



Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación

***Aplicación web de apoyo para
abogados litigantes en Venezuela***

Trabajo Especial de Grado
presentado ante la Ilustre
Universidad Central de Venezuela
Por el bachiller:

ANNGY RAKEL DORDELLY RUIZ
C.I.: 13.535.033

para optar al título de Licenciado de Computación

Tutores: Prof. Zenaida Castillo
Prof. Carlos Acosta

Caracas, octubre 2014

Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación



ACTA DEL VEREDICTO

Quienes suscriben, miembros del Jurado designado por el Consejo de la Escuela de Computación para examinar el Trabajo Especial de Grado, presentado por el bachiller Anngy Rakel Dordelly Ruiz, C.I.: 13.535.033 con el título “**Aplicación web de apoyo para abogados litigantes en Venezuela**”, a los fines de cumplir con el requisito legal para optar al título de Licenciado en Computación, dejan constancia de lo siguiente:

Leído el trabajo por cada uno de los miembros del Jurado, se fijó el día 30 de Octubre de 2014, a las 07:30 am, para que su autor lo defendiera en forma pública, en Caracas, lo cual este realizó mediante una exposición oral de su contenido, y luego respondieron satisfactoriamente a las preguntas que les fueron formuladas por el Jurado, todo ello conforme a lo dispuesto en la Ley de Universidad y demás normativas vigentes de la Universidad Central de Venezuela. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlo.

En fe de lo cual se levanta la presente acta, en Caracas el día 30 de Octubre de 2014, dejándose también constancia de que actuó como Coordinador del Jurado el Profesora Zenaida Castillo.

Prof. Zenaida Castillo
(Tutor Firmante)

Prof. Carlos Acosta
(Tutor)

Prof. Rhadames Carmona
(Jurado Principal)

Prof. Antonio Silva
(Jurado Principal)

Dedicatoria

Dedico este trabajo especial de grado...

A Dios, por acompañarme en cada segundo de mi vida y brindarme oportunidades maravillosas. En particular esta, la finalización de esta carrera.

A mi papa, mi mamá y mi abuelita, por inculcarme buenos valores, por ser los mejores padres y abuela del mundo.

A Thania, por enseñarme, animarme e impulsarme desde pequeña en mis estudios.

A Atenas, porque sin su empeño, esto no sería realidad.

¡Los amo!

Anngyta

Agradecimiento

Agradezco...

A Dios por sobre todas las cosas. Por escucharme y permitirme finalizar mis estudios de pregrado con éxito.

A la majestuosa UCV, a la Facultad de Ciencias, a la Escuela de Computación y a todos los profesores, porque a ella y a ellos les debo la profesional que soy hoy.

A Zenaida y a Carlos, por dejarme aprender de sus invaluable conocimientos, guiarme a lo largo de la carrera y en esta última etapa.

A mi mamá, mi papá, mi abuela, Thania y Atenas, por insistir y no dejarme caer.

A Max, por enseñarme a enamorarme de Python.

Por último, pero no menos importante, agradezco a los abogados que colaboraron en el desarrollo del software objeto de este trabajo.

¡A todos mil gracias!

Anngyta

Resumen

Título: Aplicación web de apoyo para abogados litigantes en Venezuela

Autor:

Anngy Rakel Dordelly Ruiz

Tutores:

Prof. Zenaida Castillo

Prof. Carlos Acosta

El presente trabajo especial de grado tiene como objetivo general, el diseño y desarrollo de una aplicación web de apoyo para abogados litigantes de Venezuela, justificado principalmente por dos elementos: la identificación de la gestión que un abogado litigante debe coordinar y ejecutar –derivado del análisis de un juicio ordinario simple en Venezuela-, y la carencia de herramientas tecnológicas especializadas de carácter gratuito, hechas en el país, que apoyen dicha gestión. Para determinar esa carencia se realizó una revisión de las herramientas disponibles en el mercado, así como sus características y limitaciones.

En cuanto a las herramientas tecnológicas consideradas y utilizadas en el desarrollo de la aplicación web, destacan: Python, como lenguaje de programación seleccionado, Django, como framework backend, JavaScript, jQuery y Bootstrap como frameworks frontend y Heroku como “Plataforma como Servicio”.

Para el desarrollo de la aplicación web se siguió la metodología ágil AgilUs, propuesta por la Prof. Eleonora Acosta de la Escuela de Computación de la UCV, basada en el concepto de usabilidad.

El resultado de las pruebas de aceptación determinó una clara intención de los abogados encuestados a usar la aplicación para apoyar su gestión, con lo que se concluye que el desarrollo de la aplicación cumplió con los objetivos planteados.

Palabras claves: aplicación web, gestión jurídica, abogado litigante.

Tabla de contenido

INTRODUCCIÓN	10
CAPÍTULO 1. PLANTEAMIENTO DEL PROBLEMA	12
1.1 OBJETIVO GENERAL	12
1.2 OBJETIVOS ESPECÍFICOS	13
1.3 JUSTIFICACIÓN	13
1.4 ALCANCE Y DELIMITACIÓN	14
CAPÍTULO 2. GESTIÓN LEGAL DE UN ABOGADO LITIGANTE EN VENEZUELA	15
2.1 PROCEDIMIENTOS DE UN JUICIO ORDINARIO	15
2.2 REQUISITOS DE LA DEMANDA	17
2.3 ESTIMACIÓN DE LA DEMANDA	17
2.4 ADMISIÓN DE LA DEMANDA	17
2.5 CITACIÓN DEL DEMANDADO	18
2.6 CONTESTACIÓN DE LA DEMANDA	18
2.7 ETAPA PROBATORIA	18
2.8 EVACUACIÓN DE PRUEBAS	18
2.9 INFORMES	19
2.10 SENTENCIA DE LA DEMANDA	19
2.11 ACTIVIDADES REALIZADAS POR UN ABOGADO LITIGANTE	19
CAPÍTULO 3. SISTEMAS DE APOYO A LA GESTIÓN JURÍDICA	23
3.1 ANTECEDENTES	23
3.2 HERRAMIENTAS DISPONIBLES EN EL MERCADO	23
CAPÍTULO 4. METODOLOGÍA Y DESARROLLO DE LA APLICACIÓN	25
4.1 METODOLOGÍAS ÁGILES	25
4.2 METODOLOGÍA AGILUS	27
4.3 MEJORES PRÁCTICAS DE AGILUS	28
4.4 CICLO DE VIDA	29
4.5 APLICACIÓN DE LA METODOLOGÍA AGILUS	30
4.6 ETAPA 1: REQUISITOS	30
4.7 ETAPA 2: ANÁLISIS	38
4.8 ETAPA 3: PROTOTIPAJE	58
4.9 ETAPA 4: ENTREGA	66
CAPÍTULO 5. HERRAMIENTAS TECNOLÓGICAS	74
5.1 APLICACIONES WEB	74
5.2 ARQUITECTURA CLIENTE-SERVIDOR	75

5.3	PATRÓN DE DISEÑO DE SOFTWARE MODELO VISTA CONTROLADOR	76
5.4	BASE DE DATOS	80
5.5	POSTGRESQL (BD).....	81
5.6	LENGUAJE DE MARCADO DE HIPERTEXTO (HMTL).....	83
5.7	LENGUAJES DE PROGRAMACIÓN WEB	84
5.8	PYTHON	85
5.9	DJANGO	86
5.10	JAVASCRIPT	88
5.11	BOOTSTRAP	89
5.12	HEROKU	89
CONCLUSIONES Y RECOMENDACIONES.....		91
REFERENCIAS BIBLIOGRÁFICAS		94
ANEXOS		101

Índice de Figuras

FIGURA 1: DIAGRAMA DE UN JUICIO ORDINARIO	16
FIGURA 2: EL MÉTODO AGILUS: ETAPA, ACTIVIDADES Y ARTEFACTOS [79]	30
FIGURA 3: TARIFAS DE TANTUM SISTEMA DE GESTIÓN JURÍDICA [17]	31
FIGURA 4: ESTATOS DE UN SERVIDOR LEX-DOCTOR v9.1 [15].....	32
FIGURA 5: MÓDULO CLIENTES DE MNPROGRAM ABOGADOS [20]	33
FIGURA 6: OPCIÓN PUBLICACIÓN DE ESTATOS DE ABOGAFIN DE NETFINCAS [13].....	33
FIGURA 7: DIAGRAMA CASO DE USO NIVEL 0	39
FIGURA 8: DIAGRAMA CASO DE USO NIVEL 1	40
FIGURA 9: DIAGRAMA DE CASO DE USO EXPEDIENTES, NIVEL 2	42
FIGURA 10: DIAGRAMA CASO DE USO CLIENTES, NIVEL 2.....	45
FIGURA 11: DIAGRAMA CASO DE USO ORGANISMOS, NIVEL 2	46
FIGURA 12: DIAGRAMA CASO DE USO AGENDA, NIVEL 2	47
FIGURA 13: DIAGRAMA CASO DE USO PLANTILLAS, NIVEL 2	49
FIGURA 14: MODELO OBJETO DEL DOMINIO DE LA GESTIÓN DE UN ABOGADO LITIGANTE	50
FIGURA 15: PROTOTIPO EN PAPEL DEL INICIO DE SESIÓN DEL SISTEMA.....	52
FIGURA 16: PROTOTIPO EN PAPEL DE PANTALLA DE INICIO DE LA APLICACIÓN.....	52
FIGURA 17: PROTOTIPO EN PAPEL DE PANTALLA DEL MÓDULO DE CLIENTES	53
FIGURA 18: PROTOTIPO EN PAPEL DE PANTALLA DE INGRESO DE INFORMACIÓN DE CLIENTE NATURAL	53
FIGURA 19: PROTOTIPO EN PAPEL DE PANTALLA DE INGRESO DE INFORMACIÓN DE CLIENTE JURÍDICO.....	54
FIGURA 20: BOTONES DE LA APLICACIÓN WEB	57
FIGURA 21: ÍCONOS DE OPERACIONES DE LA APLICACIÓN WEB	57
FIGURA 22: MENÚ DE LA APLICACIÓN WEB.....	58
FIGURA 23: VENTANA FLOTANTE DE LA APLICACIÓN WEB	58
FIGURA 24: PANTALLA DE PROTOTIPO EJECUTABLE DE LA APLICACIÓN.....	59
FIGURA 25: MÓDULO DE CLIENTES DE LA APLICACIÓN WEB DESARROLLADA.....	66
FIGURA 26: MÓDULO DE EXPEDIENTES DE LA APLICACIÓN WEB DESARROLLADA.....	67
FIGURA 27: MÓDULO DE AGENDA DE LA APLICACIÓN WEB DESARROLLADA	67
FIGURA 28: MÓDULO DE ORGANISMOS DE LA APLICACIÓN WEB DESARROLLADA.....	68
FIGURA 29: MÓDULO DE PLANTILLAS DE LA APLICACIÓN WEB DESARROLLADA.....	68
FIGURA 30: GENERADOR DE PLANTILLAS DE LA APLICACIÓN DESARROLLADA	69
FIGURA 31: EJEMPLO DE UNA PLANTILLA GENERADA POR LA APLICACIÓN DESARROLLADA	69
FIGURA 32. GRÁFICO DE RESULTADOS DE LA ENCUESTA DE ACEPTACIÓN DE LA APLICACIÓN WEB	72
FIGURA 33: ARQUITECTURA CLIENTE-SERVIDOR	76
FIGURA 34: MODELO VISTA CONTROLADOR	78
FIGURA 35: LISTA DE POSICIONES DE MAYOR DE USO DE LOS SMBD [42].....	82
FIGURA 36: SISTEMA DE ARCHIVOS DEL PROYECTO DE LA APLICACIÓN EN DJANGO	101
FIGURA 37: ARCHIVO MODELS.PY DE LA APLICACIÓN CLIENTES EN DJANGO	101

Índice de Tablas

TABLA 1: TABLA COMPARATIVA DE LOS SISTEMAS DE ABOGADOS EN EL MERCADO	34
TABLA 2: ENCUESTA PARA LEVANTAR INFORMACIÓN DE FUNCIONALIDADES DE LA APLICACIÓN WEB.....	36
TABLA 3: CASO DE USO NIVEL 0 – APLICACIÓN WEB DE APOYO PARA ABOGADOS LITIGANTES EN VENEZUELA	39
TABLA 4: CASO DE USO NIVEL 1 – EXPEDIENTES	40
TABLA 5: CASO DE USO NIVEL 1 - CLIENTES	41
TABLA 6: CASO DE USO NIVEL 1 - AGENDA	41
TABLA 7: CASO DE USO NIVEL 1 - ORGANISMOS	41
TABLA 8: CASO DE USO NIVEL 1 - PLANTILLAS.....	42
TABLA 9: CASO DE USO NIVEL 2 – CREAR EXPEDIENTE.....	42
TABLA 10: CASO DE USO NIVEL 2 – EDITAR EXPEDIENTE	43
TABLA 11: CASO DE USO 2 – SUBIR DOCUMENTO	43
TABLA 12: CASO DE USO NIVEL 2 – EDITAR DOCUMENTO	43
TABLA 13: CASO DE USO NIVEL 2 – CREAR ACTUACIÓN	44
TABLA 14: CASO DE USO NIVEL 2 – EDITAR ACTUACIÓN	44
TABLA 15: CASO DE USO NIVEL 2 – CREAR CLIENTE	45
TABLA 16: CASO DE USO NIVEL 2 – EDITAR CLIENTE.....	46
TABLA 17: CASO DE USO NIVEL 2 – CREAR ORGANISMO	46
TABLA 18: CASO DE USO NIVEL 2 – EDITAR ORGANISMO	47
TABLA 19: CASO DE USO NIVEL 2 – CREAR EVENTO	48
TABLA 20: CASO DE USO NIVEL 2 – EDITAR EVENTO	48
TABLA 21: CASO DE USO NIVEL 2 – MARCAR DÍA NO LABORABLE	48
TABLA 22: CASO DE USO NIVEL 2 – MARCAR DÍA NO DESPACHO.....	49
TABLA 23: CASO DE USO NIVEL 2 – CREAR PLANTILLA.....	49
TABLA 24. CASO DE USO NIVEL 2 – EDITAR PLANTILLA	50
TABLA 25: PATRÓN DE INTERACCIÓN – CREACIÓN DE EXPEDIENTE	55
TABLA 26: PATRÓN DE INTERACCIÓN – SUBIR DOCUMENTO	56
TABLA 27: PATRÓN DE INTERACCIÓN – AGREGAR ACTUACIÓN	56
TABLA 28: PATRÓN DE INTERACCIÓN – LISTADO DE CLIENTES	57
TABLA 29: LISTA RESULTADO DE EVALUACIÓN HEURÍSTICA	63
TABLA 30: LISTA DE PROBLEMAS Y SOLUCIONES GENERADAS EN LA PRIMERA PRESENTACIÓN.....	65
TABLA 31: ENCUESTA APLICADA	71
TABLA 32: RESULTADOS DE LA ENCUESTA APLICADA.....	71
TABLA 33: TABULACIÓN DE RESULTADOS DE LA ENCUESTA DE ACEPTACIÓN DE LA APLICACIÓN WEB	71
TABLA 34: FRAMEWORKS QUE SOPORTAN EL PATRÓN MVC [38]	80
TABLA 35: DIFERENCIAS ENTRE LA PROGRAMACIÓN DEL LADO DEL CLIENTE Y DEL SERVIDOR	85
TABLA 36: LISTA DE APLICACIONES USADAS DE DJANGO	88

Introducción

El trabajo especial de grado descrito en este documento, tiene como finalidad el diseño y desarrollo de una aplicación web de apoyo a abogados que ejercen la profesión en Venezuela. Dicha aplicación, tiene como propósito primordial brindar apoyo a la gestión de los abogados que trabajan de manera independiente, la cual, debido a una serie factores puede resultar inmanejable y/o afectar el desempeño del abogado, poniendo en riesgo el éxito en los casos que llevan.

Un estudio de mercado realizado arrojó que no existen herramientas tecnológicas de carácter gratuito que apoyen de forma sistemática la gestión del abogado en Venezuela.

La aplicación fue desarrollada en Python usando el framework backend de Django que se apega al patrón de diseño de desarrollo de software MVC (Modelo Vista Controlador), el cual propone la construcción de los componentes del software de manera separada.

Como metodología de desarrollo de software se seleccionó Agile, la cual se basa, Esta metodología basada en la usabilidad y propone a lo largo de las etapas del ciclo de vida de desarrollo, métodos, técnicas y actividades, que producen artefactos utilizados en desarrollo incremental e iterativo para asegurar que el software desarrollado sea usable.

El documento ha sido dividido en capítulos para brindar el lector un mejor entendimiento del trabajo investigado y realizado. De esta forma, en el Capítulo 1 se expone la contextualización del problema, incluyendo en él, el planteamiento del problema, el objetivo general y los objetivos específicos, así como también el alcance de la propuesta.

En el Capítulo 2 se describe en forma general el ciclo de un juicio ordinario en Venezuela. Comenzando desde los requisitos que se deben cumplir para introducir una demanda en el tribunal, hasta que se emite la sentencia de la demanda. También en este capítulo se identifican las tareas y actividades que el abogado debe realizar durante el juicio. Este último punto resulta de gran importancia para obtener una visión más clara de las innumerables actividades que un abogado litigante en Venezuela debe planificar, controlar y ejecutar, para garantizar el éxito de sus casos.

El Capítulo 3, contempla los sistemas de apoyo a la gestión jurídica disponibles en el mercado, con sus respectivas características, señalando ventajas y desventajas que se presentan en su uso.

En el Capítulo 4 se listan y describen todas las herramientas tecnológicas consultadas y usadas en el desarrollo de la aplicación web propuesta.

Se expone en el Capítulo 5, los fundamentos de la metodología ágil AgilUs, seleccionada como disciplina metodológica para desarrollar la aplicación web.

Finalmente, se dispone de un capítulo para conclusiones y recomendaciones, en el cual se exponen los resultados e ideas conclusivas del trabajo realizado, así como recomendaciones que pueden considerarse para expandir, mejorar y continuar la aplicación web.

Capítulo 1. Planteamiento del problema

Los profesionales del derecho, en particular los abogados litigantes asisten y representan a personas naturales o jurídicas en procesos legales. En Venezuela, muchos de estos profesionales, ejercen su profesión de manera independiente, es decir, no pertenecen a algún escritorio de abogados o bufete, lo que implica, coordinar y controlar una gran cantidad de actividades en las diferentes etapas del proceso legal, ante un tribunal o ente determinado.

En ocasiones, cuando son varios los casos que atienden, el control de la información de cada caso de algún cliente, puede llegar a ser inmanejable de manera manual, y el apoyo que pueden brindar las herramientas ofimáticas no resulta suficiente.

Por otro lado, los abogados litigantes no tienen disponible la información. A diario y con frecuencia, les ocurre que requieren información que no tienen a la mano, bien sea porque la tienen en documentos físicos en algún lugar (casa, oficina o despacho) o en digital en un computador. Si el requerimiento es de urgencia, el abogado deberá interrumpir su agenda del día, para dirigirse al sitio donde tiene la información, lo cual, resulta engorroso y poco factible, si la distancia no es corta, o si se desenvuelven en una ciudad con altos índices de congestión vehicular, teniendo que dejar dicha actividad para el día siguiente. En situaciones extremas, se han presentado casos en donde información muy valiosa e irrecuperable, se llega a extraviar.

Al contrario de países como España y Argentina, que poseen una gran cantidad de sistemas dirigidos a bufetes, en nuestro país son muy escasas las aplicaciones que apoyen a la gestión jurídica de los abogados litigantes, y las que existen son de carácter comercial, lo que implica, pagar para usar.

Finalmente, los sistemas disponibles en el mercado suelen tener muchas funciones, presentando al usuario final una interfaz gráfica repleta de botones y menús, que terminan siendo una barrera de uso, por temor a realizar algo indebido, por no llegar a dominarlo o sencillamente creer que apoyarse en el sistema, les traerá más tareas de las que ya tienen por la naturaleza de su profesión.

1.1 Objetivo general

Desarrollar una aplicación web para apoyar la gestión jurídica de abogados litigantes de Venezuela.

1.2 Objetivos específicos

- Describir y analizar los procesos y procedimientos básicos y relevantes de un juicio ordinario en Venezuela.
- Identificar la gestión que el abogado litigante lleva a cabo durante un juicio ordinario en Venezuela, tanto a nivel jurídico como administrativo.
- Diseñar la aplicación web de apoyo para abogados litigantes de Venezuela, de modo que permita al abogado navegar y hacer seguimiento simplificado del proceso civil de cada uno de los casos que registra en la aplicación.
- Determinar los requerimientos de la aplicación web de apoyo para abogados litigantes de Venezuela.
- Diseñar las funcionalidades de la aplicación web de apoyo para abogados litigantes de Venezuela, a partir de los requerimientos definidos.
- Diseñar la interfaz de la aplicación web de apoyo para abogados litigantes de Venezuela.
- Diseñar la aplicación web usando el patrón de arquitectura de software Modelo Vista Controlador.
- Diseñar la aplicación web de apoyo para abogados litigantes de Venezuela siguiendo la metodología ágil AgilUs.
- Implementar la aplicación web de apoyo para abogados litigantes de Venezuela.
- Evaluar la aplicación web de apoyo para abogados litigantes de Venezuela.

1.3 Justificación

La carencia de herramientas tecnológicas gratuitas que apoyen la actividad de los abogados litigantes, es una realidad en Venezuela. Actualmente un gran número de abogados deben combinar herramientas ofimáticas y registros manuales, para poder administrar y controlar toda la información que manejan, desaprovechando las facilidades que la tecnología puede ofrecerles.

Resulta conveniente diseñar y desarrollar una aplicación web que facilite la gestión de los abogados litigantes, haciendo énfasis en la automatización de algunos procesos de gestión, administración, planificación y control de dichas actividades e información de los diferentes casos legales que llevan. Además, disponer de esta información desde cualquier sitio, para eliminar la dependencia con una estación de trabajo, permitiendo de esta manera, ejercer su profesión de forma controlada, eficiente y oportuna.

Una aplicación web a la que puedan acceder desde cualquier parte y a cualquier hora, donde dispongan ordenada y centralizadamente de toda la información sobre

clientes, actuaciones, documentos, citas, entes de carácter legal, entre otros; sería una potencial alternativa y aportaría de múltiples ventajas a los abogados litigantes en sus labores diarias.

1.4 Alcance y delimitación

El presente trabajo contempla el diseño y desarrollo de una aplicación web de apoyo a la gestión jurídica. La aplicación está dirigida a profesionales del derecho, específicamente, para abogados litigantes de Venezuela. Dicha aplicación, es de uso libre y gratuito bajo la modalidad web.

La aplicación web está alojada en una plataforma gratuita, que provee todos los recursos y servicios necesarios para su implementación. Por lo tanto, no es objeto de investigación de este trabajo, considerar los detalles de implementación, tales como: sistema operativo del servidor que aloja la aplicación web, configuración del servidor y de los servicios web, selección de base de datos, configuración de base de datos y publicación de la aplicación web.

Los módulos y funciones básicas que componen la aplicación web, son:

- Clientes: Registro, modificación y consulta de clientes.
- Expedientes: Registro, modificación y consulta de expedientes, así como de las actuaciones y documentos.
- Agenda: Registro, modificación y consulta de eventos. Registrar los días no laborales y no despacho de organismos. Consultar de forma filtrada los calendarios de los organismos con marcas en los días no laborables y no despacho.
- Organismos: Registro, modificación y consulta de organismos.
- Plantillas: Carga, modificación y consulta de documentos. Generación, modificación y consulta de plantillas.

El proyecto en cuestión se encuentra delimitado, en primer lugar al contexto nacional, por lo tanto, los términos usados para hacer referencia a objetos o acciones dentro de la aplicación, serán familiares a abogados litigantes que laboran en Venezuela.

Capítulo 2. Gestión legal de un abogado litigante en Venezuela

Un abogado litigante en Venezuela ejerce profesionalmente la defensa o acusación jurídica de otra persona en un juicio o algún proceso judicial, apoyándose en la legislación venezolana, la cual está conformada básicamente por: la Constitución Bolivariana de Venezuela y el sistema de leyes que rigen las actividades legales en el país.

El poder judicial venezolano, se rige por el Tribunal Supremo de Justicia y Tribunales Superiores, entre otros; el Tribunal Supremo de Justicia está conformado por seis salas, la Sala de Casación Civil, Sala de Casación Penal, Sala de Casación Social, Sala Político-Administrativa y Sala Electoral [2].

Comprender la manera en que los abogados litigantes llevan a cabo y controlan todas las actividades inherentes a un caso, requiere ineludiblemente de un análisis de los procedimientos involucrados en un juicio, en donde los principales actores y ejecutores son: el Juez, las partes y contrapartes, los terceros, el fiscal del Ministerio Público y los auxiliares de justicia en determinado tiempo y lugar.

A continuación, se describe el proceso y procedimientos de un juicio ordinario sencillo, obviando algunos detalles y vertientes que pueden presentarse en el desarrollo del proceso y que no son objetivo de esta investigación. Posteriormente, se listarán las actividades que el abogado litigante realiza durante un juicio con las características antes mencionadas.

2.1 Procedimientos de un juicio ordinario

El proceso, en términos del Derecho, es la controversia que se suscita entre dos o más personas (de carácter natural o jurídico) ante una autoridad judicial, bajo las formas procedimentales establecidas en la ley, con el fin de obtener un pronunciamiento sobre una determinada situación jurídica.

Todo proceso judicial se inicia cuando un demandante, otorga poderes especiales a un abogado a través de un documento notariado, para que este lo represente e introduzca un “Libelo”, que no es más que un escrito de la demanda, redactado por el abogado, que debe consignarse en el Tribunal.

El objeto de la demanda, es decir, la razón que la sustenta, se denomina causa. A ésta, se le determina el proceso apropiado para ser atendida, según lo que contemple

la Ley específica. Sin embargo, existe un procedimiento preestablecido en el Código de Procedimiento Civil (el cual de ahora en adelante se señalará en este documento por sus iniciales CPC) que puede aplicarse en todo caso jurídico y en materia civil, siempre y cuando, la Ley no indique lo contrario.

En la Figura 1, se puede apreciar un diagrama que resume las etapas más relevantes de un juicio ordinario, las cuales se describirán en las próximas secciones.

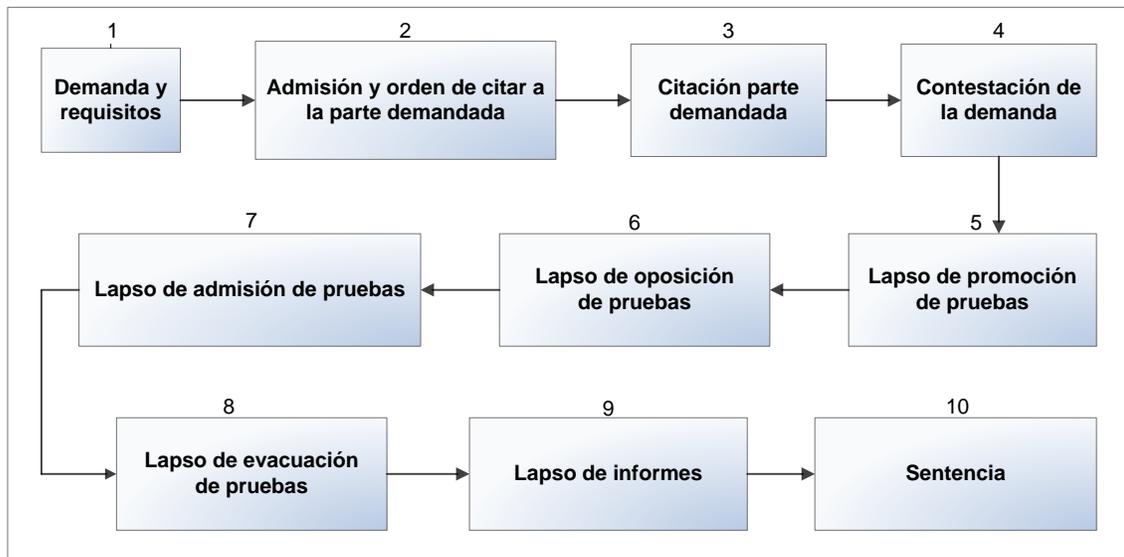


Figura 1: Diagrama de un juicio ordinario

En líneas generales, el proceso comienza cuando se ha introducido una demanda en el Tribunal. Dicha demanda será evaluada y analizada para determinar si es o no admitida. En caso de ser admitida, el Tribunal deberá citar a la parte demandada, quien está en el deber de contestar dicha citación. Luego se inicia la etapa de promoción de pruebas, en donde cada una de las partes debe anunciar todas las pruebas que intenta presentar, las cuales serán o no, objetadas por cada una de las partes. La decisión de cuales son legales y procedentes, la toma y la anuncia el Juez y con este acto se inicia el lapso de la evacuación de las pruebas admitidas, que no es más que la presentación de las mismas. Posteriormente, cada una de las partes tiene la oportunidad de presentar un informe que expone las razones por las cuales considera que el Juez debe tomar una posición favorable a la causa defendida. Por último, el Juez, anuncia la(s) decisión(es) final(es) en la sentencia.

2.2 Requisitos de la demanda

En virtud de la importancia que reviste la demanda como acto procesal y las consecuencias que tienen en el juicio, la Ley somete su forma y contenido al cumplimiento de determinados requisitos.

El CPC contempla que el libelo de la demanda deberá expresar:

- La indicación del Tribunal ante el cual se propone e la demanda.
- El nombre completo y domicilio del demandante y del demandado y el carácter que tienen.
- Si el demandante o el demandando fuera una persona jurídica, la demanda deberá contener la denominación o razón social y los datos relativos a su creación o registro
- El objeto de la pretensión, el cual deberá determinarse con precisión, indicando su situación y linderos, si fuere inmueble; las marcas, colores, o distintivos, si fuere semoviente; los signos, señales y particularidades que puedan determinar su identidad, si fuere mueble; y los datos, títulos y explicaciones necesarios si se tratare de derechos u objetos incorporeales.
- La relación de los hechos y los fundamentos de derecho en que se basa la pretensión, esto es, aquellos de los cuales se derive inmediatamente el derecho deducido, los cuales deberán producirse con el libelo.
- Si se demandare la indemnización de daños y perjuicios, la especificación de éstos y sus causas.
- El nombre y apellido del mandatorio y la consignación del poder.
- La sede o dirección del demandante [9].

2.3 Estimación de la demanda

En razón de las consecuencias procesales que produce la demanda, ésta debe ser estimada. La estimación tendrá relevancia en cuanto a: la competencia del Tribunal que conocerá el pleito, los recursos que concede la ley, lo relativo al procedimiento aplicable y en cuanto a las costas del proceso.

2.4 Admisión de la demanda

Presentada la demanda, el Tribunal tendrá tres (3) días para rechazarla o admitirla. En caso de que la admita, se debe proceder con el procedimiento de la citación del demandado.

2.5 Citación del demandado

Admitida la demanda, conforme lo dispone el CPC, se debe notificar al demandado, para que éste se dé por enterado de la misma, y comparezca en el respectivo Tribunal. La citación puede ser llevada a cabo por el demandante o por su abogado, y esta puede hacerse en varias formas. Una vez efectuada, se debe notificar al Tribunal, el resultado de la citación.

2.6 Contestación de la demanda

Citado el demandado, deberá comparecer ante el Tribunal pertinente dentro de un lapso de veinte (20) días o al término del mismo para realizar la contestación de la demanda, la cual, consiste en exponer las razones del hecho y de derecho sobre los alegatos, afirmaciones y peticiones formuladas por el actor en su libelo.

La contestación de la demanda debe presentarse por escrito al Secretario del Juez, quien estampará una nota indicando que la misma, es la contestación presentada en fecha y hora del acto.

2.7 Etapa probatoria

La etapa probatoria en el proceso civil venezolano, se cumple en cuatro momentos:

1. Lapso de promoción de pruebas, que es de quince (15) días de despacho, contados a partir del vencimiento del lapso de contestación.
2. Lapso de incorporación de las pruebas al expediente, que también es llamado lapso de oposición, y que es de tres (3) días de despacho, contados a partir del día siguiente al vencimiento del último día del lapso de promoción.
3. Lapso de admisión de las pruebas, que es de tres (3) días de despacho contados a partir del vencimiento del lapso de oposición.
4. Lapso de evacuación que es de treinta (30) días de despacho, contados a partir del día siguiente al auto de admisión de las pruebas.

2.8 Evacuación de pruebas

Admitidas las pruebas, o dadas por admitidas, comenzarán a computarse los treinta (30) días de despacho destinados a la evacuación; pero si hubieren de practicarse alguna de éstas en otro Tribunal, se dará la correspondiente comisión.

2.9 Informes

El acto de informes es un acto procesal complejo y de gran importancia para las partes. La Ley fija la oportunidad para la presentación de los informes, los cuales son escritos en donde las partes tienen la oportunidad de exponer congruentemente, con base en los hechos y pruebas del expediente, las razones jurídicas que consideren pertinentes con miras a una decisión favorable a la posición que sostienen en la causa.

Los informes constituyen una especie de autosentencia de la causa, dictada por el interesado, y aunque el Tribunal no está obligado a examinar la cuestión de hecho y de derecho planteadas por las partes, en muchos casos, unos informes bien ordenados, donde se exponen correctamente los hechos y se le den una motivación legal suficiente, con indicación de doctrina y de jurisprudencia, donde se trate y se haya decidido casos semejantes, contribuyen a orientar al Juez para tomar una decisión más ajustada a derecho, y esa es precisamente una de las finalidades de este acto procesal [5].

Los informes de las partes deben presentarse en el decimoquinto día siguiente al vencimiento del lapso probatorio. Y si cada una de las contrapartes no tiene observaciones que hacer a los informes presentados, a partir de ese día de presentación, comienza a correr el lapso para dictar sentencia. Si por el contrario, existen observaciones, comienza a correr un lapso de ocho (8) días para formularlas.

2.10 Sentencia de la demanda

La sentencia es un pronunciamiento judicial; un acto procesal dictado por el Juez mediante el cual decide en forma definitiva la causa.

Cuando la sentencia ha quedado definitivamente firme, el Tribunal a petición de la parte interesada pondrá un decreto ordenando su ejecución, en el cual se indicará un lapso que no será menor a tres (3) días ni mayor de diez (10), para que el deudor cumpla voluntariamente y no se comenzará la ejecución hasta que haya transcurrido la totalidad del lapso.

2.11 Actividades realizadas por un abogado litigante

Son muchas las actividades que un abogado litigante en Venezuela debe llevar a cabo para finalizar con éxito una disputa legal. En términos de la demanda, todas y cada una de ellas son distintas por el fin que las justifica y la información asociada, pero en términos de tareas o acciones, resultan ser parecidas, repetidas en distintos

contextos. Y entre ellas se identifican las siguientes: acordar citas, llevar a cabo reuniones, redactar documentos, realizar diligencias y trámites, recolectar y recordar información, realizar cálculos, entre otras.

Expuesto el proceso y procedimientos de un juicio ordinario sencillo en orden cronológico, se identificaron las siguientes actividades claves que los abogados deben realizar:

1. Acordar una cita con el cliente para que éste le relate la situación por la cual requiere de sus servicios. Muchos de los abogados toman nota en este encuentro de todos los detalles que el cliente le brinda en la narración, otros se apoyan en grabaciones con el propósito de consultarlas posteriormente.
2. Si el abogado considera que la situación merece ser llevada a una disputa legal, debe recolectar todos los datos del cliente, tales como:
 - Nombre y apellidos completos, si es persona natural. En caso de que sea persona jurídica, datos de la razón social, incluyendo la creación y registro.
 - Número de Identificación (para persona natural).
 - Número de RIF (opcional para persona natural y obligatorio para persona jurídica).
 - Nivel de estudios, profesión, actividad a la que se dedica (para persona natural y jurídica).
 - Dirección completa de habitación, indicando parroquia, municipio, ciudad, estado y código postal (persona natural y jurídica).
 - Dirección completa del empleador (donde ejerce la actividad a la que se dedica), o de la persona jurídica. Debe ser completa, indicando parroquia, municipio, ciudad, estado y código postal.
 - Fecha de nacimiento (para persona natural).
 - Estado civil (para persona natural).
 - Teléfonos de contacto: habitación, móvil, trabajo, fax (para persona natural y jurídica)
 - Correo electrónico (para persona natural y jurídica).
3. Creación de un documento, donde el demandado o demandante otorga poderes especiales al abogado, para que éste lo represente ante instancias judiciales. Por lo general, los abogados recurren a un documento que han realizado anteriormente, es decir, usan otro documento de plantilla.
4. Creación del Libelo de la demanda. Al igual que el poder, se apoyan en un documento creado con anterioridad, para usarlo como plantilla. Cabe destacar, que para crearlo deben recolectar los siguientes datos:
 - Identificación del Tribunal competente.

- Nombre completo y domicilio del demandante y demandado, así como el carácter que tienen.
 - Si el demandante o el demandado fuera una persona jurídica, la demanda deberá contener la denominación o razón social y los datos relativos a su creación o registro
 - El objeto de la pretensión.
 - La relación de los hechos y los fundamentos de derecho en que se basa la pretensión.
 - Si se demandare la indemnización de daños y perjuicios, la especificación de éstos y sus causas.
 - El nombre y apellido del mandatorio y la consignación del poder.
 - La sede o dirección del demandante.
5. Cálculo de la estimación de la demanda.
 6. Consignación de la demanda en el Tribunal.
 7. Consignada la demanda, el Tribunal tiene tres (3) días para aceptar o rechazar la demanda, por lo tanto, el abogado debe recordar esta cuenta de días y fechas, para consultar el estatus de la demanda, bien sea a través de un servicio en línea, o de forma presencial en el Tribunal.
 8. Ejecutar la citación del demandado bajo los medios permitidos en el CPC y en los lapsos establecidos, es decir, el abogado debe llevar la cuenta de días y fechas.
 9. Notificar al Tribunal el resultado de la citación en el lapso establecido.
 10. Citado el demandado, éste tiene veinte (20) días para comparecer al Tribunal y realizar mediante un escrito la contestación de la demanda, por lo que el abogado demandante, deberá consultarla para conocer el resultado y/o acudir al Tribunal, para conocer si el demandado alegó cuestiones previas o realizó alguna reconvencción.
 11. En caso de que no se aleguen cuestiones previas, se inicia la etapa probatoria, en la cual, el abogado debe recordar la cuenta de días y fechas para realizar específicos actos en determinados procedimientos y momentos:
 - a. Lapso de promoción de pruebas: quince (15) días de despacho.
 - b. Lapso de oposición de pruebas: tres (3) días de despacho. Tomando en cuenta que el primer día se debe expresar si conviene en alguno o de los hechos que alegue la contraparte.
 - c. Lapso de admisión de pruebas: tres (3) días de despacho.
 - d. Lapso de evacuación de pruebas: treinta (30) días de despacho.
 12. Creación del informe. En la cual el abogado debe valerse de todos los hechos de la causa, los expuestos en el juicio, las pruebas, los actos ejecutados, y hasta de casos aislados pero semejantes, para lo cual, el abogado debe

realizar una revisión historial de otros casos, al igual que doctrinas y jurisprudencia.

13. Presentar el informe en el Tribunal luego de quince (15) días de la finalización de la etapa probatoria.
14. En caso de que el abogado desee hacer una observación, debe formularla en un lapso de ocho (8) días. Caso contrario se dicta sentencia, y el abogado demandante debe mantener la cuenta de días y fechas, pues el lapso de cumplimiento de sentencia para el demandado es, no menor a tres (3) días y no mayor a diez (10).

Capítulo 3. Sistemas de apoyo a la gestión jurídica

En este capítulo se señalarán algunas herramientas previas a este trabajo, algunas privativas y otras gratuitas.

3.1 Antecedentes

En la Escuela de Computación de la Universidad Central de Venezuela se han desarrollado un par de aplicaciones con objetivos cercanos a los propuestos en este trabajo. A continuación se reseña cada uno de ellos:

- Rubén Darío Castillo Morales en el 2001, realizó un trabajo de grado titulado “Sistema Colaborativo para la Gestión, Planificación y Control de casos Jurídicos como apoyo al Ejercicio del Derecho”. El mismo consistió en desarrollar una aplicación basada en la colaboración, comunicación y coordinación para la gestión, planificación y control de casos jurídicos en firmas de abogados.
- Y Carlos José Peña en el 1994 presentó su trabajo titulado “Prototipo de Ambiente Interactivo para la administración automatizada de un Bufete de Abogados”. El objetivo perseguido con este prototipo fue la agilización de las tareas administrativas de un bufete de abogados, como: el manejo de información asociado a los casos jurídicos, la manipulación y generación de correspondencia, y el control de la nómina y la contabilidad.

3.2 Herramientas disponibles en el mercado

En la actualidad, son muchos los sistemas existentes en el mercado, desarrollados para facilitar la gestión legal de abogados. La mayoría de ellos, al igual que los trabajos de grado mencionados anteriormente, están dirigidos a despachos jurídicos que requieren administrar, manejar y compartir la información jurídica de sus clientes y sus expedientes, de forma organizada y controlada para todos los colaboradores.

Básicamente, existen dos esquemas de implementación de estas soluciones dirigidas a despachos jurídicos. Una, basada en la arquitectura cliente-servidor, y otra, que provee el servicio en la nube, como actualmente se les referencia a los sistemas o aplicaciones cargados en la web.

En el primer esquema, el sistema consiste en una combinación entre uno o varios servidores que aloja uno o varios servicios con toda la información del despacho jurídico, y aplicaciones clientes que deben ser instaladas de forma independiente en

distintos computadores, los cuales, realizan peticiones al primer componente en una misma red. Algunos ejemplos de estos sistemas son:

- **Abogafin Software para Abogados**, creado por Netfincas Software, empresa española de desarrollo de software [13].
- **Gedex Software de Gestión de Expedientes Jurídicos para Escritorios**, Estudios Jurídicos y Asesorías Letradas, creado por Brindys, empresa española de desarrollo de software [19].
- **Tantum Sistema de Gestión Jurídica**, creado en España [17].
- **Lex-Doctor Gestión Jurídica**, creado por Lex-Doctor · Sistemas Jurídicos S.R.L., empresa argentina de desarrollo de software [15].

El segundo esquema de sistemas jurídicos para despachos de abogados, basado como ya se mencionó, en un servicio alojado en la nube, al cual los usuarios acceden a través de un navegador web, bajo un método de autenticación que proporciona seguridad a la información alojada en el sistema.

Algunos sistemas que usan este esquema son:

- **Mnprogram Abogados**, creado por MNprogram, una compañía de software de gestión para micropymes y pequeñas empresas de España [20].
- **Sistema de Gestión Legal** (Tu escritorio jurídico en la Web), creado por Grupo ESW, C.A. [14].

Una descripción de características de estas herramientas se detalla en el próximo Capítulo, específicamente en la Etapa de Análisis de la metodología aplicada.

Capítulo 4. Metodología y desarrollo de la aplicación

Las metodologías de desarrollo de software son marcos de trabajos usados para estructurar, planificar y controlar el proceso de desarrollo de un software. A lo largo del tiempo, una gran variedad de métodos han sido desarrollados en el área de Ingeniería de Software.

En este proyecto se decidió utilizar la metodología AgilUs para el logro exitoso de los objetivos planteados. Ella, pertenece al grupo de metodologías ágiles, las cuales se mantienen vigentes desde hace más de dos décadas.

4.1 Metodologías ágiles

El término ágil aplicado al desarrollo de software nace en el 2001 después de la reunión de 17 expertos de la industria y se refieren al desarrollo iterativo e incremental del software, donde los requisitos y soluciones evolucionan mediante la colaboración de grupos auto organizados y multidisciplinarios, con la finalidad de acortar tiempos de entrega y dar respuesta a los cambios en el desarrollo del software [63].

Después de esta reunión nace La Alianza Ágil (The Agile Alliance), una organización sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y apoyar a las organizaciones para que adopten dichos conceptos. Asimismo publican el "Manifiesto de desarrollo ágil de software", cuyos principios se basan en valorar:

- ***Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas.*** Las personas son el principal factor de éxito de un proyecto de software. Es más importante construir un buen equipo que construir el entorno. Muchas veces se comete el error de construir primero el entorno y esperar que el equipo se adapte automáticamente. Es mejor crear el equipo y que éste configure su propio entorno de desarrollo en base a sus necesidades.
- ***Desarrollar software que funcione más que conseguir una buena documentación.*** La regla a seguir es no producir documentos a menos que sean necesarios de forma inmediata para tomar una decisión importante. Estos documentos deben ser cortos y centrarse en lo fundamental.
- ***La colaboración con el cliente más que la negociación de un contrato.*** Se propone que exista una interacción constante entre el cliente y el equipo de desarrollo. Esta colaboración entre ambos será la que marque la marcha del proyecto y asegure su éxito.

- **Responder a los cambios más que seguir estrictamente un plan.** La habilidad de responder a los cambios que puedan surgir a lo largo del proyecto (cambios en los requisitos, en la tecnología, en el equipo, etc.) determina también el éxito o fracaso del mismo. Por lo tanto, la planificación no debe ser estricta, sino flexible y abierta.

Los principios del “Manifiesto de desarrollo ágil de software” son:

- I. La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporte un valor.
- II. Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva.
- III. Entregar con frecuencia software que funcione. Desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas.
- IV. La gente del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.
- V. Construir el proyecto en torno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir finalizar el trabajo.
- VI. El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.
- VII. El software que funciona es la medida principal de progreso.
- VIII. Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.
- IX. La atención continua a la calidad técnica y al buen diseño mejora la agilidad.
- X. La simplicidad es esencial.
- XI. Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos.
- XII. En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo, y según esto se ajusta su comportamiento [63].

Algunas de las metodologías ágiles identificadas son:

- Adaptive Software Development (ASD).
- Agile Unified Process (AUP).
- Crystal Clear.
- Essential Unified Process (EssUP).
- Feature Driven Development (FDD).
- Lean Software Development (LSD).
- Kanban.

- Open Unified Process (OpenUP).
- Programación Extrema (XP).
- Método de desarrollo de sistemas dinámicos (DSDM).
- Scrum.
- G300.
- AgilUs

4.2 Metodología AgilUs

Tal y como se describe en la publicación de Eleonora Acosta, AgilUs es un método de desarrollo ágil de software, resultado de una de las líneas de investigación llevada a cabo en el Centro de Ingeniería de Software y Sistemas (ISYS) de la Escuela de Computación, de Universidad Central de Venezuela.

Esta metodología se basa en el concepto de usabilidad, en la necesidad de desarrollar software usables. Se fundamenta en el análisis centrado en el usuario y en la participación de especialistas, con el objetivo de evolucionar el software, a fin de que éste alcance el mayor grado de usabilidad una vez culminado su desarrollo.

Una de las principales características del método, es el de desarrollo iterativo e incremental que pone el mayor peso del desarrollo en la consecución de la usabilidad. Se centra en que la construcción y desarrollo de las interfaces de usuario no debe ser una adición estética que se da al final del desarrollo del sistema sino, muy por el contrario, el desarrollo de interfaces de usuario debe guiar las decisiones en Ingeniería de Software. En AgilUs son los usuarios, y no el cliente ni los programadores quienes guían el desarrollo del proyecto.

El método AgilUs busca proporcionar un conjunto de actividades organizadas para construir la usabilidad en el diseño de interfaces de usuario durante el desarrollo de un producto de software. El proceso de desarrollo de software engloba las actividades de requisitos, análisis, prototipaje y entrega; así como las evaluaciones de usabilidad correspondientes a cada etapa del proceso. Se realizan en ciclos iterativos hasta alcanzar el producto final. En cada etapa del proceso de desarrollo de software, se incluyen actividades propias para la construcción de la usabilidad.

Principios de la metodología AgilUs

AgilUs centra el desarrollo de software en los siguientes principios:

- Integra la Interacción Humano Computador (IHC) y la Ingeniería de Software (IS). IHC e IS son complementarias, no son disciplinas excluyentes. Un diseño

centrado en el usuario impacta positivamente en la calidad del software (ISO 9126-1).

- La usabilidad debe considerarse desde el principio del desarrollo. Si la IS y la IHC son complementarias y no excluyentes, y si la usabilidad aumenta la calidad del software, entonces es conveniente incluir la usabilidad desde el principio en el desarrollo como uno de los requisitos para impactar positivamente la calidad del producto final.
- La usabilidad determina la utilidad. Un software se considera útil en la medida que pueda ser usado a fin de producir resultados, en forma eficiente, intuitiva y satisfactoria para los usuarios.
- El usuario determina la usabilidad. La usabilidad no es una propiedad abstracta. Un software sólo será considerado usable en un contexto específico y por un tipo de usuario específico. El objetivo es lograr que todos los usuarios del software encuentren usables las tareas que pueden realizar.

4.3 Mejores prácticas de AgilUs

Las “buenas prácticas” del método están orientadas en satisfacer las demandas del usuario y el desarrollo iterativo e incremental, procurando la usabilidad en cada paso del proceso de desarrollo. Algunas de estas “buenas prácticas” son:

- Diseño centrado en el usuario (DCU). Este es un enfoque de diseño y desarrollo que se centra en los deseos, limitaciones y necesidades de los usuarios finales de un software. En las técnicas de DCU es relevante que los desarrolladores realicen pruebas constantes para verificar el curso que lleva el desarrollo del sistema y su interfaz de usuario. De este modo, el usuario guía indirecta pero influyentemente el proceso de desarrollo del sistema.
- Diseño basado en prototipos. AgilUs busca la generación alta fidelidad y la evaluación de los mismos por los usuarios y por especialistas en usabilidad. Los desarrolladores producen un primer prototipo, los especialistas y el usuario lo evalúan, los analistas preguntan directamente al usuario sus opiniones sobre el desarrollo, y con esa retroalimentación, los desarrolladores se disponen a producir un siguiente prototipo. Este ciclo continúa hasta que se tiene un producto listo para la entrega, cuando las evaluaciones de usabilidad, requerimientos y calidad del software están completamente satisfechas.
- Desarrollo ágil, iterativo e incremental. Se recomienda desarrollar el sistema más simple que satisfaga las necesidades actuales de los usuarios, preparándose para cambios futuros. El desarrollo por incrementos permite proveer resultados sin necesidad de esclarecer todo los requisitos de una vez al inicio del desarrollo. La iteratividad permite regresar a etapas anteriores una vez recibida la retroalimentación producto de las evaluaciones realizadas.

- Usabilidad como atributo de la calidad. Como se indica en el estándar ISO/IEC 9126-1, la usabilidad es considerada un atributo de la calidad interna y externa del software. AgilUs hace énfasis en la producción de software usable, siguiendo la recomendación de este estándar internacional.
- Interacción continua con el usuario, propiciando un intercambio cara a cara. La idea es privilegiar la usabilidad, por ello el equipo de desarrollo tomará decisiones tras realizar evaluaciones de usabilidad, ya que la usabilidad del sistema sólo puede ser determinada por el usuario.

4.4 Ciclo de Vida

Como ya se mencionó, AgilUs está basado en el desarrollo iterativo e incremental de prototipos de alta fidelidad hasta que se convierten en el producto final para entrega. El ciclo de vida de AgilUs contempla 4 etapas y en cada una de ellas hace énfasis en las evaluaciones del usuario. Las etapas son: requisitos, análisis, prototipaje y entrega. Cada una de ellas contempla lo siguiente:

- **Requisitos.** Se realiza el análisis global del problema a solucionar, se estudian productos similares existentes, se genera un perfil de usuario, y se define la lista de requerimientos a desarrollar. Esta etapa es importante en el desarrollo del software, ya que un mal análisis de requisitos traería como consecuencia un software que no cumple con las necesidades del usuario.
- **Análisis.** Se lleva a cabo el análisis de la solución a desarrollar, se emplean diagramas de casos de uso y modelo de objetos del dominio, siguiendo la notación UML, para definir las funcionalidades que tendrá el producto a desarrollar.
- **Prototipaje.** Se implementa un prototipo rápido de la interfaz de usuario a partir de los patrones de interacción, el cual va evolucionando hasta convertirse en el producto final, se genera la guía de estilo, y se realizan evaluaciones de usabilidad apropiadas a esta etapa: las evaluaciones heurísticas y las listas de comprobación.
- **Entrega.** Se aplican las pruebas al sistema para certificar que la aplicación desarrollada sea un software usable y sin errores, finalmente se pone en producción la aplicación.

La siguiente figura representa gráficamente el ciclo de vida. La cual debe comenzar a leerse desde la izquierda avanzando hacia la derecha con un giro completo de 360°, pasando desde la fase de requisitos hacia al análisis, luego el prototipaje y finalmente hasta la entrega.



Figura 2: El método AgilUs: etapa, actividades y artefactos [79]

4.5 Aplicación de la metodología AgilUs

A continuación se exponen las técnicas usadas en cada una de las etapas que componen de la metodología.

4.6 Etapa 1: Requisitos

Los requisitos de la aplicación se fueron construyendo usando en primera instancia la técnica de la tormenta de ideas. Luego se evaluaron las herramientas existentes con sus respectivas características y limitaciones. Posteriormente se aplicó una encuesta para conocer cómo trabajan los abogados en la actualidad y así ir definiendo adecuadamente los beneficios que la herramienta aportaría a la gestión de los abogados. Inmediatamente se definió el perfil de usuario y para cerrar la etapa se determinaron los requerimientos funcionales y no funcionales.

Tormenta de ideas

La idea central de este trabajo nació de múltiples reuniones llevadas a cabo entre la autora, tutores y abogados litigantes que trabajan de forma independiente. Con esas reuniones se obtuvo visión clara de la necesidad de los abogados de contar con una herramienta que apoyará su gestión.

La investigación realizada en el Capítulo 2 acerca de la gestión de los abogados litigantes, apoyó significativamente el levantamiento de la información.

Evaluación de sistemas existentes

Una gran parte de los sistemas de apoyo a la gestión jurídica disponibles en el mercado, son desarrollados en España y Argentina. Estos, están dirigidos a escritorios jurídicos y todos tienen como propósito primordial, brindar a sus clientes una solución que soporte la automatización y organización de expedientes, clientes, documentos y actividades. Algunas de las características básicas y comunes de estos sistemas, son:

- **Licenciamiento de software propietario.** Debido a que este tipo de sistemas son desarrollados por empresas dedicadas a la creación de software, su uso implica un único pago previo o periódico (mensual o anual) por la licencia. El acuerdo adquirido en el contrato, prohíbe la reproducción no autorizada, limita el acceso al código fuente y la cantidad de equipos en donde se puede instalar.

1 Seleccione los productos que desea adquirir

Producto	Seleccionar	Precio unitario	Subtotal
 Tantum 2.1 Server (Incluye un usuario)	<input checked="" type="checkbox"/>	\$ 1300	\$ 1300
▸ Usuarios Adicionales	<input type="text" value="Ninguno"/>	\$ 650	\$ 0
▸ Servicio de Acceso Remoto (Abono Anual)	<input type="checkbox"/>	\$ 0	\$ 0
▸ Servicio Backup	<input type="checkbox"/>	\$ 700	\$
▸ Servicio de mensajería	<input checked="" type="checkbox"/>	(Bonificado)	\$ 0
▸ Ayuda On Line	<input checked="" type="checkbox"/>	(Bonificado)	\$ 0
Total:			\$ 1300

Figura 3: Tarifas de Tantum Sistema de Gestión Jurídica [17]

- **Arquitectura de tipo cliente-servidor.** Basado en el procesamiento cooperativo de la información por medio de un conjunto aplicaciones, este esquema de trabajo permite a los escritorios jurídicos, centralizar, clasificar, ordenar y compartir adecuadamente toda la información de los expedientes gestionados por los distintos abogados que pertenecen a él. Dichas aplicaciones cliente, pueden estar distribuidas en una red corporativa en una misma localidad o en inclusive en distintas.

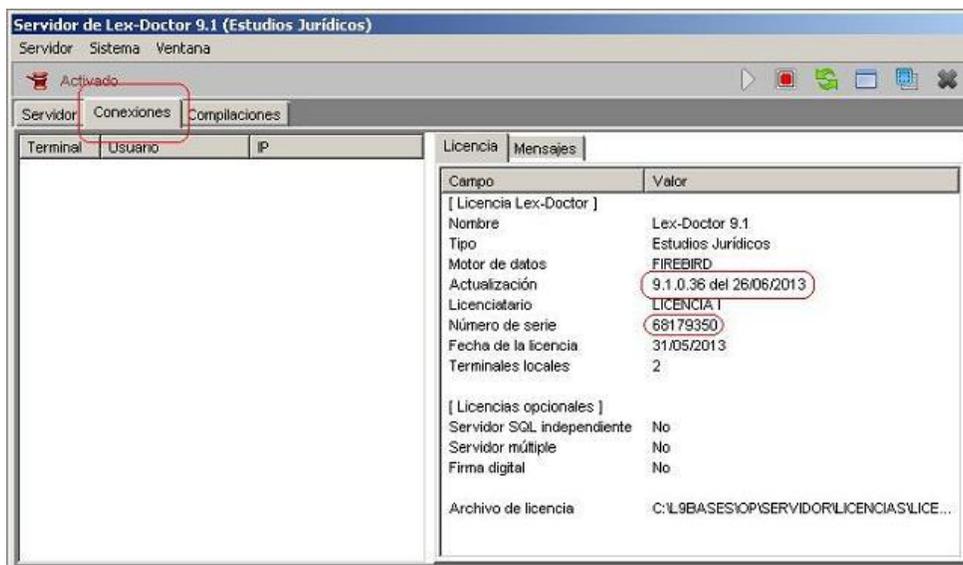


Figura 4: Estatus de un servidor Lex-Doctor v9.1 [15]

- **Interfaz gráfica.** Está compuesta por un área de trabajo en donde se muestran las respectivas ventanas del módulo que se esté trabajando, con botones e íconos intuitivos, y menús descriptivos que permiten realizar acciones en el sistema.
- **Formularios para insertar, modificar y eliminar datos.** La forma de agregar, consultar, modificar o eliminar información en cada uno de los distintos módulos (clientes, expedientes, documentos, etc.) es a través de formularios.
- **Capacidades del sistema separadas en módulos.** Los módulos que se encuentran con más frecuencia en este tipo de sistemas son: clientes, expedientes, documentos, organización, agenda, facturación y contabilidad. Los módulos de clientes, expedientes y documentos, permiten agregar, modificar y eliminar registros. El de organización y agenda, brinda al abogado un espacio donde registrar en fechas determinadas citas, recordatorios o eventos de relevancia. Los módulos de facturación y contabilidad se usan para facturar y llevar un control de los servicios prestados por el abogado; resultan muy útiles en aquellos casos en donde más de un abogado de un escritorio jurídico atiende a un cliente.

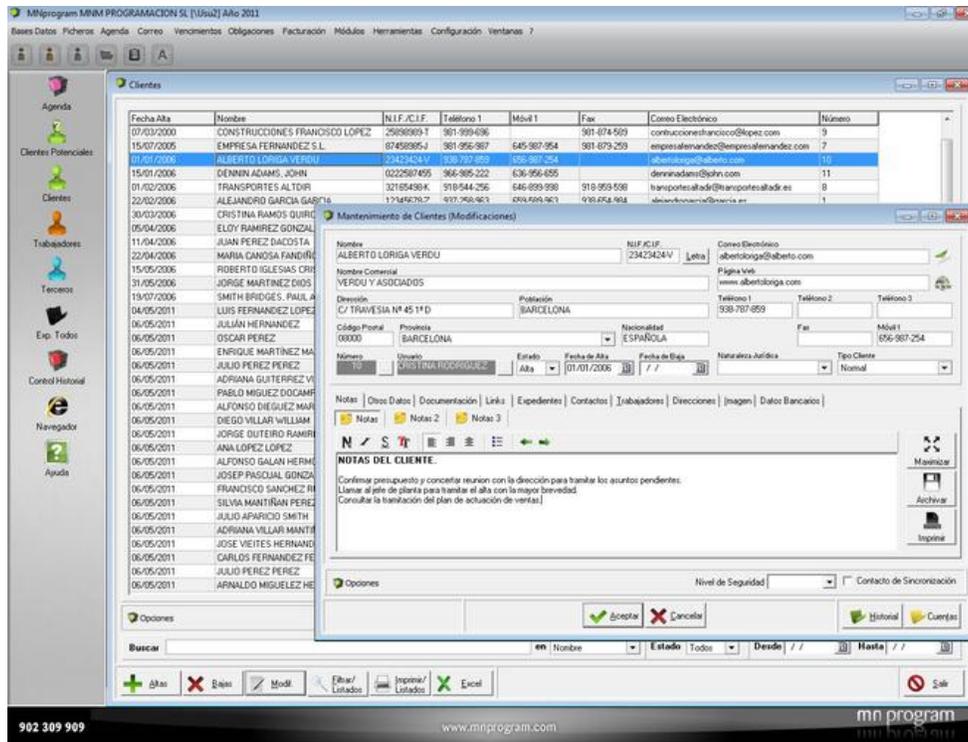


Figura 5: Módulo Clientes de MNProgram Abogados [20]

- Publicación del estatus del caso a los clientes.** Esta capacidad se refiere a la publicación del estatus del caso en un sitio en internet. El escritorio asigna una cuenta de usuario y una clave a los clientes, para que estos puedan acceder y consultar el estatus y la gestión que se realiza de su caso.

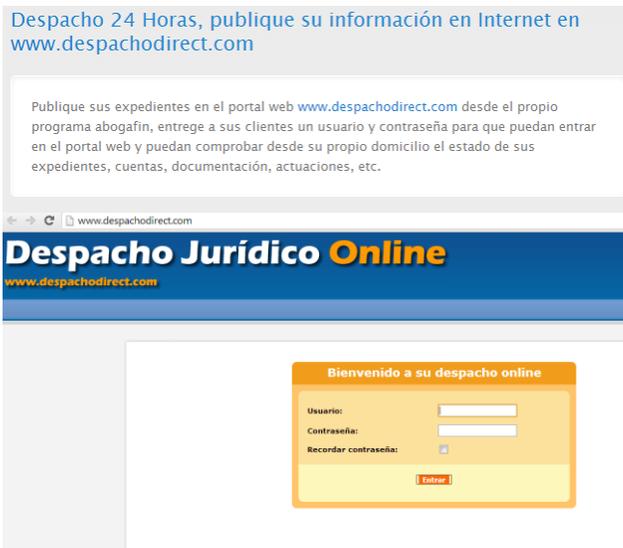


Figura 6: Opción publicación de estatus de Abogafin de Netfincas [13]

Por otro lado, se construyó una tabla comparativa de otras capacidades de algunas de las herramientas evaluadas:

	Abogafin	Gedex	Tantum	Lex-Doctor
Totalmente en internet	No	No	No	SI
Facilidad de carga cualquier formato de archivos	SI	Si	No	Si
Generación de plantillas	SI	No	Si	Si
Uso intuitivo	No	No	No	No
Interfaz amigable	SI	Si	Si	Si
Manejo de agenda	SI	Si	Si	Si
Licenciamiento propietario	No	Si	No	No
Manejo de ingresos	SI	Si	Si	Si

Tabla 1: Tabla comparativa de los sistemas de abogados en el mercado

Algunas de las limitaciones encontradas a las herramientas evaluadas son:

- No están dirigidos a abogados que ejercen en Venezuela de forma independiente. Por el contrario, están dirigidos a bufetes, lo que origina que muchas de las funcionalidades y beneficios de dichos sistemas, resulten desaprovechadas y en algunos casos inutilizadas.
- Otra limitación importante es de tipo económica. Los sistemas basados en la arquitectura cliente-servidor, implican la compra de una licencia de un software, y en ocasiones requiere otros pagos por actualizaciones y mejoras que se le agreguen al sistema. Y por servicios en la nube, se deben realizar pagos periódicos como renta, para continuar usando el servicio. Cabe destacar, que por los sistemas desarrollados fuera de Venezuela se debe cancelar en moneda extranjera.
- Sistemas con muchas funcionalidades, resultan siendo una barrera de uso y por lo general, requieren de cursos de capacitación para conocerlo y aprender a operarlo.

Encuesta aplicada

La encuesta realizada es la siguiente:

Encuesta de levantamiento de información

Para cada una de las preguntas de la siguiente encuesta, responda con una X la opción de su preferencia o complete en los casos que sea necesario.

Preguntas:

En relación a la información de sus clientes:

1. ¿En dónde guarda los datos de sus clientes, tales como: nombre completo, cédula, estado civil, dirección, teléfonos, etc.?
 - a) De forma manual/escrita []
 - b) En la computadora (agenda, documentos en Word, notas, etc.) []
 - c) En una agenda en algún dispositivo móvil (celular, tableta, etc.) []
 - d) En una aplicación de Internet (Gmail, Outlook, etc.) []
 - e) Poseo un sistema especializado instalado en una computadora []
 - f) Otros, indique: _____

En relación a la información de los expedientes/casos que lleva:

2. ¿De qué forma organiza/ordena/guarda toda la información que se genera?
 - a) De forma manual/escrita []
 - b) En la computadora/otros dispositivos (documentos en Word, etc.) []
 - c) En una aplicación de Internet (Gmail, Outlook, etc.) []
 - d) Poseo un sistema especializado instalado en una computadora []
 - e) Servicio de secretaria contratado []
 - f) Otros, indique: _____
3. ¿De qué forma registra o lleva control de todas las actividades/gestiones/diligencias/actuaciones realizadas de un expediente/caso?
 - a) De forma manual/escrita []
 - b) En la computadora/otros dispositivos (documentos en Word, notas, etc.) []
 - c) En una aplicación de Internet (Gmail, Outlook, etc.) []
 - d) Poseo un sistema especializado instalado en una computadora []
 - e) Servicio de secretaria contratado []
 - f) Otros, indique: _____
4. ¿Cómo establece Ud. el costo de sus servicios a sus clientes?
 - a) Asigno un costo global y único dependiendo del expediente/caso []
 - b) Asigno un costo a la actividad/diligencia/actuación []
 - c) Mezcla entre los dos primeros []
 - d) Otros, indique: _____

En relación a los documentos de un expediente/caso:

5. ¿De qué forma organiza/ordena/guarda los documentos generados/producidos de un expediente/caso?
- a) De forma manual []
 - b) En la computadora (carpetas, documentos en Word, notas, etc.) []
 - c) Poseo un sistema especializado instalado en una computadora []
 - d) Servicio de secretaria contratado []
 - e) Otros, indique: _____
6. ¿En dónde almacena la mayoría de los documentos digitales?
- a) En la computadora []
 - b) En dispositivos de almacenamiento []
 - c) En internet []
 - d) Otros, indique: _____
7. ¿Posee respaldo de esos documentos?
- a) Si []
 - b) No []
8. ¿Realiza usted almacenamiento ordenado y organizado de plantillas de documentos/diligencias/formatos (Word, pdf, notas, etc.)?
- a) Si []
 - b) No []

En relación a la información de calendarios de tribunales, registros, notarias, entes en general:

9. ¿Lleva usted en algún calendario los días laborables, de despacho y no despacho de los tribunales y de qué forma?
- a) Sí, de forma manual []
 - b) Sí, me apoyo con un sistema []
 - c) No []

En relación a las aplicaciones de software de apoyo a la gestión jurídica:

10. ¿Le gustaría evaluar una aplicación web gratuita que apoye sus actividades diarias con bondades para administrar la información de: clientes, documentos de sus expedientes, actuaciones, costo de sus servicios, plantillas, agenda y calendario?
- a) Si []
 - b) No []

Tabla 2: Encuesta para levantar información de funcionalidades de la aplicación web

Análisis de Encuesta

De las respuestas se obtuvo que:

- Los abogados realizan en su mayoría un manejo manual de la información de sus clientes.
- La mayoría de los abogados controlan la información generada por y de sus expedientes, de forma manual, a la vez que se apoyan de herramientas ofimáticas. Ninguno cuenta con un sistema automatizado que lo apoye en su gestión.
- Los abogados no están familiarizados con sistemas especializados que apoyen su gestión y en su mayoría el trabajo es manual o se apoyan con herramientas ofimáticas.
- Los abogados manejan una agenda de actividades, pero en su mayoría es una agenda física, es decir, de forma manual.
- La mayoría de los abogados fijan el precio de sus servicios de forma manual y la mayoría lo hace en base a un precio único por caso, ya que se acuerda con el cliente un porcentaje del monto demandado o un monto único.
- Los abogados en su mayoría poseen una computadora en un sitio físico en la que graban toda su información y es raro el uso de un repositorio en la nube o dispositivos de almacenamiento auxiliar.
- En su mayoría los abogados no poseen respaldo de los documentos y los mismos están por lo general en su computadora.
- Los abogados consideran que es importante mantener plantillas organizadas ya que los documentos legales muchas veces tienden a ser repetitivos.
- Todos llevan un control manual de los días laborables de los tribunales de su interés en agendas físicas.
- Los abogados sienten que existe una necesidad de apoyarse en un sistema que les brinde control sobre sus documentos y precios de los mismos.

Perfil de usuario

La aplicación web está dirigida a usuarios que sean abogados litigantes que trabajan de forma independiente en Venezuela, quienes no necesitan de un conocimiento avanzado en el uso de sistemas, pero deben conocer los fundamentos de trabajar con herramientas de ofimática e interactuar con páginas en Internet.

El objetivo de la aplicación es llegar únicamente a los usuarios finales, es decir, los abogados. La administración de la misma no es objeto de estudio del presente trabajo.

Lista de requerimientos funcionales y no funcionales

La recolección de toda la información previa a esta sección, da lugar a las premisas que deben tomarse en cuenta para construir la aplicación propuesta.

Requerimientos funcionales

Manejo de cuentas independientes para cada uno de los abogados que usen la aplicación web, administrar información de clientes de los abogados, cargar documentos de los casos, cargar y controlar información producida en un caso, registrar actividades, diligencias, actuaciones del caso, visualización y carga de citas en una agenda, mantener un registro de los pagos y montos que se cobran a documentos y en general de los gastos de los casos, crear plantillas, registrar información de organismos y personas contactos, almacenar documentos de forma ordenada y organizada. Contar con un calendario donde se puedan marcar las fechas destacadas de organismos.

Requerimientos no funcionales

De fácil uso, que no se requiera de conocimientos avanzados para el manejo. Uso de colores pasteles, agradables y armoniosos. Estilo de letras simple, con uso de negritas para resaltar. Íconos y botones de tamaño mediano, no muy pequeños.

4.7 Etapa 2: Análisis

Para el desarrollo de la solución se elaboraron los casos de uso, el modelo de objeto de dominio, el prototipo en papel y los patrones de interacción de la interfaz de usuario.

Modelo de Casos de Uso

Los modelos de caso de uso definen la secuencia de las interacciones del sistema con los actores. En el presente proyecto el actor principal es el abogado, el cual tiene la posibilidad de operar con todas las funciones de la aplicación web.

El diagrama completo del modelo de casos de uso, se compuso de 3 niveles. En el nivel 0, el usuario interactúa con la aplicación web como un todo. El nivel 1 muestra la interacción del actor con los módulos de la aplicación web. Y el nivel 2, muestra la interacción del actor con cada uno de los módulos de la aplicación.

A continuación se muestran los diagramas de cada uno de los niveles y a continuación su respectiva tabla de descripción.

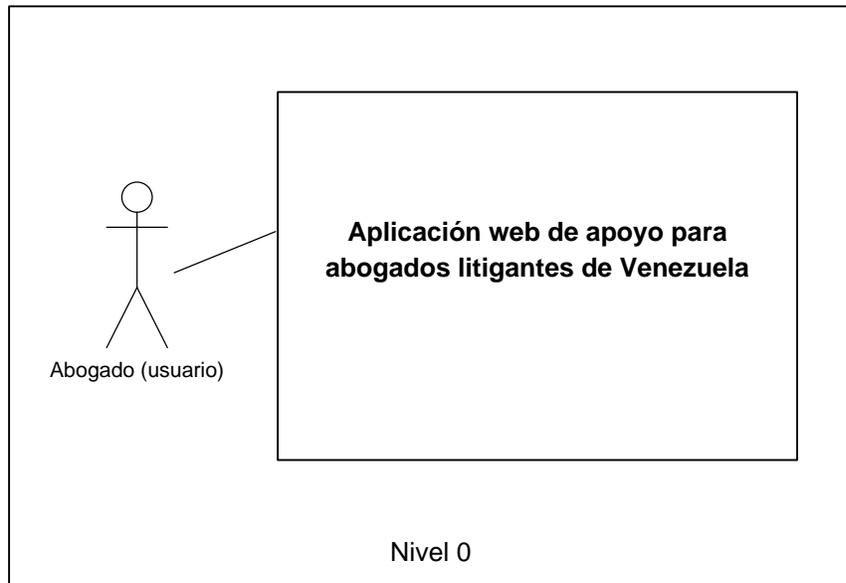


Figura 7: Diagrama caso de uso nivel 0

Nombre	<u>Aplicación web de apoyo para abogados litigantes en Venezuela</u>
Actor	Abogado
Acción	Permite al abogado interactuar con la aplicación y todos sus módulos.
Precondición	El abogado debe estar registrado en la aplicación y haber iniciado sesión.
Post condición	El abogado tiene una sesión iniciada y puede acceder a cualquiera de los módulos de la aplicación.
Nivel	0

Tabla 3: Caso de uso nivel 0 – Aplicación web de apoyo para abogados litigantes en Venezuela

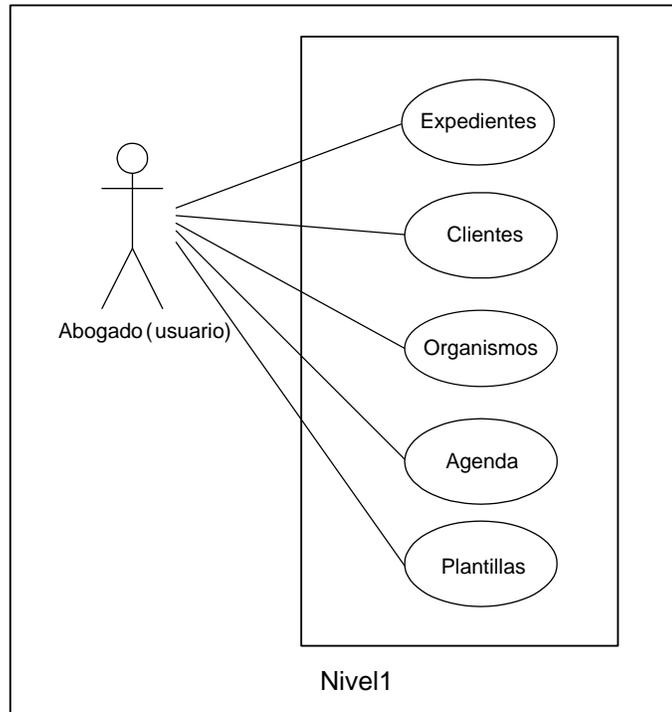


Figura 8: Diagrama caso de uso nivel 1

Nombre	<u>Expedientes</u>
Actor	Abogado
Acción	Permite al abogado interactuar con el módulo Expedientes, en el cual se muestra, registra y modifica la información y documentos de los expedientes que lleva el abogado.
Precondición	El abogado debe haber iniciado sesión en la aplicación.
Post condición	El abogado tendrá acceso a la información y documentos de los expedientes y actuaciones, así como modificarla o eliminarla.
Nivel	1

Tabla 4: Caso de uso nivel 1 – Expedientes

Nombre	<u>Clientes</u>
Actor	Abogado
Acción	Permite al abogado interactuar con el módulo Clientes, en el cual se muestra, registra y modifica la información de todos los clientes que lleva el abogado.
Precondición	El abogado debe haber iniciado sesión en la aplicación.

Post condición	El abogado tendrá acceso a la información de los clientes que lleva, así como modificarla o eliminarla.
Nivel	1

Tabla 5: Caso de uso nivel 1 - Clientes

Nombre	<u>Agenda</u>
Actor	Abogado
Acción	Permite al abogado interactuar con el módulo Agenda, en el cual se muestran, registran y se modifican las citas.
Precondición	El abogado debe haber iniciado sesión en la aplicación.
Post condición	El abogado tendrá acceso a la información de la agenda así como modificarla o eliminarla.
Nivel	1

Tabla 6: Caso de uso nivel 1 - Agenda

Nombre	<u>Organismos</u>
Actor	Abogado
Acción	Permite al abogado interactuar con el módulo Organismos, en el cual se muestra, registra y se modifica la información de los organismos con los que interactúa.
Precondición	El abogado debe haber iniciado sesión en la aplicación.
Post condición	El abogado tendrá acceso a la información de los organismos con los que interactúa, así como modificarla o eliminarla.
Nivel	1

Tabla 7: caso de uso nivel 1 - Organismos

Nombre	<u>Plantillas</u>
Actor	Abogado
Acción	Permite al abogado interactuar con el módulo Organismos, en el cual se muestra, registra y se modifica la información de los organismos con los que interactúa.
Precondición	El abogado debe haber iniciado sesión en la aplicación.
Post condición	El abogado tendrá acceso a la información de los organismos con los que interactúa, así como modificarla o eliminarla.

Nivel	1
-------	---

Tabla 8: Caso de uso nivel 1 - Plantillas

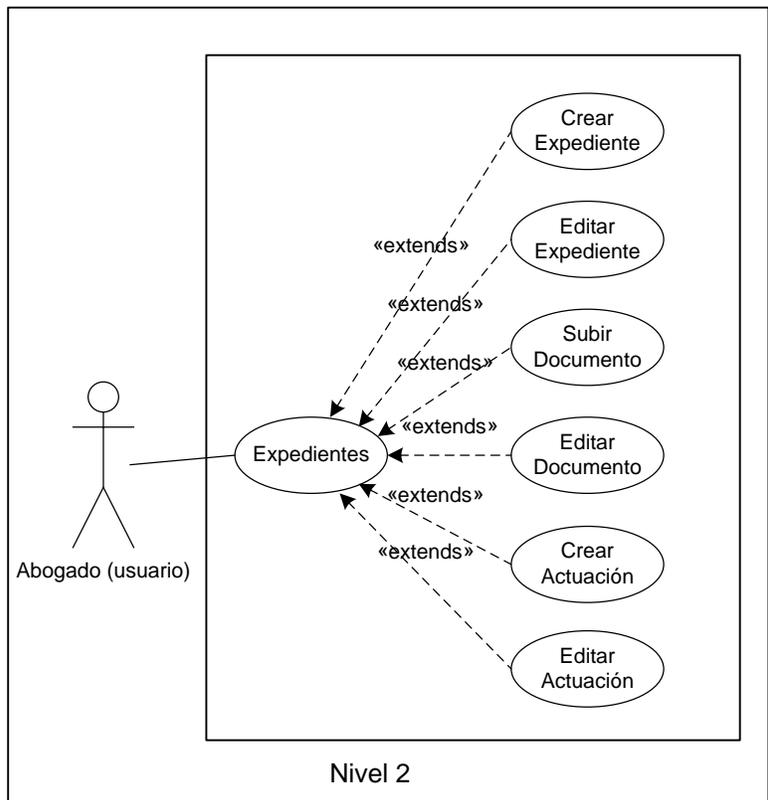


Figura 9: Diagrama de caso de uso Expedientes, nivel 2

Nombre	Crear Expediente
Actor	Abogado
Acción	Permite al abogado registrar los datos asociados a un expediente.
Precondición	Haber iniciado sesión en la aplicación, haber accedido al módulo de Expedientes y haber hecho clic en el botón de agregar un nuevo expediente.
Post condición	La aplicación debe mostrar un mensaje que indica que el expediente se creó satisfactoriamente.
Nivel	2

Tabla 9: Caso de uso nivel 2 – Crear expediente

Nombre	<u>Editar Expediente</u>
Actor	Abogado
Acción	Permite al abogado modificar o eliminar un expediente.
Precondición	El abogado debe haber ingresado en la aplicación, haber accedido al módulo de Expedientes y al expediente que se desea modificar.
Post condición	La aplicación debe mostrar un mensaje indicando que se guardaron los cambios.
Nivel	2

Tabla 10: Caso de uso nivel 2 – Editar expediente

Nombre	<u>Subir Documento</u>
Actor	Abogado
Acción	Permite al abogado agregar un documento dentro de un expediente.
Precondición	El abogado debe haber ingresado en la aplicación, haber accedido al módulo de Expedientes y haber hecho clic en el botón que permite agregar un documento al expediente.
Post condición	La aplicación debe mostrar un mensaje indicando que se guardaron los cambios.
Nivel	2

Tabla 11: Caso de Uso 2 – Subir documento

Nombre	<u>Editar Documento</u>
Actor	Abogado
Acción	Permite al abogado modificar o eliminar un documento agregado a un expediente.
Precondición	El abogado debe haber ingresado en la aplicación, haber accedido al módulo de Expedientes y al expediente donde desea modificar el documento.
Post condición	La aplicación debe mostrar un mensaje indicando que se guardaron los cambios.
Nivel	2

Tabla 12: Caso de uso nivel 2 – Editar documento

Nombre	<u>Crear Actuación</u>
Actor	Abogado
Acción	Permite al abogado agregar una actuación a un expediente.
Precondición	El abogado debe haber ingresado en la aplicación, haber accedido al módulo de Expedientes y al expediente donde desea agregar la actuación.
Post condición	La aplicación debe mostrar un mensaje indicando que se guardaron los cambios.
Nivel	2

Tabla 13: caso de uso nivel 2 – Crear actuación

Nombre	<u>Editar Actuación</u>
Actor	Abogado
Acción	Permite al abogado modificar o eliminar una actuación.
Precondición	El abogado debe haber ingresado en la aplicación, haber accedido al módulo de Expedientes y al expediente donde desea modificar la actuación.
Post condición	La aplicación debe mostrar un mensaje indicando que se guardaron los cambios.
Nivel	2

Tabla 14: Caso de uso nivel 2 – Editar actuación

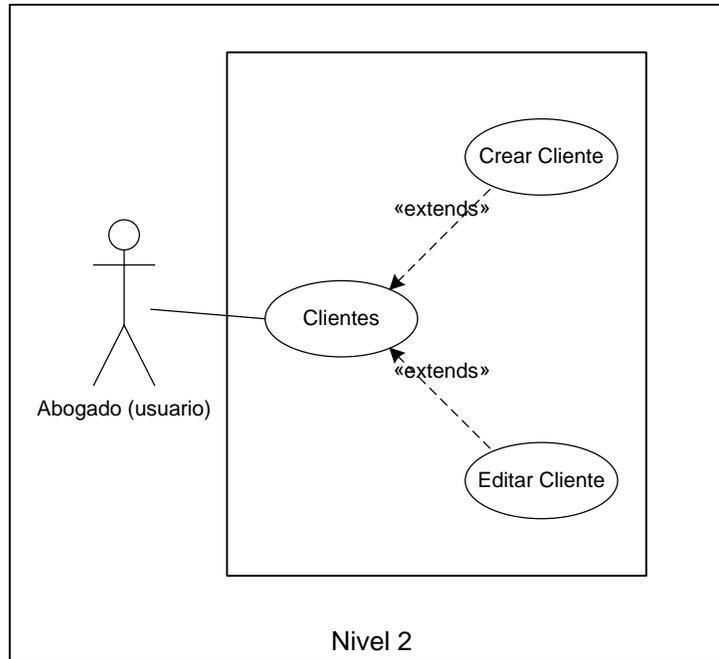


Figura 10: Diagrama caso de uso Clientes, nivel 2

Nombre	<u>Crear Cliente</u>
Actor	Abogado
Acción	Permite al abogado agregar un nuevo cliente, de tipo jurídico o natural a la aplicación.
Precondición	Haber iniciado sesión en el sistema, haber accedido al módulo de Clientes y haber hecho clic en el botón de agregar un nuevo cliente.
Post condición	La aplicación debe mostrar un mensaje que indica que el cliente se creó satisfactoriamente.
Nivel	2

Tabla 15: Caso de uso nivel 2 – Crear Cliente

Nombre	<u>Editar Cliente</u>
Actor	Abogado
Acción	Permite al abogado modificar o eliminar un cliente
Precondición	El abogado debe haber ingresado en la aplicación, haber accedido al módulo de Clientes y haber hecho clic en el botón/ícono de editar.

Post condición	La aplicación debe mostrar un mensaje indicando que se guardaron los cambios.
Nivel	2

Tabla 16: Caso de uso nivel 2 – Editar Cliente

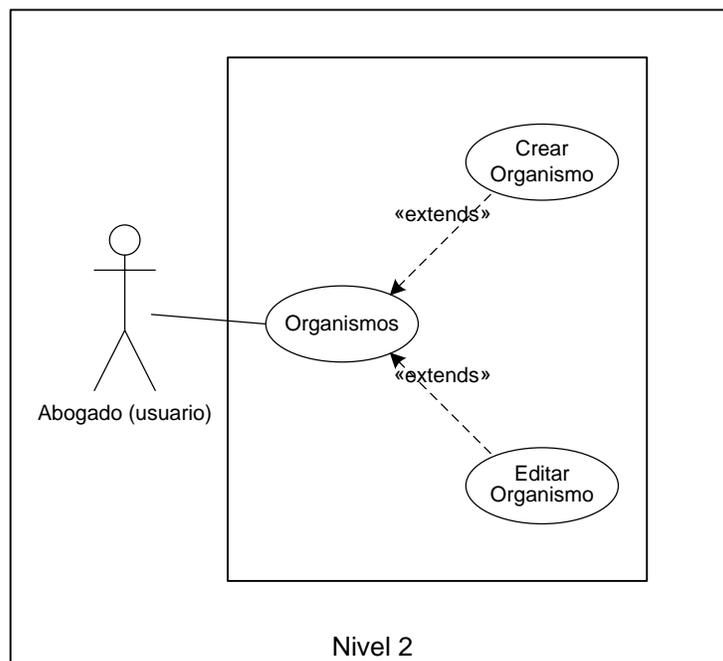


Figura 11: Diagrama caso de uso Organismos, nivel 2

Nombre	<u>Crear Organismo</u>
Actor	Abogado
Acción	Permite al abogado agregar un nuevo organismo a la aplicación.
Precondición	Haber iniciado sesión en la aplicación, haber accedido al módulo de Organismos y haber hecho clic en el botón de agregar un nuevo organismo.
Post condición	La aplicación debe mostrar un mensaje que indica que el organismo se creó satisfactoriamente.
Nivel	2

Tabla 17: Caso de uso nivel 2 – Crear organismo

Nombre	<u>Editar Organismo</u>
Actor	Abogado
Acción	Permite al abogado modificar o eliminar un organismo.
Precondición	El abogado debe haber ingresado en la aplicación, haber accedido al módulo de Organismos y haber hecho clic en el botón/ícono de editar.
Post condición	La aplicación debe mostrar un mensaje indicando que se guardaron los cambios.
Nivel	2

Tabla 18: Caso de uso nivel 2 – Editar organismo

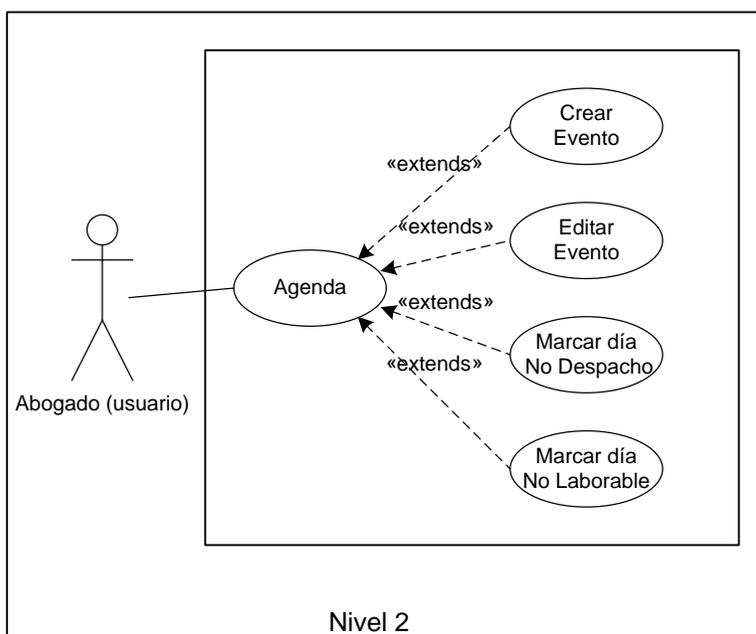


Figura 12: Diagrama caso de uso Agenda, nivel 2

Nombre	<u>Crear Evento</u>
Actor	Abogado
Acción	Permite al abogado registrar un evento como una cita, una reunión, entre otros, con fecha y hora.
Precondición	Haber iniciado sesión en la aplicación, haber accedido al módulo de Agenda y haber hecho clic en el botón de agregar un nuevo evento.

Post condición	La aplicación debe mostrar un mensaje que indica que el evento se creó satisfactoriamente.
Nivel	2

Tabla 19: caso de uso nivel 2 – Crear evento

Nombre	<u>Editar Evento</u>
Actor	Abogado
Acción	Permite al abogado modificar o eliminar un evento.
Precondición	Haber iniciado sesión en la aplicación, haber accedido al módulo de Agenda y haber hecho clic en el botón de agregar un nuevo evento.
Post condición	La aplicación debe mostrar un mensaje indicando que se guardaron los cambios.
Nivel	2

Tabla 20: Caso de uso nivel 2 – Editar evento

Nombre	<u>Marcar día no laborable</u>
Actor	Abogado
Acción	Permite al abogado marcar un día del calendario como no laborable.
Precondición	Haber iniciado sesión en la aplicación, haber accedido al módulo de Agenda y haber hecho clic en el botón de marcar día como no laborable.
Post condición	La aplicación debe mostrar un mensaje indicando que se guardaron los cambios.
Nivel	2

Tabla 21: Caso de uso nivel 2 – Marcar día no laborable

Nombre	<u>Marcar día no despacho</u>
Actor	Abogado
Acción	Permite al abogado marcar un día del calendario como no despacho.

Precondición	Haber iniciado sesión en la aplicación, haber accedido al módulo de Agenda y haber hecho clic en el botón de marcar día como no despacho.
Post condición	La aplicación debe mostrar un mensaje indicando que se guardaron los cambios.
Nivel	2

Tabla 22: caso de uso nivel 2 – Marcar día no despacho

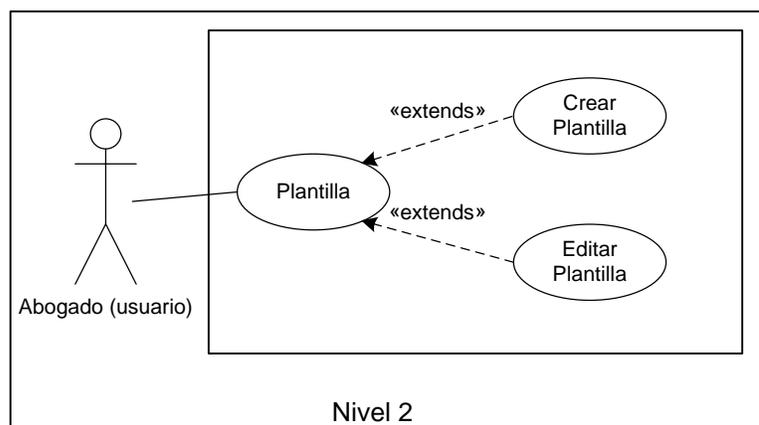


Figura 13: Diagrama caso de uso Plantillas, nivel 2

Nombre	<u>Crear Plantilla</u>
Actor	Abogado
Acción	Permite al abogado crear una plantilla de documento.
Precondición	Haber iniciado sesión en la aplicación, haber accedido al módulo de Plantillas y haber hecho clic en el botón de agregar un nuevo plantilla.
Post condición	La aplicación debe mostrar un mensaje que indica que la plantilla se creó satisfactoriamente.
Nivel	2

Tabla 23: Caso de uso nivel 2 – Crear plantilla

Nombre	<u>Editar Plantilla</u>
Actor	Abogado
Acción	Permite al abogado modificar o eliminar una plantilla.

Precondición	El abogado debe haber ingresado en la aplicación, haber accedido al módulo de Plantillas y haber hecho clic en el botón/ícono de editar.
Post condición	La aplicación debe mostrar un mensaje indicando que se guardaron los cambios.
Nivel	2

Tabla 24. Caso de uso nivel 2 – Editar plantilla

Modelo Objetos del Dominio

Un modelo objeto del dominio describe los objetos del dominio de la aplicación y las relaciones estáticas entre estos. Captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las "cosas" que existen o los eventos que suceden en el entorno en el que trabaja el sistema. El objetivo del modelado del dominio es comprender y describir las clases más importantes dentro del contexto del sistema.

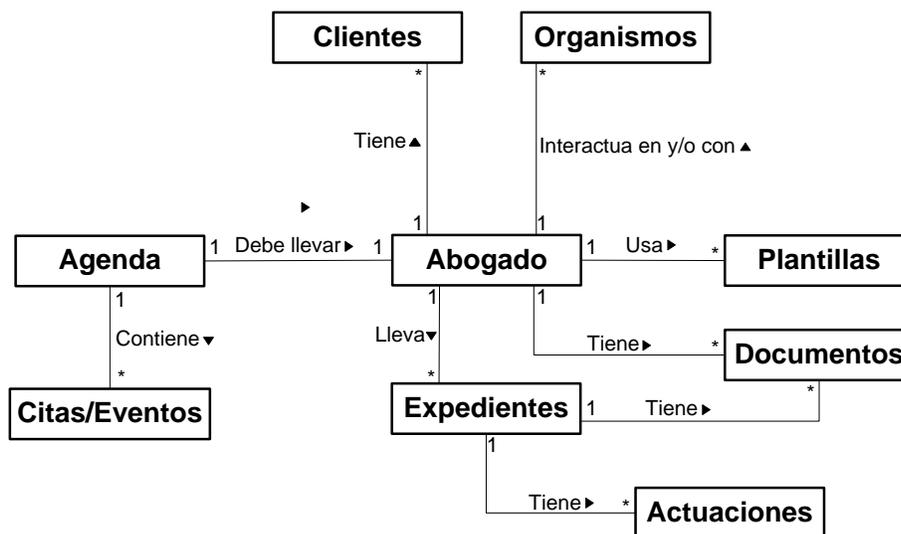


Figura 14: Modelo Objeto del Dominio de la gestión de un abogado litigante

Los objetos de dominio se detallan a continuación

- **Abogado:** Persona legalmente autorizada para asesorar y defender los derechos e intereses de otra persona en materia jurídica y representarla en un pleito.
- **Clientes:** Persona que requiere de los servicios de un abogado.

- **Expediente:** Serie de procedimientos de carácter judicial o administrativo. También se refiere al conjunto de los documentos que corresponden a un determinado caso.
- **Actuación:** Cualquier actividad de valor legal que realiza el abogado en la defensa de los intereses de sus clientes.
- **Documento:** Todo escrito de valor legal realizado por el abogado, producido en el caso, público o privado. Por ejemplo: Documento de compra y venta, testamento, títulos de propiedad, libelo, etc.
- **Agenda:** Es un organizador del tiempo ocupado y disponible del abogado.
- **Evento:** Se refiere a citas, compromisos que tiene el abogado los cuales se les puede añadir lugar, fecha y hora.
- **Organismos:** Todo ente con el que el abogado tiene interactuar. Por ejemplo: Tribunales, Juzgados, Notarias, Registros, Institutos, Ministerios, entre otros.
- **Plantilla:** Es un escrito con los formatos y contenido preestablecidos para un determinado fin. Por ejemplo: una diligencia, que es un escrito corto que el abogado debe componer para solicitar algo en un organismo.

La relación existente entre todos los objetos del dominio puede describirse de la siguiente forma:

Un abogado tiene clientes a quienes presta sus servicios. Dependiendo del servicio, el cual puede ser desde un simple documento hasta una representación en un proceso de un juicio ante un organismo. De dicho proceso se crea un expediente en un organismo, el cual contiene actuaciones y otros documentos generados y producidos en el mismo. Para crear algún documento o solicitar ante los organismos otros documentos, y en vista de que mucha información se puede reusar, los abogados se valen de plantillas que contienen texto ya formateado, ahorrándose así tiempo en componer el documento desde cero. Una herramienta de alta utilidad para los abogados es la agenda. En ella los abogados pueden llevar un control de eventos o citas que requieran recordar, agregando fecha, hora y lugar.

Prototipo en papel

Los distintos prototipos iniciales de las interfaces de usuario que se realizaron en papel son:

- **Ingreso a la aplicación web:** Para acceder a la aplicación web, se diseña una página de inicio de sesión, en la cual se solicitará el nombre de usuario y clave.

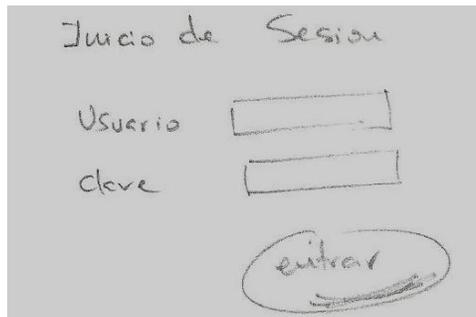


Figura 15: Prototipo en papel del inicio de sesión del sistema

- Pantalla de inicio de la aplicación web:** La pantalla de inicio presentará la información asociada al usuario que inició sesión en la aplicación. Se mostrará del lado derecho una imagen referencial al Derecho. Habrá un menú horizontal que provee acceso a los distintos módulos. En el lado derecho habrá un panel con un calendario. Dependiendo del módulo al que se acceda, se muestra un listado de los elementos que la componen. Por ejemplo, si se está en el módulo de expedientes, se mostrará lista de todos los expedientes. En la parte superior derecha se podrá salir de la aplicación con el botón Salir.

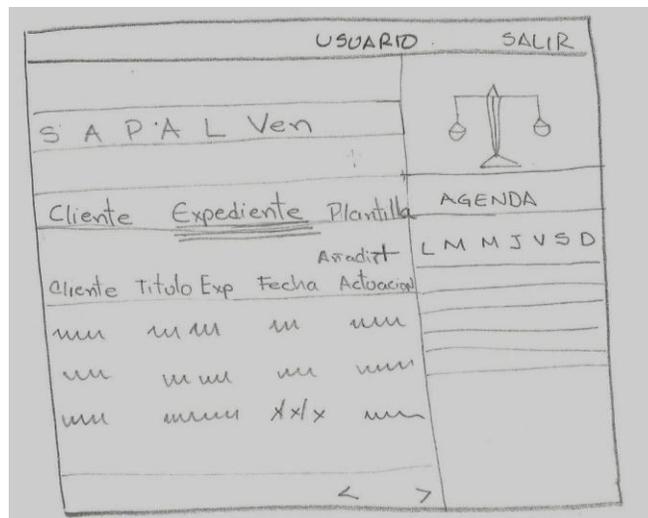


Figura 16: Prototipo en papel de pantalla de inicio de la aplicación

- Pantalla del módulo de Clientes:** Se mantiene lo definido en la pantalla de inicio de la aplicación, pero se muestra la lista de los clientes, además de sus respectivas operaciones.

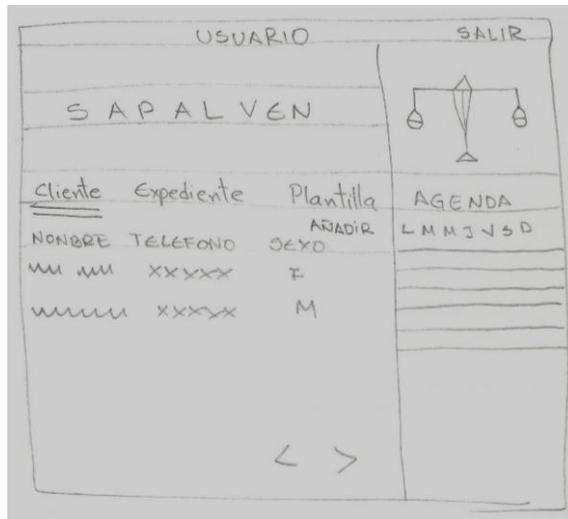


Figura 17: Prototipo en papel de pantalla del módulo de Clientes

- **Pantallas de Ingreso de Información:** Se mostrarán los formularios en ventanas flotantes que contendrá dos columnas de campos con sus respectivas etiquetas.
 - **Cliente Natural**

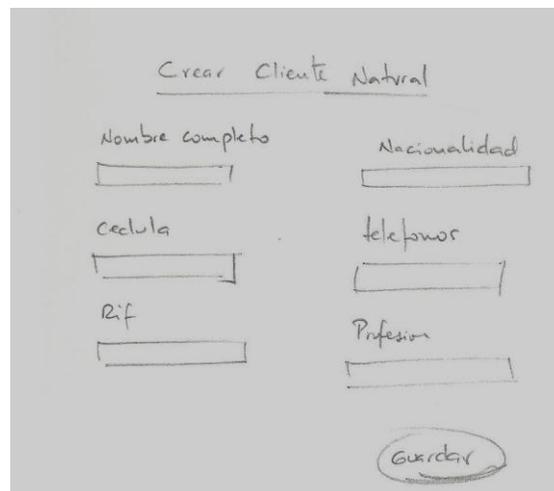


Figura 18: Prototipo en papel de pantalla de ingreso de información de cliente natural

- **Cliente Jurídico**

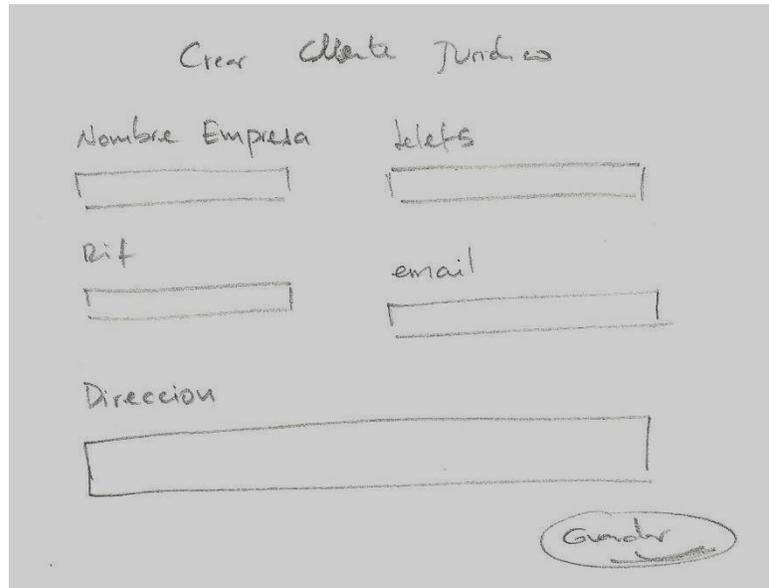


Figura 19: Prototipo en papel de pantalla de ingreso de información de cliente jurídico

Patrones de interacción

Se refiere a la forma en que la aplicación da solución a los problemas de interacción frecuentes en la interfaz de usuario en un lugar y contexto determinado.

La interfaz de la aplicación web implica una gran variedad de interacciones de parte del abogado (usuario), lo cual determina los patrones de búsqueda. Se identificaron los siguientes patrones: registrar al usuario, crear un cliente, editar un cliente, consultar listado de clientes, crear un expediente, editar un expediente, consultar listado de expedientes, subir un documento al expediente, modificar un documento, listar los documentos, agregar una actuación, modificar una actuación, listar las actuaciones, agregar un evento a la agenda, marcar un día no laborable, marcar un día de no despacho, subir una plantilla, crear una plantilla.

A continuación se describirán los patrones de interacción de la creación de un expediente, subir un documento, agregar actuación y consulta de listado de clientes.

Nombre	<u>Creación de Expediente</u>
Problema	El abogado necesita agregar un expediente en la aplicación.

Solución	<p>La aplicación tiene la funcionalidad de agregar un expediente en la pantalla principal del módulo de Expedientes. El patrón de interacción es el siguiente:</p> <ul style="list-style-type: none"> ▪ Se situará un botón de Agregar junto al listado de expedientes. ▪ Es necesario llenar sólo los campos requeridos y presionar el botón guardar. ▪ Una vez guardada la información del expediente presione el botón el enlace que dice que no requiere crear más expedientes.
Contexto	Aplicación web de apoyo para abogados litigantes en Venezuela, módulo de Expedientes.
Utilidad	Lograr que el abogado registre los datos del expediente.
Fuerzas	El usuario debe conocer como interactuar con aplicaciones Internet.
Consecuencias	<ul style="list-style-type: none"> ▪ El usuario puede crear otro expediente. ▪ El usuario visualiza el expediente creado. ▪ El usuario puede acceder a cualquier otro módulo de la aplicación.
Patrones relacionados	Ninguno.

Tabla 25: Patrón de interacción – Creación de expediente

Nombre	<u>Subir Documento</u>
Problema	El abogado necesita agregar un documento a un expediente.
Solución	<p>La aplicación web de apoyo para abogados litigantes en Venezuela provee la capacidad de subir un documento luego de creado un expediente.</p> <p>Se situará un ícono para agregar un documento a cada uno de los expedientes listados.</p> <p>Se deberán agregar los datos del documento y utilizar el botón guardar.</p>
Contexto	Aplicación web de apoyo para abogados litigantes en Venezuela, módulo de Expedientes.
Utilidad	Lograr que el abogado guarde en digital los documentos asociados al expediente.

Fuerzas	El usuario debe conocer como interactuar con aplicaciones Internet.
Consecuencias	La aplicación web indicará que el documento se cargó con éxito y se dejará abierta la posibilidad para subir otro.
Patrones relacionados	Creación de Expediente.

Tabla 26: Patrón de interacción – Subir documento

Nombre	<u>Agregar Actuación</u>
Problema	El abogado necesita agregar una actuación a un expediente.
Solución	La aplicación web de apoyo para abogados litigantes en Venezuela provee la capacidad de agregar una actuación luego de creado un expediente. Se situará un ícono para agregar una actuación a cada uno de los expedientes listados. Se deberán agregar los datos de la actuación y utilizar el botón guardar.
Contexto	Aplicación web de apoyo para abogados litigantes en Venezuela, módulo de Expedientes.
Utilidad	Lograr que el abogado registre los datos de la actuación asociada al expediente.
Fuerzas	El usuario debe conocer como interactuar con aplicaciones Internet.
Consecuencias	La aplicación web indicará que el documento se guardó con éxito la actuación y se dejará abierta la posibilidad para subir otro.
Patrones relacionados	Creación de Expediente.

Tabla 27: Patrón de interacción – Agregar actuación

Nombre	<u>Listado de Clientes</u>
Problema	El abogado necesita ver los clientes.
Solución	La aplicación web de apoyo para abogados litigantes en Venezuela provee la capacidad de mostrar el listado de clientes. El listado se situará en el módulo de Clientes en una tabla con capacidad de navegación para aquellos casos en donde la lista es muy grande. En dicha tabla se podrá ordenar según los

	campos que la componen y se agregará un campo de texto para filtrar en caso de búsquedas.
Contexto	Aplicación web de apoyo para abogados litigantes en Venezuela, módulo de Clientes.
Utilidad	Lograr que el abogado visualice la lista de los clientes con algunos datos como: identificación, nombres y apellidos, email y teléfonos.
Fuerzas	El usuario debe conocer como interactuar con aplicaciones Internet.
Consecuencias	Se muestra la lista de clientes y el usuario puede ejecutar alguna operación sobre alguno de los clientes o acceder a otro módulo.
Patrones relacionados	Creación de Cliente.

Tabla 28: Patrón de interacción – Listado de clientes

Guía de estilo

Definir las guías de estilo y adoptarlas a lo largo del desarrollo de la aplicación web, facilita enormemente el diseño de las interfaces de usuario.

Los acuerdos definidos en cuanto a la Interfaz de usuarios fueron:

- Colores primarios de la aplicación serán: blanco, gris y azul, con excepciones.
- La tipografía es de la familia: Verdana, Arial, sans-serif.
- Los botones serán grises y azules, dependiendo de la operación.



Figura 20: Botones de la aplicación web

- Los íconos de las operaciones disponibles para los objetos, son negros para distinguirlos de cualquier otra operación.



Figura 21: Íconos de operaciones de la aplicación web

- Al situar el ratón sobre algunos de los íconos se deben mostrar tooltips, que son descripciones que despliegan bajo el puntero del mouse sobre un objeto.
- El menú de la aplicación se repetirá en todas y cada una de las páginas de la aplicación.



Figura 22: Menú de la aplicación web

- Las operaciones de agregar datos a objetos, se caracterizarán por separarse de la aplicación en una ventana flotante que aparece en primer plano, dejando en segundo plano la aplicación.

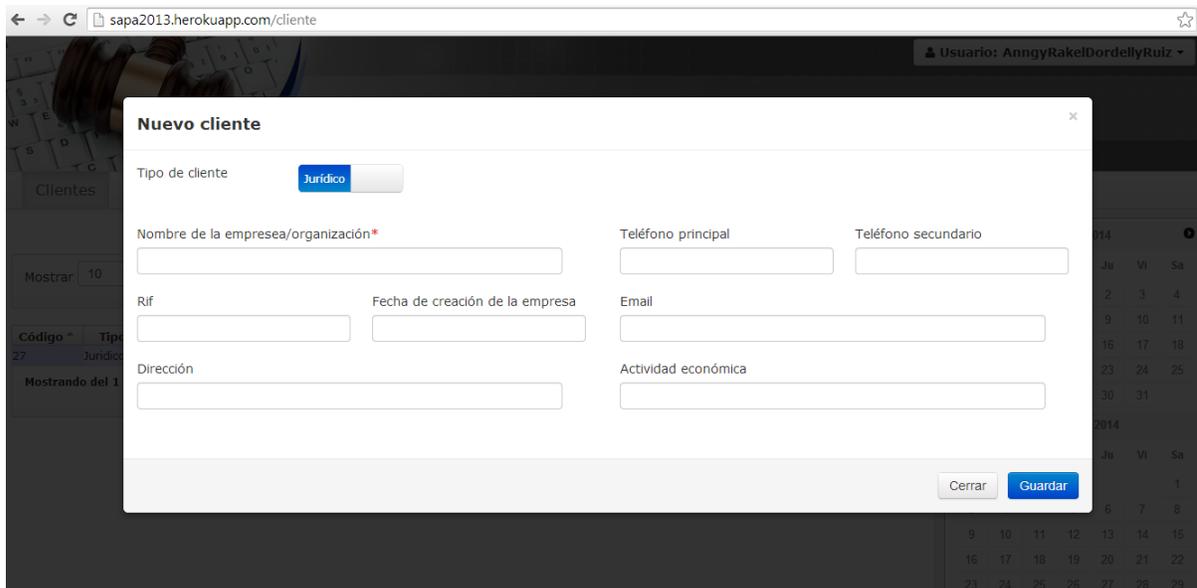


Figura 23: Ventana flotante de la aplicación web

4.8 Etapa 3: Prototipaje

En esta etapa se produjo un prototipo ejecutable y se utilizó la evaluación heurística para determinar problemas.

Prototipo Ejecutable

El prototipo ejecutable se puede apreciar en la próxima figura.

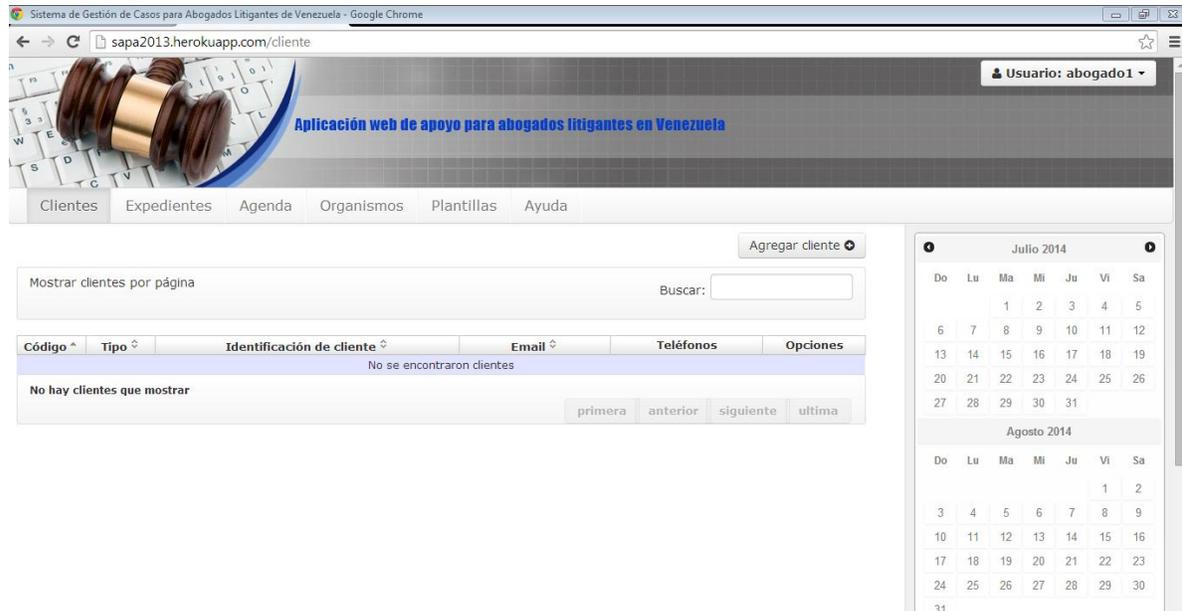


Figura 24: Pantalla de prototipo ejecutable de la aplicación

Evaluación Heurística

La evaluación heurística es un método que consta de 10 conceptos a los cuales se le asigna un valor en una escala del 0 a 4, donde 0 es menos importante y 4 catastrófico, con los cuales se evalúan distintos problemas de usabilidad en el diseño de interfaz de usuario, tal que pueden ser incluidas en el proceso de prototipaje. Estos conceptos fueron puntualizados por Nielsen [64].

Heurísticas de Nielsen:

1. **Visibilidad del estado del sistema.** El sistema debe informar a los usuarios del estado del sistema, dando una retroalimentación apropiada en un tiempo razonable.
2. **Utilizar el lenguaje de los usuarios.** El sistema debe utilizar el lenguaje de los usuarios, con palabras o frases que le sean conocidas, en lugar de los términos que se utilizan en el sistema, para que al usuario no se le dificulte utilizar el sistema.

3. **Control y libertad para el usuario.** En casos en los que los usuarios elijan una opción del sistema por error, éste debe contar con las opciones de deshacer y rehacer para proveer al usuario de una salida fácil sin tener que utilizar diálogo extendido.
4. **Consistencia y estándares.** El usuario debe seguir las normas y convenciones de la plataforma sobre la que está implementando el sistema, para que no se tenga que preguntar el significado de las palabras, situaciones o acciones del sistema.
5. **Prevención de errores.** Es más importante prevenir la aparición de errores que generar buenos mensajes de error. Hay que eliminar acciones predispuestas al error o, en todo caso, localizarlas y preguntar al usuario si está seguro de realizarlas.
6. **Minimizar la carga de la memoria del usuario.** El sistema debe minimizar la información que el usuario debe recordar mostrándola a través de objetos, acciones u opciones. El usuario no tiene por qué recordar la información que recibió anteriormente. Las instrucciones para el uso del sistema deberían ser visibles o estar al alcance del usuario cuando se requieran.
7. **Flexibilidad y eficiencia de uso.** Los aceleradores permiten aumentar la velocidad de interacción para el usuario experto tal que el sistema pueda atraer a usuarios principiantes y experimentados. Es importante que el sistema permita personalizar acciones frecuentes para así acelerar el uso de éste.
8. **Diálogos estéticos y diseño minimalista.** La interfaz no debe contener información que no sea relevante o se utilice raramente, pues cada unidad adicional de información en un diálogo compite con las unidades relevantes de la información y disminuye su visibilidad relativa.
9. **Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de los errores.** Los mensajes de error deben expresarse en un lenguaje claro, indicar exactamente el problema y ser constructivos.
10. **Ayuda y documentación.** A pesar de que es mejor un sistema que no necesite documentación, puede ser necesario disponer de ésta. Si así es, la documentación tiene que ser fácil de encontrar, estar centrada en las tareas del usuario, tener información de las etapas a realizar y no ser muy extensa [64].

La evaluación heurística del prototipo ejecutable se realizó a abogados dispuestos a probar el aplicativo, los mismos a los que se les aplicó la encuesta. Se realizó de forma grupal y se sugirieron en ese mismo momento las soluciones.

Problema	Heurística	Valoración	Solución
El calendario común a todos los módulos está en inglés.	H2	4	Colocar el calendario en español.
Los ayudantes para seleccionar fechas están en inglés.	H2	4	Colocar los ayudantes de fechas en español.
Información en la página de bienvenida de la aplicación innecesaria.	H8	1	Reemplazar el área con los campos de inicio de sesión.
Los botones y campos de inicio de sesión son muy pequeños.	H1	2	Agrandar los botones y campos, y hacerlos más llamativos.
El título del sistema no debería estar en gris, ni estar en doble línea.	H8	0	Cambiar el color del título por azul y dejarlo en una sola línea.
En los formularios de cliente, el botón del tipo de Cliente no debe estar a la derecha.	H4, H8	1	Mover el botón del tipo de Cliente a la izquierda.
En los formularios de cliente natural, la etiqueta del campo dirección está mal.	H6, H2	0	Cambiar la etiqueta del campo dirección, por "Dirección de residencia".
En los formularios de cliente natural, no existe un campo para la dirección del empleador.	H6	2	Se debe agregar un campo adicional para la "Dirección Laboral" del cliente.
En los formularios de cliente natural varias etiquetas están mal.	H6, H2	0	Cambiar las etiquetas: Nombre por Nombres, Apellido por Apellidos.
En los formularios de cliente natural, no hay un campo para el lugar de nacimiento.	H6	2	Se debe agregar un campo para el lugar de nacimiento que sea una lista de países.

En los formularios de cliente natural la etiqueta actividad económica no aplica a este tipo de cliente.	H6, H2	0	Cambiar la etiqueta del campo por "Ocupación".
En los formularios de cliente jurídico falta un campo para el sitio web de la organización.	H6	2	Agregar un campo para el sitio web.
En los formularios de cliente jurídico, faltan campos para registrar los datos del representante legal	H6	2	Agregar los siguientes campos para registrar al representante legal: nombres, apellidos, cédula, rif, estado civil, fecha de nacimiento, lugar de nacimiento, nacionalidad, dirección de residencia, email, teléfono principal y secundario, profesión, ocupación, dirección laboral, rol en la organización.
En la tabla que muestra la lista de los clientes, los títulos están mal.	H8	1	Cambiar los títulos por: código, tipo, identificación del cliente, email, teléfonos y opciones.
En los formularios de expediente, el cliente está mal.	H6, H2	0	Cambiar cliente por: cliente u organismo asociado.
En los formularios de expediente, la etiqueta de título está mal.	H6, H2	0	Cambiar la etiqueta por identificación del expediente.
En los formularios de expediente, la etiqueta de tipo de expediente está mal.	H6, H2	0	Cambiar la etiqueta por materia del caso.
En los formularios de expediente la etiqueta de la fecha de inicio está mal,	H6, H2	0	Cambiar etiqueta por fecha de ingreso del caso y la de cierre, por fecha de egreso del caso.

al igual que la fecha de cierre.			
En los formularios de expediente no hay un campo para la fecha de admisión del caso en el Tribunal, así como de recepción del caso.	H6	2	Agregar 2 campos para la fecha de admisión del caso y la de recepción.
En los formularios de creación de expediente aparece la fecha de cierre y culminación.	H8	1	Ocultar estos campos.
En los formularios de expediente el campo Antecedentes no es necesario.	H8	2	Eliminar el campo.
Algunos botones dicen "ok" y se usan para cerrar ventanas.	H2	0	Reemplazar el texto del botón por "cerrar".

Tabla 29: Lista resultado de evaluación heurística

Se implementaron las funcionalidades al prototipo ejecutable a partir de los casos de usos y patrones de interacción. Se corrigieron los problemas de las heurísticas identificadas y se obtuvo una versión de la aplicación. Esta versión se sometió a una revisión adicional.

Revisión adicional

La aplicación se sometió a una revisión por parte de los abogados y surgieron otros problemas y funcionalidades adicionales. Por lo que se realizó una lista de problemas y nuevas funcionalidades, con soluciones propuestas. Se acordó realizar una lista parecida a la de la evaluación heurística, sin clasificación, ni valoración.

Problema	Solución
No hay posibilidad de verificar si el cliente esta creado previamente a la creación del expediente.	Colocar un mensaje que recuerde al usuario, crear el cliente si no se ha creado previamente.
Los iconos de operaciones de los elementos son muy pequeños	Colocar los iconos más grandes.

No es posible identificar bien el documento desde las pantallas de detalle y edición del expediente.	Colocar el nombre del documento y el enlace al mismo.
No es posible ver la semántica de los campos o una descripción de lo que debe ir allí.	Incluir en la validación los formatos de llenado de los campos.
Al generar la plantilla sólo es posible salir de la opción guardando el documento no hay forma de deshacer la generación.	Colocar el botón de salir sin salvar.
Proceso de registro muy largo.	Proveer otro método de registro y/o inicio de sesión.
El título del sistema ha cambiado.	Cambiar el título por “Aplicación web de apoyo para abogados litigantes en Venezuela”.
En los formularios de cliente natural, el campo Nacionalidad no está bien, debe ser una lista de países.	Cambiar la lista del campo de venezolano y extranjero, por una lista de países.
Al crear o editar un cliente natural o jurídico los campos requeridos no son los correctos.	En el formulario de cliente natural, marcar los campos requeridos: nombres, apellidos y cedula. En el de cliente jurídico: nombre de la organización.
En todos los formularios de creación y edición, los teléfonos no tienen validación.	Agregar validación a los campos de los teléfonos, con formato XXX-XXXXXXX.
En todos los formularios de creación y edición, los campos de email no tienen validación.	Agregar validación a los campos de emails, con formato usuario@dominio.ext
En el formulario de creación de expediente aparecen las secciones de agregar actuación y documentos.	Eliminar estas secciones y obligar al usuario a crear primero un expediente.
En el módulo de expediente, los documentos cargados deben categorizarse (digital o digitalizado).	Agregar campo para clasificar el tipo del documento (digital o digitalizado).
En los expedientes, los precios de los documentos no deben aparecer como campo.	Eliminar el campo del precio de los documentos de un expediente.

El campo organismo en los expedientes no debe estar enlazado con los organismos del módulo Organismos.	Eliminar campo y crear uno nuevo de tipo texto, para que el usuario pueda agregar información del organismo.
El costo total de las actuaciones en los expedientes no debe aparecer.	Eliminar el costo total y crear una sección para ver las actuaciones que tienen precio y ver el monto total.
El sistema no provee recordatorios o envíos de información vía correo electrónico.	Agregar a la agenda una funcionalidad que permita al menos, enviar la información del evento por correo electrónico.
En la agenda los días no laborales y no despacho se marcan para un solo calendario.	Los días no laborales y no despacho deben ser distintos para distintos organismos. Se requiere capacidad de diferenciar el organismo.
En la agenda no se tiene la capacidad de seleccionar el calendario de los distintos organismos.	Agregar la capacidad en la agenda de seleccionar un organismo y que se muestre el calendario correspondiente.
En los formularios de la actuaciones, todos los campos aparecen como requeridos.	Eliminar la opción requerido a todos los campos a excepción del de identificación.
En los formularios de la documentos, todos los campos aparecen como requeridos.	Eliminar la opción requerido a todos los campos a excepción del título y el archivo.
En los formularios de organismos el nombre, teléfono y dirección se muestran como requeridos.	Eliminar la opción requerido a todos los campos a excepción del nombre.
El tamaño de las letras esta pequeño.	Subir el tamaño de las letras.
El menú horizontal no se distingue bien.	Cambiar el color del menú para resaltarlo.
Hay un espacio vacío en la mayoría de las páginas.	Aprovechar mejor el espacio en al área de trabajo.

Tabla 30: Lista de problemas y soluciones generadas en la primera presentación

Se solucionaron los problemas, se agregaron las funcionalidades solicitadas y se consideró listo para su presentación y entrega.

4.9 Etapa 4: Entrega

La dirección en Internet de la aplicación publicada es <http://sapa2013.herokuapp.com>.

Liberación de la aplicación

A continuación se listan las funciones de cada uno de los módulos de la aplicación con su respectiva pantalla.

En el módulo de Clientes, el abogado puede crear, modificar o eliminar clientes, así como los datos asociados a los mismos. La aplicación provee un mecanismo para diferenciar a los clientes naturales de los clientes jurídicos, debiendo seleccionar el campo “Tipo de Cliente” en la creación del cliente.

En todas las pantallas de la aplicación se dispone de un calendario en donde se pueden apreciar días destacados, resultado de algún evento creado o de haber marcado alguna fecha como día no laborable o de no despacho para un organismo. Se colocó un selector de calendario para filtrar y visualizar los eventos personales y los días destacados de los organismos, por separado.

Sistema de Gestión de Casos para Abogados Litigantes de Venezuela - Google Chrome

sapa2013.herokuapp.com/cliente

Usuario: AnngyRakelDordellyRuiz

Aplicación web de apoyo para abogados litigantes en Venezuela

Cientes Expedientes Agenda Organismos Plantillas Ayuda

Agregar cliente

Mostrar 10 clientes por página

Buscar:

Código	Tipo	Identificación de cliente	Email	Teléfonos	Opciones
14	Juridico	MAKRO	anngyta@gmail.com	0212-4710309	
15	Juridico	Inversiones JORMI	anngyta@hotmail.com	0212-3734444	
16	Natural	La Riva, Atenas	atenas_lariva@hotmail.com	0212-2353990	
17	Juridico	Compumall CA	anngyta@gmail.com	0212-9775544	

Mostrando del 1 al 4 (4 en total)

primera anterior 1 siguiente ultima

Eventos personales

- Dia de no despacho
- Dia no laborable
- Evento

Octubre 2014

Do	Lu	Ma	Mi	Ju	Vi	Sa
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25

Figura 25: Módulo de clientes de la aplicación web desarrollada

En el módulo de Expedientes, el abogado puede crear, modificar o eliminar expedientes, así como los datos asociados, actuaciones y documentos. La aplicación provee un mecanismo para diferenciar los datos del expediente, las actuaciones, los documentos y los costos registrados.

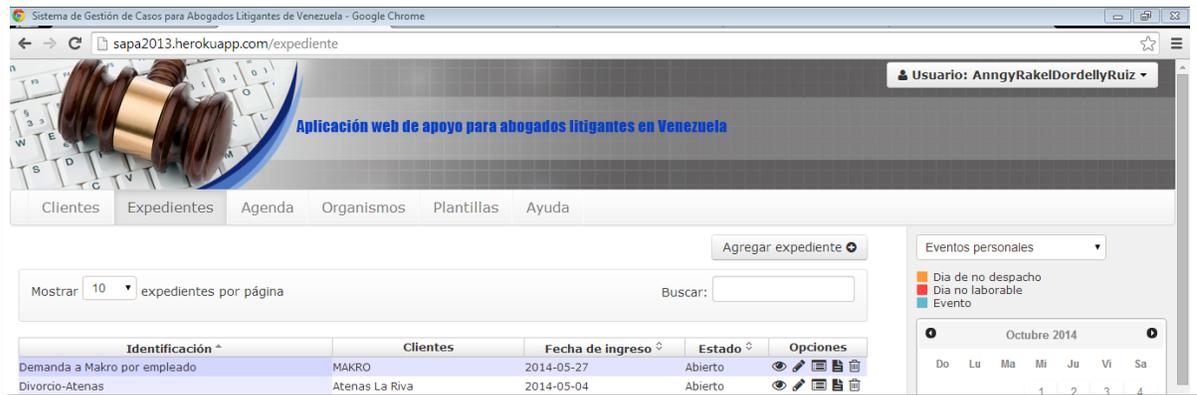


Figura 26: Módulo de expedientes de la aplicación web desarrollada

En el módulo de la Agenda, el abogado puede crear, modificar o eliminar eventos, así como los datos asociados a los mismos. La aplicación provee un mecanismo para diferenciar los eventos y las fechas marcadas como día no laborables y de no despacho, debiendo seleccionar el campo "Tipo de Evento" en la creación del evento.

Este módulo posee varias vistas. Cuando se accede al módulo la vista por defecto que se muestra es la diaria, pudiendo visualizar los eventos creados para en la fecha actual. Las otras vistas son la mensual, la anual y la de todos los eventos creados.

En la creación de un evento se dispone de un mecanismo que envía un correo electrónico con los datos del evento, a los clientes involucrados y participantes adicionales.

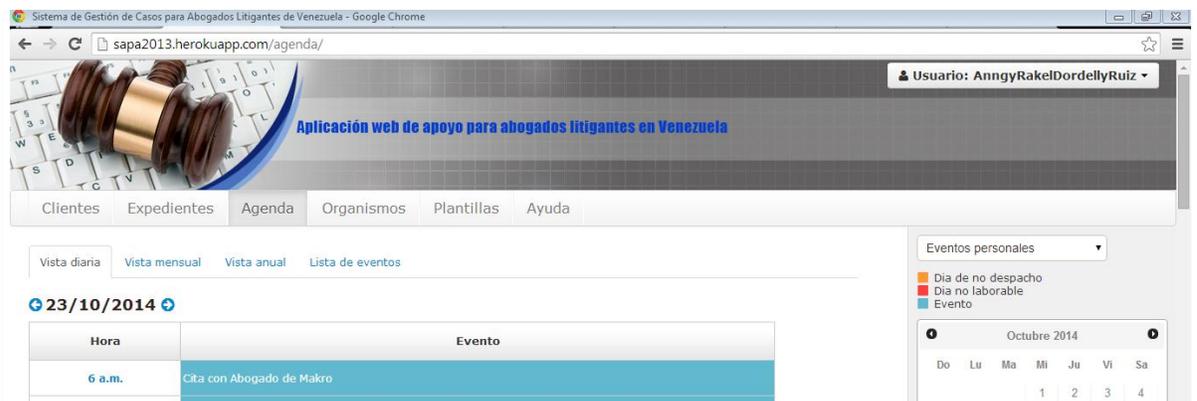


Figura 27: Módulo de agenda de la aplicación web desarrollada

En el módulo de Organismos, el abogado puede crear, modificar o eliminar organismos, así como los datos asociados a los mismos.

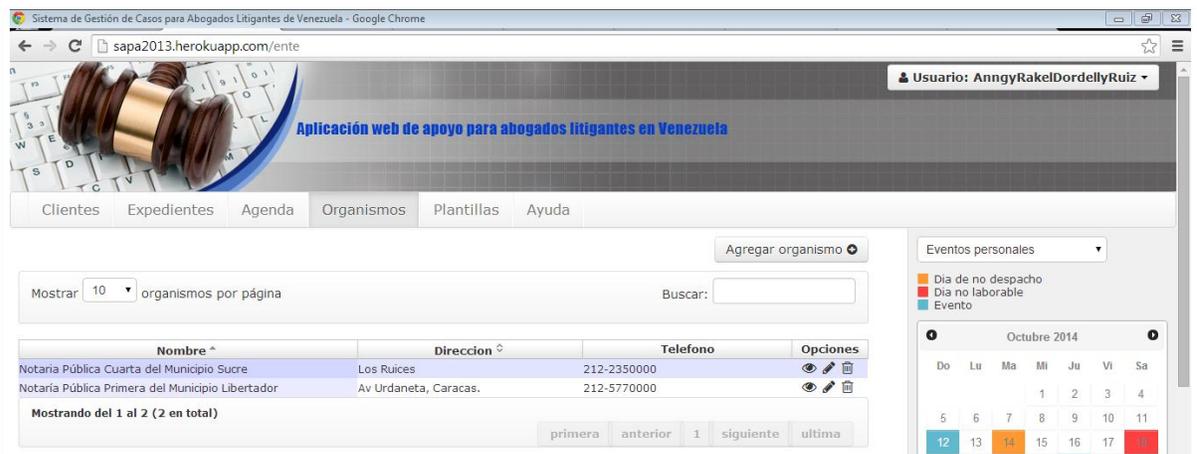


Figura 28: Módulo de organismos de la aplicación web desarrollada

En el módulo de Plantillas, el abogado puede crear, modificar o eliminar plantillas, así como los datos asociados a las mismas. La aplicación provee un mecanismo para diferenciar las plantillas de tipo documento y las que se pueden generar usando la aplicación.

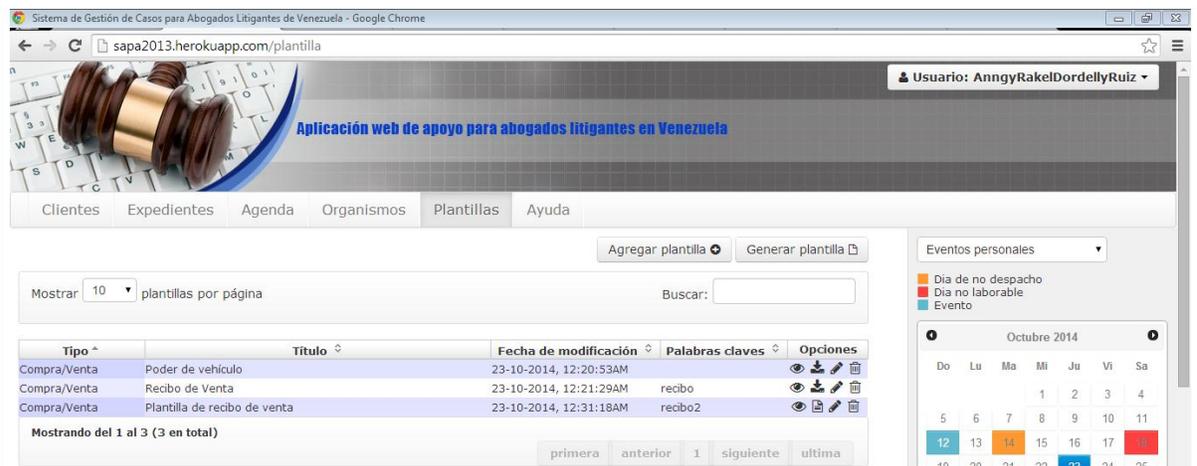


Figura 29: Módulo de plantillas de la aplicación web desarrollada

Encuesta de aceptación de la aplicación

Se decidió aplicar una encuesta de aceptación de la aplicación a los abogados colaboradores, a quienes se les aplicó la primera encuesta. Gracias a que la aplicación está publicada en un sitio público de Heroku, los abogados tuvieron la opción de acceder a través de la dirección web desde cualquier parte con acceso a Internet y a cualquier hora.

La encuesta realizada fue la siguiente:

Encuesta					
Responda la siguiente encuesta colocando una "X" en la opción de su preferencia.					
	Nunca	Raras veces	A veces	A menudo	Siempre
1. ¿Utilizaría Ud. la Aplicación Web de Apoyo para Abogados Litigantes en Venezuela para almacenar los datos de sus clientes?					
2. ¿Utilizaría Ud. la Aplicación Web de Apoyo para Abogados Litigantes en Venezuela para guardar la información que se genera en un caso?					
3. ¿Utilizaría Ud. la Aplicación Web de Apoyo para Abogados Litigantes en Venezuela para registrar actividades, gestiones, diligencias, actuaciones de un expediente?					
4. ¿Utilizaría Ud. la Aplicación Web de Apoyo para Abogados Litigantes en Venezuela para establecer el costo de sus servicios?					
5. ¿Utilizaría Ud. la Aplicación Web de Apoyo para Abogados Litigantes en Venezuela para guardar los documentos generados de un caso o expediente?					
6. ¿Utilizaría Ud. la Aplicación Web de Apoyo para Abogados Litigantes en Venezuela para guardar sus plantillas de documentos diligencias y formatos?					
7. ¿Utilizaría Ud. la Aplicación Web de Apoyo para Abogados Litigantes en Venezuela para guardar información de los días no laborables y de no despacho de los tribunales?					

8. ¿Dada las capacidades de la Aplicación Web de Apoyo para Abogados Litigantes en Venezuela utilizaría Ud. la aplicación como herramienta de apoyo de sus actividades diaria?							
--	--	--	--	--	--	--	--

Tabla 31: Encuesta aplicada

Los resultados de la encuesta aplicada fueron:

Pregunta	Respuesta
1	El 60% a menudo y 40% siempre.
2	Un 20% de los casos a veces, 40% a menudo y 40% siempre.
3	Un 40% usaría la herramienta a menudo y 60% siempre.
4	El 40% usaría a menudo y 60% siempre.
5	20% a veces, 40% a menudo y 40% siempre.
6	El 60% indicó que siempre usaría el sistema para guardar las plantillas y un 40% lo usaría a menudo.
7	Un 20% a veces, 20% a menudo y 60% siempre.
8	40% a menudo y 60% siempre usaría la herramienta. Esto es un alto grado de aceptación del sistema.

Tabla 32: Resultados de la encuesta aplicada

Tabulando los resultados se tiene que:

	N° 1	N° 2	N° 3	N° 4	N° 5	N° 6	N° 7	N° 8
Siempre	40%	40%	60%	60%	40%	60%	60%	60%
A menudo	60%	40%	40%	40%	40%	40%	20%	40%
A veces		20%			20%		20%	
Raras veces								
Nunca								

Tabla 33: Tabulación de resultados de la encuesta de aceptación de la aplicación web

Para mejor visualización y comprensión de los resultados de la encuesta, se configuró un gráfico de barras que a continuación se presenta:

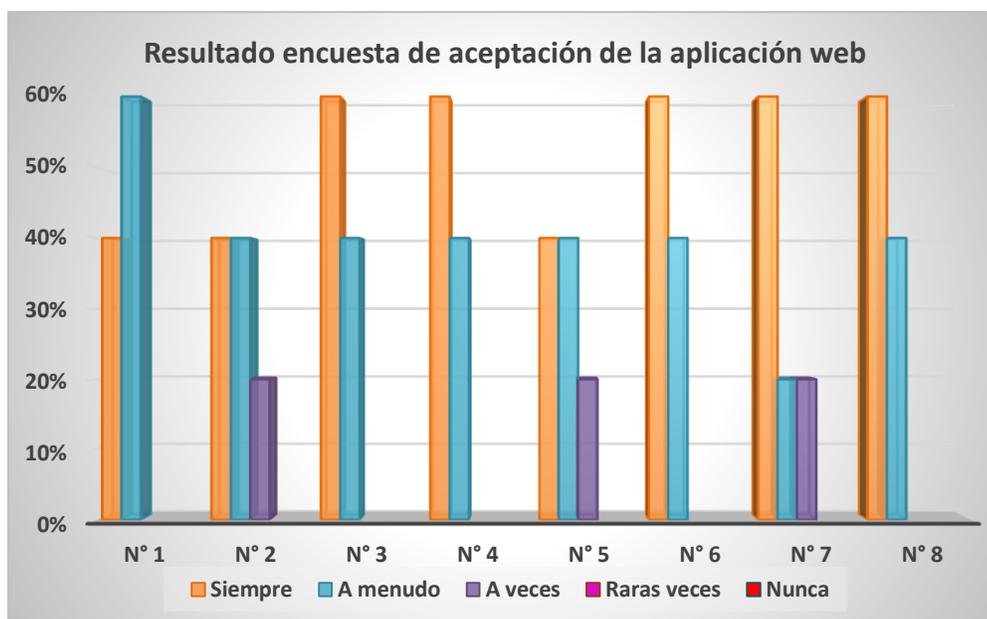


Figura 32. Gráfico de resultados de la encuesta de aceptación de la aplicación web

En el gráfico se puede apreciar que:

- En la pregunta N° 1, un 60% de los abogados encuestados afirman que **a menudo** utilizarían la Aplicación Web de Apoyo para Abogados Litigantes en Venezuela, para almacenar los datos de sus clientes. Y un 40% afirma que lo utilizaría **siempre**.
- En la pregunta N° 2, un 40% de los abogados encuestados afirman que **siempre** utilizarían la Aplicación Web de Apoyo para Abogados Litigantes en Venezuela, para guardar la información que se genera en un caso. Un 40% afirma que lo utilizaría **a menudo** y un 20% que lo utilizaría **a veces**.
- En la pregunta N° 3, un 60% de los abogados encuestados afirman que **siempre** utilizarían la Aplicación Web de Apoyo para Abogados Litigantes en Venezuela, para registrar actividades, gestiones, diligencias y actuaciones de expediente. Y un 40% afirma que lo utilizaría **a menudo**.
- En la pregunta N° 4, un 60% de los abogados encuestados afirman que **siempre** utilizarían la Aplicación Web de Apoyo para Abogados Litigantes en Venezuela, para establecer el costo de sus servicios. Y un 40% afirma que lo utilizaría **a menudo**.
- En la pregunta N° 5, un 40% de los abogados encuestados afirman que **siempre** utilizarían la Aplicación Web de Apoyo para Abogados Litigantes en Venezuela, para guardar los documentos generados de un caso. Un 40% afirma que lo utilizaría **a menudo** y un 20% que lo utilizaría **a veces**.
- En la pregunta N° 6, un 60% de los abogados encuestados afirman que **siempre** utilizarían la Aplicación Web de Apoyo para Abogados Litigantes en Venezuela, para guardar plantillas de documentos, diligencias y formatos. Y un 40% afirma que lo utilizaría **a menudo**.
- En la pregunta N° 7, un 60% de los abogados encuestados afirman que **siempre** utilizarían la Aplicación Web de Apoyo para Abogados Litigantes en Venezuela,

para guardar información de los días no laborables y de no despacho de los tribunales. Un 20% afirma que lo utilizaría **a menudo** y un 20% que lo utilizaría **a veces**.

- En la pregunta N° 8, un 60% de los abogados encuestados afirman que **siempre** utilizarían la Aplicación Web de Apoyo para Abogados Litigantes en Venezuela, para como herramienta de apoyo de sus actividades diarias. Y un 40% afirma que lo utilizaría **a menudo**.

Capítulo 5. Herramientas tecnológicas

En este capítulo se describen las herramientas tecnológicas utilizadas en el desarrollo de la aplicación.

5.1 Aplicaciones web

La web o Internet, como también se le suele denominar, ha sido adoptada en la última década por millones de personas, organizaciones y empresas, como un canal de comunicación, una vía para el intercambio de información y una herramienta para ejecución de innumerables operaciones de cualquier índole.

La humanidad ha sido testigo de una evolución y aumento progresivo del uso de las aplicaciones web en muchas de las tareas cotidianas, tanto en el ámbito personal como en lo laboral.

Una aplicación web, es un programa que se diseña para usarse a través de un navegador de internet. Su propósito fundamental, es facilitar el cumplimiento de una o más tareas [24].

Extendiendo el término al área de Ingeniería de Software, se denomina aplicación web, a aquellas herramientas que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una Intranet mediante un navegador. En otras palabras, es una aplicación de software que se codifica en un lenguaje soportado por navegadores web en la que se confía la ejecución al navegador [22].

Ventajas de las aplicaciones web

Las principales ventajas de las aplicaciones web son: su facilidad de manejo y desarrollo, accesibilidad y portabilidad. Y en base a ellas se deriva lo siguiente:

- Son fáciles de manejar, ya que la interacción con el usuario se establece en base a elementos a los que está cada vez más acostumbrado, como lo son las páginas web, las cuales le permiten conocer la funcionalidad del sistema con poca dificultad.
- Poseen alta disponibilidad, pues una de las premisas con la que se diseñan y se crean es la continuidad del servicio que ofrecen.
- No existen requerimientos de hardware para los usuarios finales de una aplicación web. Con el uso de un navegador en un computador, laptop, tableta o teléfono inteligente, es suficiente para poder utilizarlas.

- Son accesibles desde cualquier punto de una red con acceso a Internet o Intranet.
- Existe un costo efectivo de desarrollo, debido a que la usabilidad de la herramienta se limita a navegadores de internet, minimizando así el número de pruebas posibles que deben realizarse para asegurar su buen funcionamiento.
- Son económicas de desarrollar por la cantidad de herramientas gratuitas disponibles en el campo.
- Cambios y actualizaciones que se requieran realizar, tienden cada día a ser más fáciles y sencillos de agregar.
- Facilitan la centralización y el acceso a los datos.
- Dado que las aplicaciones web son gestionadas por un tercero, las actualizaciones son inmediatas, como resultado, el usuario final siempre usa la última versión publicada.

5.2 Arquitectura Cliente-Servidor

El término cliente-servidor fue usado por primera vez en la década de los 80 para referirse a computadores en red. Su uso se fue expandiendo y al final de esa misma década, empezó a ser aceptado y adoptado en el mundo de las comunicaciones [29].

La arquitectura cliente-servidor es un modelo de aplicación distribuida en donde se reparten tareas y cargas de trabajo entre proveedores de recursos o servicios, llamados servidores, y demandantes, llamados clientes [22].

La dinámica entre el cliente y el servidor es sencilla: el cliente envía un mensaje (petición) solicitando un determinado servicio a un servidor, y este envía uno o varios mensajes de vuelta al cliente con la respuesta (provee el servicio). Cabe destacar, que no es necesario que ambos componentes se encuentren en distintas computadoras, sin embargo, es el modelo más usado por la ventaja que brinda, al ofrecer la capacidad de distribuir varios recursos en una red.

La mayoría de los servicios de Internet son de tipo cliente-servidor. La acción de visitar una página en Internet requiere una arquitectura cliente-servidor, ya que el servidor web sirve las páginas web al navegador (cliente). Cuando se lee un artículo en www.wikipedia.com, la computadora y el navegador web del usuario serían considerados un cliente; y las computadoras, bases de datos y servicios que componen Wikipedia, serían considerados el servidor. Cuando el navegador web del usuario solicita un artículo de Wikipedia, el servidor de Wikipedia recopila la

información a mostrar en la base de datos de Wikipedia, la articula en una página web, y la envía de nuevo al navegador web del cliente [30].

Morales Pedro (1999) define a las aplicaciones web como aplicaciones basadas en el modelo cliente-servidor. Y esto es, debido a que el navegador web realiza peticiones a una aplicación remota accesible a través de Internet (o a través de una Intranet) que reside en un servidor, y éste envía una respuesta que se muestra dicho navegador.

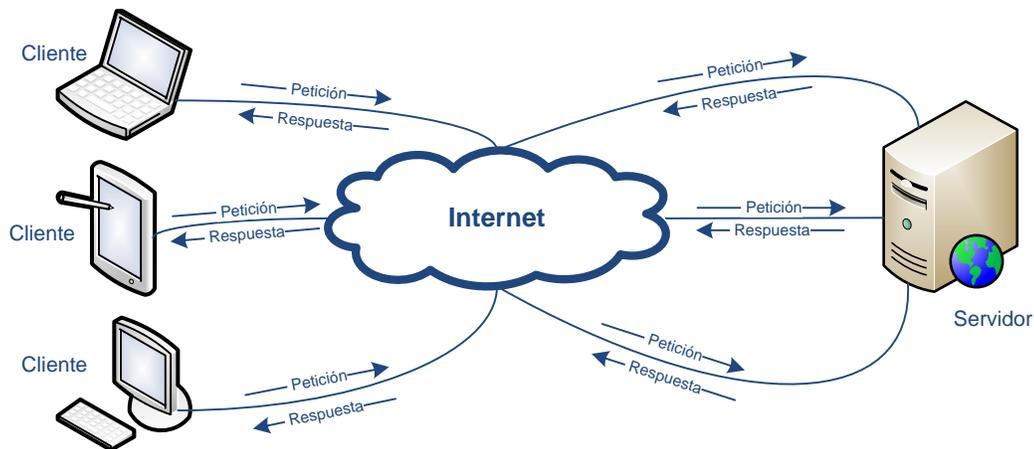


Figura 33: Arquitectura Cliente-Servidor

En un ambiente web, el servidor web es un software o programa que atiende y responde a las diferentes peticiones de los navegadores web, proporcionando los recursos que se soliciten usando el protocolo HTTP (Hypertext Transfer Protocol – Protocolo de Transferencias de Hipertexto), el cual, define la sintaxis y semántica que utilizan los elementos de software de la arquitectura web para comunicarse. Otro protocolo usado es el HTTPS (Hypertext Transfer Protocol Secure - Protocolo Seguro de Transferencia de Hipertexto), que basado en el protocolo HTTP agrega seguridad a la transferencia de datos de Hipertexto, es decir, es la versión segura del HTTP.

5.3 Patrón de diseño de software Modelo Vista Controlador

En la década de los 70's, lenguajes de programación como SmallTalk y Simula I, fueron construyendo gradualmente el paradigma de programación orientada a objetos y estableciendo conceptos tales como: objetos, clases, encapsulación, herencia y polimorfismo [37]. En la actualidad, dichos lenguajes no son usados para crear aplicaciones, pero los conceptos concebidos en su desarrollo y evolución permanecen aún vigentes y son la base de lenguajes como C++, Java o C#.

SmallTalk fue el primer lenguaje de programación que permitió diseñar interfaces de usuario con múltiples ventanas desplegadas en una misma pantalla. El concepto central detrás de las librerías de interfaz de usuario provistas por SmallTalk está basado en el patrón de diseño Modelo Vista Controlador, creado por el profesor Trygve Reenskaug [38], el cual se señalará de ahora en adelante en este documento por su acrónimo MVC.

El patrón de diseño MVC plantea un método formal para separar los módulos de entrada, de procesamiento y de salida de datos en una aplicación, así como la forma de comunicación entre ellos. Es decir, separa la lógica de negocio de la interfaz de usuario, facilitando así la evolución por separado de ambos aspectos e incrementando la reutilización y la flexibilidad.

El MVC se define como un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello, propone la construcción de tres componentes distintos que son el Modelo, la Vista y el Controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario. Este patrón de diseño se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento [37].

Descripción del patrón

- **El Modelo:** Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). Envía a la Vista aquella parte de la información en cada momento que se le solicita que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al Modelo a través del Controlador.
- **El Controlador:** Responde a eventos (usualmente acciones del usuario) e invoca peticiones al Modelo cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos). También puede enviar comandos a su Vista asociada si se solicita un cambio en la forma en que se presenta el Modelo (por ejemplo, desplazamiento en un documento o en diferentes registros de una base de datos), por tanto, se dice que el Controlador es intermediario entre la Vista y el Modelo.

- **La Vista:** Presenta el Modelo (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario) por tanto requiere del Modelo la información que debe representar como salida [38].

Interacción de los componentes

Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo de control que se sigue generalmente es el siguiente:

1. El usuario interactúa con la interfaz de usuario de alguna manera, es decir, genera un evento (por ejemplo, presiona un botón, enlace, etc.).
2. El Controlador recibe el evento y lo gestiona.
3. El Controlador notifica al Modelo de la acción del usuario y posiblemente esto resulte en una consulta o modificación del Modelo (por ejemplo, actualización de un carrito de compras en línea).
4. El Controlador envía la respuesta a la Vista y esta reacciona generando la interfaz apropiada (por ejemplo, la vista produce una pantalla con una lista del contenido de carrito de compras). La Vista obtiene sus propios datos a partir del modelo. El Modelo no tiene conocimiento directo de la Vista.
5. La interfaz de usuario espera una nueva acción del usuario, repitiendo el ciclo de nuevo [37].

La figura 7 representa gráficamente los componentes y la interacción descrita anteriormente.

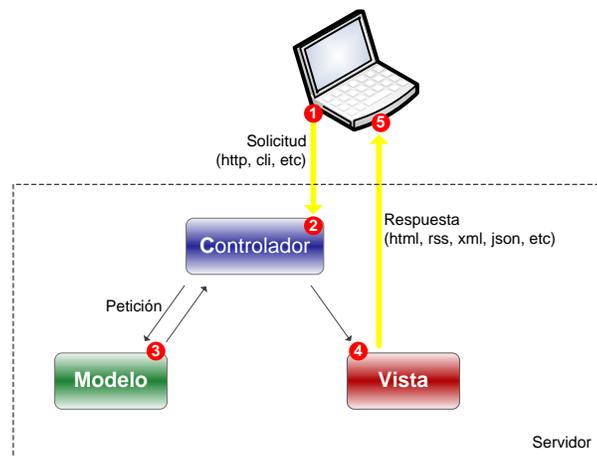


Figura 34: Modelo Vista Controlador

Frameworks que soportan MVC

Un framework –que en español significa marco de trabajo- es una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación [40]. Es un conjunto estandarizado de conceptos, prácticas y criterios para hacer frente a un tipo común de problema, que puede ser usado para ayudar a resolverlo de forma rápida y eficaz.

Los frameworks se conforman una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que puede servir de base para la organización y desarrollo de software. Típicamente, pueden incluir soporte de programas y librerías que son reutilizables, y un lenguaje interpretado, entre otras herramientas, que facilitan y agilizan el desarrollo de una aplicación web [41]. Su objetivo, en otras palabras, es proporcionar una estructura común, de modo que los desarrolladores no tienen que crear código desde cero, sino que reúsan los ya creados.

Básicamente, existen 2 tipos de frameworks a diferenciar: backend y frontend. Estos son términos fueron adoptados por los desarrolladores de software para indicar el lado donde reside el framework, hablando claro está, en un contexto de arquitectura cliente-servidor. De este modo, un framework backend es aquel que reside del lado del servidor, parte lógica de la aplicación web. Y el frontend reside en el lado del cliente y se refiere a la parte visual de la aplicación web.

El MVC ha sido adoptado ampliamente como arquitectura para diseñar e implementar aplicaciones web. De hecho, se han desarrollado multitud de frameworks, comerciales y no comerciales, que implementan este patrón.

Los frameworks, pueden diferenciarse por la interpretación de como las funciones MVC se dividen entre cliente y servidor [38]. Algunos de los frameworks más conocidos y usados, que implementan MVC, son:

Nombre del framework	Lenguaje de programación	Creador
Cocoa	Objective C	Apple
Ruby on Rails	Ruby	MIT
Struts	Java	Apache
Spring	Java	Apache
ExtJS 4	JavaScript	GPLv3
Zend Framework	PHP	BSD
CakePHP	PHP	MIT

Django	Python	BSD
ASP .NET MVC	.NET	MS-PL

Tabla 34: Frameworks que soportan el patrón MVC [38]

5.4 Base de datos

No existe una única definición de lo que es una base de datos, pero la mayoría de los autores la resumen como una colección organizada de datos; siendo la palabra “organización” la clave para entender su significado, ya que ella se refiere al orden que se le pongan a las cosas, al establecimiento o reforma de algo para lograr un fin.

Desde el punto de vista informático, una base de datos es un conjunto exhaustivo, no redundante de datos estructurados, organizados independientemente de su utilización y su implementación en máquina, accesibles en tiempo real y compatibles con usuarios concurrentes con necesidad de información distinta y no predecible en el tiempo; donde la información se encuentra almacenada en una memoria física que permite el acceso directo a un conjunto de programas que manipulan esos datos [31].

Sistemas manejadores de bases de datos

Formalmente, una base de datos se refiere a los datos en sí y el apoyo a las estructuras de los mismos. Ellas, se crean para operar grandes cantidades de información que es agregada, almacenada, recuperada y gestionada en el tiempo.

Un sistema manejador de base de datos, el cual será referido de ahora en adelante por sus siglas “SMBD”, consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos. Su objetivo principal es proporcionar una forma de almacenar y recuperar la información de una base de datos, práctica y eficientemente [43].

Los SMBD están diseñados especialmente para interactuar con el usuario, otras aplicaciones y la propia base de datos, para capturar y analizar los datos [42].

La gestión de los datos implica tanto la definición de estructuras para almacenar la información como la provisión de mecanismos para la manipulación de la información, de esto, son responsables los SMBD, así como mantener la integridad y seguridad de los datos, pues deben proporcionar la fiabilidad de la información almacenada, a pesar de fallas del sistema o intentos de acceso sin autorización. Además, si los datos van a ser compartidos entre diversos usuarios, el sistema debe evitar posibles resultados anómalos.

Una base de datos no es generalmente portable a través de diferentes SMBD, pero diferentes SMBD pueden interoperar utilizando estándares como SQL y ODBC para permitir a una aplicación trabajar con más de una base de datos [42].

Algunos de SMBD más conocidos son: MySQL, Microsoft SLQ Server, Oracle, dBASE, SAP; SQLite, PostgreSQL, FoxPro, IBM DB2, OpenOffice Base Y FileMaker Pro.

5.5 PostgreSQL (BD)

PostgreSQL es un SMBD relacional orientado a objetos, gratis y de código abierto. Su pasado está ligado al inicio mismo de los SMBD relacionales orientado a objetos. Fue desarrollado originalmente por el equipo de desarrolladores de Michael Stonebreaker en la Universidad de California, Berkeley. Posteriormente, fue publicado y liberado bajo la licencia de Distribución de Software Berkeley (BSD - Berkeley Software Distribution).

Actualmente, PostgreSQL es dirigido por una comunidad de desarrolladores denominada el Grupo de Desarrollo Global de PostgreSQL (PostgreSQL Global Development Group - PGDG). Todos los integrantes de esta comunidad, trabajan de forma desinteresada, altruista y libre, a la vez que son apoyados por organizaciones comerciales [44].

Según la compañía austriaca de desarrollo de software, Solid IT, la cual publica mensualmente en su página web www.DB-engines.com, una lista de posiciones de los SMBD más usados en Internet, PostgreSQL, se encuentra posicionado en el 4to lugar entre los 10 primeros. La recolección y consolidación de esta información se basa en las menciones de los SMBD en los siguientes criterios: número de menciones en los sitios web, mediciones de resultados de búsqueda en internet, interés general en el sistema (medido a través de Google Trends¹), frecuencia de temas en foros, número de ofertas de empleo en las que se mencionan y número de perfiles en las redes profesionales en los que se mencionan. A continuación se muestra una captura de pantalla de la lista, tomada en el mes de enero de 2014.

¹ Google Trends es una herramienta de Google Labs que muestra los términos de búsqueda más populares del pasado reciente en Internet. <http://www.google.com/trends/?hl=es>

Rank	Last Month	DBMS	Database Model	Score	Changes
1.	1.	Oracle ↗	Relational DBMS	1500.23	+32.43
2.	2.	MySQL ↗	Relational DBMS	1288.39	-8.53
3.	3.	Microsoft SQL Server ↗	Relational DBMS	1214.27	-11.75
4.	4.	PostgreSQL ↗	Relational DBMS	230.45	+2.20
5.	 6.	MongoDB ↗	Document store	195.17	+16.94
6.	 5.	DB2 ↗	Relational DBMS	188.46	+0.15
7.	7.	Microsoft Access ↗	Relational DBMS	152.88	-22.11
8.	8.	SQLite ↗	Relational DBMS	93.00	-4.29
9.	9.	Sybase ASE ↗	Relational DBMS	87.88	-6.62
10.	10.	Cassandra ↗	Wide column store	80.31	-0.87

Figura 35: Lista de posiciones de mayor de uso de los SMBD [42]

Principales características de PostgreSQL

Algunas de las principales características de PostgreSQL son:

- *Cumple con las características ACID²*. PostgreSQL cuenta con las funcionalidades necesarias para que sus transacciones cumplan la “Atomicidad”³, la “Consistencia”⁴, el “Aislamiento”⁵ y la “Durabilidad”⁶. Estas

²ACID es un acrónimo de Atomicity, Consistency, Isolation and Durability, que en español significa: Atomicidad, Consistencia, Aislamiento y Durabilidad. Todas ellas, son un conjunto de características necesarias para que una serie de instrucciones puedan ser consideradas como una transacción. Un sistema de gestión de bases de datos se dice que es ACID, cuando cuenta con las funcionalidades necesarias para que sus transacciones tengan las características ACID.

³Atomicidad es una propiedad que asegura que una transacción se realizó o no. Las transacciones atómicas son indivisibles, de forma exclusiva: se ejecutan todas las operaciones que la componen o ninguna.

⁴Consistencia es una propiedad que asegura que cualquier transacción llevará a la base de datos de un estado válido a otro estado válido. Cualquier dato que se escriba en la base de datos tiene que ser válido de acuerdo a las reglas definidas.

⁵Aislamiento es una propiedad que asegura que la ejecución concurrente de las transacciones resulte en un estado del sistema que se obtendría si estas transacciones fueran ejecutadas en secuencia. Es decir, dos o más transacciones sobre los mismos datos no generarán un problema.

⁶Durabilidad es la propiedad que garantiza la persistencia de una transacción. Es decir, que las actualizaciones hechas a la base de datos producto de transacciones ejecutadas, no se pierdan, incluso ocurriendo un fallo en el sistema.

características son necesarias para que una serie de instrucciones puedan ser consideradas como una transacción en una base de datos.

- *Soporte a alta concurrencia.* Mediante un sistema denominado Control de Acceso Concurrente Multiversión (Multi-Version Concurrency Control – MVCC), PostgreSQL permite que, mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo “commit”⁷.
- *Amplia variedad de tipos de datos.* PostgreSQL provee numerosos tipos de datos y posibilidad de definir nuevos tipos. Además de los tipos estándares en cualquier base de datos, están disponibles los tipos geométricos, direcciones de red, cadenas binarias, matrices, entre otros. Adicionalmente, se pueden crear tipos de datos propios.
- Provee funciones/procedimientos almacenados (stored procedures) en numerosos lenguajes de programación, bloques anónimos de código de procedimientos, soporte al almacenamiento de objetos binarios grandes (gráficos, videos, sonido, etc.), claves foráneas (foreign keys), disparadores (triggers), vistas, integridad transaccional, herencia de tablas, soporte para transacciones distribuidas, entre otras [46].

5.6 Lenguaje de Marcado de Hipertexto (HMTL)

El Lenguaje de Marcas de Hipertexto (HyperText Markup Language – HTML), sirve de estándar de referencia para la elaboración de páginas web y establece una estructura básica y un código (denominado HTML) para la definición de contenido, texto, imágenes, videos, etc. Tal y como su nombre lo indica, es un lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web.

Basándose en la filosofía de desarrollo de la referenciación, añade elementos externos (imagen, video, script, etc.) a una página web, a través de referencias a las ubicaciones de dichos elementos mediante texto, en vez de colocarlos directamente en el código de la página. De modo que, la página web sólo contiene texto y el navegador web (interpretador del código) se encarga de unir todos los elementos y mostrarlos en la página web final [49].

Cabe destacar, que el navegador web es una aplicación que sólo comprende las etiquetas HTML, y con ellas, da formato a la página y la presenta en pantalla. Podría

⁷Commit: sentencia que termina una unidad de trabajo y confirma los cambios de la base de datos que ha realizado esa unidad de trabajo.

decirse, que el HTML es un servicio esencial que permite la creación de páginas web con información estática.

5.7 Lenguajes de programación web

Para eliminar esa limitante que el HTML presenta, y en general, para agregar elementos de mayor funcionalidad y dinamismo a las páginas web, nacieron los lenguajes de programación web, que no son más que lenguajes de programación específicos que permiten desarrollar páginas y aplicaciones que corren sobre Internet.

La forma en que se integran el HTML y un lenguaje de programación es literalmente incrustando al código HTML un programa, o lo que es lo mismo, otro código –o pedazos de código- creado con el lenguaje de programación web. Dicho programa, puede correr en el cliente o en el servidor web. De allí, que la programación web se divida en dos grandes grupos: la del lado del servidor y la del lado del cliente [48].

Programación web del lado del servidor

Los lenguajes del lado servidor son aquellos que permiten crear programas que son reconocidos, ejecutados e interpretados por el propio servidor web y que envían datos al cliente en un formato comprensible para él.

Los usos de la programación web del lado del servidor son:

- Procesamiento de entradas del usuario.
- Presentación de las páginas solicitadas.
- Permite estructurar las aplicaciones web.
- Provisión de interacción con servidores y almacenamiento.
- Provisión de interacción con bases de datos, incluyendo la consulta y modificación de los datos.
- Provisión de la codificación de datos en páginas HTML.

Algunos ejemplos de los lenguajes del lado del servidor son: PHP, ASP.NET (C#), C++, Java, Python, Ruby, entre otros.

Programación web del lado del cliente

En contraposición a los lenguajes del lado del servidor, con los lenguajes del lado del cliente se crean secuencias de comandos que son incrustados en la página web y se

procesan en el navegador web. Vale señalar, que el navegador web debe admitir estos lenguajes para que la secuencia de comando se pueda ejecutar.

Los usos de la programación web del lado del cliente son:

- Permite la creación de páginas web más interactivas.
- Permite que los elementos que la componen, puedan trabajar dinámicamente.
- Permite la interacción con almacenamiento temporal y local.
- Funciona como una interfaz entre el usuario y el servidor
- Envío de peticiones al servidor.
- Recuperación de datos del servidor.
- Proporciona acceso remoto para el programa de servidor de cliente.

Algunos lenguajes del lado del cliente son: JavaScript, JQuery, VBScript, AJAX, etc.

Las diferencias básicas entre la programación del lado del cliente y del servidor, se aprecian en la siguiente tabla:

Programación del lado del cliente	Programación del lado del servidor
Procesos complejos son a menudo más eficientes que en el lado del cliente, ya que el programa y los recursos asociados no se descargan en el navegador web.	La respuesta a la interacción puede ser más rápida que en el lado del servidor, una vez que se ha descargado el código del programa.
Hay consideraciones de seguridad al enviar información sensible.	Los servicios son seguros, ya que no se envía información desde el navegador.
No se basa en el navegador web.	Hay dependencia del navegador que el usuario use en el computador.
El procesamiento es afectado por el rendimiento del servidor.	La velocidad del procesamiento es afectada por el rendimiento de la computadora del usuario.

Tabla 35: Diferencias entre la programación del lado del cliente y del servidor

5.8 Python

Python es un lenguaje de programación con licencia de código abierto creado en Holanda a principios de la década de los 90 por Guido van Rossum. En la actualidad,

Python es dirigido por Guido van Rossum y administrado por la Fundación de Software Python (Python Software Foundation – PSF) [54].

Se trata de un lenguaje interpretado, de alto nivel, basado en objetos. Sus estructuras de datos de alto nivel, combinadas con tipado dinámico y ligadura dinámica, lo hacen especialmente atractivo para el desarrollo de aplicaciones rápidas así como lenguaje de scripts o medio de conexión de otros componentes existentes [60].

Además de la claridad y sencillez de su sintaxis mejora notablemente la legibilidad de los programas y por lo tanto reduce el costo de mantenimiento de software.

Python, soporta además módulos y paquetes, lo que incrementa la modularidad y la reutilización de código. Permite incorporar rutinas compiladas en C para realizar funciones críticas a alta velocidad. También implementa estructuras de datos avanzadas (listas, tuplas, diccionarios) que pueden ser combinadas para crear otras estructuras más complejas [55].

Su intérprete y la librería de Python están disponibles para las principales plataformas (Linux, Windows, Unix, etc.) en código fuente o binario y pueden ser distribuidas libremente. Asimismo, Python se caracteriza por disponer de una excelente documentación y una de las mejores guías de aprendizaje.

5.9 Django

Django es un framework web backend de código abierto escrito en Python, que permite construir aplicaciones web más rápido y con menos código, fomentando el desarrollo rápido y el diseño limpio y pragmático. Así es como lo define la comunidad Fundación de Software Django, quien es la encargada de mantener este framework [65].

Sus principales características son:

- Se adscribe al patrón de diseño de software MVC, por lo que, las diferentes partes del sitio están claramente separadas.
- Facilita la creación de sitios web complejos. Django pone énfasis en el re-uso, la conectividad y extensibilidad de componentes, apegándose al principio DRY (Don't Repeat Yourself), que en español significa No Te Repitas.
- Usa Python en todas las partes del framework, incluso en configuraciones, archivos, y en los modelos de datos.

- Enfocado en resolver problemas de desarrollo Web con los que los desarrolladores de Django se han encontrado.
- Aplicaciones que son fáciles de mantener y rindan bajo mucha carga.
- URL's limpias (amigables).
- Módulos poco dependientes.
- El sistema de archivos que lo componen en la instalación sin comenzar a desarrollar, es muy pequeño en comparación con otros frameworks que se desenvuelven en el mismo ámbito. Django crea en una (1) carpeta 4 archivos. Mientras que frameworks como Rail y Symfony, crea más de, veinticinco (25) carpetas y ciento quince (115) archivos.
- La comunidad de desarrolladores que lo mantiene aporta mejoras casi a diario, motivados por propios deseos egoístas de ahorrar tiempo a ellos mismos [67].

Algunos de los sitios web que usan Django como framework web backend, son: Google, Yahoo, Discovery Communication, Instagram, Pinterest, National Geographic, Walt Disney, The New York Times, Mozilla Fundation, entre otros.

Como ya se mencionó en secciones anteriores, los frameworks proveen librerías que agilizan el desarrollo de software. En Django, a estas librerías se les llama aplicaciones. Las aplicaciones de mayor relevancia utilizadas en la creación de la presente aplicación web y su respectiva función, se aprecian en la siguiente tabla:

Nombre de la aplicación	Función
Django Crispy Forms	Permite construir, personalizar y reusar formularios. Con el uso de esta aplicación se tiene el control total del HTML, sin necesidad de escribir código HTML [68].
Django Guardian	Permite la implementación de permisos de objetos de Django, proporcionando un motor adicional de autenticación [69].
Django Registration	Permite implementar una funcionalidad de registro de usuario, de aplicación sencilla pero extensible para Django potenciada por sitios web. Facilita el proceso de registro a una aplicación web en dos fases. Se realiza el registro inicial seguido de un correo electrónico de confirmación con instrucciones para la activación de la cuenta nueva [70].
Django Social Auth	Permite configurar mecanismos de autenticación o autorización social para los proyectos de Django.

	Implementa una interfaz común para definir los nuevos proveedores de autenticación de terceros, es decir, proporciona el registro del usuario y de inicio de sesión utilizando las credenciales de sitios sociales [71].
--	--

Tabla 36: Lista de aplicaciones usadas de Django

5.10 JavaScript

JavaScript es un lenguaje de programación web interpretado, basado en objetos, basado en prototipos, imperativo, multiplataforma, débilmente tipado y dinámico, cuyo código se incluye directamente en el mismo documento, usado para el desarrollo de aplicaciones web cliente-servidor en páginas HTML [72].

Se utiliza mayormente en su forma del lado del cliente, implementado como parte de un navegador web, permitiendo incorporar mejoras en la interfaz de usuario y páginas web dinámicas.

Algunas de las facilidades que el uso de JavaScript proporciona son:

- Interactividad y modificaciones de los objetos de dominio, incluyendo soporte para CSS⁸.
- Manejo de eventos. Capturar eventos generados por el usuario y responder a ellos sin crear una petición al servidor.
- Manipulación de la hoja de estilos CSS.
- Incorporar efectos y animaciones personalizadas.
- Controlar ventanas del navegador y contenido que se muestra.
- Programar páginas dinámicas simples.
- Evitar dependencia del servidor web para cálculos sencillos.
- Comprobar los datos que el usuario introduce en un formulario antes de enviarlos.
- Comunicación con el usuario mediante diversos métodos.

⁸ CSS (Cascading Style Sheets) y en español Hoja de Estilo en Cascada, es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML. Con el uso de este lenguaje se ejerce control sobre los textos, tablas, gráficos y otros, tales como: tipo, tamaño, énfasis, color, fondos, bordes, relleno, etc.

Gracias a que JavaScript es considerado un framework frontend de desarrollo web, existe una gran cantidad de librerías que agilizan el desarrollo y ahorran tiempo a los programadores.

jQuery

Una de las librerías de JavaScript usadas en el desarrollo de la aplicación web propuesta en el presente trabajo, es jQuery. Esta, permite simplificar la manera de interactuar con los documentos HTML. Es software libre y de código abierto, por lo que puede usarse en proyectos libres y privados [74].

Al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

5.11 Bootstrap

Es un framework frontend creado originalmente por dos desarrolladores de Twitter, para acelerar el diseño de nuevas aplicaciones web. Su objetivo principal es el de facilitar la creación de interfaces web con hojas de estilo en cascada y JavaScript. Su característica principal y objeto de su creación, es la adaptación de la interfaz del sitio web al tamaño del dispositivo en que se visualice, es decir, el sitio web se adapta automáticamente al tamaño de la resolución del dispositivo donde se muestre la página web. Esta técnica de diseño y desarrollo se conoce como “diseño adaptativo”.

Los diseños de interfaces web creados con Bootstrap son simples, limpios e intuitivos, esto les da agilidad a la hora de cargar y al adaptarse a otros dispositivos. El framework tiene varios elementos con estilos predefinidos fáciles de configurar, tales como: botones, menús desplegables, formularios incluyendo todos sus elementos e integración con jQuery [75].

5.12 Heroku

Heroku es una “Plataforma como Servicio” (Platform as a Service – PaaS), la cual, es un modelo de prestación de servicio que ofrece al usuario desarrollador, una plataforma de desarrollo y herramientas de programación que puede utilizar para desarrollar y controlar aplicaciones propias, sin considerar la infraestructura. Este servicio, es la encapsulación de una abstracción de un ambiente de desarrollo y el empaquetamiento de una serie de módulos o complementos que proporcionan, normalmente, una funcionalidad horizontal (persistencia de datos, autenticación,

mensajería, etc.). De esta forma, un arquetipo de plataforma como servicio podría consistir en un entorno que contiene una pila básica de sistemas, componentes preconfigurados y listos para integrarse sobre una tecnología concreta de desarrollo. Ejemplos de este modelo son un sistema Linux, un servidor web o un ambiente de programación. Ejemplos comerciales incluyen Google App Engine, que sirve aplicaciones de la infraestructura Google, Windows Azure de Microsoft, que permite el desarrollo y ejecución de aplicaciones codificadas en varios lenguajes y tecnologías como .NET, Java y PHP.

Heroku soporta distintos lenguajes de programación tales como: Ruby, Java, Node.js, Scala, Clojure, Python y PHP [82].

En Heroku se define una aplicación como el código fuente de la aplicación web junto a sus dependencias, las cuales se indican en un archivo con sus respectivas versiones. En el caso de Python, este archivo debe ser llamado "requirements.txt".

El despliegue de la aplicación web se hace a través de "Git". Git es un software de control de versiones que contempla una serie de comandos con los que se controlan las versiones que se desarrollan.

Heroku fue seleccionado como PaaS para la creación de la presente aplicación web, debido a que es una plataforma gratuita que provee y gestiona todos los recursos necesarios para publicar una aplicación web, lo que evita a los desarrolladores considerar factores, tales como: el sistema operativo del servidor que alojará la aplicación web, configuración del servidor y de los servicios web, selección de base de datos, configuración de base de datos y publicación del sitio web. Todos estos factores resultaron transparentes en el momento de la implementación.

Conclusiones y recomendaciones

Culminado el presente trabajo especial de grado, se concluye que los objetivos planteados fueron alcanzados en su totalidad, debido a que:

- Se describieron y se analizaron los procesos y procedimientos básicos y relevantes de un juicio ordinario en Venezuela. Y esto conllevó, a identificar las actividades que un abogado litigante que trabaja de forma independiente, desempeña durante un proceso judicial.
- Con el módulo de Expedientes y la sección de actuaciones, el abogado puede navegar y hacer seguimiento simplificado del desarrollo de cada uno de los casos que registra en la aplicación.
- El uso de las técnicas propuestas por la metodología AgilUs hizo que se determinaran efectivamente los requerimientos, y estos se convirtieron en las funciones y capacidades de la aplicación, claramente separados en los módulos que la componen.
- La interfaz diseñada, fue considerada amigable y armoniosa.
- Las herramientas tecnológicas y el uso del Modelo Vista Controlador, permitieron desarrollar un código limpio y sencillo. Además, se comprobaron las facilidades que ofrece el patrón, desarrollando en Python y usando Django. Todas ellas, fueron una combinación eficaz al momento de corregir problemas y añadir funcionalidades al sistema.
- La aplicación se implementó correctamente, ya que luego de que se subió a Heroku y se indicaron en esa plataforma todos los requerimientos, se comportó en la forma esperada.

La aplicación web permite de forma efectiva centralizar y controlar información producida en la gestión jurídica de los abogados litigantes en Venezuela, siguiendo los pasos propuestos en la metodología AgilUs.

Una capacidad puesta en práctica desde el inicio de la investigación y sobretodo en las entrevistas con los abogados, fue la abstracción. A pesar de lo amplio y complejo del tema, se logró simplificar y dar forma a la solución de las principales necesidades identificadas en la gestión jurídica de los abogados. Sin esta capacidad, el desarrollo de la aplicación hubiese sido muy extenso para un solo desarrollador.

Se concluye además, de acuerdo a la investigación realizada, que no se ha desarrollado y no existe en Venezuela, un software especializado de carácter gratuito que cumpla con las necesidades que presentan los abogados litigantes en su gestión

jurídica. En otras palabras, existe un vacío en el mercado nacional de este tipo de herramientas, que la aplicación web desarrollada cubrirá.

Durante el diseño de esta aplicación se evaluó la posibilidad de ofrecer una herramienta de tipo standalone, sin embargo, la problemática, el propósito y luego los requisitos establecidos por los abogados, reforzaron la premisa de que fuera una aplicación orientada a un servicio web, para evitar la dependencia con computadoras. Además, manifestaron dar un paso adelante con la tendencia de tener todo en Internet.

En cuanto a la metodología AgilUS, la cual jugó un papel clave en la flexibilidad de las técnicas y mecanismos que propone. En el desarrollo de esta aplicación, las conversaciones entre los abogados y el desarrollador resultaron ser una de las mejores formas para comprender y plasmar en papel las necesidades los abogados. Otra de las herramientas claves, fueron las encuestas aplicadas para recolectar información y evaluar el grado de aceptación de la aplicación y el posible uso futuro. Cabe mencionar, que el enfoque dado a estas encuestas fue de carácter cualitativo y no cuantitativo. La cantidad de abogados encuestados fue 5 y todos con una amplia experiencia en el manejo de herramientas ofimáticas y navegación en internet, así como en su profesión.

Los resultados en general de la encuesta de aceptación fueron positivos y satisfactorios, por lo que se puede concluir, que existe un claro interés e intención de usar la aplicación para los propósitos establecidos.

Finalmente, con el fin de aumentar funcionalidades, enriquecer y expandir la aplicación web, se sugieren las siguientes recomendaciones:

- Agregar dinamismo en las funcionalidades. Por ejemplo: Para crear un expediente y completar toda la información solicitada por la aplicación, debe previamente crearse el cliente. Sería ideal que sin necesidad de abandonar el formulario de creación de Expedientes y los datos ya introducidos, se pueda crear desde ese módulo, el cliente, al menos con los datos más básicos (nombres, apellidos, cédulas, o en el caso de que sea de tipo jurídico, nombre de la organización y rif).
- Enriquecer el módulo de agenda para crear eventos recurrentes y recordatorios que pueden ser enviados por correo electrónico al abogado, recordando lapsos del proceso legal. Asimismo, se recomienda agregar la funcionalidad de configuración de envío de mensajes de textos a demanda a dispositivos móviles.

- Expandir la aplicación para convertirla en una suite que tenga aplicaciones clientes para tipos distintos de dispositivos y todos puedan sincronizar en dos direcciones contra un repositorio principal.

Referencias bibliográficas

- [1] OLASO, Luis M. *Introducción al derecho*. Tomo I. Universidad Católica Andrés Bello. 1986.
- [2] Wikipedia. (2013). *Poder Judicial de Venezuela*. [Consultado 25/04/2013]. http://es.wikipedia.org/wiki/Poder_Judicial_de_Venezuela
- [3] OSSORIO, Manuel. *Diccionario de Ciencias Jurídicas, Políticas y Sociales*. Heliasta. 1999.
- [4] CABANELLAS, Guillermo. *Diccionario de Derecho Usual*. Tomos I A-D. Editorial Heliasta. 1976.
- [5] CABANELLAS, Guillermo. *Diccionario de Derecho Usual*. Tomo II E-M. Editorial Heliasta. 1976.
- [6] CABANELLAS, Guillermo. *Diccionario de Derecho Usual*. Tomo III N-R. Editorial Heliasta. 1976.
- [7] CABANELLAS, Guillermo. *Diccionario de Derecho Usual*. Tomo IV S-Z. Editorial Heliasta. 1976.
- [8] CONGRESO DE LA REPÚBLICA DE VENEZUELA. Gaceta Oficial N° 1.081. *Ley de Abogados*. 23 de enero de 1967.
- [9] CONGRESO DE LA REPÚBLICA DE VENEZUELA. Gaceta Oficial N° 4.209. *Código de Procedimiento Civil*. 18 de septiembre de 1.990.
- [10] FEDERACIÓN DE COLEGIOS DE ABOGADOS DE VENEZUELA. *Código de Ética Profesional del Abogado Venezolano*. 3vde agosto de 1.985.
- [11] GARCÍA MAYNEZ, E. *Introducción al Estudio del Derecho*. Editorial Porrúa, México, 1955.
- [12] MONTES, Félix y otros. *Sistema de Gestión de Pasantías para la Escuela de Computación*. Trabajo especial de grado. Escuela de Computación de la UCV. Mayo 2013.
- [13] NetFincas Software. 1999. *Abogafin*. [Consultado 01/06/2013]. <http://www.netfincas.com/netfincas+empresa.html>
- [14] Grupo ESW, C.A. 2013. *Sistema de Gestión Legal*. [Consultado 25/08/2013]. <http://www.sistemadegestionlegal.com/>

- [15] Sistemas Jurídicos SRL. 2012. *Lex-Doctor, Gestión jurídica*. [Consultado 25/08/2012]. http://www.lex-doctor.com/sobre_lex-doctor.php
- [16] Jurisoft. 2012. *Infolex*. [Consultado 11/08/2013]. <http://www.jurisoft.es/ie/infolex.aspx?menu=3>
- [17] Tantum. 2005. *Tantum, Sistema de Gestión Jurídica*. [Consultado 25/08/2013]. <http://www.tantum.com.ar/producto.php>
- [18] The Cloud Gate S.L. 2013. *Nubbius, Gestión eficiente de bufetes*. [Consultado 18/08/2013]. <http://nubbius.com/>
- [19] Brindys. 2012. *Gedex*. [Consultado 18/08/2012]. <http://www.brindys.com/es/>
- [20] MNProgram. (s.f.). *MNProgram Abogados*. [Consultado 25/08/2013]. <http://www.softwareabogado.net/>
- [21] Wikipedia. Julio 2012. *Presunción*. [Consultado 08/09/2013]. [http://es.wikipedia.org/wiki/Presunci%C3%B3n_\(derecho\)](http://es.wikipedia.org/wiki/Presunci%C3%B3n_(derecho))
- [22] JMPereda's Weblog. Agosto 2007. *¿Qué es una aplicación web?* [Consultado 15/11/2013]. <http://jimpereda.wordpress.com/2007/08/24/definiendo-la-plantilla/>
- [23] Boxesandarrows. Junio 2003. *What is a web application?* [Consultado 15/11/2013]. <http://boxesandarrows.com/what-is-a-web-application/>
- [24] Wikipedia. (s.f.). *Aplicación Web*. [Consultado 15/11/2013]. http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web
- [25] Login desarrollos. (s.f.). *Aplicaciones online*. [Consultado 15/11/2013]. <http://www.logindesarrollos.com/es/Servicios-aplicaciones-web-21>
- [26] Zef.me. (2005). *More Advantages of web Applications*. [Consultado 15/11/2013]. <http://zef.me/987/more-advantages-of-web-applications>
- [27] Magic Web solutions. *The benefits of web-based applications*. (s.f.). [Consultado 15/11/2013]. <http://www.magicwebsolutions.co.uk/blog/the-benefits-of-web-based-applications.htm>
- [28] LUJAN, Sergio. 2002. *Programación de aplicaciones web. Historia, principios básicos y cliente web*. [Consultado 16/11/2013]. <http://rua.ua.es/dspace/handle/10045/16995>

- [29] UDLAP Bibliotecas. Enero de 2004. *Implementación de un reconocedor de voz gratuito para el sistema de ayuda a invidentes Dos-Vox en español*. [Consultado 16/11/2013]. http://catarina.udlap.mx/u_dl_a/tesis/documentos/lis/marquez_a_bm
- [30] Wikipedia. (s.f.). *Client-server model*. [Consultado 16/11/2013]. http://en.wikipedia.org/wiki/Client%E2%80%93server_model
- [31] Dataprix. (s.f.). *Introducción al DB2*. [Consultado 23/11/2013]. <http://www.dataprix.com/introducci-n-db2>
- [32] Wikipedia. (s.f.). *Base de datos*. [Consultado 23/11/2013]. http://es.wikipedia.org/wiki/Base_de_datos
- [33] Wikipedia. (s.f.). *Database*. [Consultado 23/11/2013]. <http://en.wikipedia.org/wiki/Database>
- [34] SILBERSCHATZ, Abraham y otros. 2002. *Fundamentos de Bases de Datos*. [Consultado 23/11/2013]. https://docs.google.com/file/d/0BzOqmEzk7_bVZTgwYjkzZGYtMDQ4ZC00MDI2LWExYzAtYjdiNDFIYzdINTJh/edit?pli=1
- [35] CAMPS, Rafael y otros. *Base de Datos*. Primera edición. Eureka Media, SL. Barcelona, Mayo 2005. [Consultado 23/11/2013]. http://www.aulaexperta.com/ve/ae/usuarios/OLIVERSAN/BD_913.pdf
- [36] BURBECK, Steve. (1992). *How to use Model-View-Controller (MVC)*. [Consultado 15/12/2013]. <http://st-www.cs.illinois.edu/users/smarch/st-docs/mvc.html>
- [37] DEACON, Jhon. (2013). *Model View Controller (MVC) Architecture*. [Consultado 15/12/2013]. <http://www.idl.co.uk/briefings/MVC.pdf>
- [38] Wikipedia. (2013). *Modelo Vista Controlador*. [Consultado 15/12/2013]. http://es.wikipedia.org/wiki/Modelo_Vista_Controlador
- [39] Code Project. (2008). *Simple Example of MVC (Model Controller View) Design Pattern for Abstraction*. [Consultado 18/15/2013]. <http://codeproject.com/Articles/25057/Simple-Example-of-MVC-Model-View-Controller-Design>
- [40] GUTIERREZ, Javier. (s.f.). *¿Qué es un framework web?* [Consultado 18/12/2013]. http://lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf

- [41] Wikipedia. (2013). *Framework*. [Consultado 18/12/2013]. <http://es.wikipedia.org/wiki/Framework>
- [42] DB-engines. (2014). *Knowledgebase Base of Relational and NoSQL Database Management Systems*. [Consultado 01/01/2014]. <http://db-engines.com/en/ranking>
- [43] SERES, Andrei. (2010). *Three Database Management Systems (DBMS) compared*. [Consultado 02/01/2014] <http://www.opensourcejournal.ro/2010-Volume02/number04/paper005-fullpaper.pdf>
- [44] Wikipedia. (2014). *PostgreSQL*. [Consultado 20/02/2014]. <http://es.wikipedia.org/wiki/PostgreSQL>
- [45] Comunidad PostgreSQL-es. (2014). *PostgreSQL-es*. [Consultado 20/02/2014]. <http://www.postgresql.org/es/>
- [46] Wikipedia. (2014). *ACID*. [Consultado 01/01/2014]. <http://es.wikipedia.org/wiki/ACID>
- [47] IBM. (2014). *IBM Knowledge Center*. [Consultado 01/01/2014]. <http://www-01.ibm.com/support/knowledgecenter/?lang=es>
- [48] GOMEZ, Ramiro A. (s.f.) *White paper: Lenguajes de Programación*. [Consultado 03/01/2014]. <http://www.ilustrados.com/documentos/lenguajes-programacion-020608.pdf>
- [49] Wikipedia. (2014). *HTML*. [Consultado 03/01/2014]. http://es.wikipedia.org/wiki/HTML#Accesibilidad_web
- [50] JAISWAL, Abhishek. (2014). *Client-Side vs Server-Side Programming Languages*. [Consultado 17/01/2014]. <http://www.c-sharpcorner.com/UploadFile/2072a9/client-side-vs-server-side-programming-languages/>
- [51] Motive Glossary. (2004). *The Motive Web Design Glossary*. [Consultado 18/01/2014]. <http://www.motive.co.nz/glossary/client-server-php?ref>
- [52] FOTE, Bill. (2013). *Client-Side vs. Server-Side Code: What's the Difference?* [Consultado 19/01/2014]. <http://www.seguetech.com/blog/2013/05/01/client-side-server-side-code-difference>

- [53] Python Foundation Software. (2014). *Python*. [Consultado 12/04/2013]. <http://www.python.org>
- [54] Wikipedia. (2013). *Python*. [Consultado 20/04/2013]. <http://es.wikipedia.org/wiki/Python>
- [55] ALVAREZ, Miguel. (2013). *¿Qué es Python?* [Consultado 20/04/2013]. <http://www.desarrolloweb.com/articulos/1325.php>
- [56] Python Foundation Software. (2013). *The Python Tutorial*. [Consultado 12/04/2013]. <http://docs.python.org/3/tutorial/index.html>
- [57] Python Foundation Software. (2013). *Beginners Guide*. [Consultado 10/04/2013]. <https://wiki.python.org/moin/BeginnersGuide>
- [58] Python Foundation Software. (2013). *The Python Language Reference*. [Consultado 10/04/2013]. <http://docs.python.org/3/reference/index.html>
- [59] Universitat Jaume I. (2010). *Python: Conceptos básicos y ejercicios*. [Consultado 20/04/2013]. <http://www3.uji.es/~vjimenez/PL/python-by-vilar.pdf>
- [60] CASADO, Manuel y otros. (2003). *El lenguaje de programación Python*. [Consultado 25/04/2013]. <http://zarza.fis.usal.es/~fgarcia/docencia/poo/02-03/trabajos/S1T8.pdf>
- [61] BECERRA, Andrés. (2009). *Introducción a la programación con Python*. [Consultado 20/04/2013]. <http://cic.puj.edu.co/wiki/lib/exe/fetch.php?media=webpages:abecerra:introprog-py.pdf>
- [62] CANOS, Penadés. (2013). *Metodologías ágiles en el desarrollo de software*. [Consultado 30/03/2013]. <http://noqualityinside.com.ar/nqi/nqifiles/XP:Agil.pdf>
- [63] Wikipedia. (2014). *Metodologías ágiles de desarrollo de software*. [Consultado 20/03/2013]. http://es.wikipedia.org/wiki/Desarrollo_%C3%A1gil_de_software
- [64] NIELSEN, Jacob. (1995). *10 Usability Heuristics for User Interface Design*. [Consultado 25/04/2014]. <http://www.nngroup.com/articles/ten-usability-heuristics/>
- [65] Django Software Foundation. (2013). *Django en español*. [Consultado 22/06/2013]. <http://django.es>

- [66] Django Software Foundation. (2013) *Django*. [Consultado 21/06/2013]. <http://www.djangoproject.com>.
- [67] Django Software Foundation. (2013). Django documentation. [Consultado 28/06/2013]. <https://docs.djangoproject.com/en/1.7/intro/tutorial01/>
- [68] ARAUJO, Miguel y Daniel Greenfeld. (2013). *Forms have never been this crispy*. [Consultado 10/10/2013]. <http://django-crispy-forms.readthedocs.org/en/latest/>
- [69] BALCERZAK, Lukasz. (2013). *Django-guardian*. [Consultado 10/5/2014]. <http://django-guardian.readthedocs.org/en/v1.2/>
- [70] BENNET, James. (2013). Django-registration. [Consultado 01/03/2014]. <http://django-registration.readthedocs.org/en/latest/>
- [71] AGUIRRE, Matías. (2011). *Django Social Auth*. [Consultado 25/03/2014]. <https://django-social-auth-readthedocs.org/en/latest/>
- [72] Wikipedia. (2014). *JavaScript*. [Consultado 02/01/2014]. <http://es.wikipedia.org/wiki/JavaScript>
- [73] The JQuery Foundation. (2014). *jQuery write less, do more*. [Consultado 02/01/2014]. <http://jquery.com>
- [74] Wikipedia. (2014). *jQuery*. [Consultado 02/01/2014]. <http://es.wikipedia.org/wiki/jQuery>
- [75] Bootstrap contributors. (2014). *Bootstrap*. [Consultado 02/01/2014]. <https://getbootstrap.com>
- [76] Wikipedia. (2014). *Computación en la nube*. [Consultado 13/09/2013]. http://es.wikipedia.org/wiki/Computacion_en_la_nube
- [77] Heroku Inc. (s.f.). *Heroku*. [Consultado 31/08/2013]. <https://www.heroku.com>
- [78] Wikipedia. (2014). *Casos de uso*. [Consultado 01/02/2014]. http://es.wikipedia.org/wiki/Caso_de_uso
- [79] ACOSTA, Alecia Eleonora. (2011). *Un método ágil de desarrollo de software que incorpora la usabilidad*. [Consultado 20/02/2013]. http://www.ciens.ucv.ve:8080/genasig/sites/interaccion-humano-comp/archivos/234_CLEI_Acosta_Paper.pdf

- [80] TOGNINI, Juan Carlos. (2013). *Cloud Computing: Servicios actuales en la Nube*. [Consultado 15/09/2013]. <http://www.informatica-hoy.com.ar/la-nube/Cloud-Computing-Servicios-actuales-en-la-Nube.php>.
- [81] Heroku.com. (s.f.). *Heroku Dev Center*. [Consultado 01/09/2013]. <https://devcenter.heroku.com>
- [82] Wikipedia. (2013). *Heroku*. [Consultado 01/09/2013]. <http://es.wikipedia.org/wiki/Heroku>.

Anexos

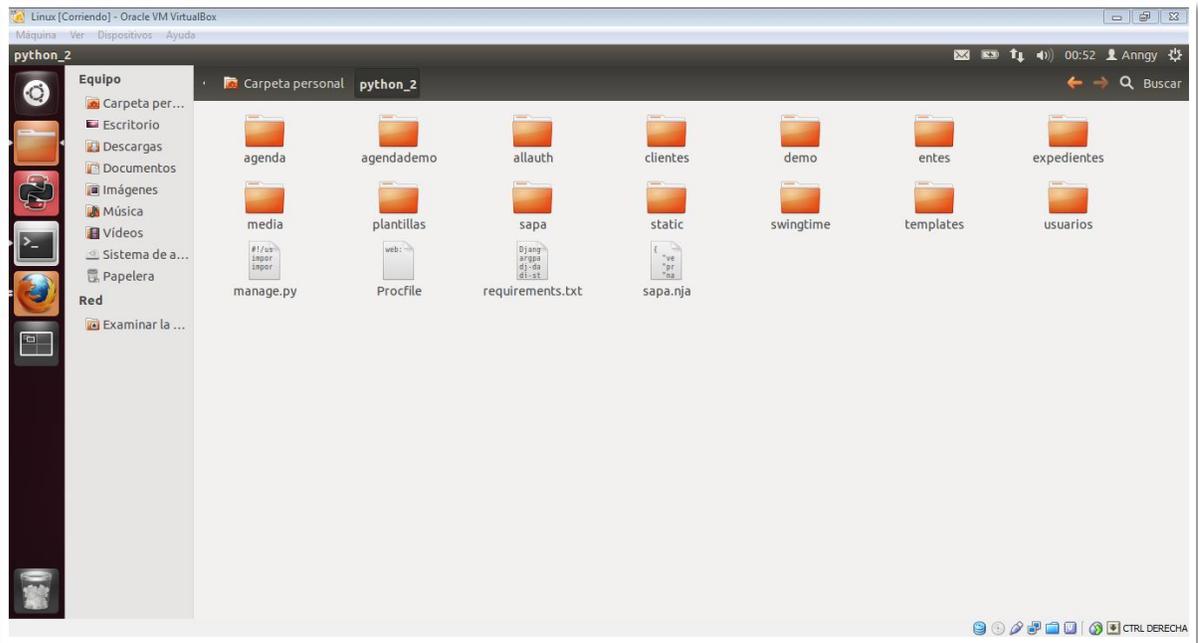


Figura 36: Sistema de archivos del proyecto de la aplicación en Django

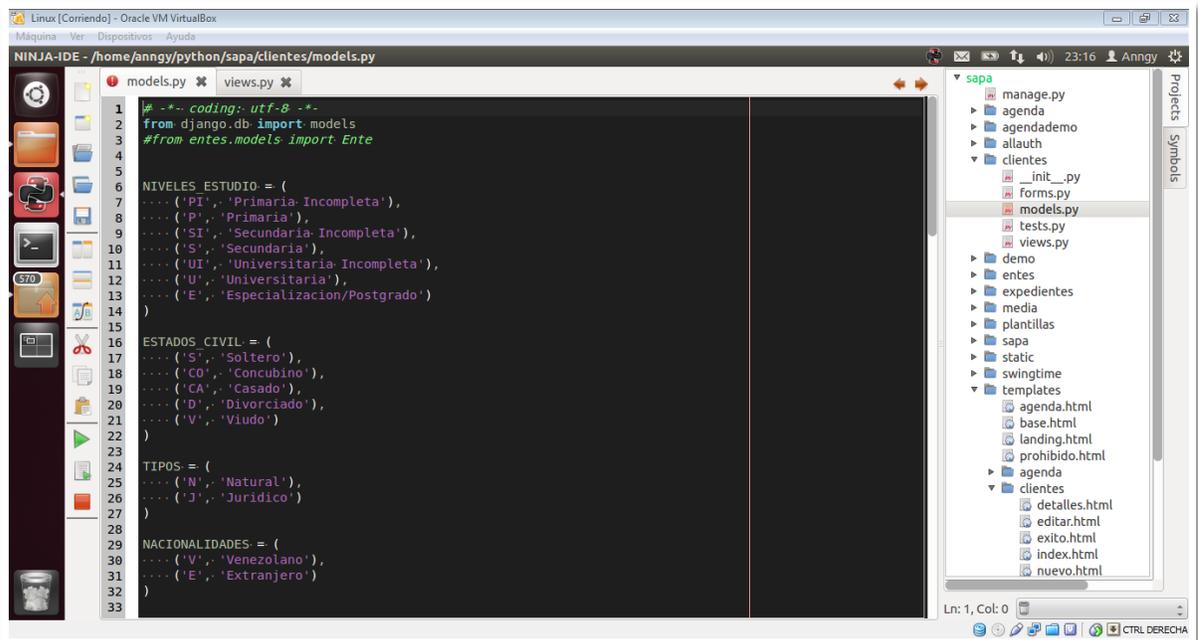


Figura 37: Archivo models.py de la aplicación clientes en Django