			X						X		
4	ONBC	X			X						X	X	X
5	Sirec		X				X						
6	BES	X			X						X	X	X
7	CLAIM S.A	X			X						X	X	X
	Total	6	3	2	5	0	2	0	0	0	5	3	3
	Por ciento(%)	75	37.5	25	62.5	0	25	0	0	0	62.5	37.5	37.5

Fuente: Elaboración propia



*Fundamentación teórica v tendencias tecnológicas.*

Tabla 2.1 Muestra el porcentaje de cumplimiento de las herramientas analizadas con los requisitos planteados por el cliente



Fuente:  
Elabora

ción propia

En la investigación realizada muestra que ninguna de las herramientas analizadas cumple al menos el 50% de los requisitos que el cliente solicita. Además, se han analizado que los softwares que existen a nivel mundial no son softwares gratuitos y al mismo tiempo no siguen las legislaciones, regulaciones y normas cubanas por lo que no constituyen un elemento para su consulta. Prevalecen aplicaciones antiguas que no cumplen los estándares establecidos y el contenido que muestran está obsoleto o desactualizado. Ninguna de las herramientas analizadas gestiona la ubicación y el estado de los procesos (recurso de casaciones), ni planifica el entrenamiento para los futuros egresados, así como su evaluación y no controla el proceso de certificación de los documentos docentes y no docentes antes del MINREX. Requisitos todos enumerados por el cliente.

### 1.2 Objeto de estudio

La Organización Nacional de Bufetes Colectivos (ONBC) surge como una organización autónoma para que todo ciudadano natural o no tuvieran una persona capacitada para representarlos en todos los ámbitos jurídicos y

*Fundamentación teórica y tendencias tecnológicas.*

encontraran confianza en quien los representaba. A raíz que fue creciendo la organización se situó direcciones en los municipios cabeceras, así como bufetes en resto de la provincia por la demanda creciente de la población. Esta organización surge como idea del Comandante en Jefe de la Revolución Cubana con sus normas y procedimientos los cuales la hacen única en el país. La dirección provincial de Matanzas se encuentra ubicada en Contreras entre Jovellanos y Matanzas, municipio Matanzas, provincia Matanzas, su actividad fundamental es la contratación de personas jurídicas para representar en procesos litigiosos o no.

La misión de la empresa consiste en satisfacer las necesidades del cliente eje fundamental y razón de ser, así como la representación de iguales de los distintos sectores y contratación de los servicios especializados, directamente o como contrapartida de firmas extranjeras.

La visión de la empresa está enfocada en ser una organización empeñada en el desarrollo integral del sector jurídico, de manera que alcance y el desarrollo de la organización sea en un ambiente ecuánime, productivo y desarrollador.

Su objeto social consiste en:

- Brindar servicios de asesoramiento legal a clientes nacionales y extranjeros.
- Brindar servicios de traducción, certificación en las embajadas y consulados ubicados en el país en pesos convertibles.
- Brindar servicios asesoramiento legal a empresas en pesos cubanos.
- Brindar servicios de representación a los ciudadanos ante los órganos judiciales en pesos cubanos.
- Brindar servicios de oficios y de tramites notariales.

*Fundamentación teórica y tendencias tecnológicas.*

### 1.3 Descripción de los procesos que serán objeto de automatización.

Los procesos que serán objeto de automatización tenemos:

- Administrar la seguridad y el control de usuarios y su acceso a datos.
- Gestión de los contratos para su funcionamiento en tiempo real de la información.
- Generar diferentes reportes basados en el nivel de contratación y a petición del cliente.
- Gestión de las infracciones cometidas por los letrados.
- Gestión de los municipios y bufetes.
- Gestión del estado de los contratos.
- Gestión de las notificaciones.
- Gestión de las distintas fases de los contratos.

### 1.4 Tecnologías consideradas para la implementación del sistema.

Para el desarrollo de una solución eficiente es necesario realizar un estudio previo de las tecnologías existentes proporcionada a su vez por el cliente a fin de cumplir con las normas y procedimientos que rigen a la organización la cual fue objeto de estudio, para así seleccionar las más adecuadas para la tarea en cuestión. Sobre esta base se seleccionaron las herramientas tecnológicas que a continuación se detallan.

#### 1.4.1 Lenguaje del lado del cliente

Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento, en el proyecto se emplearon las siguientes tecnologías:

## HTML 5

Del lado cliente se desea utilizar HTML (*Hyper Text Markup Language*), lenguaje con el que se escriben las páginas web. Es un lenguaje de hipertexto, es decir, un lenguaje que permite escribir texto de forma estructurada, y que está compuesto por etiquetas, que marcan el inicio y el fin de cada elemento del documento. Un documento hipertexto no sólo se compone de texto, puede contener imagen, sonido, video, etc., por lo que el resultado puede considerarse como un documento multimedia. Los documentos HTML deben tener la extensión html o htm (*HyperText Markup*), para que puedan ser visualizados en los navegadores que se encargan de interpretar el código y mostrar a los usuarios las páginas web resultantes del código interpretado. (Lujan, 2014). HTML5 promete cambiar la forma en que las aplicaciones web se desarrollan a través de su abanico de funcionalidades de lado del cliente incluyendo hojas de estilos de código (CSS, por sus siglas en inglés) y otras características compatibles con los navegadores actuales. (Collins, 2012). Este lenguaje como su antecesor HTML 4 hace uso también de las librerías de JavaScript y la combinación de estos lenguajes hacen del diseño y maquetación de la web una experiencia más fácil y amena, de aquí la justificación para su uso en este trabajo.

## CSS 3

CSS son las siglas de *Cascade Style Sheet* que traducido al español significa hojas de estilo en cascada. Las hojas de estilo es una tecnología que permite controlar la apariencia de una página web. En un principio, los sitios web se concentraban más en su contenido que en su presentación, tal como sucede con HTML que no pone atención en la apariencia del documento, sino en la estructura. CSS describe cómo los elementos dispuestos en la página son presentados al usuario, siendo es un gran avance que complementa el HTML y la Web en general. Con las hojas estilo se pueden especificar el tamaño, fuentes, color, espaciado entre textos y recuadros, así como el lugar donde disponer texto

*Fundamentación teórica y tendencias tecnológicas.*

e imágenes en la página. (Gauchat, 2012). La elección de CSS3 para desarrollar este proyecto viene dada debido a la gran variedad de propiedades que incorpora con respecto a su antecesor. En esta nueva versión según (Gauchat, 2012) se añaden propiedades como *border-radius* (genera esquinas redondeadas). *Box-shadow* y *text-shadow* se encargan de proporcionar sombra a la caja de texto y al propio texto. *Border-image* (crea un borde con una imagen personalizada), estas entre otras propiedades dotan al diseño de un resultado elegante y atractivo.

### *JavaScript*

Java Script es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página web. Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado. Con Java Script se pueden crear efectos especiales en las páginas y definir interactividades con el usuario. Java Script permite ejecutar instrucciones como respuesta a las acciones del usuario (eventos), con lo que se pueden crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo. (Gauchat, 2012). Además, Java Script pone a disposición del programador todos los elementos que forman la página web, para que éste pueda acceder a ellos y modificarlos dinámicamente. Se hará uso de este lenguaje en el proyecto debido a que es un lenguaje pensado para mejorar la interfaz de usuario y proveer dinamismo a la web, se complementa además con otro de los lenguajes que serán utilizados en este trabajo (HTML5).

#### 1.4.2 Lenguaje del lado del servidor

### *C#*

Este lenguaje se basa en una fuerte herencia de C++.y como se convierte en inmediatamente familiar para los programadores de C++ y Java. “C# es un lenguaje de programación moderno e intuitivo orientado a objetos que ofrece

*Fundamentación teórica y tendencias tecnológicas.*

importantes mejoras, incluidos un sistema de tipos unificados, código no seguro para un control máximo del programador y nuevas estructuras de lenguaje muy eficaces y fáciles de entender por la mayoría de los programadores". (Kovacs, 2011). En varios sitios y bibliografías se define como un lenguaje moderno, potente, flexible. Para este trabajo la autora optará por el empleo de este lenguaje por su sintaxis fácil y rápida de asimilar, unido a que la curva de aprendizaje de este lenguaje ha sido más elevada con respecto a otros. C# es además desarrollado por Microsoft como parte de su plataforma .NET, por tanto, se integra perfectamente al Entorno de Desarrollo Integrado (IDE) Visual Studio que se utilizará en este proyecto.

*Motor de vista Razor*

La sintaxis Razor es una sintaxis de programación de ASP.NET simplificada que facilita la operación de incrustar código basado en servidor en el formato HTML de una página web. Razor es mucho más fácil para los principiantes que ASP. Los desarrolladores experimentados de ASP.NET verán que con la sintaxis Razor son más productivos (Álvarez, 2009). La autora se decanta por este motor de vistas ya que su sintaxis tiene como base lenguajes como C# y HTML, ambos utilizados en el proyecto. Este motor permite además minimizar el número de caracteres y pulsaciones de tecla que se necesitan en un archivo. A diferencia de la mayoría de las sintaxis de programación web basada en servidor, no necesita interrumpir la codificación para denotar los bloques de servidores en HTML explícitamente. El analizador es capaz de deducir los bloques del código. Esto hace que la sintaxis sea más compacta y fácil escribir.

1.4.3 Frameworks

*Bootstrap 3*

Bootstrap 3 y sus antecesores Bootstrap 2 y Bootstrap son un conjunto de herramientas gratuitas para crear sitios web y aplicaciones web. Contiene HTML y basados en CSS plantillas de diseño para tipografía, formas, botones, navegación

*Fundamentación teórica y tendencias tecnológicas.*

y otros componentes de la interfaz, así como las extensiones de JavaScript opcionales. Bootstrap fue desarrollado por Mark Otto y Jacob Thornton en Twitter como un Frameworks para fomentar la coherencia entre los instrumentos internos. Antes de Bootstrap, se utilizaron varias bibliotecas para el desarrollo de la interfaz, lo que llevó a inconsistencias y una alta carga de mantenimiento (Gauchat, 2012). Desde la aparición de Bootstrap 3 el framework se ha vuelto bastante más compatible con desarrollo web responsive, de ahí una de las ventajas que aparece en su nueva versión. Con el desarrollo de webs responsables se logra que las páginas sean adaptables a todo tipo de dispositivos y pantallas, sea cual sea su tamaño. La utilización de este framework en este proyecto viene dada por la anteriormente expuesta ventaja sumada a las siguientes:

- Es compatible con la mayoría de navegadores web del mercado como Google Chrome, navegador que se utilizará en el desarrollo del software.
- Se interpola con CSS y JavaScript (lenguajes que se utilizan en el trabajo) haciendo uso de las librerías de este último.
- Permite velocidad al programar ya que basta con seleccionar algunas de sus ya implementadas clases y lograr casi sin esfuerzo grupos de botones, barras de navegación, dropdowns, formularios etc.

### *JQuery*

JQuery es una poderosa biblioteca de Java Script que ayuda a desarrolladores y diseñadores a agregar elementos interactivos y dinámicos a sus sitios, limando inconsistencias en los navegadores y reduciendo grandemente el tiempo de desarrollo (Chaffer & Swedberg, 2010).

### ASP.Net

Constituye un entorno para aplicaciones web desarrollado y comercializado por Microsoft. Es usado por programadores y diseñadores para construir sitios web dinámicos, aplicaciones web y servicios web XML (Freeman, 2015). Apareció en enero de 2002 con la versión 1.0 del .NET Framework, y es la tecnología sucesora de la tecnología *Active Server Pages* (ASP). ASP.NET está construido sobre el *Common Language Runtime*, permitiendo a los programadores escribir código ASP.NET usando cualquier lenguaje admitido por el .NET Framework

#### Características de ASP.NET

Esposito (2014) plantea que uno de los objetivos más importantes considerados para creación de la tecnología .NET es el de otorgar un gran rendimiento. Es por ello que se plantean las principales características que combina .NET para convertirse en una de las mejores opciones para el desarrollo de aplicaciones. Se mencionan las características más sobresalientes.

- **Eficiencia:** para que la tecnología tenga éxito al ser utilizada por las Empresas, el personal especializado debe estar capacitado para migrar las aplicaciones y así evitar un rendimiento deficiente al ejecutar el código, ya que el Common Language Runtime (CLR) ejecuta el código de manera especial. El CLR compila en un lugar particular todos los códigos de aplicaciones en códigos naturales de máquinas, esto con la finalidad de asegurar un rendimiento óptimo. (Esposito, 2014). La conversión mencionada anteriormente por el autor, puede realizarse en el momento en que se ejecuta la aplicación o bien cuando se instala la aplicación por primera vez. La compilación realizada en el proceso hace uso de todas las características del microprocesador disponible en diferentes plataformas, logrando así superar a las aplicaciones tradicionales de Windows.



*Fundamentación teórica y tendencias tecnológicas.*

- **Soporte de lenguajes:** esta característica es quizás considerada como una de las más importantes para beneficio de los desarrolladores, debido que ASP.NET ofrece la posibilidad de escribir códigos en diversos lenguajes. ASP.NET soporta la programación en lenguajes como, Visual Basic.Net y C#.
- **Código y contenido por separado:** en la versión tradicional ASP, se presenta el inconveniente de tener que crear la interfaz de usuario y el código ASP de manera conjunta, es decir se realiza una combinación de código de imágenes, botones y tablas de HTML y secciones en VBScript o JavaScript pocas prácticas. La versión actual ASP .NET soluciona el problema separando la interfaz de usuario con el código.
- **Compatibilidad con navegadores:** permite la creación de páginas Web que funcionan correctamente en todos los navegadores. Esta mejora se incluye gracias a los controles de servidor que posee la nueva versión. En el momento en el que un control es procesado, automáticamente se realiza un chequeo del tipo de navegador que se está ejecutando y se genera una página adecuada para el navegador.
- **Código Compilado:** la compilación no interpreta el código como lo hace la versión ASP. Sino que dentro del entorno New Generation Windows Services (NGWS) el código es compilado just- in- time, aumentando el rendimiento mediante el soporte nativo y servicios de caché.
- **Controles de Servidor:** dentro de los aspectos importantes de ASP.NET, se encuentra su librería de clases, la cual brinda al programador una herramienta para creación de aplicaciones multiplataforma, permitiendo también un considerable ahorro en las líneas de código empleado.

*Fundamentación teórica y tendencias tecnológicas.*

Ventajas de ASP.NET

De acuerdo con Freeman (2015) ASP. NET ofrece varias ventajas importantes referentes a los modelos de programación Web empleados antes del surgimiento de esta tecnología:

- **Mejor rendimiento:** A diferencia de sus predecesores es capaz de aprovechar las ventajas del enlace anticipado, la compilación just-in-time, la optimización nativa y los servicios de caché desde el primer momento. Por lo tanto, existe un incremento importante del rendimiento desde el inicio, donde se comienza a escribir el código.
- **Compatibilidad con herramientas de primer nivel:** El ambiente de trabajo se complementa con un diseño y una caja de herramientas muy completas. Los controles de servidor de arrastrar y colocar, y la implementación automática ejemplifican la eficacia de las herramientas empleadas.
- **Eficacia y flexibilidad:** Se encuentra disponible para los programadores de aplicaciones web, debido a que la biblioteca de clases, la mensajería y las soluciones de acceso a datos se encuentran accesibles desde la Web de manera uniforme. Es posible elegir el lenguaje que mejor se adapte a la aplicación para desarrollar o incluso implementar varios lenguajes.
- **Simplicidad:** Facilita la realización de tareas como el envío de formularios, autenticación del cliente y la implementación y configuración de sitios. El ambiente de trabajo permite generar interfaces de usuario, que separan la lógica de aplicación del código de presentación, además de controlar eventos de forma sencilla a través del modelo de procesamiento de formulario de tipo Visual Basic. CLR simplifica la programación con

*Fundamentación teórica y tendencias tecnológicas.*

servicios de código administrado como el recuento de referencia automático y el recolector de elementos no utilizados.

- **Facilidad de uso:** Utiliza una configuración jerárquica basada en texto, lo cual simplifica la aplicación de la configuración al entorno del servidor y a las aplicaciones Web. La información de configuración es almacenada como texto sin formato, por lo que se puede aplicar la nueva configuración sin la ayuda de herramientas de administración local. Una aplicación ASP.NET se implementa en un servidor de forma sencilla mediante una copia de los archivos necesarios, por lo que no se requiere reinicio de servidor ni para implementar o reemplazar el código compilado en ejecución
- **Escalabilidad y disponibilidad ASP .NET:** Brown (2016) expone que fue diseñado tomando en cuenta la escalabilidad con características específicamente a medida, con el fin de mejorar el rendimiento en entornos agrupados y de múltiples procesadores. También controla y administra los procesos de tal manera que, si uno no se comporta adecuadamente, se pueda crear un nuevo proceso ayudando a mantener la aplicación disponible.
- **Posibilidad de personalización y extensibilidad:** ASP .NET cuenta con una arquitectura que permite a los programadores insertar código en el nivel adecuado. Y es posible extender o sustituir un subcomponente del motor de tiempo de ejecución con un componente escrito personalizado.
- **Seguridad:** Con la configuración por aplicación y la autenticación de Windows, es posible obtener una seguridad completa de las aplicaciones.

#### 1.4.4 Entity Framework y Object Relational Mapping

*Entity Framework* (EF) es un *Object Relational Mapping* (ORM) que permite a los desarrolladores de .NET trabajar con datos relacionales usando objetos específicos del dominio (Lerman & Rowan, 2011)

*Object Relational Mapping* (ORM) es una técnica de programación para convertir las tablas de una base de datos, en una serie de entidades que simplifiquen las tareas básicas de acceso a los datos para el programador. (Lerman & Rowan, 2011).

- **Persistencia transparente:** los objetos del dominio no saben nada acerca de la base de datos donde son persistidos, el framework lo resuelve en forma automática utilizando archivos de *mapping* expresados en XML.
- 
- **Soporte de polimorfismo:** puede cargar jerarquías de objetos en forma polimórfica. Soporte de los 3 niveles de mapeo de herencia: mapear toda una jerarquía de clases a una sola tabla, crear una tabla por cada clase concreta o crear una tabla por cada escalón de la jerarquía.
- 
- **Soporte completo de asociaciones:** los *frameworks* de ORM soportan el mapeo de todos los tipos de relaciones que pueden existir en un modelo de objetos del dominio (asociaciones 1...1, 1...N, N...M, unidireccionales y bidireccionales).
- 
- **Soporte de caching:** en el contexto de una transacción, se pueden disminuir la cantidad de veces en que se recupera información a la base de datos cacheando en memoria los objetos que son accedidos varias veces.
- 
- **Soporte de múltiples dialectos SQL:** se puede independizar completamente del tipo de base de datos utilizada. La aplicación puede

*Fundamentación teórica y tendencias tecnológicas.*

persistir sus datos en SQL Server, en Oracle, en MySQL, etc. simplemente cambiando la configuración correspondiente.

- 

Existen tres formas diferentes de trabajo del ORM en EF: *Database First* (permite primero crear la base de datos con sus tablas y luego incorporarlas a la aplicación), *Model First* (define las clases en la aplicación y a partir de ello crear la base de datos asociada) y *Code First* (utilizada para generar una nueva base de datos a partir de un modelo o mapear su modelo para una base de datos existente). En este trabajo se utilizó *Model First*.

#### 1.4.5 Técnicas de desarrollo web Ajax

Es el acrónimo de *Asynchronous JavaScript and XML*, la cual se considera una técnica de desarrollo web que permite implementar una comunicación asíncrona entre el cliente y el servidor. Ajax no es una tecnología sino un conjunto de tecnologías formada por JavaScript, HTML, XML, CSS, y XML HTTP *Request Object*, aplicadas de forma concreta, nos permite crear aplicaciones web más eficientes en la interacción con el usuario. Gracias a esta técnica, muchas de las tareas que habitualmente se realizan desde el lado servidor (por ejemplo, consultas a bases de datos) pueden realizarse parcialmente desde el lado cliente, evitando recargar la página y brindando una experiencia de navegación más fluida (Krall, 2013). La utilidad de esta técnica en el proyecto viene dada por sus principales ventajas como brindar la posibilidad de que las páginas se modifiquen sin tener que volver a cargarse, dándole al usuario la sensación de que los cambios se producen instantáneamente y como consecuencia al no recargarse la página se reduce el tiempo implicado en cada transacción. La nueva capa intermedia de AJAX mejora la respuesta de la aplicación, ya que el usuario nunca se encuentra con una ventana del navegador vacía esperando la respuesta del servidor.

#### 1.4.6 Herramientas de desarrollo

##### *Gestor de base de datos: Oracle 11g y Toad*

Oracle es una herramienta cliente/servidor para la gestión de Bases de Datos. Se considera uno de los sistemas gestores de base de datos más completos, destacando el soporte de transacciones, la estabilidad, la escalabilidad y el soporte multiplataforma. (Arthur, 2015).

Este ha sido diseñado para que las organizaciones puedan controlar y gestionar grandes volúmenes de contenidos no estructurados en un único repositorio con el objetivo de reducir los costes y los riesgos asociados a la pérdida de información. Esta herramienta hace que el proceso de recopilación de datos sea más completo, fácil y sencillo.

En la realización del sistema para la gestión del proceso de información del estado de los servicios en la DPBC Matanzas se utilizó la versión Oracle 11g. Esta puede ayudar a crear aplicaciones de bases de datos personalizadas. En lugar de utilizar un programa de bases de datos estandarizado, se puede obtener un programa personalizado que se ajusta a las necesidades del cliente pues permite crear una base de datos que sólo tiene las funciones que se necesitan. También permite eliminar cualquier función no necesaria en el sistema de bases de datos del negocio.

Utiliza estándares de la industria del código abierto para ayudar a que las funciones estén creadas específicamente para la realización de la aplicación. Oracle Database 11g también ayuda con la minería de datos. Esta aplicación de bases de datos hace que sea posible encontrar las relaciones en los datos. Para los trabajos con grandes cantidades de datos regularmente, esto puede ahorrarte tiempo y encontrar las relaciones que necesitas en los datos. También tiene la habilidad de descubrir patrones en los datos a través del tiempo, además de que cuenta con un sistema de gestión y desarrollo de la base de datos **TOAD** la cual

*Fundamentación teórica y tendencias tecnológicas.*

es una aplicación informática de desarrollo SQL y administración de base de datos, considerada una herramienta útil para los Oracle DBAs (administradores de base de datos). Actualmente está disponible para las siguientes bases de datos: Oracle Database, Microsoft SQL Server, IBM DB2, y MySQL y es una aplicación gratuita.

*Microsoft Visual Studio 2013 Express y ReSharper 8.0*

*Microsoft Visual Studio* es un Entorno de Desarrollo Integrado (IDE, por sus siglas en inglés) para sistemas operativos *Windows*. Soporta varios lenguajes de programación, tales como *Visual C++*, *Visual C#*, *Visual J#*, y *Visual Basic .NET*, al igual que entornos de desarrollo web como ASP.NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.

*Visual Studio* permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión .NET 2002). Así se pueden crear aplicaciones que se intercomunican entre estaciones de trabajo, páginas web y dispositivos móviles.

El *ReSharper* es un complemento para *Visual Studio* implementado por *JetBrains* para la auditoría de código de aplicaciones (en tiempo de diseño) que permite la refactorización del código fuente, define estándares de calidad en el software que se está desarrollado.

Además, instruye al desarrollador que lo utiliza al mostrar formas mejoradas para el código escrito por el desarrollador, al tiempo que fomenta el uso de buenas prácticas.

Este complemento comprueba la sintaxis del código, detectando código muerto/inalcanzable (*unreachable*), sugiere el uso de convenciones en el llamado de las variables (*UpperCamel Case*, *LowerCamel Case*), revisa la accesibilidad de clases y atributos y detecta los conflictos. (Microsoft Corporation, 2013)

### Modelo Vista Controlador (MVC)

El patrón de arquitectura Modelo Vista Controlador (MVC) es un patrón que define la organización independiente del Modelo (Objetos de Negocio), la Vista (interfaz con el usuario u otro sistema) y el Controlador (controlador del *workflow* de la aplicación).

De esta forma, dividimos el sistema en tres capas donde, tenemos la encapsulación de los datos, la interfaz o vista por otro y por último la lógica interna o controlador.

El patrón de arquitectura "modelo vista controlador", es una filosofía de diseño de aplicaciones, compuesta por:

- Modelo
  - Contiene el núcleo de la funcionalidad (dominio) de la aplicación.
  - Encapsula el estado de la aplicación.
  - No sabe nada / independiente del Controlador y la Vista.
- Vista
  - Es la presentación del Modelo.
  - Puede acceder al Modelo, pero nunca cambiar su estado.
  - Puede ser notificada cuando hay un cambio de estado en el Modelo.
- Controlador
  - Reacciona a la petición del Cliente, ejecutando la acción adecuada y creando el modelo pertinente.

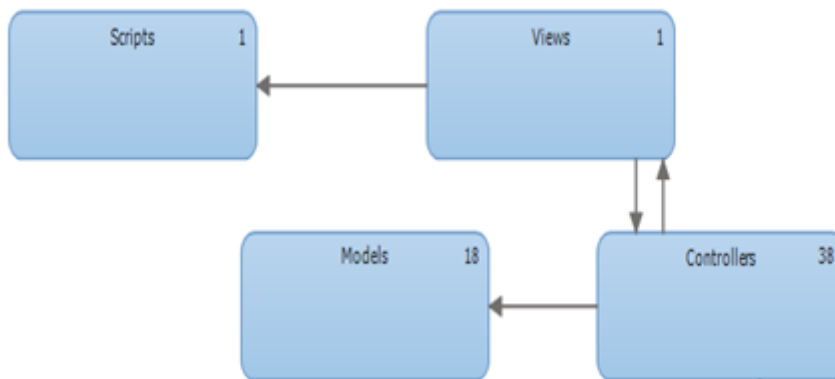
Para entender cómo funciona nuestro patrón MVC, se debe entender la división a través del conjunto de estos tres elementos y como estos componentes se comunican unos con los otros y con otras vistas y controladores externos al modelo principal. Para ello, es importante saber que el controlador interpreta las



*Fundamentación teórica y tendencias tecnológicas.*

entradas del usuario (tanto teclado como el ratón), enviado el mensaje de acción al modelo y a la vista para que se proceda con los cambios que se consideren adecuados. (\_\_, 2010)

Las Figura 1 y 2, visualiza la arquitectura en detalle, observándose las relaciones entre las capas de la arquitectura. Los diagramas son creados en Visual Studio.



*Ilustración 1. 1 capas de MVC*

Fuente: Elaboración propia.

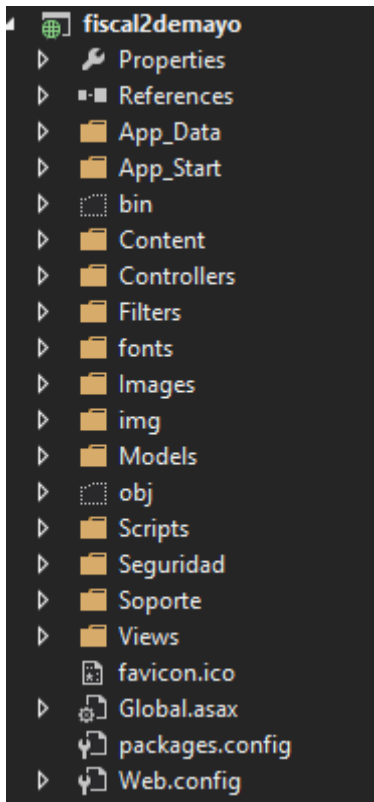


Ilustración 2.1

Fuente: Elaboración propia.

### 1.5 Metodología de desarrollo de software: *Extreme Programming (XP)*

Según (Borja López, 2017) las metodologías de desarrollo describen los pasos a seguir para desarrollar un producto de software, definen qué es lo que se debe hacer, cómo y quién debe hacerlo. Precisan el orden de las tareas, los artefactos que van a ser generados, quiénes son los responsables y cómo deben hacerse cada una de estas tareas durante el ciclo de desarrollo del proyecto.

Para la elección de la metodología a utilizar se deben tener en cuenta las características que reúne el equipo, las necesidades específicas de la situación y las prioridades demandadas por el cliente. Deben analizarse además dos factores fundamentales: el tipo de proyecto que se desea desarrollar y el tiempo que se dispone para desarrollar el mismo.

*Fundamentación teórica y tendencias tecnológicas.*

Actualmente existen dos grandes grupos de metodologías de desarrollo, las metodologías tradicionales o no ágiles y las metodologías ágiles: De acuerdo con Yolanda Borja (2017) una metodología de desarrollo es el conjunto de métodos que especifican quién debe hacer qué cosa, cuándo debe hacerse, y el tiempo en que debe hacerse, estableciendo un conjunto de roles (el quién), actividades (la cosa) y un ciclo de vida que establece las diferentes fases. Esta puede seguir uno o varios modelos de ciclo de vida. Según esta autora el ciclo de vida muestra qué es lo que hay que lograr a lo largo del desarrollo del proyecto, pero no cómo hacerlo, enfatiza en que la metodología indica cómo hay que obtener los distintos productos parciales y finales y como imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. La realización de aplicaciones y sistemas más complejos en la actualidad, ha llevado a la búsqueda de métodos para coordinar el trabajo, dado el tipo de tarea a realizar se puede seleccionar la metodología más exacta y adecuada.

Para el desarrollo de software tenemos las llamadas Metodologías Tradicionales o no ágiles y Metodologías Ágiles. Ver Anexo 1 (Tabla comparativa entre metodologías)

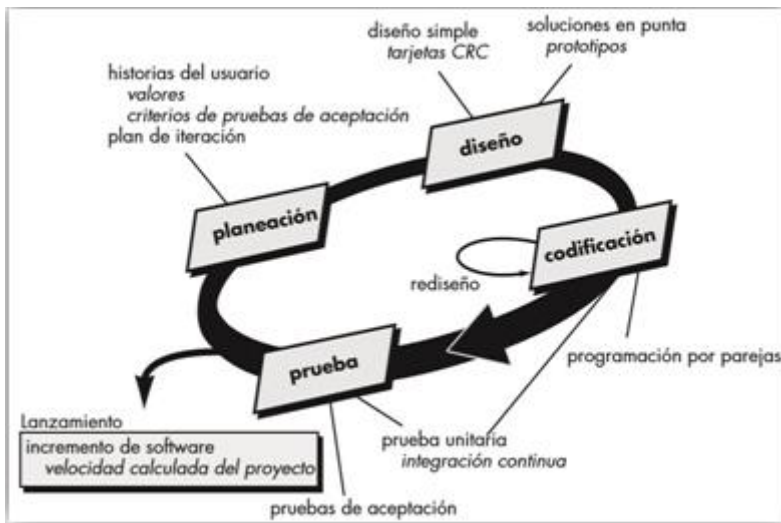
Para el desarrollo del sistema se escogió la Programación Extrema (XP), pues XP es el enfoque más utilizado de desarrollo de software ágil (*Pressman, 2010*). Es una metodología ágil utilizada para proyectos de corto plazo y propensos a cambios, equipos reducidos, cuya particularidad es tener como parte de éste al usuario final.

En todo trabajo realizado como parte de XP debe existir (Beck K., 2004):

- **Comunicación:** Entre los propios desarrolladores y entre estos y los clientes.
- **Simplicidad:** Crear un diseño sencillo que se implemente con facilidad en forma de código.

*Fundamentación teórica y tendencias tecnológicas.*

- **Retroalimentación:** Se obtiene del software implementado, del cliente y otros miembros del equipo de software.
- **Disciplina:** Por parte del equipo de desarrollo al ser muy probable que los requerimientos cambien, lo que traerá repeticiones sustanciales del diseño y del código implementado.
- El proceso de XP consta de cuatro actividades estructurales: **Planeación, Diseño, Codificación y Pruebas**, como muestra la Figura 3.



*Ilustración 3.1 Proceso de Programación Extrema(XP)*

Fuente: Pressman,2010

- **Planeación:** Entender el contexto del negocio y obtener los requerimientos necesarios. Se crean las primeras Historias de Usuario (HU) y el plan de entrega.
- **Diseño:** Es sencillo, lo cual se prefiere sobre una representación más compleja. Ocurre tanto antes como después de que comienza la codificación.
- **Codificación:** Se realizan pruebas unitarias a cada HU antes de iniciar la codificación, después de estas, el desarrollador está mejor

*Fundamentación teórica y tendencias tecnológicas.*

capacitado para centrarse en lo que debe implementarse para pasar la prueba.

- **Pruebas:** Además de las pruebas unitarias, en XP se desarrollan las pruebas de aceptación, son especificadas por el cliente y se centran en las funcionalidades generales del sistema.

### 1.6 Conclusiones del capítulo.

Luego de estudiar los antecedentes del proceso objeto de estudio y los aspectos generales de la entidad, así como las herramientas y tecnologías y la metodología de desarrollo de software, se concluye que:

1. Los sistemas existentes no resuelven la situación polémica planteada.
2. La combinación de las herramientas de desarrollo *Microsoft Visual Studio 2013* y *ReSharper 8.0*, utilizando las tecnologías antes mencionadas y la metodología XP, es la apropiada para desarrollar el sistema que dará solución al problema de esta investigación.
3. Se demuestra la necesidad de la utilización de software libre como una tendencia a asumir.
4. Para el desarrollo de las etapas de la Ingeniería de Software es factible utilizar XP.

En sentido general se ha contribuido a la mejor comprensión del objeto de estudio y se han establecido las bases para las siguientes fases de la investigación.

## Capítulo 2. Descripción de la solución propuesta.

En este capítulo se expone el diagnóstico de la situación real de la entidad, se identifica el flujo de trabajo, los requerimientos funcionales y no funcionales, apoyándose en la metodología de desarrollo de software Programación Extrema (XP), se abordarán los elementos necesarios para describir la solución propuesta y la seguridad del sistema.

### 2.1 Descripción de la propuesta.

La propuesta se basa en la creación de un sistema Web, programada utilizando la arquitectura MVC, las capas utilizadas son:

- **Capa de Acceso a Datos:** está formada por el *Entity Data Model*(EDM) generado por el motor de *ADO .Net Entity Framework* disponible en el *Microsoft Framework 4.5*.
- **Capa de Negocio:** está formada por clases controladores que se encargarán de dar respuesta a las acciones de los usuarios mediante las clases modelos y las vistas.
- **Capa de Interface:** está formada por vistas. *cshtml* (motor *Razor*) de cada área y controlador.

Dentro de las funcionalidades principales de la propuesta están:

- Administrar la seguridad y el control de usuarios y su acceso a datos.
- Gestión de los contratos para su funcionamiento en tiempo real de la información.
- Generar diferentes reportes basados en el nivel de contratación y a petición del cliente.
- Gestión de las infracciones cometidas por los letrados.

Para lograr almacenar y manipular de forma eficiente la información del sistema se utilizará una Base de Datos en el servidor de datos Oracle 11g. La membresía se controlará mediante el servidor de dominio de Windows de la organización en su nodo central.

## 2.2 Equipo de trabajo y roles.

La metodología XP está diseñada para ser aplicada en un equipo de trabajo formado por el cliente y más de un desarrollador según los diferentes roles; en éste caso el equipo de trabajo está formado por el cliente, un entrenador y un desarrollador. Entre las características fundamentales del equipo de trabajo de XP tenemos:

- El cliente forma parte del equipo, escribe las Historias de Usuario, las pruebas funcionales para validar su implementación, asigna la prioridad de la HU y propone cuál se implementará en cada iteración.
- Se trabaja cuarenta horas semanales.
- Se trabaja desarrollando pequeñas versiones funcionales que se entregan al final de cada iteración.

El equipo de trabajo se encuentra formado por:

1. Celia Fuentes Gómez, Autora.
2. Liz Pérez Martínez, Tutora.
3. Sergio Jorge Pagés Valdés, Cliente.

*Tabla 3.2 Equipo de trabajo y roles*

Miembros	Roles que desempeña
Celia Fuentes Gómez	Programador, Encargado de pruebas,

Descripción de la solución propuesta

T		Encargado de seguimiento.
a	Liz Pérez Martínez	Entrenador, Gestor, Consultor.
T		
a	Sergio Jorge Pagés Valdés	Cliente, Encargado de pruebas.

bla

Fuente: Elaboración propia.

Tabla 4.2 Usuarios y permisos

Usuarios	Permisos
<b>Administrador</b>	<p>Administra el sistema y tiene acceso a la gestión de usuarios, municipios, contratos, bufetes, fase y codificador de asuntos. Inserta, modifica y elimina usuarios (clientes, letrados, director y técnico tramitador).</p> <p>Inserta, modifica y elimina municipios, contratos, bufetes, fase y los codificadores de asuntos.</p>
<b>Letrado o Abogado</b>	<p>Gestiona todo lo relacionado con los contratos y notificaciones. Se encarga de la gestión del proceso de contratación.</p> <p>Inserta, modifica y elimina contratos.</p>
<b>Director</b>	<p>Gestiona todo lo relacionado con los contratos e infracciones. Se encarga de la gestión del proceso de</p>



*Descripción de la solución propuesta*

	<p>contratación.</p> <p>Inserta, modifica y elimina contratos e infracciones. Consulta toda la información que se gestiona en el sistema.</p>
<b>Técnico Tramitador</b>	<p>Gestiona todo el flujo informativo acerca del estado de los contratos.</p> <p>Se ocupa de la gestión del estado de la fase de los contratos, así como la información oportuna al cliente.</p>
<b>Cliente</b>	<p>Consulta toda la información referente a su contrato.</p>

Fuente: Elaboración propia

### 2.3 Análisis de Requerimientos

El levantamiento de los requerimientos constituye el paso primordial en el proceso de construcción de software, pues estos darán origen a las funcionalidades principales a implementar. Dichos requerimientos se especifican en las historias de usuarios (HU), las cuales, pueden diferir de los requerimientos iniciales, ya que un mismo requerimiento puede dividirse en varias HU, o pueden agruparse varios en una misma HU. Los requerimientos funcionales iniciales son los siguientes los cuales pueden ser modificados de ser necesarios:

1. Crear y almacenar la información en Bases de Datos (BD)
2. Seguridad y autenticación de usuarios
3. Gestionar usuarios

4. Gestionar roles
5. Gestionar contratos
6. Gestionar fase de los contratos
7. Gestionar fase
8. Gestionar codificador de asuntos
9. Gestionar letrados(Abogados)
10. Gestionar clientes
11. Gestionar infracciones
12. Gestionar municipios
13. Gestionar notificaciones
14. Gestionar carta
15. Generar reportes informativos
16. Trazas de Seguridad

#### Requisitos no Funcionales

Son requerimientos que imponen restricciones en el diseño o implementación de un sistema como estándares de calidad. Son propiedades o cualidades que el producto debe tener. Los siguientes requisitos no funcionales, están orientados a la seguridad, estandarización, y centralización de datos, para asegurar así, el rendimiento y calidad del software.

- Los tiempos de respuesta del software deben ser mínimos.
- El software deberá permitir el ingreso solo al personal autorizado.
- El software deberá mostrarle a cada usuario solo sus datos.
- El software debe poder ser usado por personas con conocimientos básicos en el manejo de computadoras.

Descripción de la solución propuesta

- El software debe contar con un manual de usuario que guía al mismo de cómo debe trabajar en el software.
- El software debe presentar una interfaz amigable y sencilla.

Tabla 5.2 Requisitos de hardware

	Procesador	RAM (Random Access Memory)	Disco Duro
<b>Servidor</b>	Core i3 4 GHz	4 GB	80 GB libres
<b>Cliente</b>	Core Duo 1+ GHz	1 GB	40 Mb libres

Fuente: Elaboración propia.

Tabla 6.2 Requisitos de Software

Servidor	Cliente
<p><b>Sistemas Operativos:</b></p> <ul style="list-style-type: none"> <li>➤ Windows Server 2008 R2 SP1 (64 bits)</li> <li>➤ Windows Server 2008 SP2 (32 y 64 bits)</li> </ul> <p><b>Sistema de Gestión de Base de Datos:</b> Oracle 11g 2005 /2008</p>	<p><i>Navegadores:</i></p> <ul style="list-style-type: none"> <li>➤ Mozilla Firefox 15.0 o superior</li> <li>➤ Google Chrome 30.0 o superior</li> </ul>

Fuente: Elaboración propia.

## 2.4 Artefactos de la metodología

Las HU son la técnica utilizada en XP para especificar los requisitos del software., se utilizan para estimar tiempos de desarrollo del sistema, también se hace uso de ellas en la fase de pruebas, para verificar si el sistema cumple con lo que especifica la historia de usuario. “Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales” (Borja López, 2017). Resumiendo, una historia de usuario es un texto de una o dos frases en las que se dice algo que debe hacer el sistema. Las historias de usuario poseen los detalles mínimos que se requieren para que los programadores puedan realizar una estimación poco riesgosa del tiempo que llevará su desarrollo.

La metodología XP presenta escalas para evaluar la complejidad de las Historias de Usuario, dichas escalas se presentan a continuación:

- Alta: Asignada a las Historias de Usuario que corresponden a funcionalidades esenciales en el desarrollo del proyecto, a las que el cliente define como primordiales.
- Media: Dada a las Historias de Usuario que resultan para el cliente como funcionalidades a tener en cuenta, sin que estas tengan una afectación directa sobre el proyecto que se esté desarrollando.
- Baja: Se le otorga a las Historias de Usuario que constituyen funcionalidades que sirven de ayuda al control de elementos asociados al equipo de desarrollo, a la estructura y no tienen nada que ver con el proyecto en desarrollo.

La Escala Nominal de Riesgo en Desarrollo definida por el programador:

- Alta: Cuando para la implementación de la Historia de Usuario se considera la posible existencia de errores que lleven a inoperatividad del código.

*Descripción de la solución propuesta*

- Media: Cuando pueden aparecer errores en la implementación de la Historia de Usuario que puedan retrasar la entrega de la versión.
- Baja: Cuando pueden aparecer errores que serán tratados con relativa facilidad sin que traigan perjuicios para el desarrollo del proyecto.

Estas ayudan en la comunicación entre el cliente y los desarrolladores y pueden ir cambiando a medida que avanza el proyecto y que el cliente vea nuevas posibilidades y soluciones. Cuando llega la hora de implementar una historia de usuario, el cliente y los desarrolladores se reúnen para concretar y detallar lo que tiene que hacer dicha historia.

*Tabla 7.2 Historias de usuario, prioridad, riesgo y dependencia.*

Nº	Nombre de HU	Prioridad	Riesgo	Dependencia
1	Diseño y creación Base de Datos (BD)	Alta	Alto	
2	Autenticación de usuarios	Baja	Alto	1,13
3	Gestionar roles	Media	Medio	1
4	Gestionar contratos	Media	Medio	8,9
5	Gestionar fase de los contratos	Media	Medio	4, 5
6	Gestionar fase	Media	Medio	1
7	Gestionar codificador de asunto	Media	Medio	1
8	Gestionar Letrados	Media	Medio	1,15
9	Gestionar Clientes	Media	Medio	1,15
10	Gestionar infracciones	Media	Medio	4,8
11	Gestionar municipios	Baja	Medio	1

*Descripción de la solución propuesta*

<b>12</b>	Gestionar notificaciones	Media	Medio	8, 9
<b>13</b>	Gestionar carta	Media	Medio	4
<b>14</b>	Generar reportes informativos	Baja	Bajo	
<b>15</b>	Gestionar usuarios	Baja	Alto	1,3
<b>16</b>	Generar Trazas de seguridad	Baja	Alto	
<b>17</b>	Diseño de la Interfaz Usuario	Alto	Alto	1,3,15

Fuente: Pressman,2010

**2.4.1 Plan de Entregas**

Está creado teniendo en cuenta la información anterior, se observa en la siguiente tabla que las entregas se realizan al final de cada iteración y que las estimaciones de esfuerzo la establecen los programadores utilizando como medida una semana ideal de programación. Las historias se estiman entre 1 y 3 semanas donde su planificación para cada una de las iteraciones se realizó en entrevistas concebidas con el cliente. Estas entrevistas se efectuaron antes de comenzar a trabajar en una nueva iteración, en la mayoría de los casos en la reunión de entrega de la iteración precedente.

Cada iteración incluye la realización de una o más historias de usuarios, y pueden realizarse tantas iteraciones como sean necesarias y estime el programador. Al final de cada iteración no es obligatorio realizar una entrega, aunque se recomienda que las entregas se realicen una vez concluidas estas. El plan de iteraciones especifica exactamente qué HU serán implementadas en cada iteración del sistema. En la tabla se incluye además de las iteraciones los puntos estimados y la fecha de entrega de cada iteración.

**Puntos Estimados o Estimaciones:** Los puntos estimados, no son más que la estimación de duración de una HU. Las estimaciones no sobrepasan las 3 semanas por lo que:

---

*Descripción de la solución propuesta*

- 1 punto estimado equivale a 1 semana de trabajo.
- 2 puntos estimados equivalen a 2 semanas de trabajo.
- 3 puntos estimados equivalen a 3 semanas de trabajo.

Cada semana de trabajo equivale a 4 días laborales.

**Fecha de entrega:** Las fechas de entrega están definidas por HU en cada iteración. En la siguiente tabla se ha hecho coincidir la fecha de entrega de cada HU al terminar la misma. Debe tenerse en cuenta que, como define la metodología, al finalizar cada iteración se dedica una semana de pruebas a las historias implementadas en la misma. Por tanto, al finalizar cada iteración deberán contarse 4 días de prueba hasta la próxima entrega de HU.

Se tuvo en cuenta todo el tiempo que la planificación podía variar pues estaba sujeta a los cambios que surgieran en las historias de usuario que incluye cada iteración.

*Tabla 8.2: Plan de entregas*

Nº	Nombre de HU	Estimación	Iteración	Fecha de entrega
1	Diseño y creación Base de Datos (BD)	3	1	3/10/17- 28/11/17
2	Autenticación de Usuarios	1	1	3/10/17- 28/11/17
3	Gestionar roles	2	1	3/10/17- 28/11/17
4	Gestionar contratos	2	2	5/12/17- 23/1/17
5	Gestionar fase de los contratos	2	2	5/12/17- 23/1/17

## Descripción de la solución propuesta

<b>6</b>	Gestionar fase	2	2	5/12/17- 23/1/17
<b>7</b>	Gestionar codificador de asunto	2	4	27/3/18- 22/5/18
<b>8</b>	Gestionar Letrados	1	3	30/1/18- 22/3/18
<b>9</b>	Gestionar Clientes	2	3	30/1/18- 22/3/18
<b>10</b>	Gestionar infracciones	1	3	30/1/18- 22/3/18
<b>11</b>	Gestionar municipios	1	3	30/1/18- 22/3/18
<b>12</b>	Gestionar notificaciones	1	4	27/3/18- 22/5/18
<b>13</b>	Gestionar carta	2	2	5/12/17- 23/1/17
<b>14</b>	Generar reportes informativos	2	4	27/3/18- 22/5/18
<b>15</b>	Gestionar usuarios	1	4	27/3/18- 22/5/18
<b>16</b>	Generar Trazas de seguridad	2	1	3/10/17- 28/11/17
<b>17</b>	Diseño de la Interfaz Usuario	1	4	

Fuente: Pressman, 2010



---

*Descripción de la solución propuesta*

Cuando se utiliza el término gestión en el nombre o descripción de una historia de usuario se hace referencia al manejo de adicionar, modificar, visualizar y eliminar la información relativa a la misma.

#### 2.4.2 Descripción de las historias de usuario.

Historias de Usuarios.

Tabla 9.2 HU17

Historia de Usuario	
Número: 17	Usuario: Administrador
Nombre historia: Diseño de la interfaz de usuario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 4
Programador responsable: Celia Fuentes Gómez	
Descripción: Crear las interfaces de usuarios del sistema.	
Observaciones: Deben ser creadas correctamente.	

Fuente: Elaboración propia.

Tabla 10.2 Crear Interfaz Principal

Tarea	
Número de tarea: 2.1	Número de historia: 2
Nombre de tarea: Crear Interfaz Principal	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Responsable: Celia Fuentes Gómez.	
Descripción: Crear la interfaz principal del sistema.	

Descripción de la solución propuesta

Fuente: Elaboración propia.

Tabla 11.2 HU2

Historia de Usuario	
Número: 2	Usuario: Administrador, Técnico Tramitador, Letrado, Director y Cliente.
Nombre historia: Autenticarse	
Prioridad en negocio: Baja	Riesgo en desarrollo: Alto
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Celia Fuentes Gómez	
Descripción: El usuario introduce su nombre y su contraseña para acceder al sistema, de ser correctos se le mostrará un entorno de trabajo acorde con su nivel de acceso y de no serlo, el sistema mostrará un error.	
Observaciones: Confirmar el nivel de acceso del usuario. Verificar la integridad de los datos.	

Fuente: Elaboración propia.

Tabla 12.2 Autenticarse

Tarea	
Número de Tarea: 3.1	Número de Historia: 2
Nombre de Tarea: Autenticarse.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Programador Responsable: Celia Fuentes Gómez.	
Descripción: El usuario introduce su nombre y su contraseña para acceder al	

Descripción de la solución propuesta

sistema, de ser correctos se le mostrará un entorno de trabajo acorde con su nivel de acceso y de no serlo, el sistema mostrará un error.

Fuente: Elaboración propia.

Tabla 13.2 HU15

Historia de Usuario	
Número:15	Usuario: Administrador y cliente
Nombre historia: Gestionar usuarios	
Prioridad en negocio: Baja	Riesgo en desarrollo: Alto
Puntos estimados: 1	Iteración asignada: 4
Programador responsable: Celia Fuentes Gómez.	
Descripción: Mostrar, eliminar, editar y agregar usuarios que puedan acceder al sitio.	
Observaciones: Solo puede acceder a esta página un usuario con rol administrativo, debidamente autenticado.	

Fuente: Elaboración propia.

Tabla 14.2 Autenticación de usuarios

Tarea	
Número de tarea: 4.1	Número de historia: 15
Nombre de tarea: Autenticación de usuario	
Tipo de tarea: Desarrollo	Puntos estimados:1
Responsable: Celia Fuentes Gómez	
Descripción: Implementar una función que permita al usuario acceder al sistema dependiendo de su rol.	

Descripción de la solución propuesta

Fuente: Elaboración propia.

Tabla 15.2 Agregar usuarios.

Tarea	
Número de tarea: 4.2	Número de historia: 15
Nombre de tarea: Agregar usuario	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Responsable: Celia Fuentes Gómez.	
Descripción: Implementar una función que permita agregar usuarios llenando los campos correspondientes a sus datos personales.	

Fuente: Elaboración propia.

Tabla 16.2 HU3

Historia de Usuario	
Número: 3	Usuario: Administrador
Nombre historia: Gestionar roles y permisos	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Celia Fuentes Gómez	
Descripción: Permitir modificar los roles del sistema.	
Observaciones: Solo puede acceder a esta página un usuario con rol administrativo, correctamente autenticado.	

Fuente: Elaboración propia.

Tabla 17.2 Roles

Tarea
-------

Descripción de la solución propuesta

Número de tarea: 5.1	Número de historia: 3
Nombre de tarea: Agregar rol	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Responsable: Celia Fuentes Gómez	
Descripción: Se implementarán las funciones que permitan agregar y retirar roles a los usuarios del sistema.	

Fuente: Elaboración propia.

Tabla 18.2 Permisos

Tarea	
Número de tarea: 5.2	Número de historia: 3
Nombre de tarea: Agregar permiso	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Responsable: Celia Fuentes Gómez	
Descripción: Se implementarán las funciones que permita agregar y retirar permisos a los roles creados.	

Fuente: Elaboración propia.

Tabla 19.2 HU 4

Historia de Usuario	
Número:4	Usuario: Cliente y técnico tramitador
Nombre historia: Gestionar contratos	
Prioridad en negocio: media	Riesgo en desarrollo: Medio
Puntos estimados:2	Iteración asignada:2

Descripción de la solución propuesta

Programador responsable: Celia Fuentes Gómez
Descripción: Se inicia si el cliente necesita realizar un contrato por un servicio determinado.
Observaciones: Es de vital importancia verificar los datos proporcionados por el cliente.

Fuente: Elaboración propia.

Tabla 20.2 Adicionar contratos

Tarea	
Número de tarea: 6.1	Número de historia: 4
Nombre de tarea: Adicionar contratos	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Responsable: Celia Fuentes Gómez	
Descripción: Implementar la función que te permita agregar los contratos proporcionados por el usuario.	

Fuente: Elaboración propia.

Tabla 21.2 HU 5

Historia de Usuario	
Número: 5	Usuario: Letrados y Técnico Tramitador
Nombre historia: Gestionar Fase de los contratos	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 2	Iteración asignada: 2
Programador responsable: Celia Fuentes Gómez	

Descripción: Se inicia cuando el usuario necesita conocer el estado del contrato.
Observaciones: Es de vital importancia verificar la integridad de los datos.

Fuente: Elaboración propia.

En cada Tarea se deben tomar en cuenta las HU faltantes, las Pruebas de Aceptación no superadas en la iteración anterior y las tareas no terminadas en la iteración anterior.

## 2.5 Etapa de Diseño

En XP solo se diseñan aquellas historias de usuario que el cliente ha seleccionado para la iteración actual por dos motivos: por un lado, se considera que no es posible tener un diseño completo del sistema y sin errores desde el principio. El segundo motivo es que, dada la naturaleza cambiante del proyecto, el hacer un diseño muy extenso en las fases iniciales del proyecto para luego modificarlo, se considera un desperdicio de tiempo. Es importante resaltar que esta tarea es permanente durante la vida del proyecto partiendo de un diseño inicial que va siendo corregido y mejorado en el transcurso del proyecto. En este acápite se expone cómo fue diseñada la solución propuesta, explicándose el patrón de arquitectura de software utilizado para su diseño, se muestran ejemplos de prototipos de interfaces principales.

## 2.6 Estudio de Factibilidad.

Una de las cuestiones a tener en cuenta en el momento de desarrollar un software, es validar los beneficios y ventajas de éste con respecto a su costo, por lo que es necesaria una estimación del costo del software, además de un análisis de los beneficios tangibles e intangibles que reportará el proyecto.

### 2.6.1 Estimación de Costo.

En el momento de la planificación del proyecto, es casi imposible determinar el número de líneas de código que poseerá la aplicación, siendo el método

tradicional para calcular los costos COCOMO (Modelo Constructivo de Costos, por su acrónimo del inglés *CO*nstructive *CO*st *MO*del) poco idóneo en este caso, ya que está orientado a la magnitud del producto final, midiendo el tamaño del proyecto, en líneas de código principalmente. Dentro de sus principales inconvenientes podemos citar:

- Los resultados no son proporcionales a las tareas de gestión ya que no tiene en cuenta los recursos necesarios para realizarlas.
- Se puede desviar de la realidad si se indica mal el porcentaje de líneas de comentarios en el código fuente.
- Es un tanto subjetivo, puesto que está basado en estimaciones y parámetros que pueden ser "vistos" de distinta manera por distintos analistas que usen el método.
- Se miden los costes del producto, de acuerdo a su tamaño y otras características, pero no la productividad.

Sin embargo, ya que estamos utilizando XP como metodología para el desarrollo del software se tiene estimado el tiempo de desarrollo del mismo, aprovechando que la fórmula de *Bohem* comprende este parámetro (Costo = Cantidad de Hombres \* Salario Medio \* Tiempo de Desarrollo), es posible obtener una estimación del salario del autor. Haciendo los cálculos pertinentes tenemos:

La Tabla 22.2 muestra el salario mensual de cada miembro del equipo de trabajo en sus respectivos períodos, para un total 7 meses de duración del proyecto, considerando que un mes equivale a 22 días de trabajo. El costo inicial obtenido es de \$4630.00.

Tabla 22.2 Estimación de costo.

Miembros	Salario mensual	TD	Costo
----------	-----------------	----	-------



*Descripción de la solución propuesta*

Sergio Jorge Pagés Valdés	\$1450.00	1 mes	\$1450.00
MSc.Liz Pérez Martínez	\$879.00	1 mes	\$879.00
Celia Fuentes Gómez	\$350.00	7 mes	\$2450.00
<b>TOTAL</b>	<b>\$2 679.00</b>		<b>\$4 779.00</b>

Fuente: Elaboración propia.

*2.6.2 Costo real*

Para calcular el costo real del proyecto, se tuvieron en cuenta determinados gastos en que se incurrió durante su desarrollo. La Tabla 23.2 muestra una relación aproximada de otros aspectos que influyeron durante los 7 meses de trabajo, donde el gasto que ocasionaron es de \$630.00 aproximadamente.

Tabla 23.2 Gastos.

Aspecto	Gasto mensual	Gasto total
<b>Alimentación</b>	\$28.00	\$196.00
<b>Tarjetas Nauta</b>	\$50.00	\$350.00
<b>Tarifa eléctrica</b>	\$12.00	\$84.00
<b>TOTAL</b>	<b>\$90.00</b>	<b>\$630.00</b>

Fuente: Elaboración propia.

Al adicionarle estos gastos al costo inicial se estima que el costo monetario real aproximado del software es de \$5409.00.

*2.6.3 Análisis de costos y beneficios.*

Por consiguiente, el beneficio de invertir en este sistema es tanto para el trabajador, como para el cliente, ya que:

- Mejora la información en estado real de los procesos.
- Aumenta el nivel de contratación.
- Mejora el clima laboral, la motivación y la participación.

- Disminuye los costes de servicios de correo.
- Disminuye el costo de telefonía.

Con estos beneficios y la estimación de costos realizada anteriormente, se evidencia que la solución propuesta presenta una buena relación entre costo y beneficios.

### 2.7 Conclusiones del capítulo.

Una vez descrita la propuesta de solución al problema científico de esta investigación, se concluye que:

1. Se dispone de un equipo de trabajo capaz de solucionar el problema existente.
2. La planificación inicial es necesaria para tener una visión general del problema que se enfrenta y pensar desde el inicio en soluciones eficaces.
3. Las HU iniciales permitieron especificar los requisitos funcionales de la aplicación.
4. El cliente conformó las HU de acuerdo a sus necesidades.
5. La planificación de las iteraciones se realiza según las características de las HU, teniendo en cuenta la prioridad y el riesgo de desarrollo.
6. El plan de entrega de los módulos de la aplicación se correspondió con el final de cada iteración.
7. La estimación del costo del software se realizó a partir de la fórmula de Bohem, apreciándose un valor racional del mismo.
8. El beneficio de este sistema es tanto para el trabajador, como para el cliente.
9. El sistema aporta un beneficio mayor que su costo de producción.

## Capítulo 3. Resultados obtenidos y pruebas realizadas al software

### 3.1 Introducción

En el presente y último capítulo se exponen las etapas Desarrollo y Pruebas de la metodología XP, abordándose los cambios realizados en la planificación inicial del proyecto y ejemplificándose un conjunto de pruebas de evaluación de software efectuadas durante su desarrollo.

### 3.2 Estándares de programación.

Los estándares de programación mantienen el código legible para los miembros del equipo, facilitando cambios. Algunos de los utilizados, se describen a continuación:

- Se utiliza una arquitectura MVC, que consta de tres capas principales: Model, view y controller.
- Los controladores comienzan con mayúscula y le sigue la palabra reservada Controller
- La implementación de todas las clases comienza con mayúscula y se nombran según su contenido.
- Las variables comienzan como indica el manual de procedimiento de la organización de informáticos de la JDN.
- Los comentarios de las clases o métodos se encuentran en la parte superior de estos.

Los resultados de la implementación del software, usando las herramientas y plataformas tecnológicas establecidas en el Capítulo I y empleando la metodología de desarrollo ágil XP arrojan los datos mostrados en la siguiente tabla:

Tabla 24.3 Resultados de la implementación de SFC

Indicador	Valor
Número de HU implementadas.	10
Nivel de aceptación.	97,63%

Fuente: Elaboración propia.

### 3.3 Pruebas

La prueba de software es un elemento crítico para la garantía de la calidad del sistema y representa una revisión final de las especificaciones del diseño y de la codificación de acuerdo a los requerimientos del usuario.

Los elementos de prueba de software para realizar las validaciones y verificaciones del software existen distintos tipos de técnicas. Las primeras se aplican sobre el sistema en reposo, tanto a requisitos como a modelos de análisis, diseño y código y las segundas generan entradas al sistema con el objetivo de detectar fallos durante su ejecución. Estas técnicas son las que conocen como pruebas del software o testing

Características de una buena prueba:

1. Debe tener alta probabilidad de encontrar fallos.
2. Debe centrarse en lo que puede o no hacer el software.
3. Tiene que jugar con el tiempo y los recursos, así que no debe ser redundante. Debe ser lo más profunda posible.
4. Una prueba superficial no alcanza a descubrir lo que puede hacer que la aplicación no funcione.
5. Debe realizarse separadamente del resto de las pruebas.

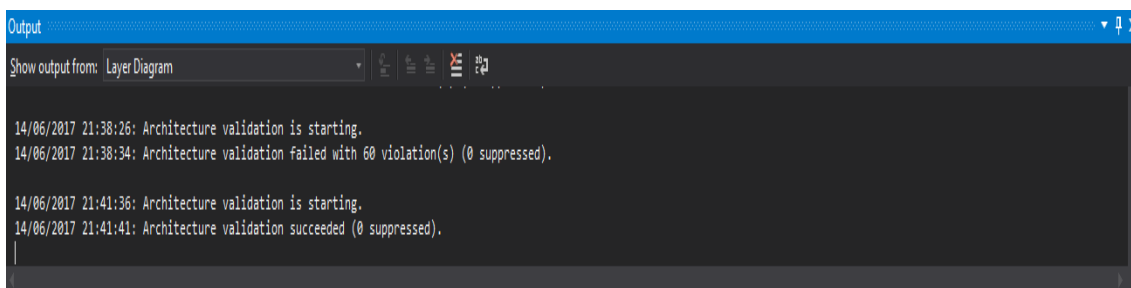
Existen dos tipos muy comunes de técnicas de prueba que son aplicadas al culminar la implementación de un producto de software: las de Caja Blanca o Estructurales y las de Caja Negra o Funcionales.

- La prueba de caja blanca del software realiza un minucioso examen de los detalles procedimentales. En ella se eligen y ejercitan una serie de caminos lógicos importantes y se examina el estado del programa en varios puntos, con el objetivo de comprobar si el estado real coincide con el esperado.
- La prueba de caja negra se lleva a cabo sobre la interfaz del software, pretendiendo demostrar que sus funciones son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto.

### 3.3.1 Pruebas de Caja Blanca

#### Pruebas de Arquitectura

Durante el desarrollo del proyecto se realizaron varias pruebas a la arquitectura para comprobar el correcto funcionamiento del software. Estas pruebas se realizaron utilizando los diagramas de MVC creados en el proyecto de modelado *Modeling Project*. En la Ilustración se visualizan las dos últimas validaciones de arquitectura realizada, en el primer caso la validación dio incorrecta debido a la falta de componentes y un segundo caso donde muestra la correcta validación de la arquitectura.



```
Output
Show output from: Layer Diagram

14/06/2017 21:38:26: Architecture validation is starting.
14/06/2017 21:38:34: Architecture validation failed with 60 violation(s) (0 suppressed).

14/06/2017 21:41:36: Architecture validation is starting.
14/06/2017 21:41:41: Architecture validation succeeded (0 suppressed).
```

Ilustración 4.3 Prueba de Arquitectura

Fuente: Elaboración propia.

#### Realización de Pruebas Unitarias.

Las pruebas unitarias fueron ejecutadas en paralelo con la codificación para lograr el correcto funcionamiento del código fuente y corregir errores en fases

tempranas. En la figura\_\_-- se muestran dos de las pruebas unitarias. Para la cual se utilizó NUnit y los ejemplos que se muestran, fueron realizados a la aplicación web. La primera de las pruebas con un resultado erróneo y la segunda al solucionar el error.

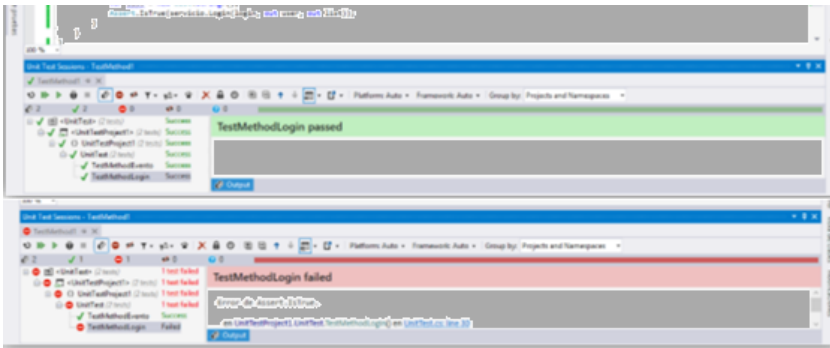


Ilustración 5.3 Prueba Unitaria.

*Fuente: Elaboración propia.*

### 3.3.2 pruebas de Caja Negra.

Las pruebas se llevan a cabo sobre la interfaz del software, y es completamente indiferente del comportamiento interno y la estructura del programa.

Los casos de prueba de la caja negra pretenden demostrar que:

- Las funciones del sistema son operativas.
- La entrada se acepta de forma adecuada.
- Se produce una salida correcta.
- La integridad de la información externa se mantiene.

Se derivan conjuntos de condiciones de entrada que ejercitan completamente todos los requerimientos funcionales del programa.

La prueba de la caja negra intenta encontrar errores de las siguientes categorías:

- Funciones incorrectas o ausentes.
- Errores de interfaz.

- Errores en estructuras de datos o en accesos a bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización y de terminación.

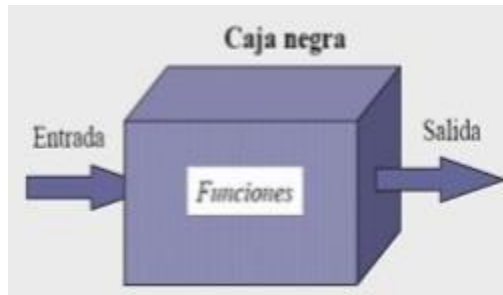


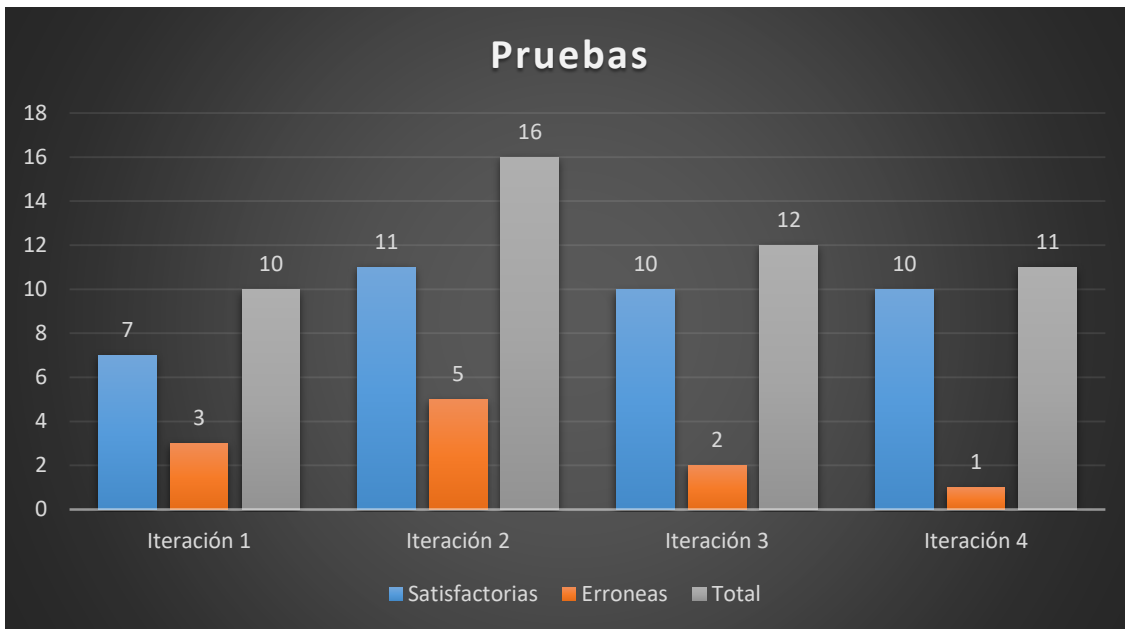
Ilustración 6.3 Prueba de Caja Negra

Fuente: Wikipedia,2016

### *Pruebas de Aceptación*

Al finalizar cada iteración del software se realizaron sus correspondientes pruebas, las cuales permitieron evaluar el cumplimiento o no de las funcionalidades requeridas por el cliente y marcar el camino a seguir en la próxima iteración. En la siguiente figura se observa una representación de las pruebas de aceptación realizadas al finalizar cada iteración.

Tabla 25.3 Prueba de Aceptación



Fuente: Elaboración propia.

A continuación, se especifican 5 de las 48 pruebas de aceptación elaboradas.

Tabla 26.3 Prueba de Aceptación 1

Pruebas de Aceptación	
Número de caso de prueba: 1	Número de Historia de Usuario: 2
Nombre del caso de prueba: Seguridad y autenticación de usuario	
Descripción: Se inserta el nombre de usuario y la contraseña para entrar al software. Se insertarán de forma incorrecta, dejando campos en blanco para verificar la validación, se tratará de entrar al software poniendo una ruta destino para verificar que entra solo cuando se autentifica correctamente. Luego se insertaran los datos correctos para comprobar esta funcionalidad.	
Condiciones de ejecución: El usuario tendrá acceso a las funcionalidades en las que tenga los permisos correspondientes	



*Resultados obtenidos y pruebas realizadas al software.*

<p>Entrada/Pasos de ejecución:</p> <p>Presionar el botón Entrar con los campos en blanco</p> <p>Fijar una ruta para entrar sin haberse autenticado</p> <p>Introducir los datos correctamente y presionar el botón Entrar</p>
<p>Resultado esperado: El software debe alertar al usuario cuando se insertan datos erróneos o no se ha autenticado. Cuando se insertan los datos correctamente el software debe entrar y mostrar las funcionalidades a las cuales el usuario tiene permiso</p>
<p>Evaluación de la prueba: Satisfactoria</p>

Fuente: Elaboración propia.

Tabla 27.3 Prueba de Aceptación 2

Pruebas de Aceptación	
Número de caso de prueba: 2	Número de Historia de Usuario: 4
Nombre del caso de prueba: Gestionar Contratos	
<p>Descripción: Se inserta el nombre y demás campos para crear un nuevo libro, se insertan de forma incorrecta dejando campos en blanco, luego se inserta de forma correcta para verificar que los datos sean almacenados.</p> <p>Se modifican los datos del libro, se modifican incorrectamente para verificar la validación, se modifican correctamente para verificar que los datos sean almacenados y cargados</p> <p>Se elimina el libro aceptando el mensaje de confirmación.</p>	
<p>Condiciones de ejecución: Los usuarios que pueden realizar esta acción son los que tienen el rol de abogado, técnico tramitador y director. Deben haberse insertado previamente el codificador de asunto a gestionar.</p>	

*Resultados obtenidos y pruebas realizadas al software.*

<p>Entrada/Pasos de ejecución:</p> <p>Dejar campos en blanco.</p> <p>Insertar datos de manera incorrecta.</p> <p>Insertar los datos correctamente.</p> <p>Modificar los datos dejando campos en blanco.</p> <p>Modificar los datos de manera incorrecta.</p> <p>Modificar los datos de forma correcta.</p>
<p>Resultado esperado: El software debe alertar al usuario cuando se insertan datos erróneos. Cuando se insertan los datos correctamente el software debe almacenarlos en la base de datos y mostrarlos</p>
<p>Evaluación de la prueba: Satisfactoria</p>

Fuente: Elaboración propia.

*Tabla 28.3 Prueba de Aceptación 3*

Pruebas de Aceptación	
Número de caso de prueba: 3	Número de Historia de Usuario: 6
Nombre del caso de prueba: Gestionar Fase de los contratos	
<p>Descripción: Se insertan los datos del contrato para ver su estado, se insertan de forma incorrecta dejando campos en blanco, luego se inserta de forma correcta para verificar que los datos sean almacenados, luego se seleccionan el contrato correspondiente.</p> <p>Se modifican los datos del contrato, se modifican incorrectamente para verificar la validación, se modifican correctamente para verificar que los datos sean almacenados y cargados</p> <p>Se elimina el estado del contrato el mensaje de confirmación.</p>	

*Resultados obtenidos y pruebas realizadas al software.*

Condiciones de ejecución: Los usuarios que pueden realizar esta acción son los que tienen el rol Técnico Tramitador.
<p>Entrada/Pasos de ejecución:</p> <p>Dejar campos en blanco.</p> <p>Insertar datos de manera incorrecta.</p> <p>Insertar los datos correctamente.</p> <p>Modificar los datos dejando campos en blanco.</p> <p>Modificar datos de manera incorrecta.</p> <p>Modificar los datos de forma correcta.</p>
Resultado esperado: El software debe alertar al usuario cuando se insertan datos erróneos. Cuando se insertan los datos correctamente el software debe almacenarlos en la base de datos y mostrarlos
Evaluación de la prueba: Satisfactoria

Fuente: Elaboración propia.

Tabla 29.3 Prueba de Aceptación 4

Pruebas de Aceptación	
Número de caso de prueba: 4	Número de Historia de Usuario: 9
Nombre del caso de prueba: Gestionar Letrados	
<p>Descripción: Se insertan los datos del letrado para crear uno nuevo, se insertan de forma incorrecta dejando el campo en blanco, luego se inserta de forma correcta para verificar que los datos sean almacenados.</p> <p>Se modifican los datos del letrado, se modifican incorrectamente para verificar la validación, se modifican correctamente para verificar que los datos sean almacenados y cargados</p>	

*Resultados obtenidos y pruebas realizadas al software.*

Se elimina el letrado aceptando el mensaje de confirmación
Condiciones de ejecución: Los usuarios que pueden realizar esta acción son los que tienen el rol Director y Administrador, deben estar creados el municipio y el bufete al cual pertenece.
Entrada/Pasos de ejecución:  Dejar campos en blanco.  Insertar datos de manera incorrecta.  Insertar los datos correctamente.  Modificar los datos dejando campos en blanco.  Modificar datos de manera incorrecta.  Modificar los datos de forma correcta.
Resultado esperado: El software debe alertar al usuario cuando se insertan datos erróneos. Cuando se insertan los datos correctamente el software debe almacenarlos en la base de datos y mostrarlos.
Evaluación de la prueba: Satisfactoria

Fuente: Elaboración propia.

Tabla 30.3 Prueba de Aceptación 5

Pruebas de Aceptación	
Número de caso de prueba: 5	Número de Historia de Usuario: 11
Nombre del caso de prueba: Gestionar Infracciones.	
Descripción: Se insertan los datos de la Infracción para crear una nueva, se insertan de forma incorrecta dejando el campo en blanco, luego se inserta de forma correcta para verificar que los datos sean almacenados.  Se modifican los datos de la Nota, se modifican incorrectamente para	

*Resultados obtenidos y pruebas realizadas al software.*

<p>verificar la validación, se modifican correctamente para verificar que los datos sean almacenados y cargados</p> <p>Se elimina la Infracción aceptando el mensaje de confirmación.</p>
<p>Condiciones de ejecución: Los usuarios que pueden realizar esta acción son los que tienen el rol Tribunal, deben haberse insertado el usuario, el abogado, el bufete, el municipio y el número del contrato</p>
<p>Entrada/Pasos de ejecución:</p> <p>Dejar campos en blanco.</p> <p>Insertar datos de manera incorrecta.</p> <p>Insertar los datos correctamente.</p> <p>Modificar los datos dejando campos en blanco.</p> <p>Modificar datos de manera incorrecta.</p> <p>Modificar los datos de forma correcta.</p>
<p>Resultado esperado: El software debe alertar al usuario cuando se insertan datos erróneos. Cuando se insertan los datos correctamente el software debe almacenarlos en la base de datos y mostrarlos</p>
<p>Evaluación de la prueba: Satisfactoria</p>

Fuente: Elaboración propia.

*Prueba de estrés*

La prueba de estrés se utiliza normalmente para romper la aplicación. Se va doblando el número de usuarios que se agregan a la aplicación y se ejecuta una prueba de carga hasta que se rompe. Este tipo de prueba se realiza para determinar la solidez de la aplicación en los momentos de carga extrema y ayuda a los administradores para determinar si la aplicación rendirá lo suficiente en caso de que la carga real supere a la carga esperada.

*Resultados obtenidos y pruebas realizadas al software.*

Primeramente, se utilizó el software *Webserver Stress Tool 8-Enterprise Edition(Freeware)* y se realizó la prueba con 200 y 500 usuario como muestra la siguiente ilustración con sus resultados:

Logfiles		Results per User (Complete Test)			Results per URL (Complete Test)	
URL No.	Name	Clicks	Errors	Errors [%]	Time Spent [ms]	Avg. Click Time [ms]
1		2.822	0	0,00	27.590.017	9.777

**Test Setup**

Test Type: RAMP (run test for 5 minutes)  
 User Simulation: ramp test with up to 200 simultaneous users - 5 seconds between clicks (Random)  
 Logging Period: Log every 5 seconds

**URLs**

Logfiles		Results per User (Complete Test)			Results per URL (Complete Test)	
URL No.	Name	Clicks	Errors	Errors [%]	Time Spent [ms]	Avg. Click Time [ms]
1		4	0	0,00	605	151

**Project and Scenario Comments, Operator**

**Test Setup**

Test Type: CLICKS (run test until 1 clicks per user)  
 User Simulation: 5 simultaneous users - 500 seconds between clicks  
 Logging Period: Log every 10 seconds

Las pruebas realizadas fueron satisfactorias el sistema no obtuvo sobrecarga y se llegó a la conclusión que el sistema soporta más de la carga esperada.

### 3.4 Validación de la solución propuesta

El sistema se desarrolló con el objetivo de facilitar la manera de obtener y gestionar la información referente a los servicios que brinda la dirección provincial de Bufetes Colectivos de Matanzas en cada uno de sus municipios. Para demostrar el cumplimiento de este objetivo, en este acápite se hace un análisis de los resultados obtenidos con la implementación de la solución propuesta, mediante una comparación con respecto a la situación inicial.

La siguiente tabla muestra en detalle el nivel de aceptación de la solución propuesta basado en las HU implementadas, las cuales obtuvieron el puntaje

*Resultados obtenidos y pruebas realizadas al software.*

correspondiente por el cliente luego de realizar las pruebas de aceptación, al finalizar cada iteración de software.

La Tabla muestra el tiempo de obtención y procesamiento de información con el uso de la solución propuesta y sin la utilización de la misma, es decir, como se hacía antes de esta investigación.

Tabla 31.3 Tiempo de obtención y procesamiento de información

<b>Descripción</b>	<b>Sin la solución propuesta (Valor aproximado)</b>		<b>Con la solución propuesta</b>
<b>Gestionar contratos de todos los municipio</b>	De 20 minutos a 48 horas		Menos de 1 minuto
<b>Consultar situación de los contratos</b>	De 25 minutos a 30 días		Menos de 1 minuto
<b>Consultar infracciones</b>	45 minutos (se recopilab información de municipios)	a todos los	Menos de 1 minuto
<b>Consultar fase de los contratos</b>	45 minutos (se recopilab información de municipios)	a todos los	Menos de 1 minutos
<b>Reporte de situación de los contratos</b>	5 minutos (se recopilab información de municipios)	a todos los	Menos de 1 minutos por reporte

*Resultados obtenidos y pruebas realizadas al software.*

<b>Reporte Cumplimiento Plan Mensual de contratación por letrados</b>	15 minutos (se información de municipios)	recopilab a todos los	Menos de 1 minutos por reporte
<b>Generar reportes informativos</b>	30 minutos (se información de municipios)	recopilab a todos los	Menos de 1 minutos por reporte

Fuente: Elaboración propia.

Puede constatarse que la búsqueda de información mejoró considerablemente, en especial los Reporte de contratación, así como la fase en que se encuentran los contratos, ya que debían reportar una inmensa cantidad de datos de todos los municipios de la provincia. Con la ayuda de la solución propuesta, los reportes generados son más confiables y precisos, además disminuyen el tiempo de su elaboración, solamente los ejecutivos de DPNC (Provincial) con solo acceder al sistema, tienen acceso a visualizar los datos de los procesos que se gestionan ya no solo a nivel municipal, sino que se visualiza lo que ocurre en cada municipio sin necesidad de las continuas salvadas de los sistemas para su posterior integración.

De lo expuesto anteriormente se puede afirmar que la utilización de la solución propuesta influye de manera positiva en el almacenamiento y procesamiento de los datos referentes al servicio de abastecimiento de agua y operaciones de mantenimiento en la DPBC Matanzas.

### 3.5 Conclusiones del capítulo.

Una vez construida la propuesta de solución al problema científico de esta investigación a partir de la metodología XP, y analizado los resultados de la misma a partir de las pruebas funcionales, se concluye que:



---

*Resultados obtenidos y pruebas realizadas al software.*

1. La planificación fue acertada, pues permitió el desarrollo del sistema según el cronograma y cumpliendo con los objetivos del cliente al iniciar esta investigación.
2. El uso de pruebas funcionales durante el desarrollo del sistema permitió tanto al cliente como al desarrollador encontrar defectos y errores en el sistema y su rápida corrección.
3. A partir del diseño de la propuesta y con el uso de las tecnologías adecuadas para su implementación se obtuvo un sistema que cumple a cabalidad con las expectativas del cliente.

## Conclusiones generales

Como resultado de esta investigación quedaron satisfechos los objetivos trazados arribando a las siguientes conclusiones:

1. El estudio realizado sobre los antecedentes, el estado actual de la temática, la bibliografía y documentos relacionados con el objeto de estudio, permitió contar con los elementos necesarios para dar solución a la problemática planteada.
2. Los sistemas automatizados encontrados, vinculados al tema no le dan solución al problema planteado por lo que no es factible su utilización.
3. Se utilizaron las herramientas de software más factibles para la construcción de la solución.
4. Se implementó el sistema web para la automatización de los procesos de contratación requeridos en la organización, cumpliendo con el cronograma de desarrollo planteado al cliente.
5. Se realizó la estimación de costo de implementación del sistema y el estudio de factibilidad, arrojando como resultado la factibilidad de la realización del sistema informático.
6. La realización de las pruebas funcionales a este sistema informático permitió detectar errores en el sistema y la rápida corrección de los mismos.
7. La implementación del sistema y la aplicación de las pruebas de validación con resultados satisfactorios demostraron que el software elaborado cumple con los requerimientos especificados constatándose, mediante avales, su aporte práctico a la organización.

Para finalizar, tras haber verificado los objetivos específicos del trabajo de diploma se puede concluir que el objetivo general se cumplió al obtener la aplicación web SFC. Facilitando de esta forma, el control del nivel de contratación y gestión de los procesos, comprobándose su validez mediante un análisis de resultados realizado con el apoyo del cliente, dándole un nivel de aceptación de un 97,6 %y a través

del criterio de los expertos, valorándolo de forma adecuada, con se afirma que la hipótesis planteada en los inicios de la investigación es verdadera.

## Recomendaciones

Desde el punto de vista del alcance del presente trabajo y teniendo en cuenta el momento de desarrollo del mismo, se proponen las siguientes recomendaciones:

1. Implementar el resto de los módulos que integran el sistema.
2. Ampliar la aplicación, incorporándole aspectos para facilitar el trabajo del cliente.
3. Generar reportes en forma de gráficas para una mejor comprensión de los mismos.
4. Incorporar una ayuda dinámica al sistema.
5. Extender la herramienta web a toda la nación para facilitar la gestión de los procesos en el resto del país.

## Bibliografía

\_ AJAX un nuevo acercamiento a Aplicaciones Web [En línea] // AJAX un nuevo acercamiento a Aplicaciones Web. - 2014. - 3 de Diciembre de 2014. - <http://www.uberbin.net/archivos/internet/ajax-un-nuevo-acercamiento-a-aplicaciones-web.php>.

\_ Especificación JSON [En línea] // Especificación JSON. - \_, 2012. - 10 de Diciembre de 2014. - <http://www.json.org/json-es.html>.

\_ Patrón de arquitectura Modelo Vista Controlador (MVC) [En línea] // Patrón de arquitectura Modelo Vista Controlador (MVC). - \_, 2010. - 12 de Diciembre de 2014. - <http://www.inf.uc3m.es/~a0080802/RAI/mvc.html>.

**Apache** Javascript a fondo [En línea] // Javascript a fondo. - Apache, 2010. - 9 de Diciembre de 2014. - <http://www.desarrolloweb.com/javascript>.

**Autores Colectivo de** Normas y procedimientos de la ONBC [En línea] // Normas y procedimientos de la ONBC. - febrero de 2018.

**Beck Kent** Extreme Programming Explained: Embrace Change [Libro]. - \_ : \_, 1999.

**Cochran David** Twitter Bootstrap Web Development [Libro]. - \_ : Packt Publishing, 2012. - Vol. 1st Edition.

**Consortio W3C** HTML5: ¿qué es HTML5? [En línea] // HTML5: ¿qué es HTML5?. - Consortio W3C, 2007. - 9 de Diciembre de 2014. - <http://desarrolloweb.dlsi.ua.es/html5-que-es-html5>.

**Corporation, Microsoft** ADO.Net Entity Framework [En línea]. - Microsoft Software Developers Network, 2013. - 1 de Diciembre de 2015. - <http://msdn.microsoft.com/en-us/data/ef.aspx..>

**Dominquez Rolo** InterCAS [Patente]. - 2014.

**Microsoft Corporation** Asp.Net Mvc [En línea]. - Corporation, Microsoft, 2014. - 2014 de Diciembre de 2. - <http://www.asp.net/mvc>.

**Microsoft Corporation** C# Web Programación [En línea] // C# Web Programación. - Microsoft Corporation, 2012. - 8 de Diciembre de 2014. - <http://www.webprogramacion.com/csharp.aspx>.

**Microsoft Corporation** Página Web de Microsoft Sql Server [En línea] // Página Web de Microsoft Sql Server. - Microsoft Corporation, 2012. - 2014 de Diciembre de 4. - <http://www.microsoft.com/spain/sql/default.aspx>.

**Microsoft Corporation** Sitio Oficial del Visual Studio [En línea] // Sitio Oficial del Visual Studio. - Microsoft Corporation, 2013. - 11 de Diciembre de 2014. - <http://www.microsoft.com/spain/visualstudio>.

**Otto Mark y Thornton Jacob** Tweeter Bootstrap [En línea]. - 2014. - 7 de mayo de 2014. - <http://twitter.github.com/bootstrap>.

**Resig John** Sitio oficial del jQuery [En línea] // Sitio oficial del jQuery. - jQuery Development Team, 2014. - 5 de Diciembre de 2014. - <http://jquery.com/>.

**Weiler** <https://Claim.SA.cu> [En línea] // <https://Claim.SA.cu>. - 2016.

Wikipedia [En línea] // Wikipedia. - 2016.

## Anexo 1. Comparación entre las metodologías ágiles y tradicionales

Tabla 32 Anexo 1

<b>Metodología Ágil</b>	<b>Metodología Tradicional</b>
Pocos artefactos.	Más artefactos.
Pocos roles, más genéricos y flexibles.	Más roles, más específicos.
No existe un contrato tradicional, debe ser bastante flexible.	Existe un contrato prefijado.
El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Grupos pequeños (< 10 integrantes) y trabajando en el mismo sitio.	Aplicables a proyectos de cualquier tamaño, pero suelen ser especialmente efectivas/usadas en proyectos grandes y con equipos posiblemente dispersos.
La arquitectura se va definiendo y mejorando a lo largo del proyecto.	Se promueve que la arquitectura se defina tempranamente en el proyecto.
Se hace énfasis en los aspectos humanos: el individuo y el trabajo en equipo.	Se hace énfasis en la definición del proceso: roles, actividades y artefactos.
Basadas en heurísticas provenientes de prácticas de producción de código.	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.
Especialmente preparadas para cambios durante el proyecto.	Cierta resistencia a los cambios.

Fuente: (Rodríguez Ramírez, 2006)

## Anexo 2 Imágenes del sistema



Registrarse

Usuario

Contraseña

Entrar

 Sistema Judicial Cubano

©2018 Asesoría Legal de forma personalizada. Todos los derechos reservados

Ilustración 7 Autenticarse

Fuente: Elaboración propia.

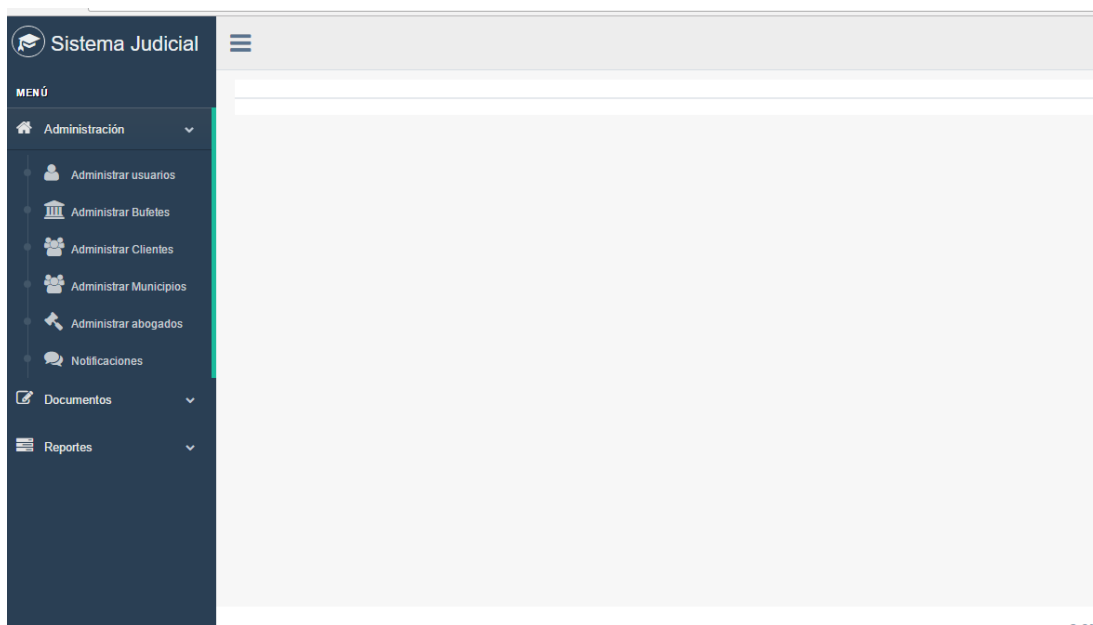


Ilustración 8 Página principal de la administración

Fuente: Elaboración propia.



## Acrónimos

MVC	Modelo Vista Controlador
HTML	<i>HyperText Markup Language</i> (Lenguaje de Marcado de Hipertexto)
ORACLE	<i>Structured Query Language</i> (Lenguaje de Consulta Estructurado)
CSS	<i>Cascading Style Sheets</i> (Hoja de Estilo en Cascada)
XP	<i>Extreme Programming</i> (XP)
HU	Historia de Usuario
BD	Base de Datos
EF	<i>Entity Framework</i>
DDL	<i>Data Definition Language</i> (Lenguaje de Definición de Datos)
URLs	<i>Uniform Resource Locator</i> (Localizador de Recursos Uniforme)
GUI	<i>Graphical User Interface</i> (Interfaz Gráfica de Usuario)
DLLs	<i>Dynamic Link Library</i> (Biblioteca de Enlace Dinámico)
IDE	<i>Integrated Development Environment</i> (Entorno de Desarrollo Integrado)
UML	<i>Unified Modeling Language</i> (Lenguaje Unificado de Modelado)

CRUD      *Create Read Update Delete* (Crear Leer Actualizar  
Eliminar)

DPBC      *Dirección Provincial de Bufetes Colectivos Matanzas*