PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
FACULTAD DE INGENIERÍA
ESCUELA DE SISTEMAS Y COMPUTACIÓN

ANÁLISIS, DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA PARA CONSULTORÍA JURÍDICA. CASO DE ESTUDIO: ESTUDIO JURÍDICO DEL DR. JAVIER IRIGOYEN HURTADO

ACOSTA IRIGOYEN ANA MARÍA

DISERTACIÓN PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

DIRECTOR: ING. FABIÁN DE LA CRUZ

QUITO, JUNIO 2019

Dedicatoria

A Dios, el más importante.

A mi abuelito, quien me enseñó a ser una persona correcta y me motivó a ser la mejor de las profesionales.

A mi madre, la persona más fuerte, valiente, inteligente y decidida que conozco.

A mi padre que siempre ha sido comprensible y ha luchado mucho.

A mi hermano, mi ejemplo a seguir de toda la vida.

Agradecimientos

A Dios y a mi familia por su ayuda incondicional, los consejos y esfuerzos.

A mi director, el Ing. Fabián, y a mis revisores, Ing. Alarcón e Ing. Chafla, quienes me ayudaron a pulir mi disertación con sus correcciones.

A mi hermano, Sebastián Acosta que me ha enseñado exigentemente desde pequeña gramática, ortografía, redacción e investigación, pero más importante aún, quien siempre está a mi lado a pesar de todo, quien me cuida, me protege y a quien amo incondicionalmente.

A mi tío Javier Irigoyen, quien generosamente me colaboró con su Estudio Jurídico.

A Jean Pierre Del Castillo, quien me ha apoyado, acompañado y aconsejado; y a mis amigos, quienes me han enseñado muchas cosas.

A los docentes de la facultad de Ingeniería, con quienes tuve el gusto de compartir varios semestres y aprender nuevos conocimientos.

Tabla de contenidos

Índice de tablas

Índice de figuras

Capítulo 1. Marco teórico

Para tener una mejor comprensión del presente proyecto, es necesario abarcar los aspectos técnicos para especificar y definir las herramientas, metodologías y arquitectura que serán de utilidad para desarrollar el producto final. También contiene los aspectos legales, donde se proporciona una definición del derecho, se relata un contexto histórico del derecho laboral y se detallan sus principios, además de la estructura del código del trabajo ecuatoriano.

1.1. Aspectos técnicos

Para el desarrollo técnico de la presente disertación se requiere de la utilización de un paradigma de desarrollo basado en tres capas, una de modelo, otra de vista y una de controlador, llamado MVC, tal como se muestra en la Figura 1. Al trabajar con este paradigma, se obtienen beneficios para la aplicación que se está desarrollando, como la programación independiente entre cada capa, mejor mantenimiento, conexión en tiempo de compilación, entre otras (Fernández y Díaz, 2012).

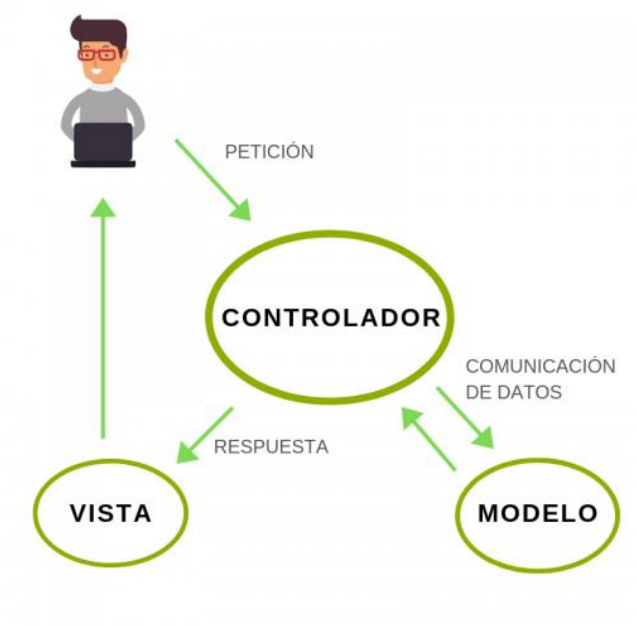

Figura 1 Bobb, N. (2020). *MVC (Modelo Vista Controlador)* [Figura]. Recuperado de https://nicobobb.com/mvc/

La capa de modelo se ocupa únicamente del manejo, control y gestión de datos de la aplicación, por lo cual no se preocupa de gestionar o conectar las demás capas, ni de la conexión entre ellas (Fernández y Díaz, 2012). Se encarga de acceder a la información almacenada en las bases de

datos y mostrar en las vistas los cambios que afectaron a los datos. Adicionalmente, para el desarrollo de la aplicación que se realizará en la presente disertación y que se llamará ***LEGAL HELP,*** se requerirá del uso de una base de datos relacional SQL mediante el motor de base de datos MySQL, que funcionará como herramienta gestora de datos para la capa de modelo.

La capa de vistas se encarga de la interfaz gráfica de usuario. Ésta presenta los datos del modelo esperando que el usuario interactúe con los mismos para que la capa de controlador realice cambios en los datos o que los presente del modo en que el usuario desee. Se espera que esta capa interactúe con la de controlador para la obtención de datos a mostrar, mas no directamente con la de modelo, ya que eso no es necesario en la interacción con el usuario. En el desarrollo de esta aplicación se hará uso del lenguaje de programación JAVASCRIPT, mediante un framework llamado ExtJS, que fungirá principalmente como capa de vista.

Finalmente, la capa de controlador es la que tiene el dominio del problema y según Fernández y Díaz (2012) le da significado a las entradas que proporciona el usuario, mediante funciones que obtienen datos o que actúan con los mismos datos a modo de parámetros para su manipulación. Para el desarrollo de esta entidad se hará uso del lenguaje de programación llamado PHP, mediante el IDE PhpStorm de JetBrains. Éste es un software con licencia, sin embargo, para el desarrollo de la aplicación del presente trabajo de disertación se hará uso de una licencia de estudiante que tiene una duración anual, es renovable en el caso de seguir siendo estudiante.

1.1.1. Metodologías de desarrollo

El desarrollo de un sistema de software debe tener calidad en sus procesos, por lo cual, si se desea obtener un sistema libre de defectos, eficaz y eficiente se requiere seguir una metodología que colabore con elementos que faciliten un control y planificación adecuados que, además sea acorde al tipo de sistema que se está desarrollando (Humphrey, 2001).

En general, cualquier metodología de desarrollo se basa en especificar los requerimientos, diseñar, construir y probar el producto. Ante esto existen varios modos de desarrollo, tomando en cuenta factores como el tiempo, el costo, la relación con el cliente, los desarrolladores disponibles, el tamaño del proyecto, entre otros. Cuando se desea llevar a cabo un proyecto de desarrollo pequeño o mediano en un corto tiempo, con pocos desarrolladores y una intervención frecuente del cliente, es recomendable usar una metodología de desarrollo ágil.

Las metodologías de desarrollo ágil están orientadas hacia productos o proyectos que son desarrollados y entregados de manera incremental. Se desarrollan de dicha manera debido a que

el cliente está involucrado en el proceso para proporcionar constantemente requerimientos nuevos que se necesiten aplicar al sistema. Es por esta necesidad frecuente de cambios que los desarrolladores del proyecto deben aceptar dichas modificaciones de manera positiva y alinearse con dicho proceso para cumplir cronogramas.

De acuerdo con Sommerville (2011): “En la práctica, los principios que subyacen a los métodos ágiles son a veces difíciles de cumplir" (p.60). Esto se debe a que generalmente el compromiso de las personas que desarrollan el proyecto y el cliente difiere por cuestión de tiempo, prioridades, presiones por parte de terceros, entre otras. Otro tipo de conflictos que se ven en la práctica se da por parte de los desarrolladores y su capacidad para adaptarse a trabajar con un equipo acorde al ritmo ágil, que sepa priorizar y acoplar los cambios en procesos definidos a priori.

Las características que más se destacan dentro de las metodologías ágiles son:

- Carencia de documentación formal. Prima el documento de requerimientos, donde se establece que hace o debería hacer la aplicación.
- El documento de requerimientos se forma a partir de las necesidades que solicita el cliente a medida que se va desarrollando la aplicación.
- Debido a que los requerimientos son dados de manera incremental, el diseño también debe serlo. Estar acorde con las entregas hasta alcanzar un producto final aprobado por el cliente es lo fundamental.
- Simplicidad en los procesos y en el desarrollo de software.
- Los desarrolladores deben poseer talentos y habilidades, como: alto conocimiento, enfoque hacia una misma meta, colaboración, capacidad para la toma de decisiones, y para la resolución de problemas, confianza, respeto, organización y sobre todo compromiso. Dichas cualidades son las que les permitirán realizar su trabajo de manera óptima aprovechando la metodología ágil como facilitador de los procesos (Pressman, 2005). Cabe mencionar que Según Pressman (2005), “(…) el proceso se ajusta a las necesidades de las personas y del equipo, y no al revés”(p.83).

Uno de los métodos ágiles más conocidos es la Programación Extrema o conocido mayormente como XP (*Xtreme Programming*). Este método fue publicado en 1999 por Kent Beck y consta de 4 fases:

- Planeación: aquí se definen *historias de usuario* por el cliente, quien le da una prioridad mientras el equipo de desarrollo otorga un valor llamado costo que se cuantifica en

semanas, luego se agrupan las historias para cada entrega próxima y se realizan acuerdos entre todos

- Diseño: aquí se hace uso de tarjetas CRC (colaborador/ responsabilidad/ clase) y se considera una fase de cambio continuo.
- Codificación, aquí se crean pruebas de unidad, se programan las historias acordadas y se unen las partes de cada miembro del equipo frecuentemente.
- Pruebas: en esta última fase se realizan pruebas de integración, de validación y de aceptación las cuales van de la mano con las historias de usuario.

A pesar de que la metodología mencionada puede ser aplicada a cualquier proyecto que guarde las características antes mencionadas, hoy en día, el desarrollo de aplicaciones de teléfono es un mercado que está creciendo con fuerza y está en mejora continua. Lo que diferencia a un sistema web de una aplicación móvil está en:

- Alto nivel de competitividad.
- Entregas en poco tiempo.
- Dificultad en identificar los requerimientos de los clientes y los interesados.

Desarrollar aplicaciones móviles implica considerar cambios frecuentes por parte de las necesidades del cliente, así como las restricciones tecnológicas por el ritmo acelerado en que las herramientas de desarrollo crecen y cambian entre cada versión. También se deben examinar los cambios de características de los distintos tipos de *hardware y software*, como por ejemplo, un dispositivo Samsung cuyo sistema operativo es Android tiene una arquitectura y software diferente a un iPhone, cuyo sistema operativo es IOS. Del mismo modo puede haber dispositivos con sistemas operativos compatibles, pero arquitecturas diferentes como Xiaomi o Samsung, quienes comparten a Android.

Es por ello que es más factible utilizar una metodología ágil orientada al desarrollo de aplicaciones móviles, en la Tabla 1 se resumen las más conocidas:

Tabla 1

Metodologías de Desarrollo de Software orientado a móviles
o Modelo Waterfall (en cascada) o Requerimientos fijos. o No posee retroalimentación entre cada fase. o Las fases se concluyen secuencialmente. o Posee fechas fijas de entrega.

- No prevé incertidumbre.

- Desarrollo rápido de aplicaciones
  - Las funcionalidades son incrementables.
  - Uso de patrones de diseño.
  - Prevé cambios en los requerimientos.
  - Plazos de entrega son cortos.
  - Usado para prototipos.

- Desarrollo ágil
  - Ritmo cambiante.
  - Interacción en el proceso controlable.
  - Buena comunicación entre los miembros del equipo.
  - Tiempos de entrega cortos para cada artefacto.
  - Se obtiene retroalimentación constante.

- Mobile-D
  - Ciclos de desarrollo rápidos.
  - Equipos de desarrollo pequeños.
  - Basado en la metodología XP.
  - Sus fases comprenden:
    - Exploración
    - Inicialización
    - Fase de producto
    - Fase de estabilización
    - Fase de pruebas

Fuente: Elaboración propia

1.1.1.1. Metodología de desarrollo Mobile-D

Para el desarrollo de la aplicación ***LEGAL HELP*** se utilizará la metodología Mobile-D, ya que, al ser una técnica orientada al desarrollo de software móvil incremental, provee una serie de ventajas adecuadas al presente proyecto como: mantenibilidad, tiempos de entrega reducidos, requiere un grupo pequeño de desarrolladores de hasta máximo 10 integrantes, entre otras.

La metodología de desarrollo Mobile-D, tiene las fases de: exploración, inicialización, de producto, de estabilización y de pruebas. En la Figura 2 se puede apreciar su secuencialidad y las partes que componen cada fase.

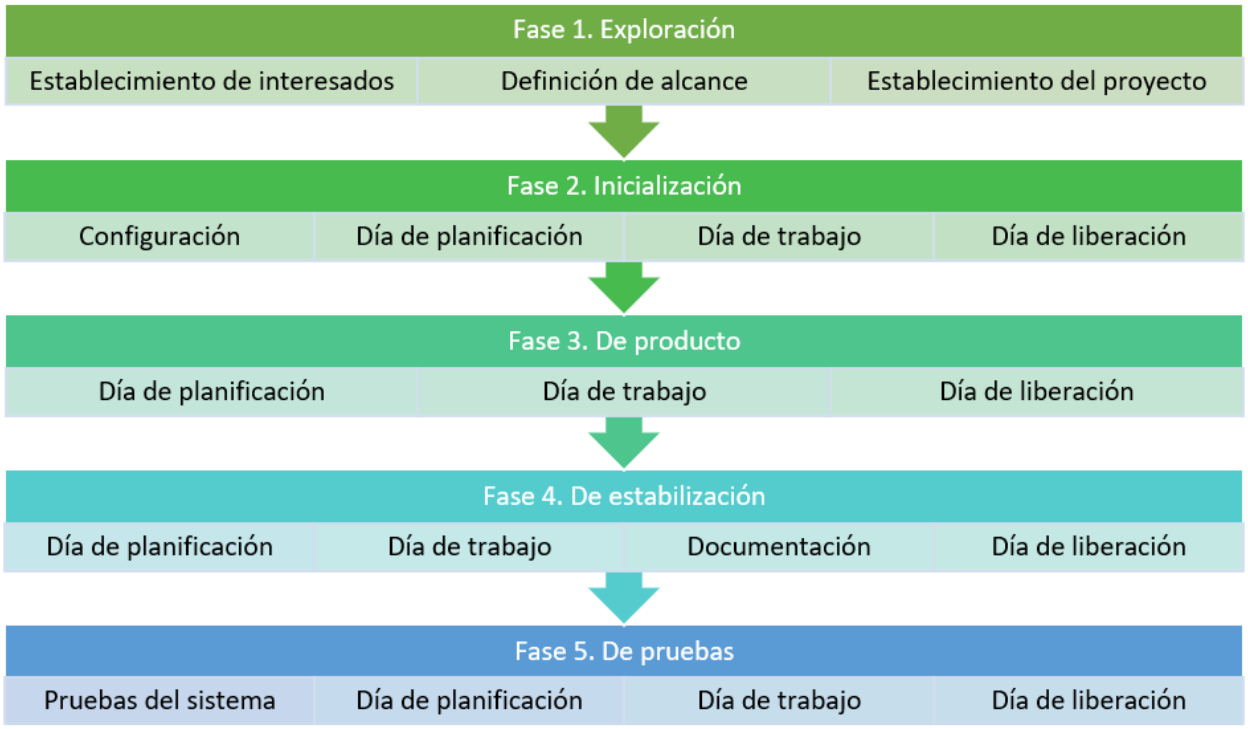

Figura 2 Fases de metodología de desarrollo Mobile-D

a. Fase de exploración:

Aquí, tanto el equipo de desarrollo como los clientes toman un papel importante para la construcción de un plan inicial gracias a la definición y entendimiento del proyecto (Gómez Medina y Hernández Zuleta, 2016). Destacan estas partes:

I. Establecimiento de interesados:

El propósito de esta sección es identificar los grupos de interés a lo largo del desarrollo del proyecto, estos son todos aquellos que se ven afectados de alguna manera con el proyecto, ya que cada uno de estos actores aporta de manera particular al progreso del proyecto (Gómez Medina y Hernández Zuleta, 2016).

Como una de las primeras fases se debe identificar cuáles son las necesidades y expectativas que tienen los clientes acerca del proyecto para establecer los requerimientos del sistema (Gómez Medina y Hernández Zuleta, 2016).

II. Definición de alcance:

En la segunda fase se debe realizar un breve levantamiento de los requerimientos y definición de los límites del proyecto, de modo que se pueda comprender ¿qué se pretende construir?, ¿para qué? y ¿por qué? (Gómez Medina y Hernández Zuleta, 2016).

III. Establecimiento del proyecto:

Aquí se define la línea base y los recursos arquitectónicos, humanos y técnicos que serán requeridos para llevar a cabo el desarrollo del proyecto. Los procesos de establecimiento consisten en:

- Establecer la línea de proceso base,
- Planificar la documentación necesaria,
- Planificar el seguimiento y la medición,
- Identificar las necesidades de formación (Gómez Medina y Hernández Zuleta, 2016).

Los documentos resultantes de esta fase son los pilares para la continuidad del proceso. El más importante es el documento de requerimientos iniciales. También se obtiene el plan de proyecto, el documento de la descripción de línea base, la lista de actividades de aseguramiento de la calidad y por último el plan de medición y de formación (Villanueva, 2015).

b. Fase de inicialización:

El propósito de esta fase es que el equipo de desarrollo adquiera una comprensión general de los requerimientos iniciales y la línea de la arquitectura; y además, se busca identificar los recursos para la planificación, desarrollo y publicación (Villanueva, 2015).

Esta fase se divide en 3 subfases, primero *la puesta en marcha del proyecto*, que se caracteriza por la configuración de los recursos físicos y técnicos, las capacitaciones del equipo y la comunicación con los clientes; luego está la *planificación inicial*, que se caracteriza por la planificación de la arquitectura y análisis de los requerimientos; y por último, en el *día de prueba*, se realizan pruebas de módulo y de funciones del sistema para asegurar la calidad de implementación del sistema (Gómez Medina y Hernández Zuleta, 2016).

Los documentos resultantes de la fase de inicialización son: el plan de proyecto actualizado, la primera versión del diseño de software, la descripción del diseño, el documento de requerimientos iniciales, la lista de desarrollo de interfaz de usuario y las pruebas de cada requerimiento (Villanueva, 2015).

c. Fase de producto:

En esta fase se repite iterativamente el proceso: planificación, desarrollo y publicación, hasta finalizar la implementación de las funcionalidades descritas en el documento de requerimientos (Gómez Medina y Hernández Zuleta, 2016).

Esta fase está compuesta de los siguientes días:

I. Día de planificación:

Comprende el análisis de requerimientos, la revisión de los factores necesarios para realizar las pruebas de aceptación y la planificación de las iteraciones y tareas para mejorar el proceso posterior para cada fase (Gómez Medina y Hernández Zuleta, 2016).

II. Día de trabajo:

En esta etapa se desarrollan las funcionalidades previstas en el documento de requerimientos de acuerdo a la prioridad más alta dada por el cliente, contemplando la programación en parejas, la integración continua y los informes al cliente, con el fin de mejorar la comunicación con el cliente y mejorar el desarrollo, la programación y la implementación (Gómez Medina y Hernández Zuleta, 2016).

III. Día de publicación:

El lanzamiento da como resultante una versión funcional del sistema, en la que constan: la integración del sistema, las pruebas de prelanzamiento y de aceptación, además de una auditoría de lanzamiento y la creación de una línea base, llamada ceremonia de lanzamiento (Gómez Medina y Hernández Zuleta, 2016).

Los documentos resultantes en la etapa de producto son: el documento de aceptación de pruebas, las notas de desarrollo, las ilustraciones de interfaz de usuario, la lista de puntos de acción, el plan de proyecto actualizado, la historia y tarjetas de tareas, la lista de defectos, el documento de requerimientos y el informe de estado diario (Villanueva, 2015).

d. Fase de estabilización:

Esta etapa juega un papel mucho más importante cuando el proyecto está conformado por varios equipos que desarrollan módulos o subsistemas para construir el producto completo, de todas maneras, es de gran utilidad para el presente proyecto, la cual es elaborada por una sola persona. En sí, la fase de estabilización se encarga únicamente de llevar a cabo la integración del sistema

para finalizar la implementación, asegurar la calidad, y finalizar la documentación del proyecto (Gómez Medina y Hernández Zuleta, 2016). Aquí se obtiene el documento del producto finalizado.

e. Fase de pruebas:

En este último paso es importante hacer una revisión paralela con los requerimientos, de modo que se pueda verificar el correcto desenvolvimiento de cada uno de ellos y realizar las pruebas respectivas de funcionamiento, pruebas de módulos, de integración, de sistema y de aceptación. En pocas palabras, esta fase se encarga de identificar, documentar y corregir los errores del sistema mientras optimiza la calidad del sistema, proporcionando una versión estable del software (Gómez Medina y Hernández Zuleta, 2016).

Finalmente se obtiene el documento de errores y de pruebas del sistema.

1.1.2. Herramientas de desarrollo

Como se mencionó anteriormente, se hará uso de diferentes herramientas que apoyarán el desarrollo de las capas del modelo, vista y controlador de la aplicación ***LEGAL HELP***.

A continuación, se puede observar en la Tabla 2 un resumen de los IDE´s y los lenguajes de programación que se usarán:

Tabla 2

*mientas de desarrollo de la aplicación **LEGAL HELP***

Capa	Herramienta	Lenguaje
Modelo	SQLyog Ultimate – MySQL GUI v11.11 (64 bit) PhpStorm 2018.1.6	MariaDB 10.1.36 Codeigniter 3.1.10
Vista	Sencha Architect 4.2.4	Ext JS 6.6.x Modern
Controlador	PhpStorm 2018.1.6	Codeigniter 3.1.10

Fuente: Elaboración propia

Figura 5 R2 Data Technology. (2018). *Sencha Ext JS* [Figura]. Recuperado de https://www.r2datatechnology.com/producto/sencha-ext-js/

Figura 4 Vergara, M. (2016). *Qué es Codeigniter y cuáles son algunas de sus ventajas* [Figura]. Recuperado de https://www.coriaweb.hosting/codeigniter-cuales-algunas-ventajas/

Figura 3 Seravo. (2015). *10 reasons to migrate to MariaDB (if still using MySQL)* [Figura]. Recuperado de https://seravo.fi/2015/10-reasons-to-migrate-to-mariadb-if-still-using-mysql

Como se puede ver en la tabla, tanto para la capa de modelo y controlador se necesitará del framework de PHP[1] llamado Codeigniter y para la capa de vista el framework de JavaScript. Los marcos de trabajo o frameworks fueron desarrollados para minimizar la carga al programar el código que podría ser reutilizado. Éstos proporcionan una estructura predefinida de componentes y demás elementos complejos para la construcción de una aplicación web completa, y se orientan bajo modelos de desarrollo como el paradigma MVC que se mencionó antes.

Según Gutiérrez (2014), la mayoría de frameworks web se encargan de ofrecer una capa de controladores acorde a un patrón de desarrollo, de modo que la integración entre ellas se facilite.

ExtJS es un framework de JavaScript[2] que como muchos otros marcos de trabajo ofrece una serie de componentes para construir una aplicación. Ahora bien, cabe recalcar que ExtJS también es adaptativo para la construcción de aplicaciones web y móviles de datos intensivos (data-intensive) para cualquier tipo de dispositivo, de acuerdo con la página oficial del framework desarrollado por Sencha.

El IDE que se utiliza es conocido como Sencha Architect y es un software con licencia. Su costo es de $1.200 dólares por una licencia *pro* individual; a pesar de ello existe una licencia *community* gratis que dura tan solo 30 días. Para el desarrollo de la aplicación de la presente disertación se dispone de una licencia *pro* otorgada gracias a la empresa Maz – Tecnologías de información.

Para realizar las pruebas del sistema se hará uso de una herramienta conocida como SonarCloud, caracterizada por ser un servicio en la nube basado en SonarQube, una plataforma para inspeccionar continuamente la calidad del código fuente, detectando defectos, vulnerabilidades y fallas de seguridad (Travis CI, 2020). SonarCloud soporta lenguajes como PHP, Javascript, Java, C#, C, C++, entre otros más.

1.2. Aspectos jurídicos

La palabra derecho se deriva del latín «directum», que deviene en el verbo «dirigere», que significa caminar recto con dirección a una meta (Robles Velasco, 2016). También se puede considerar como un conjunto de principios que rigen una serie de normas y leyes enfocadas en el comportamiento humano; con el fin de alcanzar un concepto de justicia ideal, en un contexto y en un espacio temporal dado.

[1] PHP (Preprocesador de hipertexto), es un lenguaje de programación que está vinculado con el desarrollo de páginas Web y con el lenguaje HTML (Gutiérrez Gallardo, 2004).

[2] JavaScript, es un lenguaje de programación potente de archivos de comandos o script que permite añadir unidades funcionales al sistema para simplificar y automatizar la ejecución de las funciones.

Según Lalanne (como menciona Páez Benalcázar, 2019), “los principios jurídicos son aquellas proposiciones práctico-normativas (…) que expresan criterios de conducta en forma sintética, indeterminada y abstracta, y que, (…) operan como fundamento, explicación e instancia de unificación de todas las demás normas e instituciones del orden jurídico” (Páez Benalcázar, 2019, p.15).

El derecho posee una serie de ramificaciones que se encargan de los principios, normas, reglamentos y resolución de áreas determinadas, tales como: la penal, el constitucional, el civil, la laboral, entre otros más. Para el desarrollo del presente proyecto se hará un enfoque al derecho laboral.

1.2.1. Contexto histórico del derecho laboral

Primero se destaca la revolución industrial, (siglo XVIII-XIX) época primordial para el progreso y desarrollo humano a gran escala por la inserción de las máquinas en las industrias, las mismas que reemplazaban la mano de obra del trabajador. Esta revolución generó conflictos laborales como el abuso contra el trabajador obligándolos a exceder la carga horaria bajo pésimas condiciones humanas que generalmente causaban accidentes debido a las maquinarias. Esta situación generó el inicio de protestas y la creación de partidos de obreros que trataban de velar por sus derechos, creándose las primeras normas a favor de los trabajadores (Reyes Mendoza, 2012).

En 1919, se creó la Organización Internacional del Trabajo, tras el fin de la Primera Guerra Mundial, acorde con la Conferencia de Paz de Versalles (Reyes Mendoza, 2012). Este fue un organismo común para países de modo que se pudiesen gestionar las condiciones laborales internacionales, que en esa época eran fuertemente necesarias de definir, debido a tres razones fundamentales que se explican a continuación. (Yáñez Andrade, 2000).

Primero, los trabajadores resultaron ser uno de los actores más perjudicados por la guerra, a ellos se les demandó el incremento de producción con los mismos salarios y un aumento de la jornada laboral (Yáñez Andrade, 2000). Luego aparecieron ideas de regulación, donde tanto trabajadores y empresarios dialogaron y negociaron buscando afianzar la producción para el beneficio del país (Yáñez Andrade, 2000, p.320). Por último, la cooperación entre países, que tiene como propósito forjar un orden económico entre los mismos para mantener la ventaja competitiva balanceada (Yáñez Andrade, 2000).

Paralelamente en Ecuador, en 1922, grupos sindicales reclamaron sus derechos laborales, como mejoras salariales y de condiciones laborales para los indígenas y campesinos. Tres años más tarde se creó el Ministerio de Trabajo por la primera Junta de Gobierno Plural tras la Revolución Juliana, acorde con Avilés Pino (n.d.), dicho ministerio tenía como guía la constitución de la Organización Internacional del Trabajo.

En 1928, se crearon leyes de protección social, como la ley de Desahucio del Trabajo, Jornada de Trabajo y Descanso Obligatorio, la Ley del Contrato Individual de Trabajo, entre otras más que colaboraron a la posterior creación del Código del Trabajo (Avilés Pino, n.d.).

En 1929, la Asamblea Constituyente proclamó una serie de pilares para la posterior construcción del Código de Trabajo que estipulaba la "duración máxima de la jornada, el descanso semanal, prevención de accidentes, trabajo de mujeres y menores, protección de la maternidad, jubilación, montepío civil, ahorro y cooperativa" (Pérez Ramírez, 2014). En 1937, Virgilio Guerrero Espinosa, Miguel Ángel Zambrano y Néstor Mogollón realizaron el primer Código de Trabajo, que ha sido objeto de varias codificaciones oficiales con el paso de los años.

1.2.2. Principios del derecho del trabajo

Según Pacheco (como se citó en Páez Benalcázar, 2019), los principios del derecho del trabajo "son reglas protectoras que informan la elaboración de las normas de las de carácter laboral, amén de servir de fuente de inspiración directa en la solución de conflictos, sea mediante la interpretación, aplicación o integración de normativas" (Páez Benalcázar, 2019, p.21).

Existen varios modelos que clasifican los principios del derecho del trabajo, pero principalmente se las puede categorizar como se muestra en la Tabla 3.

Tabla 3

Principios del derecho del trabajo

Principio
Tutelar o protectorio
Irrenunciabilidad de los derechos
Continuidad o conservación de la relación de trabajo
Primacía de la realidad
Razonabilidad

Buena fe

Intangibilidad

Progresividad

No discriminación

Derecho colectivo del Trabajo o Derecho Sindical

Fuente: Elaboración propia

Los principios mencionados anteriormente pueden empatarse con el código del trabajo ecuatoriano, actualizado en agosto del 2019, que contiene lo siguiente:

I. Del contrato individual de trabajo
    a. De su naturaleza y especies
    b. De la capacidad de contratar
    c. De los efectos del contrato de trabajo
    d. De las obligaciones del empleador y del trabajador
    e. De la duración máxima de la jornada de trabajo, de los descansos obligatorios y de las vacaciones
    f. De los salarios, de los sueldos, de las utilidades y de las bonificaciones y remuneraciones adicionales
    g. Del trabajo de mujeres y menores
    h. De los aprendices
    i. De la terminación del contrato de trabajo
    j. Del desahucio y del despido
    k. Del fondo de reserva, de su disponibilidad y de la jubilación
II. Del contrato colectivo de trabajo
    a. De su naturaleza, forma y efectos
    b. De la revisión, de la terminación y del incumplimiento del contrato colectivo
    c. Del contrato colectivo obligatorio
III. De las modalidades de trabajo
    a. Del servicio doméstico
    b. Del trabajo a domicilio
    c. De los artesanos

d. De los empleados privados
e. De los agentes de comercio y corredores de seguros
f. Trabajo en empresas de transporte
g. Del trabajo agrícola

IV. De los riesgos del trabajo
a. Determinación de los riesgos y de la responsabilidad del empleador
b. De los accidentes
c. De las enfermedades profesionales
d. De las indemnizaciones
e. De la prevención de los riesgos, de las medidas de seguridad e higiene, de los puestos de auxilio, y de la disminución de la capacidad para el trabajo

V. De las asociaciones de trabajadores y de los conflictos colectivos
a. De las asociaciones de trabajadores
b. De los conflictos colectivos

VI. Organización, competencia y procedimiento
a. De los organismos y de las autoridades
b. De la administración de justicia
c. De la competencia y del procedimiento

VII. De las sanciones

VIII. Del desistimiento, del abandono y de la prescripción.

(Corporación de Estudios y Publicaciones, 2019)

Capítulo 2. Estudio Jurídico del Dr. Javier Irigoyen Hurtado

En el presente capítulo se proporciona información del estudio jurídico del Dr. Javier Irigoyen Hurtado, se describe su situación actual, problemáticas, proceso de la empresa, y entender de qué manera el estudio jurídico requiere de un enfoque tecnológico en su empresa y disponer de una aplicación móvil que facilite la asesoría legal y los pagos respectivos.

2.1. Situación actual

El Dr. Javier Irigoyen Hurtado, *Licenciado en Ciencias Públicas y Sociales* y *Abogado y Doctor en Jurisprudencia* de la Universidad Central del Ecuador, con más de 35 años de experiencia y ejercicio profesional en materia civil, laboral y de inquilinato, es la cabeza del estudio jurídico de su mismo nombre.

Éste estudio jurídico se dedica a realizar consultoría jurídica y elaboración de contratos de todas las materias legales y su respectiva legislación.

2.2. Problemática

De acuerdo con el Dr. Javier Irigoyen, la problemática tiene que ver con el facilismo de muchos clientes. Este facilismo se traduce en una limitada capacidad de tiempo y recursos económicos, mientras exigen respuestas prontas o soluciones superficiales. Por esta razón los clientes recurren a la competencia que muchas veces les otorga trabajos de menor calidad, ya que no realizan estudios exhaustivos, ejerciendo un trabajo breve e ineficiente y que posiblemente derivaría en un conflicto legal mayor. Podríamos comparar lo dicho, entre la medicina curativa y la preventiva.

Adicionalmente, la situación actual económica que presenta el Ecuador, caracterizada por la falta de liquidez, es un componente más por el que la gran mayoría de empresas y clientes siguen buscando abaratar sus gastos, omitiendo muchas veces la importancia de invertir en sus servicios

Por consiguiente, dado que la competencia ofrece costos menores y tiempo de respuesta inmediata, el Estudio Jurídico del Dr. Javier Irigoyen se ve en la necesidad de encontrar la manera de proporcionar un valor agregado a sus servicios de consultoría puesto que la calidad y profesionalidad de su trabajo lo amerita, el cual requiere de mayor tiempo de estudio y costos más elevados que no son valoradas como se debería.

Las Tecnologías de Información (TIC´s), facilitan la adquisición, el manejo, la generación, la comunicación y el almacenamiento de la información en masa, mediante el uso de herramientas tecnológicas como el Internet y las telecomunicaciones. Actualmente, éstas ejercen una gran influencia en el desarrollo de la sociedad y con más ímpetu ahora, por lo que es necesario hacer uso de las mismas para obtener una ventaja competitiva dentro del mercado, por esa razón se desarrollará una aplicación de software que facilite la prestación de servicios legales en línea y que proporcione una comunicación pronta, eficaz, eficiente y comprensible entre el abogado y el cliente, como un primer ejercicio que se podría replicar a otros establecimientos.

2.3. Procesos

Los procesos del Estudio Jurídico del Dr. Javier Irigoyen implican dos etapas, la primera se encarga del *proceso de consultoría*, y la segunda se encarga del *proceso de acción.*

En la Figura 6 se presenta la primera parte, en el cual los clientes se acercan al Estudio Jurídico para ser asesorados sobre un tema legal. Antes de ser atendidos deben cancelar un valor fijo por consultas, después el cliente procede a explicar su caso al abogado a cargo y éste le proporciona una solución verbal o escrita dependiendo el caso, a menos que el abogado requiera de más tiempo para el estudio del caso y eventualmente dar una solución al mismo.

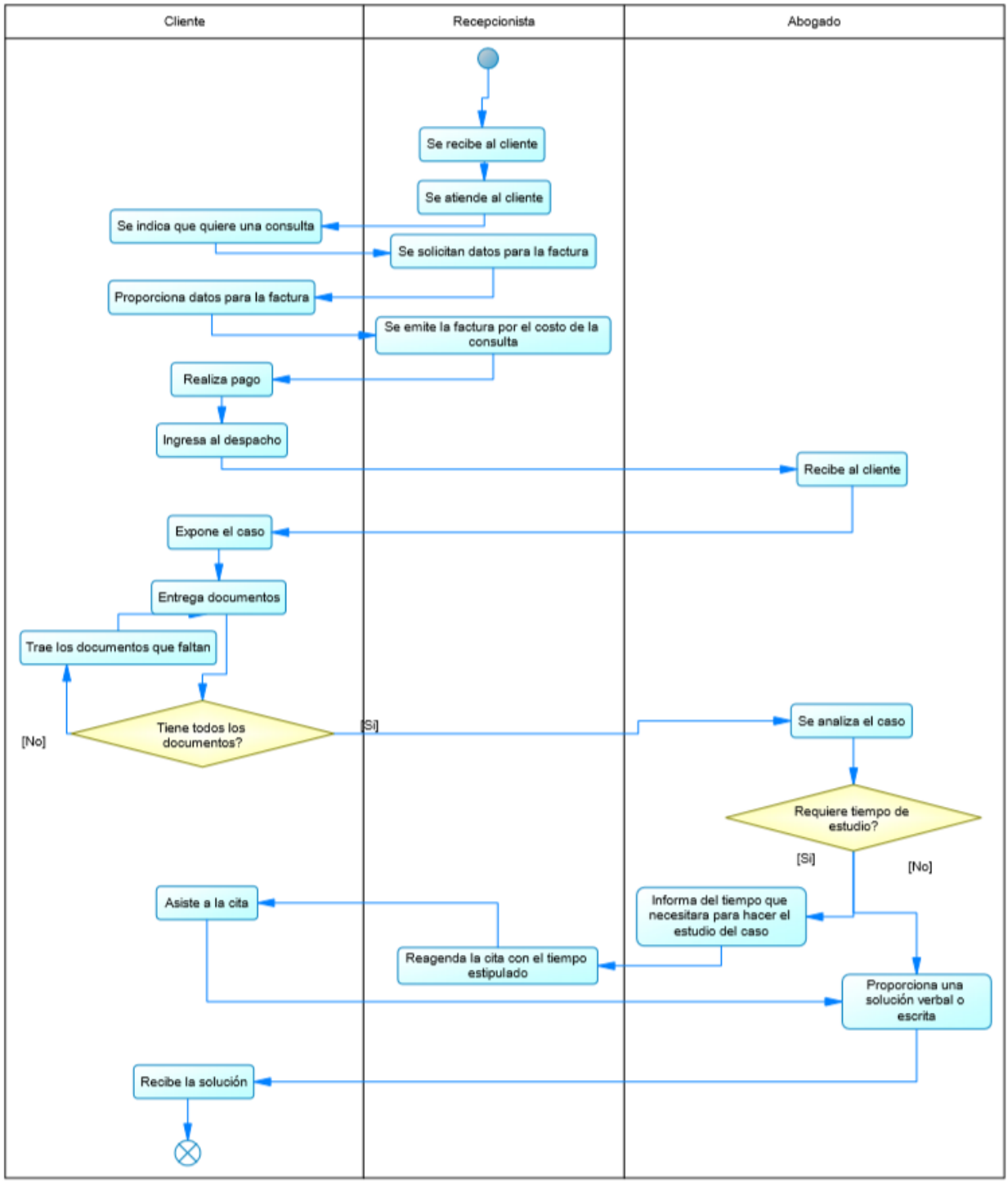

Figura 6 *Proceso de consultoría (Primera etapa)*. Fuente: Elaboración propia

Finalmente, como se puede ver en la Figura 7, el cliente, al recibir la solución, decide si gustaría que el mismo abogado realice la acción legal propuesta en la solución. Si el cliente acepta que se

lleve a cabo la acción, entonces se da lugar a la segunda etapa. Primero se debe pactar los honorarios para que el abogado se haga cargo del caso, después éste debe llevar a cabo los pasos necesarios para alcanzar la solución legal, y si en dicha transición se requiere de nuevos pagos deben ser saldados para continuar el proceso y finalizarlo.

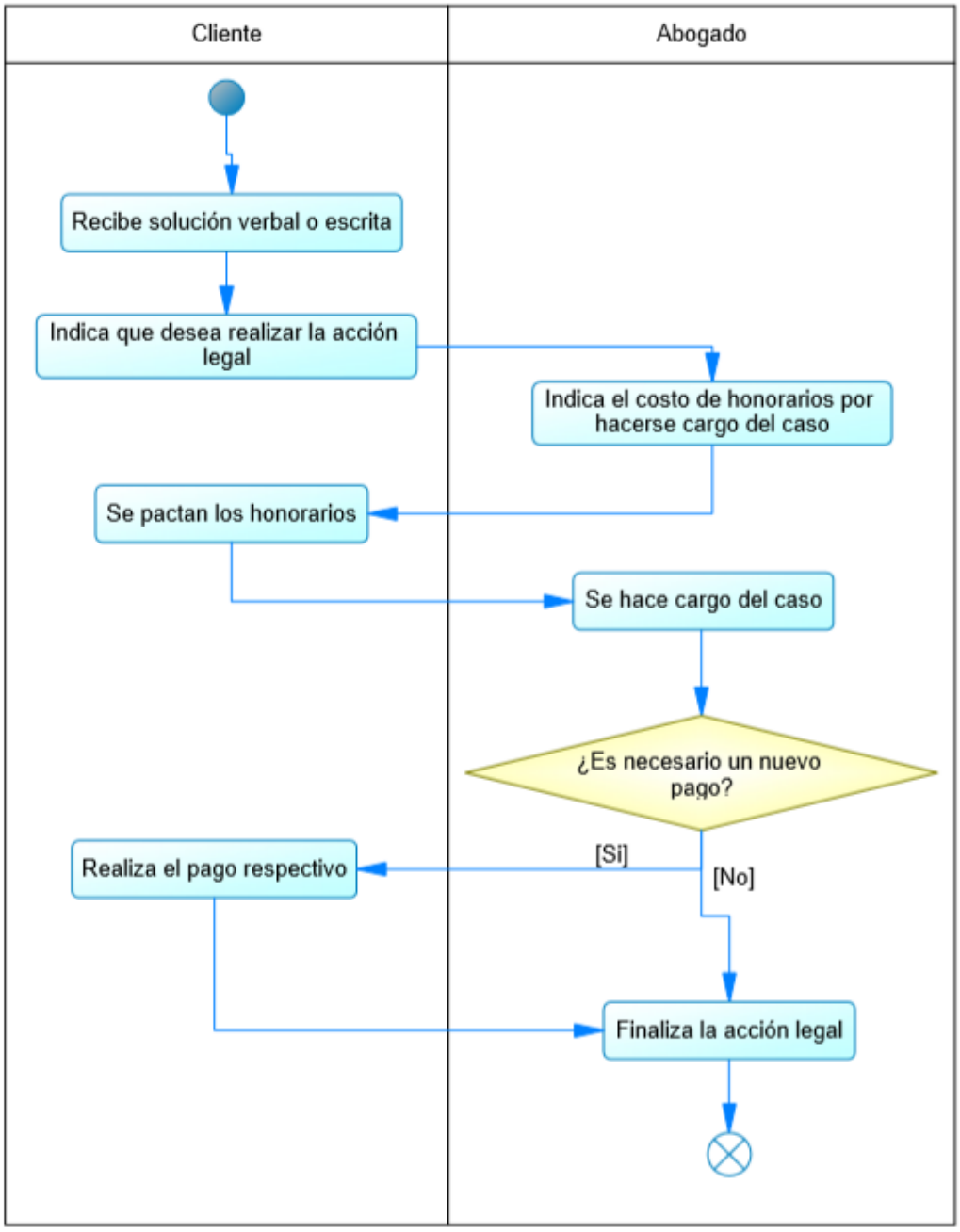

Figura 7 Proceso de acción (segunda etapa) Fuente: Elaboración propia

En el desarrollo de la aplicación ***LEGAL HELP,*** los procesos serán automatizados y mejorados. En ninguna de las dos etapas será necesaria la intervención de un recepcionista que atienda y cobre

al cliente, sino que el cliente es quien realizará las interacciones con la aplicación de modo que ésta la guiará por el proceso para identificar lo que desea, ubicarlo con el abogado correspondiente, realizar los pagos en línea con tarjeta de crédito, y facilitar la comunicación entre el cliente y abogado, convirtiéndolo en un proceso más eficiente, rápido y seguro, consiguiendo además, beneficios como adquirir públicos jóvenes, y consecuentemente más ventas al aumentar la cartera de clientes.

Capítulo 3. Primera fase

Como se describió en el capítulo 1, la metodología de desarrollo de software Mobile-D posee 5 etapas, de las cuales la exploración y la inicialización serán desarrolladas a detalle en el presente capítulo. En la exploración, se establecerán los grupos de interesados, se define el alcance y se determina el establecimiento del proyecto; mientras que en la inicialización se realiza la puesta en marcha, que comprende la planificación inicial, y se lleva a cabo el *día de prueba.*

3.1. Fase de exploración

3.1.1. Establecimiento de interesados

Los grupos de interés del proyecto son:

- El Estudio Jurídico del Dr. Javier Irigoyen Hurtado, debido a su necesidad de desarrollar una aplicación móvil para automatizar sus procesos y volverse más efectivos y eficientes en la atención, consultoría y respuesta al cliente.
- El equipo de desarrollo, quien en este caso es la autora.
- Los clientes, quienes necesitan de una asesoría legal eficiente, de fácil acceso y comprensible.

3.1.2. Definición de alcance

3.1.2.1. Planeación inicial del proyecto

El proyecto tiene como alcance el análisis, desarrollo y la implementación de una aplicación para celular multiplataforma en el Estudio Jurídico del Dr. Javier Irigoyen Hurtado, el cuál pueda ser usado por los encargados pertinentes en el estudio jurídico y por los usuarios en general que tengan acceso a un celular inteligente con capacidad para aplicaciones móviles.

Los usuarios que requieran el servicio legal son los llamados clientes y deberán poseer una capacidad económica moderada para poder pagar el servicio requerido. De todas formas, ello no implica que el cliente deba pertenecer a una categoría socioeconómica media-alta para hacer uso de la aplicación.

Por otro lado, los usuarios que proporcionen el servicio legal mediante la aplicación deberán tener conocimientos tanto generales como específicos sobre el derecho y la rama jurídica correspondiente.

La presente disertación tiene como propósito realizar un estudio exploratorio puesto que no se ha realizado con anterioridad ningún tipo de aplicación o sistema que plantee lo que se ha mencionado antes en el Estudio Jurídico del Dr. Javier Irigoyen.

Además, se realizarán estudios descriptivos ya que se pretende describir las características, perfiles y procesos respectivos al sistema y al caso de estudio para realizar un análisis adecuado.

Por último, se hará un estudio correlacional, el cual está encargado de vincular el sistema que se pretende desarrollar con el Estudio Jurídico del Dr. Javier Irigoyen.

3.1.2.2. Conjunto de requisitos iniciales (SRS)

a) Restricciones
   - La herramienta Sencha Architect requiere de una licencia pagada y la curva de aprendizaje de ExtJS es más empinada que la de JavaScript, ya que ExtJS es un framework de JavaScript robusto y multiplataforma, el cual viene equipado con cientos de componentes de interfaz de usuario de alto rendimiento preconstruidos en un lugar central (Joshi, 2020).
   - La herramienta PhpStorm requiere de una licencia.

b) Suposiciones y dependencias
   - Se depende de las licencias que proporcionan la usabilidad de las herramientas Sencha Architect y PhpStorm.
   - El desarrollo de la aplicación depende de que el servidor Apache esté habilitado y funcione correctamente.

c) Evolución previsible del sistema
   - Realizar las pruebas pertinentes a la aplicación para instalarla en producción en un servidor web.
   - Se prevé la publicación de la aplicación ***LEGAL HELP*** en tiendas en línea como AppStore o PlayStore, de modo que esté disponible para la descarga gratuita.
   - Se prevé que la aplicación realice los pagos en línea con tarjeta de crédito o débito.

d) Requisitos específicos

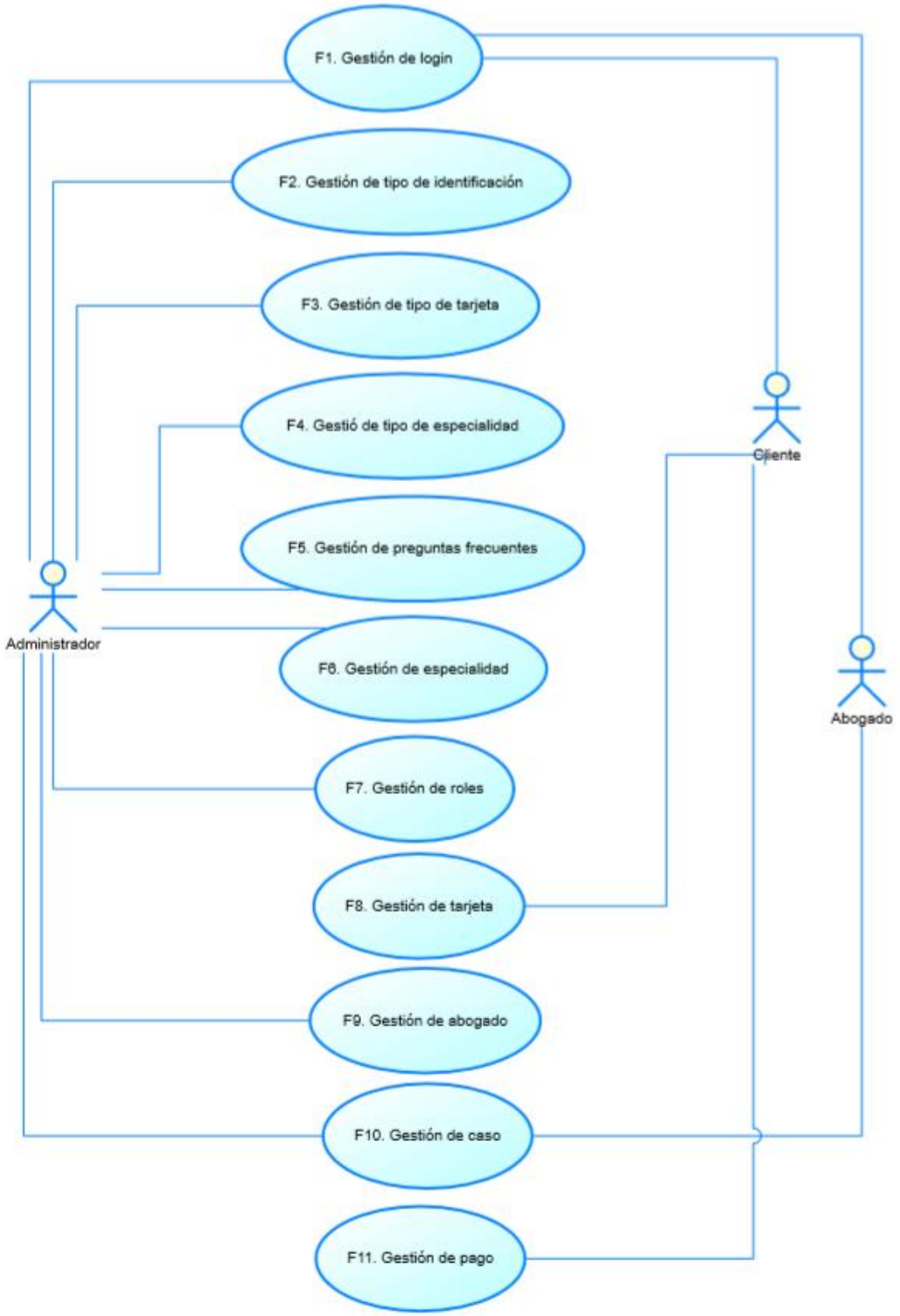

Figura 8 Requisitos específicos. *Fuente: Elaboración propia*

Tabla 4

Requisito de gestión de login

Número de requisito	F1
Nombre de requisito	Gestión de login
Descripción	Permite que los usuarios inicien sesión, registren su cuenta en la aplicación o cierren sesión.
Prioridad del requisito	Alta
Autores	Usuario

Fuente: Elaboración propia

Tabla 5

Requisito de gestión de tipo de identificación

Número de requisito	F2
Nombre de requisito	Gestión de tipo de identificación
Descripción	Permite que se gestionen los tipos de identificación.
Prioridad del requisito	Alta
Autores	Administrador

Fuente: Elaboración propia

Tabla 6

Requisito de gestión de tipo de tarjeta

Número de requisito	F3
Nombre de requisito	Gestión de tipo de tarjeta
Descripción	Permite que se gestionen los tipos de tarjeta.
Prioridad del requisito	Alta
Autores	Administrador

Fuente: Elaboración propia

Tabla 7

Requisito de gestión de tipo de especialidad

Número de requisito	F4
Nombre de requisito	Gestión de tipo de especialidad
Descripción	Permite que se gestionen los tipos de especialidad.
Prioridad del requisito	Alta
Autores	Administrador

Fuente: Elaboración propia

Tabla 8

Requisito de gestión de preguntas frecuentes

Número de requisito	F5
Nombre de requisito	Gestión de preguntas frecuentes
Descripción	Permite que se gestionen las preguntas frecuentes.
Prioridad del requisito	Alta
Autores	Administrador

Fuente: Elaboración propia

Tabla 9

Requisito de gestión de especialidad

Número de requisito	F6
Nombre de requisito	Gestión de especialidad
Descripción	Permite que se gestionen las especialidades, para su creación, modificación, eliminación y consulta.
Prioridad del requisito	Alta
Autores	Administrador

Fuente: Elaboración propia

Tabla 10

Requisito de gestión de biblioteca

Número de requisito	F7
Nombre de requisito	Gestión de biblioteca
Descripción	Permite que se gestione la biblioteca, para su creación, modificación, eliminación y consulta.
Prioridad del requisito	Alta
Autores	Administrador

Fuente: Elaboración propia

Tabla 11

Requisito de gestión de roles

Número de requisito	F8
Nombre de requisito	Gestión de roles
Descripción	Permite que se gestionen los roles de los usuarios.
Prioridad del requisito	Alta
Autores	Administrador

Fuente: Elaboración propia

Tabla 12

Requisito de gestión de tarjeta

Número de requisito	F9
Nombre de requisito	Gestión de tarjeta
Descripción	Permite que se gestionen tarjetas, para su creación, modificación, eliminación y consulta.
Prioridad del requisito	Alta

Autores	Usuario

Fuente: Elaboración propia

Tabla 13

Requisito de gestión de abogado

Número de requisito	F10
Nombre de requisito	Gestión de abogado
Descripción	Permite que se gestionen los abogados, para su modificación y consulta.
Prioridad del requisito	Alta
Autores	Usuario

Fuente: Elaboración propia

Tabla 14

Requisito de gestión de caso

Número de requisito	F11
Nombre de requisito	Gestión de caso
Descripción	Permite que se gestionen los casos, para su creación, modificación y consulta.
Prioridad del requisito	Alta
Autores	Abogados

Fuente: Elaboración propia

Tabla 15

Requisito de gestión de pago

Número de requisito	F12
Nombre de requisito	Gestión de pago
Descripción	Permite que se habiliten y se realicen los pagos.
Prioridad del requisito	Alta

Autores	Usuario y abogado

Fuente: Elaboración propia

Tabla 16

Requisito de gestión de mensaje

Número de requisito	F13
Nombre de requisito	Gestión de mensaje
Descripción	Permite que se muestren y envíen mensajes.
Prioridad del requisito	Alta
Autores	Usuario y Abogado

Fuente: Elaboración propia

e) Requisitos de las interfaces

a. Interfaces de usuario

La aplicación tendrá una ventana inicial, la cual tendrá una forma con un campo para el correo y un campo para la contraseña, junto con tres botones, uno para iniciar sesión, otro para registrar los datos y uno para reestablecer la contraseña si el usuario la olvidó, Si presiona el botón de *Registrarse* se mostrará una ventana con una forma con los campos necesarios y un botón para enviar el registro. Si se presiona el botón de *Olvide mi contraseña*, se mostrará una ventana con un campo de texto para ingresar el correo electrónico y un botón para reestablecer la contraseña.
El resto de las pantallas contendrán componentes principales como formas y tablas (grids), también componentes como campos de texto, áreas de texto, casilla de opciones (comboBox), botones, menús, y contenedores como ventanas, paneles, entre otros; para mejorar la comunicación entre los clientes/abogados y facilitar la administración de la aplicación.

b. Interfaces de hardware

Se requiere de un dispositivo hardware móvil que permita ingresar y retornar datos, que ayude como una interacción física del usuario con la aplicación, por lo que se necesita de un teléfono inteligente, con una interfaz táctil que facilite al usuario el ingreso de datos con un teclado y el software muestre la información por medio de una pantalla táctil.
La aplicación servirá para interfaces de hardware que funcionen con el sistema operativo Android, por ejemplo: para toda la gama de teléfonos Samsung, algunos teléfonos LG, HTC, Sony, Motorola, Huawei, Xiaomi, entre otros más.

c. Interfaces de software

La interfaz de usuario es la aplicación móvil que se desarrollará, ésta es la que procesará los datos y recibirá la interacción del usuario para retornar respuestas visibles en la interfaz de hardware.

d. Interfaces de comunicación

Se requiere establecer una comunicación entre el cliente y el abogado delegado, por lo que se construirán pantallas de mensajería para enviar y recibir mensajes por medio de este canal.

f) Requisitos funcionales

Los requisitos detallados a continuación presentan un flujo principal, en el cual se listan los pasos para cada requisito junto con el código de la excepción si la requiere. La tabla de las excepciones se muestra a continuación en la tabla 17.

Tabla 17

Excepciones

Código	Causa	Mensaje
E1	Problemas con BBDD	Ha ocurrido un error, comunicarse con el administrador.
E2	Problemas en la validación de formato	Ingrese los datos correctamente

Fuente: Elaboración propia

1) Gestión de usuario

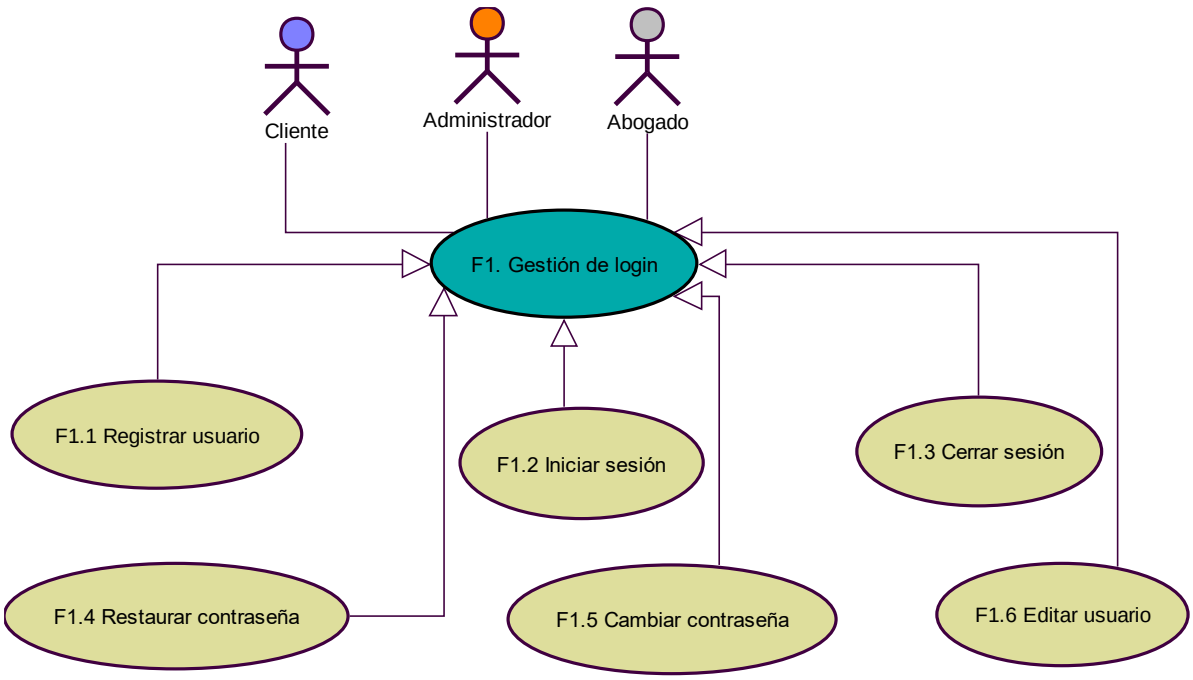

Figura 9 Gestión de login. *Fuente: Elaboración propia*

Nombre: **F1.1. Registrar usuario**

Descripción: Permite que el actor se registre en el sistema.

Usuarios: Cliente, Administrador, Abogado

Flujo principal:

1. El usuario cliquea el botón *Registrarse* de la pantalla inicial.

2. El sistema presenta la pantalla de registrar usuario.
3. El usuario ingresa el primer nombre.
4. El usuario ingresa el segundo nombre.
5. El usuario ingresa el primer apellido.
6. El usuario ingresa el segundo apellido.
7. El usuario ingresa el correo. (E2)
8. El usuario ingresa la contraseña. (E2)
9. El sistema valida que la contraseña tenga los caracteres necesarios.
10. El usuario repite la contraseña. (E2)
11. El sistema valida que la contraseña repetida coincida con la primera.
12. El usuario ingresa el número de teléfono. (E2)
13. El usuario ingresa la dirección de domicilio.
14. El usuario ingresa el tipo de identificación.
15. El usuario ingresa el número de identificación.
16. El usuario cliquea en *Enviar.*
17. El sistema almacena la información. (E1)

Nombre: **F1.2. Iniciar sesión**

Descripción: Permite que el usuario inicie sesión.

Usuarios: Cliente, Administrador, Abogado

Flujo principal:

1. El usuario ingresa el correo electrónico. (E2)
2. El usuario ingresa la contraseña. (E2)
3. El usuario cliquea el botón *Iniciar sesión* de la pantalla inicial.
4. El sistema valida que los datos sean correctos. (E1)
5. El sistema presenta la pantalla respectiva al rol de usuario.

Nombre: **F1.3. Cerrar sesión**

Descripción: Permite que el usuario cierre sesión.

Usuarios: Cliente, Administrador, Abogado

Flujo principal:

1. El usuario cliquea en el botón de menú en la barra superior y presiona el botón *Cerrar sesión.*
2. El sistema cierra sesión y muestra la pantalla inicial. (E1)

Nombre: **F1.4. Restaurar contraseña**

Descripción: Permite que el usuario restaure su contraseña en caso de haberla olvidado.

Usuarios: Cliente, Administrador, Abogado

Flujo principal:

1. El usuario cliquea el botón *Olvide mi contraseña* de la pantalla inicial.

2. El sistema presenta la pantalla de restaurar contraseña.
3. El usuario ingresa el correo. (E2)
4. El usuario cliquea en *Recuperar*.
5. El sistema envía un correo con la contraseña temporal generada. (E1)

Nombre: **F1.5. Cambiar contraseña**

Descripción: Permite que el usuario cambie su contraseña.

Usuarios: Cliente, Administrador, Abogado

Flujo principal:

1. El usuario cliquea en el botón de menú en la barra superior y presiona el botón *Cuenta.*
2. Se despliega la pantalla de cuenta.
3. El usuario cliquea el botón *Cambiar contraseña*.
4. El sistema presenta la pantalla de cambio de contraseña.
5. El usuario ingresa la clave actual. (E2)
6. El usuario ingresa la clave nueva. (E2)
7. El usuario ingresa la repetición de la clave nueva. (E2)
8. El usuario cliquea en *Cambiar.*
9. El sistema almacena la información. (E1)

Nombre: **F1.6. Editar usuario**

Descripción: Permite que el usuario edite sus datos.

Usuarios: Cliente, Administrador, Abogado

Flujo principal:

1. El usuario cliquea en el botón de menú en la barra superior y presiona el botón *Cuenta.*
2. Se despliega la pantalla de cuenta.
3. El usuario cliquea el botón *Editar perfil.*
4. El sistema presenta la pantalla de edición de usuario.
5. El usuario ingresa el primer nombre.
6. El usuario ingresa el segundo nombre.
7. El usuario ingresa el primer apellido.
8. El usuario ingresa el segundo apellido.
9. El usuario ingresa el correo. (E2)
10. El usuario ingresa la contraseña. (E2)
11. El usuario ingresa el número de teléfono. (E2)
12. El usuario ingresa la dirección de domicilio.
13. El usuario ingresa el tipo de identificación.
14. El usuario ingresa el número de identificación.
15. El usuario cliquea en *Guardar.*
16. El sistema almacena la información. (E1)

2) Gestión de tipo de identificación

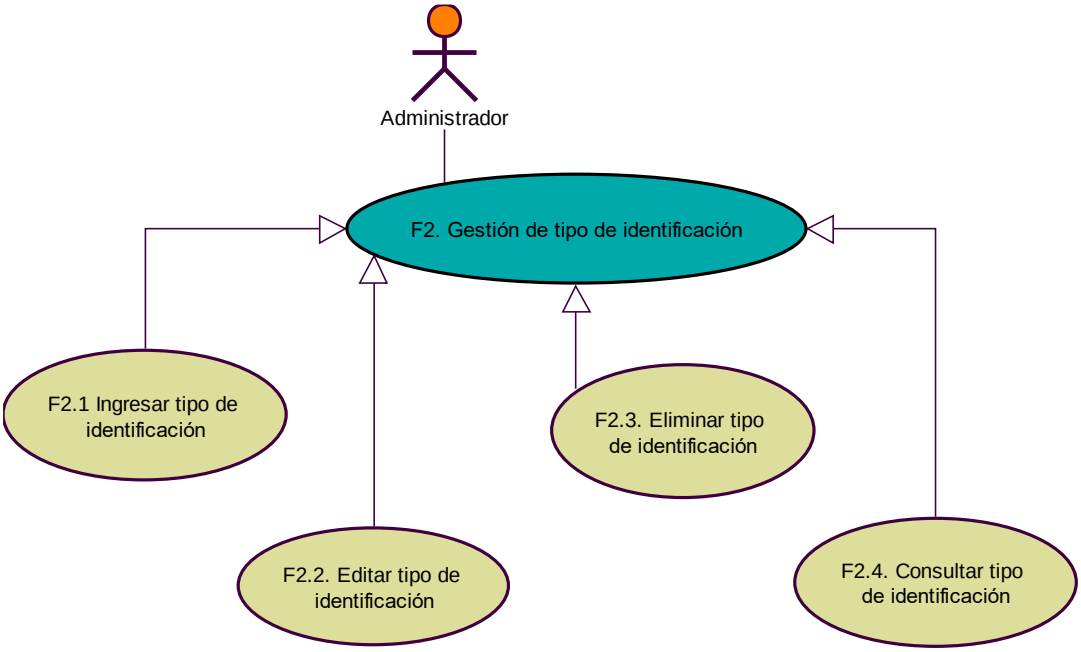

Figura 10 Gestión de tipo de identificación. *Fuente: Elaboración propia*

Nombre: **F2.1. Ingresar tipo de identificación**

Descripción: Permite que el usuario ingrese un tipo de identificación en el sistema.

Usuarios: Administrador

Flujo principal:

1. El usuario cliquea el botón *Tipos de identificación* de la pantalla principal del rol de administrador.
2. El sistema presenta la pantalla de administración de tipos de identificación.
3. El usuario cliquea en el botón de *Añadir (+).*
4. El usuario ingresa el nombre del tipo de identificación.
5. El usuario cliquea el botón *Guardar*.
6. El sistema almacena la información. (E1)

Nombre: **F2.2. Editar tipo de identificación**

Descripción: Permite que el usuario edite un tipo de identificación registrado en el sistema.

Usuarios: Administrador

Flujo principal:

1. El usuario cliquea el botón *Tipos de identificación* de la pantalla principal del rol de administrador.
2. El sistema presenta la pantalla de administración de tipos de identificación.
3. El usuario desliza a la izquierda el tipo de identificación de la lista que desee editar.
4. El usuario modifica el nombre del tipo de identificación.

5. El usuario cliquea el botón *Guardar*.
6. El sistema almacena la información. (E1)

Nombre: **F2.3. Eliminar tipo de identificación**

Descripción: Permite que el usuario elimine un tipo de identificación registrado en el sistema.

Usuarios: Administrador

Flujo principal:

1. El usuario cliquea el botón *Tipos de identificación* de la pantalla principal del rol de administrador.
2. El sistema presenta la pantalla de administración de tipos de identificación.
3. El usuario desliza a la derecha el tipo de identificación de la lista que desee eliminar.
4. El sistema elimina el registro. (E1)

Nombre: **F2.4. Consultar tipo de identificación**

Descripción: Permite que el usuario consulte los tipos de identificación registrados en el sistema.

Usuarios: Administrador

Flujo principal:

1. El usuario cliquea el botón *Tipos de identificación* de la pantalla principal del rol de administrador.
2. El sistema presenta la pantalla de administración de tipos de identificación.
3. Se despliega el listado de los tipos de identificación registrados en el sistema.

3) Gestión de tipo de tarjeta

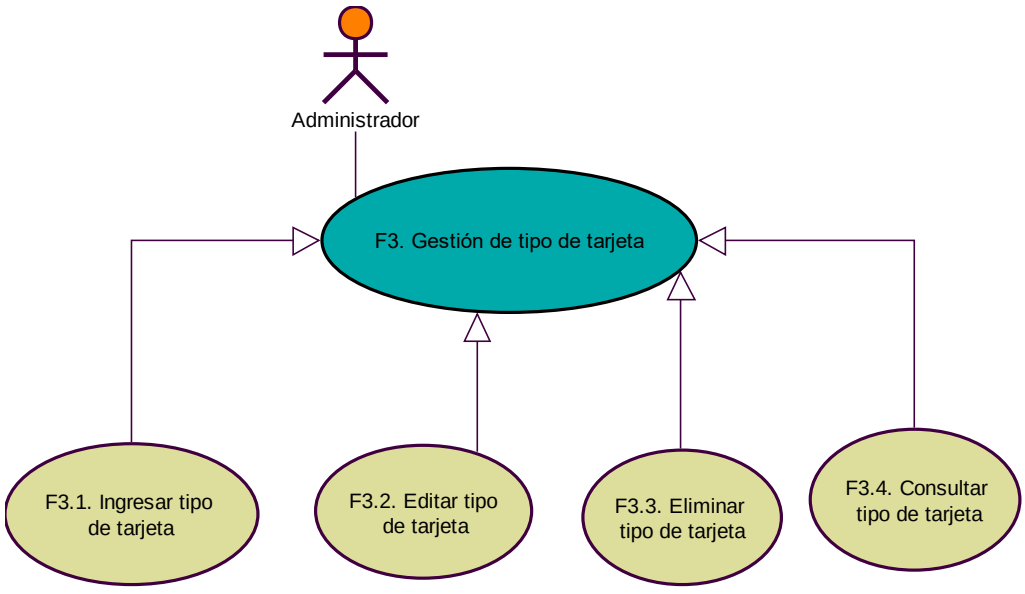

Figura 11 Gestión de tipo de tarjeta. *Fuente: Elaboración propia*

Nombre: **F3.1. Ingresar tipo de tarjeta**

Descripción: Permite que el usuario ingrese un tipo de tarjeta en el sistema.

Usuarios: Administrador

Flujo principal:

Se usa el mismo esquema del caso de uso F2.1. Revisar sección de adjuntos.

Nombre: **F3.2. Editar tipo de tarjeta**

Descripción: Permite que el usuario edite un tipo de tarjeta registrado en el sistema.

Usuarios: Administrador

Flujo principal:

Se usa el mismo esquema del caso de uso F2.2. Revisar sección de adjuntos.

Nombre: **F3.3. Eliminar tipo de tarjeta**

Descripción: Permite que el usuario elimine un tipo de tarjeta registrado en el sistema.

Usuarios: Administrador

Flujo principal:

Se usa el mismo esquema del caso de uso F2.3. Revisar sección de adjuntos.

Nombre: **F3.4. Consultar tipo de tarjeta**

Descripción: Permite que el usuario consulte los tipos de tarjeta registrados en el sistema.

Usuarios: Administrador

Flujo principal:

Se usa el mismo esquema del caso de uso F2.4. Revisar sección de adjuntos.

4) Gestión de tipo de especialidad

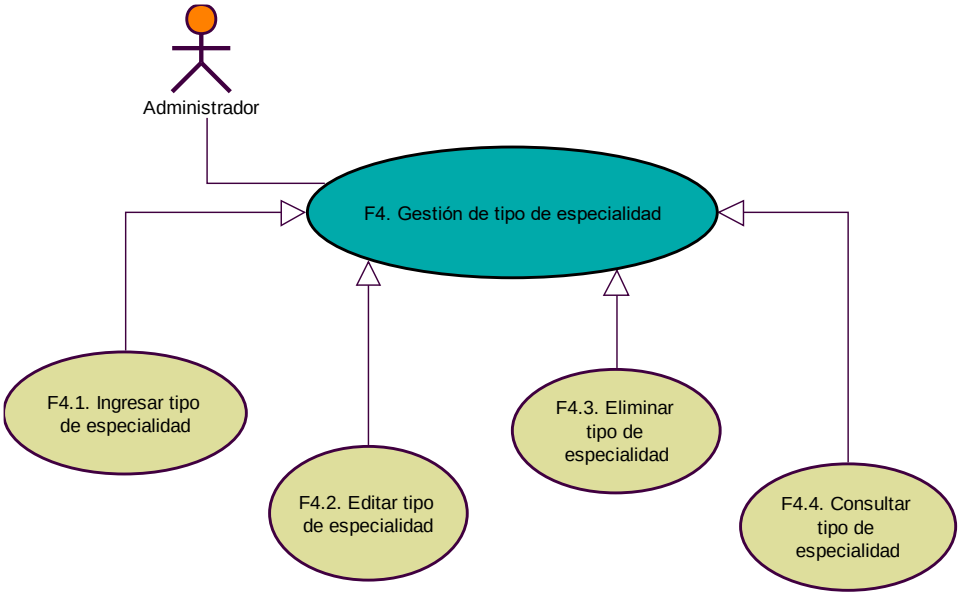

Figura 12 Gestión de tipo de especialidad. *Fuente: Elaboración propia*

Nombre: **F4.1. Ingresar tipo de especialidad**

Descripción: Permite que el usuario ingrese un tipo de especialidad en el sistema.

Usuarios: Administrador

Flujo principal:

Se usa el mismo esquema del caso de uso F2.1. Revisar sección de adjuntos.

Nombre: **F4.2. Editar tipo de especialidad**

Descripción: Permite que el usuario edite un tipo de especialidad registrado en el sistema.

Usuarios: Administrador

Flujo principal:

Se usa el mismo esquema del caso de uso F2.2. Revisar sección de adjuntos.

Nombre: **F4.3. Eliminar tipo de especialidad**

Descripción: Permite que el usuario elimine un tipo de especialidad registrado en el sistema.

Usuarios: Administrador

Flujo principal:

Se usa el mismo esquema del caso de uso F2.3. Revisar sección de adjuntos.

Nombre: **F4.4. Consultar tipo de especialidad**

Descripción: Permite que el usuario consulte los tipos de especialidad registrados en el sistema.

Usuarios: Administrador

Flujo principal:

Se usa el mismo esquema del caso de uso F2.4. Revisar sección de adjuntos.

5) Gestión de preguntas frecuentes

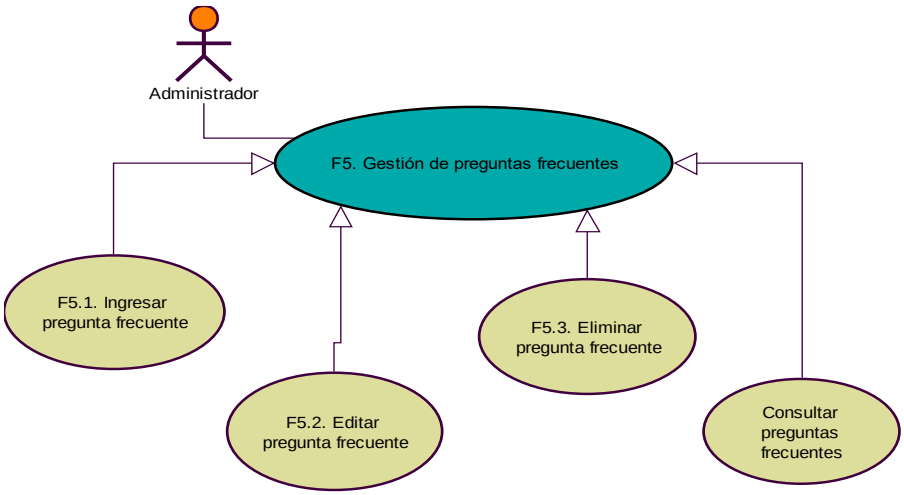

Figura 13 Gestión de preguntas frecuentes. *Fuente: Elaboración propia*

Nombre: **F5.1. Ingresar pregunta frecuente**

Descripción: Permite que el usuario ingrese una pregunta frecuente.

Usuarios: Administrador

Flujo principal:

Se usa el mismo esquema del caso de uso F2.1. Revisar sección de adjuntos.

Nombre: **F5.2. Editar pregunta frecuente**

Descripción: Permite que el usuario edite una pregunta frecuente registrada en el sistema.

Usuarios: Administrador

Flujo principal:

Se usa el mismo esquema del caso de uso F2.2. Revisar sección de adjuntos.

Nombre: **F5.3. Eliminar pregunta frecuente**

Descripción: Permite que el usuario elimine un tipo de especialidad registrado en el sistema.

Usuarios: Administrador

Flujo principal:

Se usa el mismo esquema del caso de uso F2.3. Revisar sección de adjuntos.

Nombre: **F5.4. Consultar preguntas frecuentes**

Descripción: Permite que el usuario consulte las preguntas frecuentes registradas en el sistema.

Usuarios: Administrador

Flujo principal:

Se usa el mismo esquema del caso de uso F2.4. Revisar sección de adjuntos.

6) Gestión de especialidad

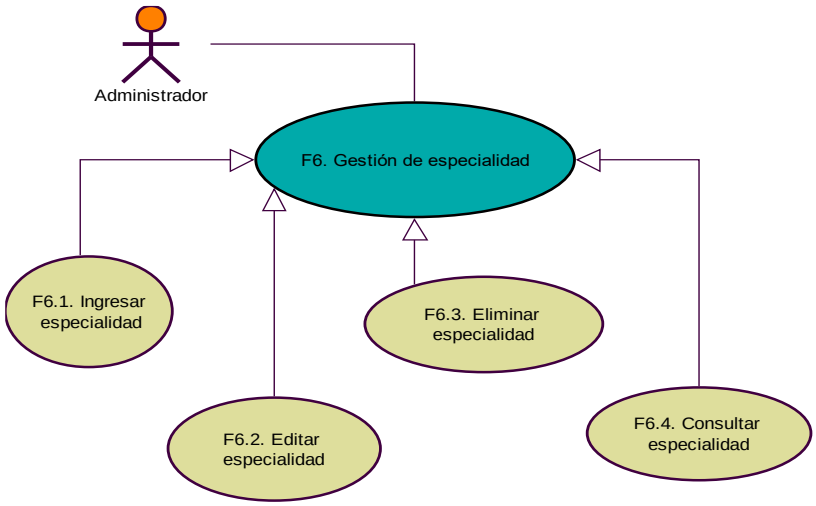

Figura 14 Gestión de especialidad. *Fuente: Elaboración propia*

Nombre: **F6.1. Ingresar especialidad**

Descripción: Permite que el usuario ingrese una especialidad en el sistema.

Usuarios: Administrador

Flujo principal:

Se usa el mismo esquema del caso de uso F2.1. Revisar sección de adjuntos.

Nombre: **F6.2. Editar especialidad**

Descripción: Permite que el usuario edite una especialidad registrada en el sistema.

Usuarios: Administrador

Flujo principal:

Se usa el mismo esquema del caso de uso F2.2. Revisar sección de adjuntos.

Nombre: **F6.3. Eliminar especialidad**

Descripción: Permite que el usuario elimine una especialidad registrada en el sistema.

Usuarios: Administrador

Flujo principal:

Se usa el mismo esquema del caso de uso F2.3. Revisar sección de adjuntos.

Nombre: **F6.4. Consultar especialidad**

Descripción: Permite que el usuario consulte las especialidades registradas en el sistema.

Usuarios: Administrador

Flujo principal:

Se usa el mismo esquema del caso de uso F2.4. Revisar sección de adjuntos.

7) Gestión de biblioteca

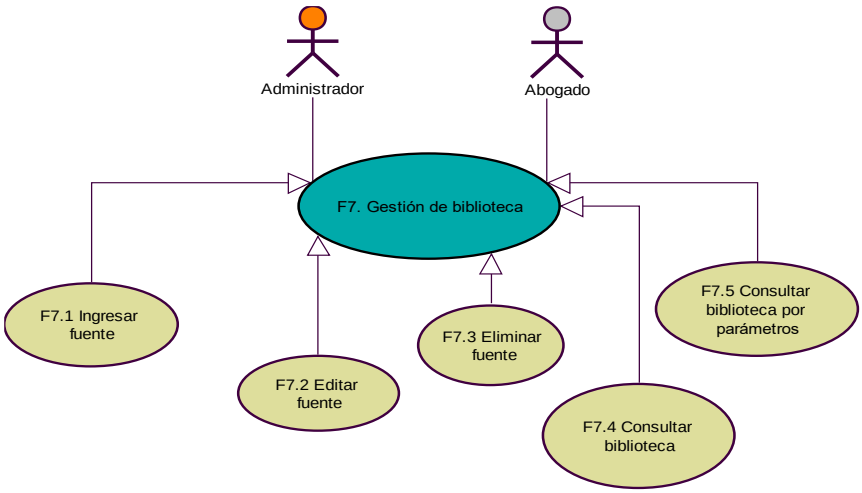

Figura 15 Gestión de biblioteca. *Fuente: Elaboración propia*

Nombre: **F7.1. Ingresar fuente**

Descripción: Permite que el usuario ingrese una fuente a la biblioteca en el sistema.

Usuarios: Administrador

Flujo principal:

Se usa el mismo esquema del caso de uso F2.1. Revisar sección de adjuntos.

Nombre: **F7.2. Editar fuente**

Descripción: Permite que el usuario edite una fuente de la biblioteca registrada en el sistema.

Usuarios: Administrador

Flujo principal:

Se usa el mismo esquema del caso de uso F2.2. Revisar sección de adjuntos.

Nombre: **F7.3. Eliminar fuente**

Descripción: Permite que el usuario elimine una especialidad registrada en el sistema.

Usuarios: Administrador

Flujo principal:

Se usa el mismo esquema del caso de uso F2.3. Revisar sección de adjuntos.

Nombre: **F7.4.1. Consultar biblioteca**

Descripción: Permite que el usuario consulte las fuentes de la biblioteca registradas en el sistema.

Usuarios: Administrador, abogado

Flujo principal:

Se usa el mismo esquema del caso de uso F2.4. Revisar sección de adjuntos.

Nombre: **F7.4.2. Consultar biblioteca por parámetros**

Descripción: Permite que el usuario consulte las fuentes de la biblioteca registradas en el sistema.

Usuarios: Administrador, abogado

Flujo principal:

1. El usuario cliquea el botón *Biblioteca* de la pantalla principal del rol de administrador.
2. El sistema presenta la pantalla de administración de biblioteca.
3. Se despliega el listado de fuentes de la biblioteca registradas en el sistema.
4. El usuario ingresa el nombre de la fuente deseada.
5. Se despliega el listado de fuentes de la biblioteca registradas en el sistema que coinciden con el nombre ingresado por el usuario.
6. El usuario hace clic en un ítem de la lista de fuentes que desee visualizar.
7. Se despliega la pantalla con la visualización de la fuente. (E1)

8) Gestión de roles

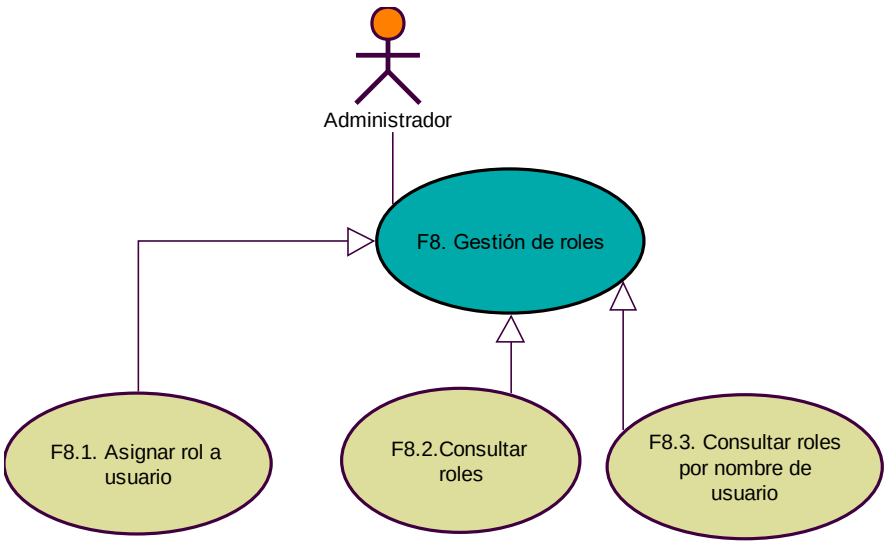

Figura 16 Gestión de roles. *Fuente: Elaboración propia*

Nombre: **F8.1. Asignar rol a usuario**

Descripción: Permite que el usuario asigne un rol a un usuario.

Usuarios: Administrador

Flujo principal:

1. El usuario cliquea el botón *Roles* de la pantalla principal del rol de administrador.

2. El sistema presenta la pantalla de administración de roles.
3. Se despliega un listado de los usuarios del sistema con su rol respectivo.
4. El usuario desliza a la izquierda el ítem de la lista de usuarios que desee asignar un rol.
5. Se despliega la pantalla con el formulario de asignación de rol.
6. El usuario selecciona el rol.
7. El usuario cliquea en *Guardar*.
8. El sistema almacena la información. (E1)

Nombre: **F8.2. Consultar roles**

Descripción: Permite que el usuario consulte los usuarios con su rol.

Usuarios: Administrador

Flujo principal:

Se usa el mismo esquema del caso de uso F2.4. Revisar sección de adjuntos.

Nombre: **F8.3. Consultar roles por nombre de usuario**

Descripción: Permite que el usuario consulte los usuarios con su rol por nombre de usuario.

Usuarios: Administrador

Flujo principal:

Se usa el mismo esquema del caso de uso F7.4.2. Revisar sección de adjuntos.

9) Gestión de tarjeta

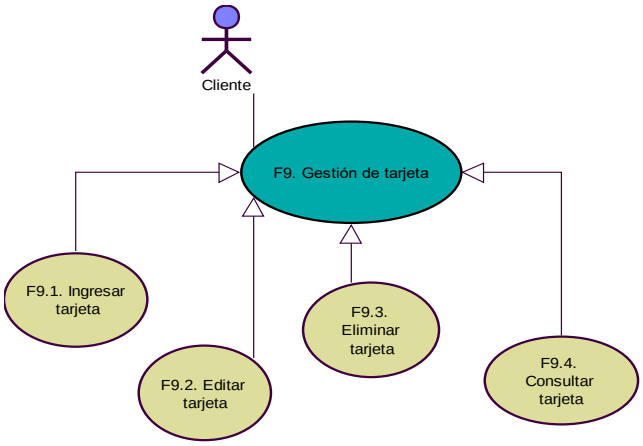

Figura 17 Gestión de tarjeta. *Fuente: Elaboración propia*

Nombre: **F9.1. Ingresar tarjeta**

Descripción: Permite que el usuario ingrese los datos de su tarjeta.

Usuarios: Cliente

Flujo principal:

1. El usuario cliquea en el botón de menú en la barra superior y presiona el botón *Cuenta.*
2. Se despliega la pantalla de cuenta.
3. El usuario cliquea el botón *Datos tarjeta*.
4. El sistema presenta la pantalla de datos tarjeta.
5. El usuario ingresa el nombre completo del titular de la tarjeta.
6. El usuario selecciona el tipo de tarjeta.
7. El usuario ingresa el número de la tarjeta. (E2)
8. El usuario cliquea en *Guardar.*
9. El sistema almacena la información. (E1)

Nombre: **F9.2. Editar tarjeta**

Descripción: Permite que el usuario edite los datos de su tarjeta.

Usuarios: Cliente

Flujo principal:

1. El usuario cliquea en el botón de menú en la barra superior y presiona el botón *Cuenta.*
2. Se despliega la pantalla de cuenta.
3. El usuario cliquea el botón *Datos tarjeta*.
4. El sistema presenta la pantalla de datos tarjeta.
5. El usuario modifica el nombre completo del titular de la tarjeta.
6. El usuario modifica el tipo de tarjeta.
7. El usuario modifica el número de la tarjeta. (E2)
8. El usuario cliquea en *Guardar.*
9. El sistema almacena la información. (E1)

Nombre: **F9.3. Eliminar tarjeta**

Descripción: Permite que el usuario elimine los datos de la tarjeta.

Usuarios: Cliente

Flujo principal:

1. El usuario cliquea en el botón de menú en la barra superior y presiona el botón *Cuenta.*
2. Se despliega la pantalla de cuenta.
3. El usuario cliquea el botón *Datos tarjeta*.
4. El sistema presenta la pantalla de datos tarjeta.
5. El usuario cliquea en el botón *Borrar datos tarjeta.*
6. El sistema elimina los datos de la tarjeta. (E1)

Nombre: **F9.4. Consultar tarjeta**

Descripción: Permite que el usuario consulte los datos de su tarjeta.

Usuarios: Cliente

Flujo principal:

1. El usuario cliquea en el botón de menú en la barra superior y presiona el botón *Cuenta*.
2. Se despliega la pantalla de cuenta.
3. El usuario cliquea el botón *Datos tarjeta*.
4. El sistema presenta la pantalla de datos tarjeta.
5. El sistema carga los datos de la tarjeta en el formulario. (E1)

10) Gestión de abogados

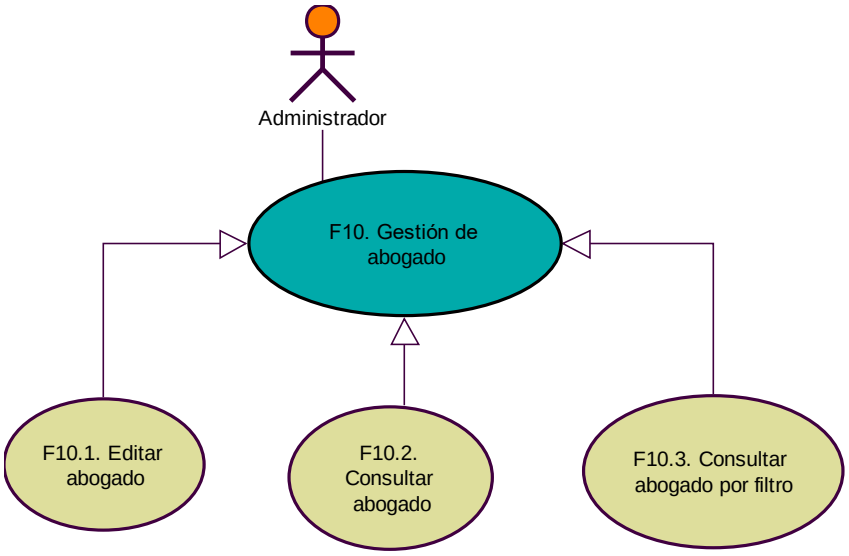

Figura 18 Gestión de abogado. *Fuente: Elaboración propia*

Nombre: **F10.1. Editar abogado**

Descripción: Permite que el usuario edite un abogado registrado en el sistema.

Usuarios: Administrador

Flujo principal:

Se usa el mismo esquema del caso de uso F2.2. Revisar sección de adjuntos.

Nombre: **F10.2. Consultar abogado**

Descripción: Permite que el usuario consulte los abogados registrados en el sistema.

Usuarios: Administrador

Flujo principal:

Se usa el mismo esquema del caso de uso F2.4. Revisar sección de adjuntos.

Nombre: **F10.3. Consultar abogado por parámetro**

Descripción: Permite que el usuario consulte los abogados registrados en el sistema por nombre de usuario, correo de usuario, número de licencia o especialidad.

Usuarios: Administrador

Flujo principal:

Se usa el mismo esquema del caso de uso F7.4.2. Revisar sección de adjuntos.

11) Gestión de casos

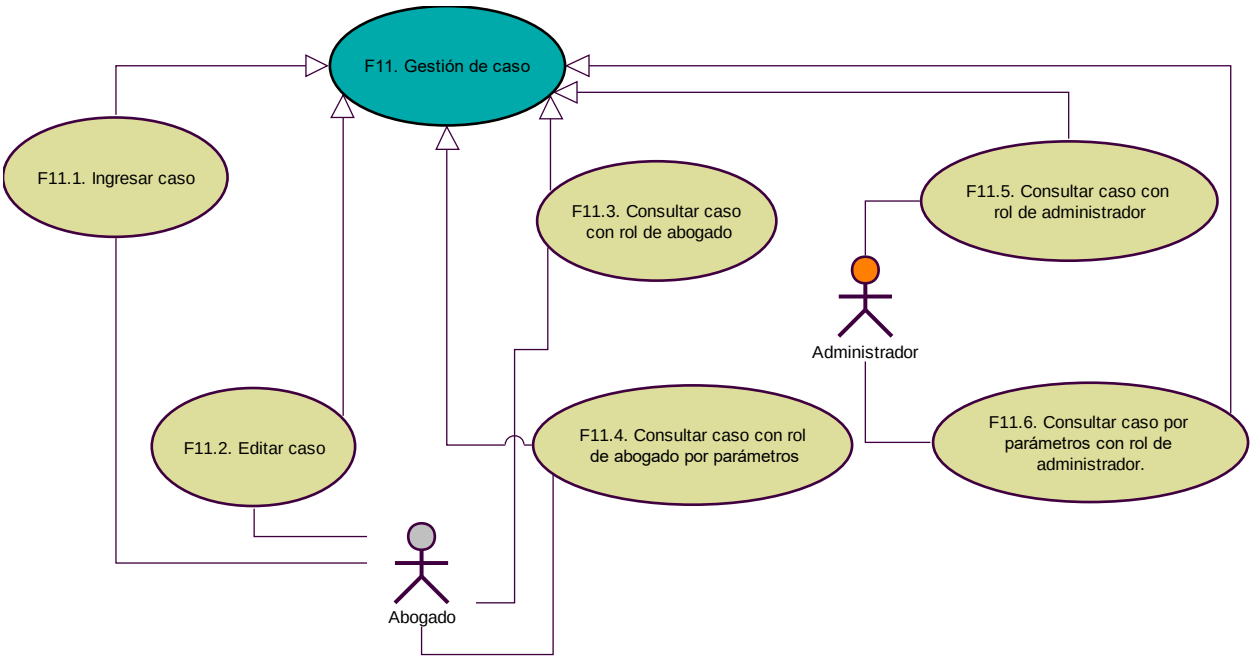

Figura 19 Gestión de caso. *Fuente: Elaboración propia*

Nombre: **F11.1. Ingresar caso**

Descripción: Permite que el usuario ingrese un caso en el sistema.

Usuarios: Abogado

Flujo principal:

1. El usuario cliquea el botón *Direccionamiento* de la pantalla principal del rol de abogados.
2. El sistema presenta la pantalla de mensajes de direccionamiento.
3. El usuario desliza a la izquierda el ítem de la lista de mensajes que desea crear el caso.
4. Se despliega la pantalla con el formulario de ingresar caso.
5. El usuario selecciona la especialidad del caso.
6. El usuario selecciona el abogado que se encargará del caso.
7. El usuario ingresa el número del caso.
8. El usuario ingresa el nombre del caso.
9. El usuario ingresa la descripción del caso.
10. El usuario cliquea en *Guardar.*
11. El sistema almacena la información. (E1)

Nombre: **F11.2. Editar caso**

Descripción: Permite que el usuario edite un caso.

Usuarios: Abogado

Flujo principal:

1. El usuario cliquea el botón *Asesoría* o *Acción* de la pantalla principal del rol de abogados.
2. El sistema presenta la pantalla de asesoría o acción respectivamente.
3. El usuario desliza a la izquierda el ítem de la lista de mensajes que desea editar el caso.
4. Se despliega la pantalla con el formulario del caso.
5. El usuario edita la especialidad del caso.
6. El usuario edita el abogado que se encargará del caso.
7. El usuario edita el nombre del caso.
8. El usuario edita la descripción del caso.
9. El usuario cliquea en *Guardar.*
10. El sistema almacena la información. (E1)

Nombre: **F11.3. Consultar caso con rol de abogado**

Descripción: Permite que el usuario visualice sus casos.

Usuarios: Abogado

Flujo principal:

1. El usuario cliquea el botón *Direccionamiento, Asesoría* o *Acción* de la pantalla principal del rol de abogados.
2. El sistema presenta la pantalla de direccionamiento, asesoría o acción respectivamente.
3. El sistema presenta en una lista los casos que le corresponden y están abiertos. (E1)

Nombre: **F11.4. Consultar caso con rol de abogado por parámetros**

Descripción: Permite que el usuario visualice sus casos por nombre del cliente.

Usuarios: Abogado

Flujo principal:

1. El usuario cliquea el botón *Direccionamiento, Asesoría* o *Acción* de la pantalla principal del rol de abogados.
2. El sistema presenta la pantalla de direccionamiento, asesoría o acción respectivamente.
3. El sistema presenta en una lista los casos que le corresponden y están abiertos.
4. El usuario ingresa el nombre del cliente, nombre del caso o número del caso deseado.
5. El sistema presenta en una lista los casos que coinciden con el filtro, le corresponden, y están abiertos. (E1)

Nombre: **F11.5. Consultar caso con rol de administrador**

Descripción: Permite que el usuario visualice los casos registrados en el sistema.

Usuarios: Administrador

Flujo principal:

Se usa el mismo esquema del caso de uso F2.4. Revisar sección de adjuntos.

Nombre: **F11.6. Consultar caso por parámetros con rol de administrador.**

Descripción: Permite que el usuario consulte los casos registrados en el sistema por nombre de cliente, código de caso, nombre de caso.

Usuarios: Administrador

Flujo principal:

Se usa el mismo esquema del caso de uso F7.4.2. Revisar sección de adjuntos.

12) Gestión de pago

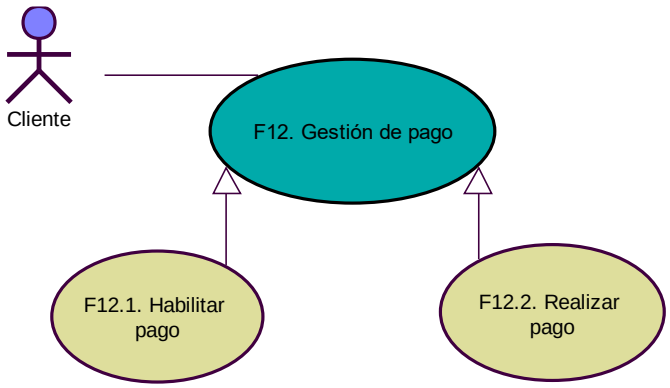

Figura 20 Gestión de pago. *Fuente: Elaboración propia*

Nombre: **F12.1. Habilitar pago**

Descripción: Permite que el usuario habilite el pago.

Usuarios: Abogado

Flujo principal:

1. El usuario cliquea el botón *Direccionamiento* de la pantalla principal del rol de abogados.
2. El sistema presenta la pantalla de mensajes de direccionamiento.
3. El usuario desliza a la derecha el ítem de la lista de mensajes que desea habilitar el pago.
4. Se despliega la pantalla con el formulario de habilitar pago.
5. El usuario ingresa el valor. (E2)
6. El usuario ingresa el porcentaje de impuesto. (E2)
7. El usuario cliquea en *Habilita el pago.*

8. El sistema almacena la información y habilita el pago. (E1)

Nombre: **F12.2. Realizar pago**

Descripción: Permite que el usuario realice el pago de la consultoría o la acción legal.

Usuarios: Cliente

Flujo principal:

1. El usuario cliquea el botón de la especialidad necesitada en la pantalla principal del rol de cliente.
2. El sistema presenta la pantalla de opciones para consultoría legal.
3. El usuario cliquea el botón de asesoría.
4. El usuario cliquea en el botón de *Pagar valor por consultoría* o *Pagar valor por acción legal*.
5. Se despliega la pantalla con la cuenta, los detalles y valores a pagar.
6. El usuario cliquea el botón *Realiza pago*.
7. El sistema envía el pago. (E1)

13) Gestión de mensajes

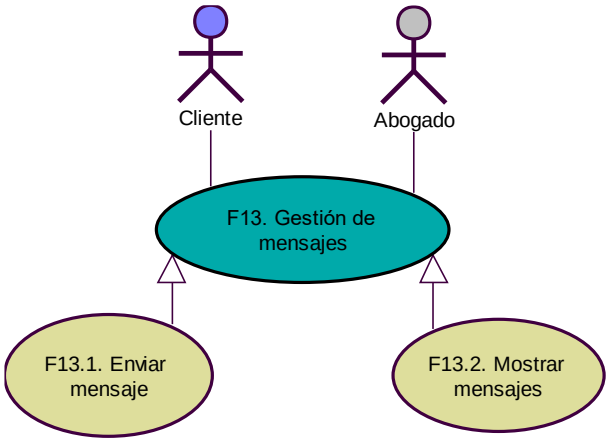

Figura 21 Gestión de mensajes. *Fuente: Elaboración propia*

Nombre: **F13.1. Enviar mensaje**

Descripción: Permite que el usuario envíe un mensaje.

Usuarios: Abogado, cliente

Flujo principal:

1. El usuario cliquea el botón *Direccionamiento, Asesoría* o *Acción legal* de la pantalla principal del rol de abogados o de clientes.
2. El sistema presenta la pantalla de mensajes.
3. El usuario ingresa el mensaje.
4. El usuario cliquea el botón de enviar.
5. El sistema almacena la información y envía el mensaje. (E1)

Nombre: **F13.2. Mostrar mensajes**

Descripción: Permite que el usuario visualice los mensajes.

Usuarios: Abogado, cliente

Flujo principal:

1. El usuario cliquea el botón *Direccionamiento, Asesoría* o *Acción legal* de la pantalla principal del rol de abogados o de clientes.
2. El sistema presenta la pantalla de mensajes.
3. Se despliega los mensajes. (E1)

g) Requisitos no funcionales

a. Requisitos de rendimiento

- La cantidad de peticiones AJAX, deben ser controladas para evitar una sobrecarga en el servidor.

b. Seguridad

- A partir del login, se debe identificar el rol del usuario, de modo que se identifiquen los permisos que poseen para las distintas acciones en el sistema.
    - El rol de administrador permite ingresar, actualizar, eliminar y consultar los abogados, tipos de especialidad, tipos de identificación, tipos de tarjeta, preguntas frecuentes, especialidades y roles; además permite visualizar los casos legales del sistema.
    - El rol de cliente permite acceder a un proceso secuencial de direccionamiento, pago por consultoría, asesoría legal, pago por llevar a cabo la acción legal correspondiente, y la acción. En cada uno de los pasos el cliente mantiene una comunicación en línea con el abogado delegado al caso.
    - El rol de abogado permite asesorar a los clientes, mediante un chat en línea para el direccionamiento, la consultoría y la acción legal. Además, este rol es quien ingresa los casos en el sistema y redirige a los clientes de una fase a otra.
- Las tablas en las bases de datos deberán tener campos de auditoría para almacenar a los usuarios que crearon y modificaron el archivo, además de la fecha y hora respectiva para cada caso.

c. Fiabilidad

- Los errores deben ser controlados y notificados mediante mensajes informativos.

- La conexión entre cada capa debe funcionar de manera correcta todo el tiempo, de modo que no existan pérdidas de datos.

d. Disponibilidad

- El servidor debe estar disponible las 24 horas del día, los 7 días de la semana para que los usuarios puedan tener un uso continuo de la aplicación.

e. Mantenibilidad

- Es necesario seguir estándares de nomenclatura para variables, funciones y componentes para facilitar el mantenimiento de la aplicación.

f. Portabilidad

- Es importante la verificación de que la aplicación sea compatible para dispositivos con sistema operativo Android y que también funcione en distintos navegadores para asegurar su portabilidad.

3.1.3. Establecimiento del proyecto

En esta subfase se requiere la configuración de la línea base que constituirá la creación del proyecto.

3.1.3.1. Establecer la línea de proceso base

La línea base consiste en los pilares que ayudarán a que el proyecto se desarrolle apropiadamente, por lo que es importante iniciar con la configuración en cada capa del paradigma MVC.
Para el establecimiento del proyecto en Codeigniter se requiere descargar los ficheros de la versión 3.1.10 en la página oficial de Codeigniter. Se deben realizar las siguientes configuraciones:

- En la ruta *../application/config/database.php* se debe configurar la conexión con la base de datos como se muestra en la figura 22.

```
'dsn' => '',
'hostname' => 'localhost',
'username' => 'root',
'password' => 'root',
'database' => 'legal_help',
'dbdriver' => 'mysqli',
'dbprefix' => '',
'pconnect' => FALSE,
'db_debug' => (ENVIRONMENT !== 'production'),
'cache_on' => FALSE,
'cachedir' => '',
'char_set' => 'utf8',
'dbcollat' => 'utf8_general_ci',
'swap_pre' => '',
'encrypt' => FALSE,
'compress' => FALSE,
'stricton' => FALSE,
'failover' => array(),
'save_queries' => TRUE
```

Figura 22 Configuración de base de datos en Codeigniter. *Fuente: Elaboración propia*

- En la ruta *../application/config/config* se configura la línea 26 como se muestra en la figura 23.

```
$config['base_url'] = 'http://127.0.0.1/servidor_legalhelp';
```

Figura 23 Configuración de la URL base. *Fuente: Elaboración propia*

- En la ruta ../application/controllers se deben crear archivos con extensión php para los controladores de cada entidad del sistema.
- En la ruta ../application/models se deben crear archivos con extensión php para la conexión del controlador con la base de datos.

Para la creación del proyecto con Sencha Architect 4.2.4 es necesario crear un proyecto en blanco con Ext JS 6.6.x Modern.

Para la creación de la base de datos es necesaria la configuración de la figura 24.

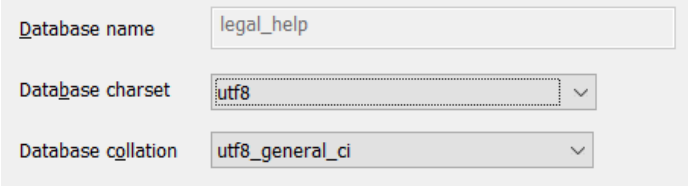

Figura 24 Creación de la base de datos. *Fuente: Elaboración propia*

Versionar el proyecto es importante tanto para almacenarlo como para mantener un orden respecto a los cambios que se realizan en el modelo, vista y controlador. Para lograr esto se utiliza GitHub como repositorio y se configura en PhpStorm 2018.1.6 seleccionando la opción *VCS* del menú principal y escoger la opción *Checkout from Version Control* y seleccionar *Git.* Se abrirá una ventana donde se ingresará la dirección URL del repositorio creado en GitHub para el proyecto y se hace clic en *Clone*. Una vez clonado el repositorio localmente se deben añadir los ficheros al repositorio en GitHub, para lo cual se hace clic derecho en la carpeta raíz y se selecciona en *Git* y *Add.* Por último, se debe subir los ficheros al repositorio haciendo clic derecho en la carpeta raíz y se selecciona en *Git* y *Commit Directory.*

3.1.3.2. Planificar la documentación necesaria

La documentación necesaria son los resultados que se obtienen en cada una de las fases.

Tabla 18

Fase	Iteración	Subfase	Salidas
Fase de exploración	Iteración 1	Establecimiento de interesados	Lista de interesados
		Definición de alcance	Documento de requerimientos iniciales
		Establecimiento del proyecto	Plan de proyecto inicial, lista de actividades de aseguramiento de la calidad, plan de formación
Fase de inicialización	Iteración 1	Puesta en marcha del proyecto	Plan de proyecto actualizado,
		Planificación inicial	documento de diseño de software, lista de desarrollo de interfaz de usuario

		Día de prueba	Pruebas de cada requerimiento
Fase de producto	Iteración 1	Día de planificación	Documento de aceptación de pruebas
		Día de trabajo	Desarrollo de las funcionalidades según su prioridad, tarjetas de tareas
		Día de liberación	Lista de defectos, informe de estado diario
Fase de estabilización	Iteración 1 Iteración 2	Día de planificación Día de trabajo	Integración del sistema
		Documentación	Documentación del proyecto
		Día de liberación	Lista de defectos, informe de estado diario
Fase de pruebas	Iteración 2	Pruebas del sistema Día de planificación Día de trabajo Día de liberación	Documento de errores y pruebas del sistema.

Fuente: Elaboración propia

3.1.3.3. Identificar las necesidades de formación

Una de las necesidades más importantes para iniciar el proyecto es el conocimiento requerido para manejar ExtJS 6.6.x, y el framework Codeigniter. El primero está basado en el lenguaje JavaScript, el cual tiene propiedades, funciones y configuraciones exclusivas de la librería. El segundo está basado en el lenguaje PHP y de igual manera tiene funciones propias del framework que lo caracteriza como un marco de trabajo útil. Por lo tanto, es importante la formación y capacitación de ExtJS y Codeigniter.

3.2. Fase de inicialización

3.2.1. Documento de diseño de software

3.2.1.1. Diagrama de actividad

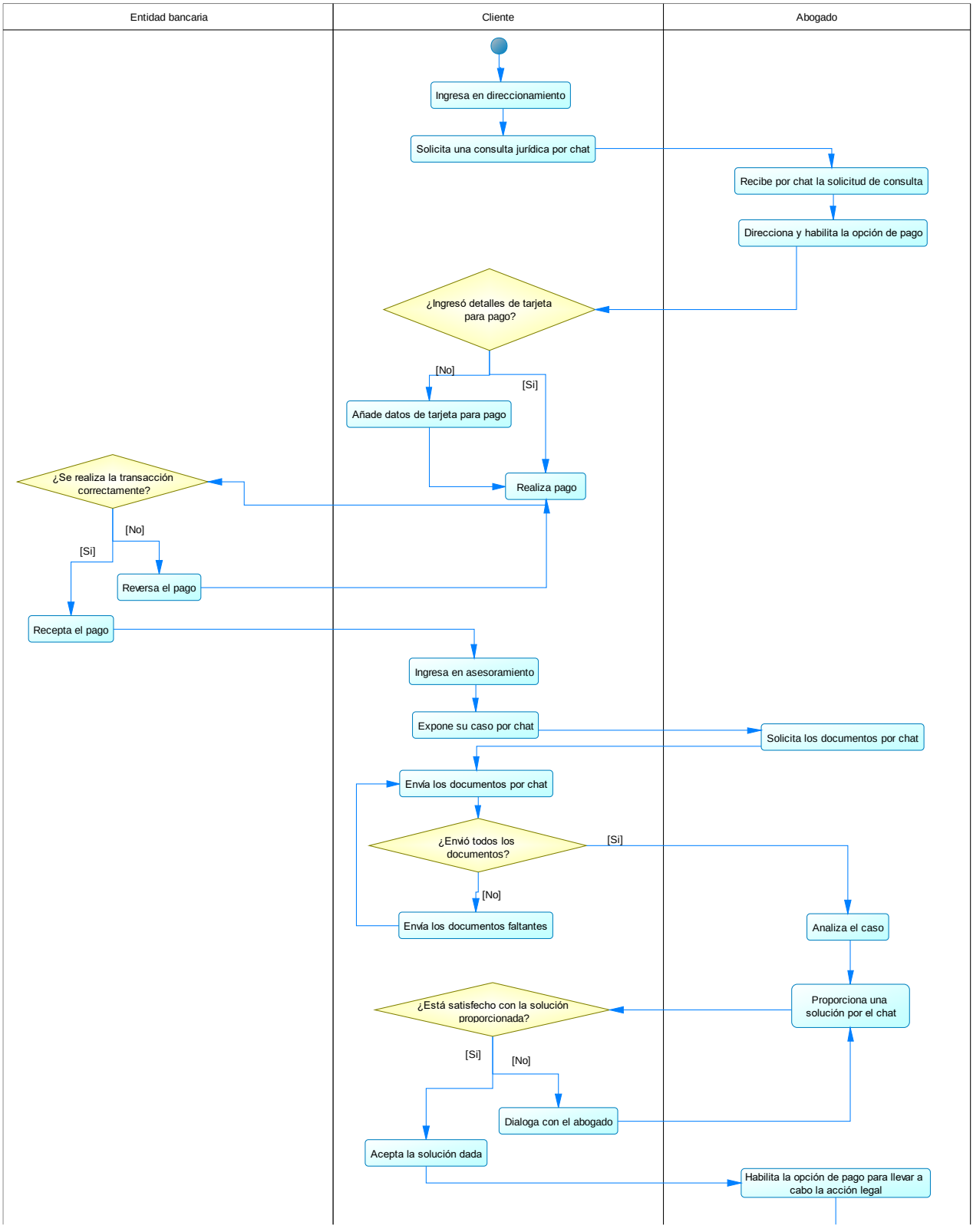

Figura 25 Diagrama de actividad parte 1. *Fuente: Elaboración propia*

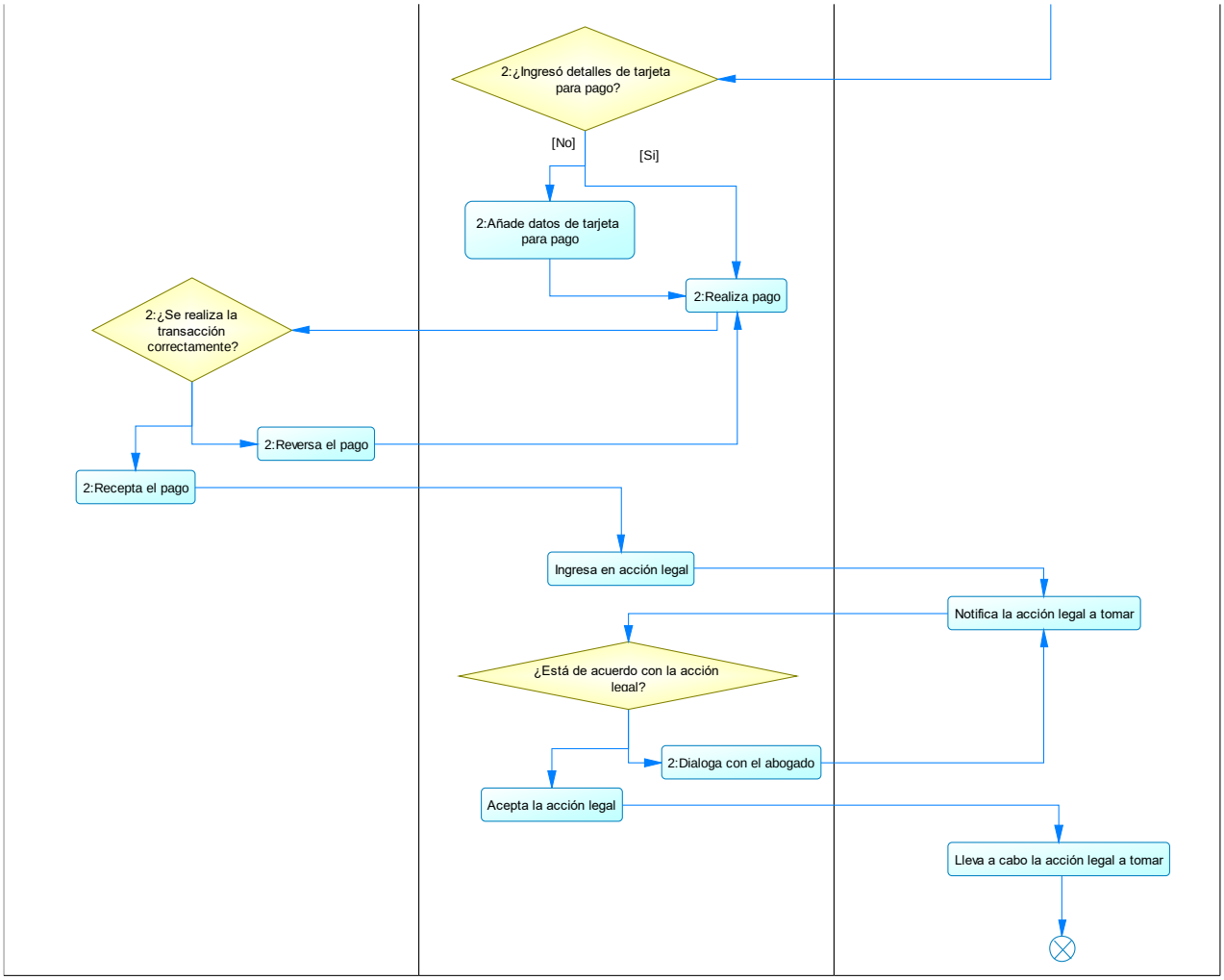

Figura 26 Diagrama de actividad parte 2. *Fuente: Elaboración propia*

3.2.1.2. Diagrama entidad relación

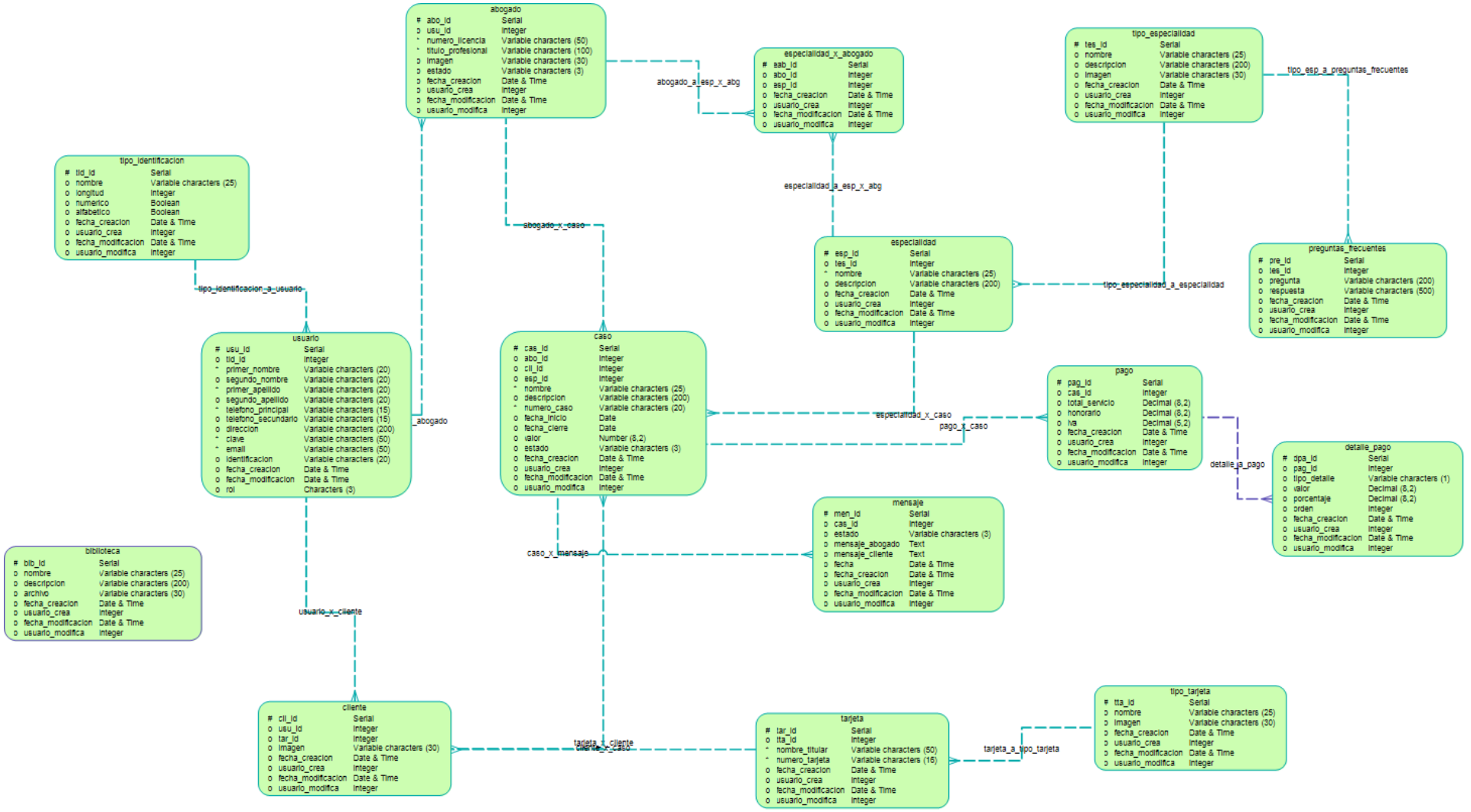

Figura 27 Diagrama entidad relación. *Fuente: Elaboración propia*

3.2.1.3. Diagrama conceptual

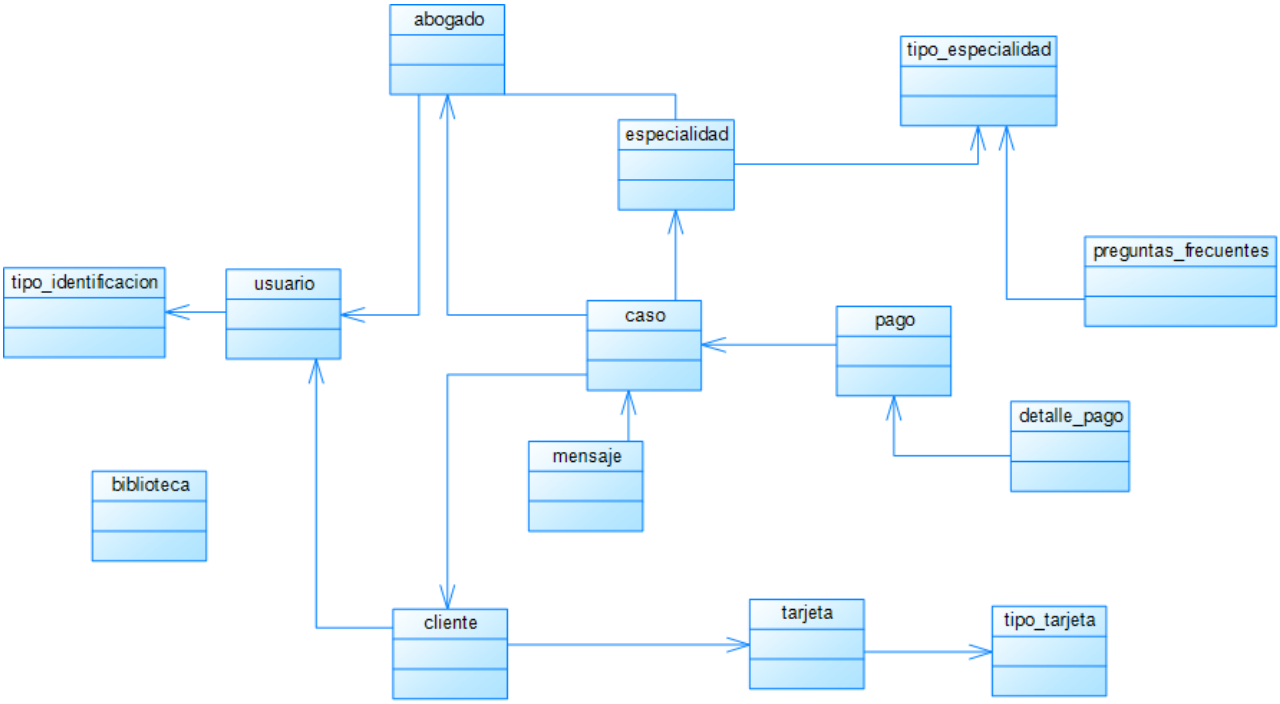

Figura 28 Diagrama conceptual. *Fuente: Elaboración propia*

3.2.2. Lista de desarrollo de interfaz de usuario

- Pantalla de inicio de sesión:

Figura 29 Pantalla inicial. *Fuente: Elaboración propia*

- Pantalla principal con rol de administrador:

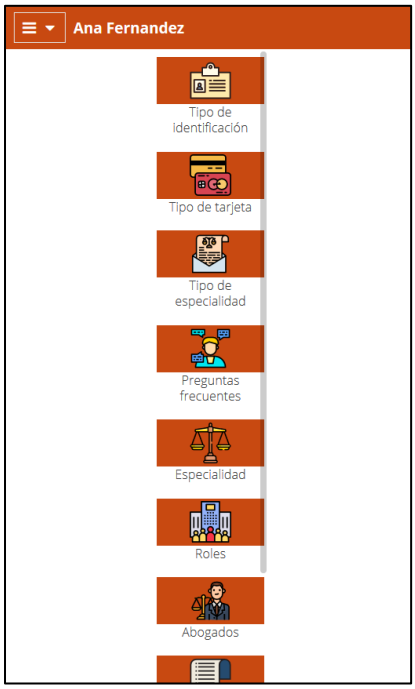

Figura 30 Pantalla principal rol de administrador. *Fuente: Elaboración propia*

- Pantalla de administración de tipo de identificación:

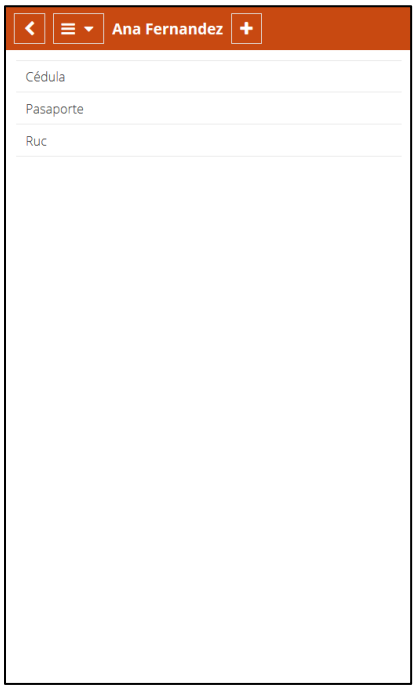

Figura 31 Pantalla de administración de tipo de identificación. *Fuente: Elaboración propia*

- Pantalla de edición de tipo de identificación:

Figura 32 Pantalla de edición de tipo de identificación. *Fuente: Elaboración propia*

- Pantalla de administración de tipo de tarjeta:

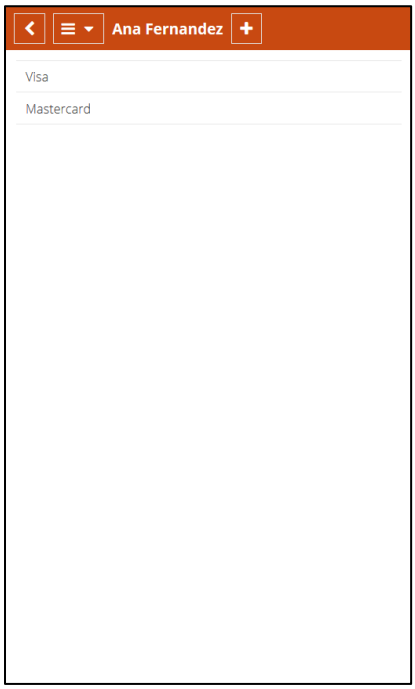

Figura 33 Pantalla de administración de tipo de tarjeta. *Fuente: Elaboración propia*

- Pantalla de edición de tipo de tarjeta:

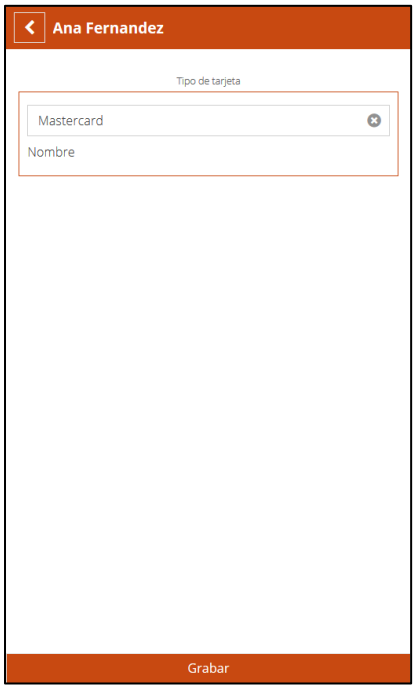

Figura 34 Pantalla de edición de tipo de tarjeta. *Fuente: Elaboración propia*

- Pantalla de administración de tipo de especialidad:

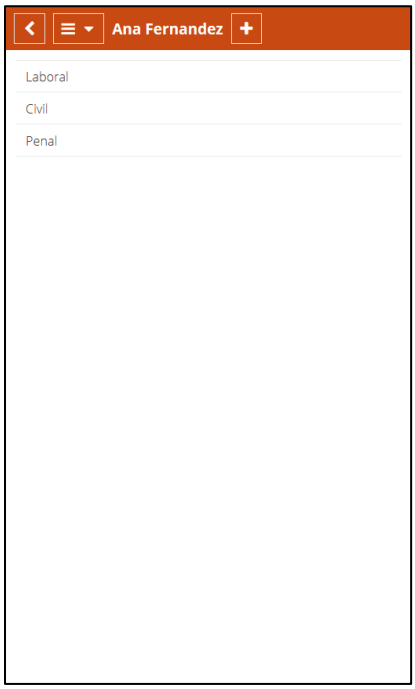

Figura 35 Pantalla de administración de tipo de especialidad. *Fuente: Elaboración propia*

- Pantalla de edición de tipo de especialidad:

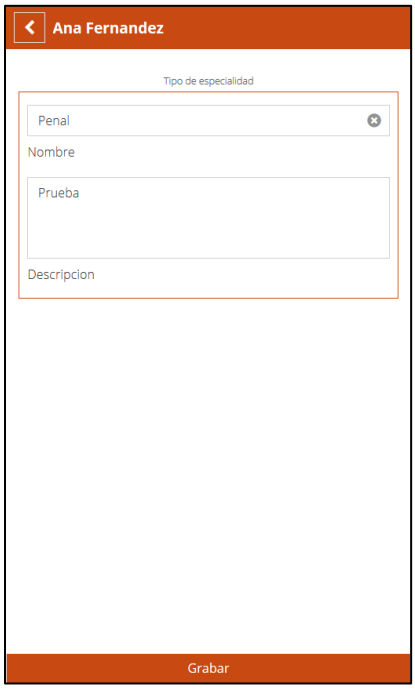

Figura 36 Pantalla de edición de tipo de especialidad. *Fuente: Elaboración propia*

- Pantalla de administración de preguntas frecuentes:

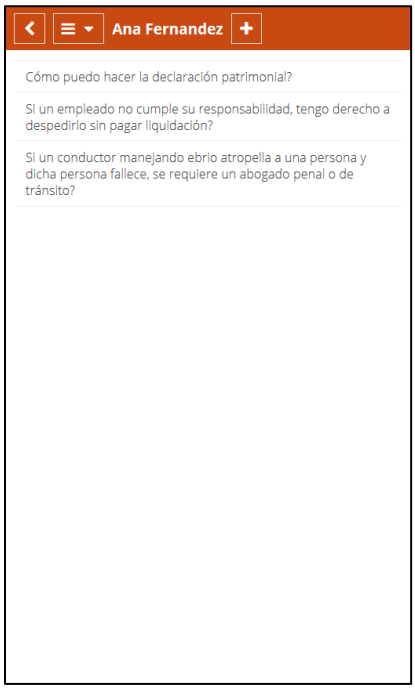

Figura 37 Pantalla de administración de preguntas frecuentes. *Fuente: Elaboración propia*

- Pantalla de edición de preguntas frecuentes:

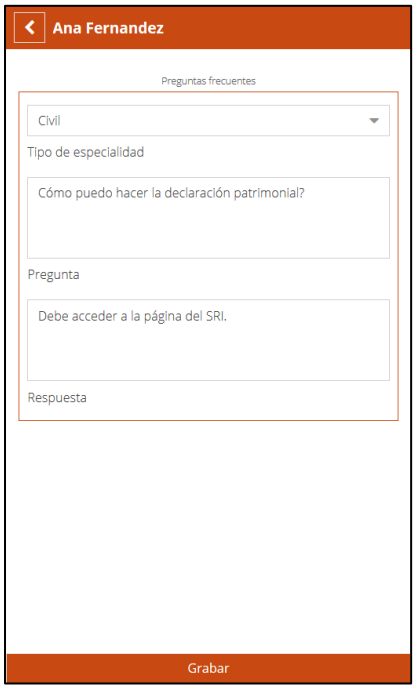

Figura 38 Pantalla de edición de preguntas frecuentes. *Fuente: Elaboración propia*

- Pantalla de administración de especialidad:

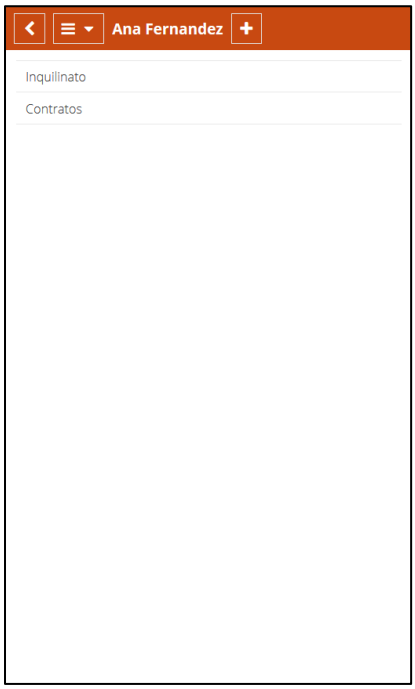

Figura 39 Pantalla de administración de especialidad. *Fuente: Elaboración propia*

- Pantalla de edición de especialidad:

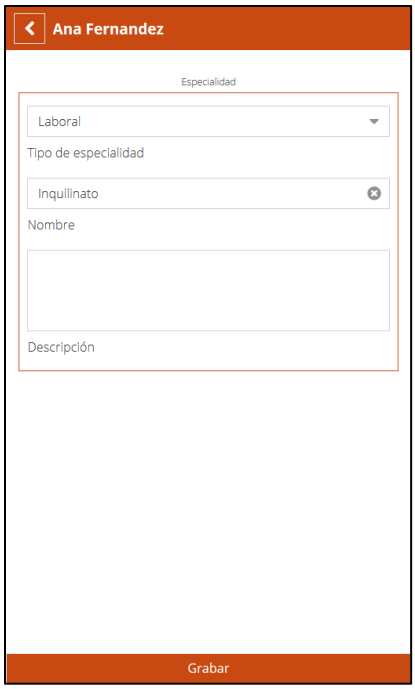

Figura 40 Pantalla de edición de especialidad. *Fuente: Elaboración propia*

- Pantalla de administración de roles:

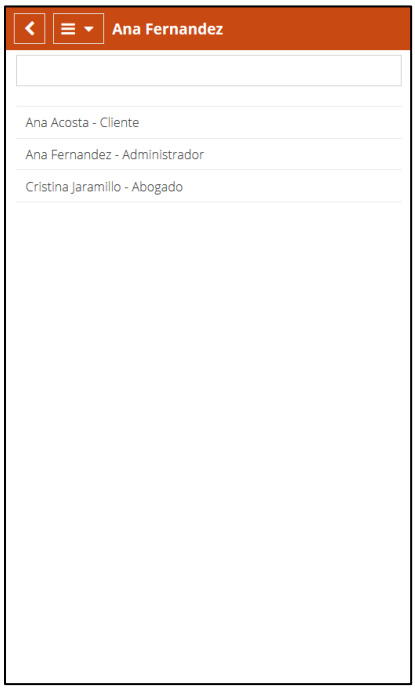

Figura 41 Pantalla de administración de roles. *Fuente: Elaboración propia*

- Pantalla de edición de roles:

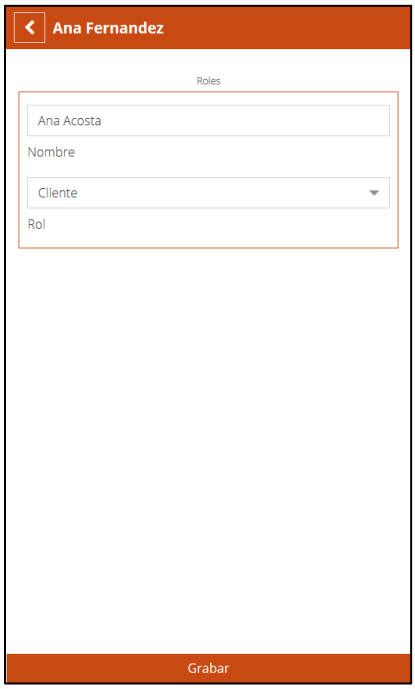

Figura 42 Pantalla de edición de roles. *Fuente: Elaboración propia*

- Pantalla de administración de abogados:

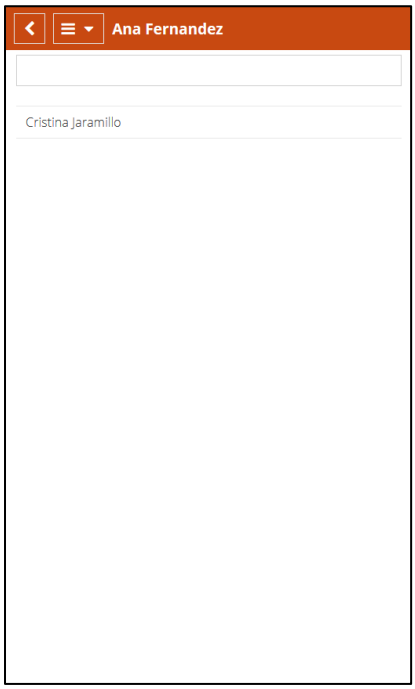

Figura 43 Pantalla de administración de abogados. *Fuente: Elaboración propia*

- Pantalla de edición de abogados:

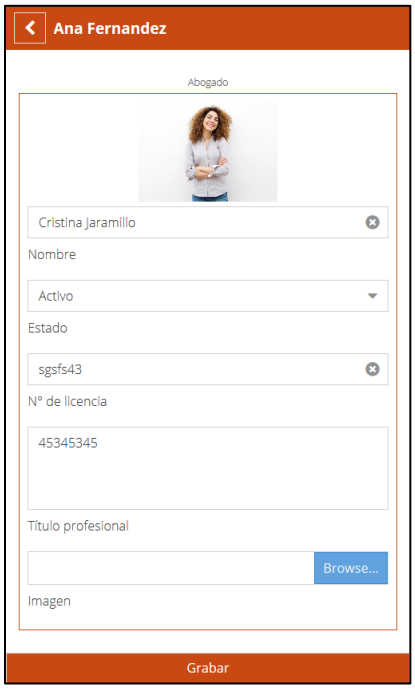

Figura 44 Pantalla de edición de abogados. *Fuente: Elaboración propia*

- Pantalla de administración de biblioteca:

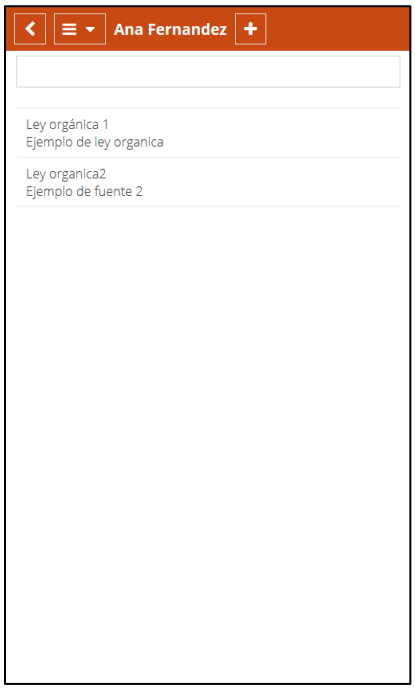

Figura 45 Pantalla de administración de biblioteca. *Fuente: Elaboración propia*

- Pantalla de edición de biblioteca:

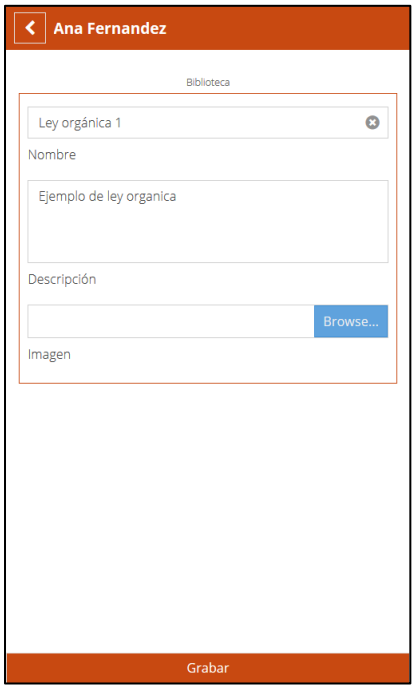

Figura 46 Pantalla de edición de biblioteca. *Fuente: Elaboración propia*

- Pantalla de visualización de biblioteca:

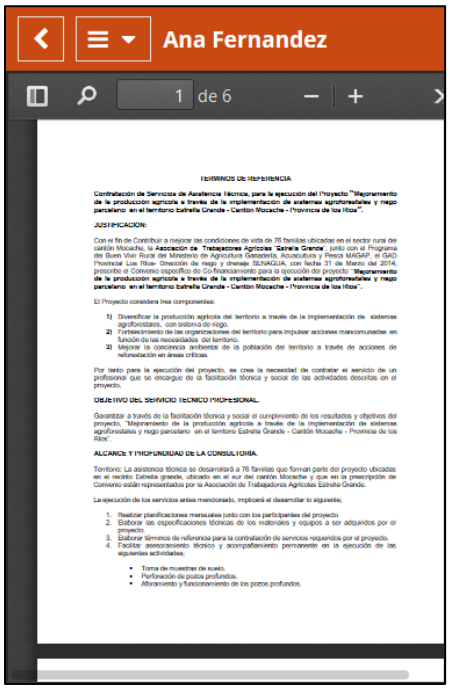

Figura 47 Pantalla de visualización de biblioteca. *Fuente: Elaboración propia*

- Pantalla principal con rol de cliente:

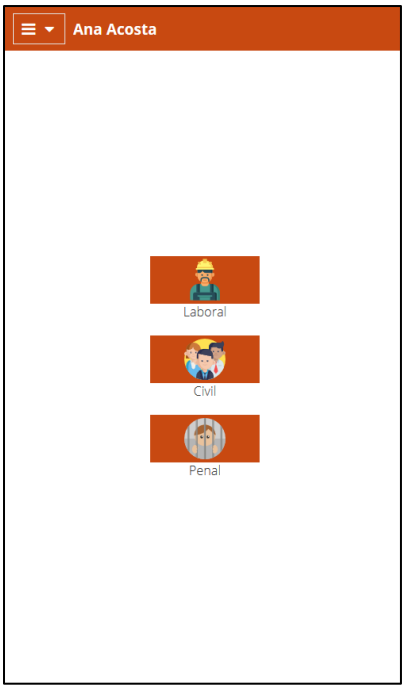

Figura 48 Pantalla principal con rol de cliente. *Fuente: Elaboración propia*

- Pantalla de asesoría o preguntas frecuentes de acuerdo con el tipo de especialidad:

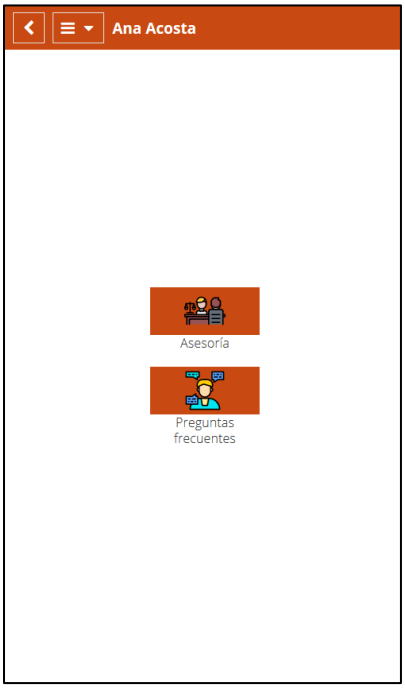

Figura 49 Pantalla de rol de cliente. *Fuente: Elaboración propia*

- Pantalla de asesoría:

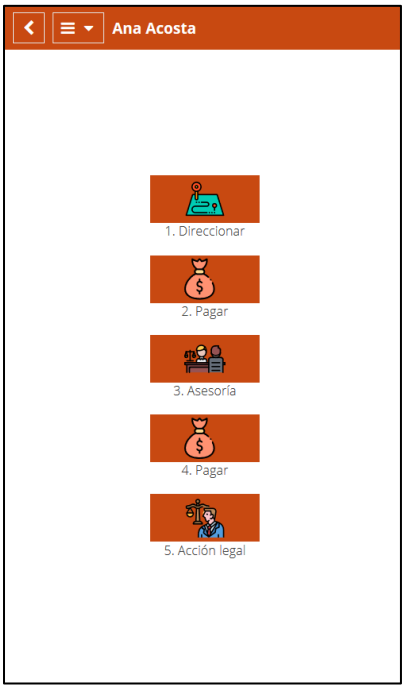

Figura 50 Pantalla de asesoría. *Fuente: Elaboración propia*

- Pantalla de preguntas frecuentes:

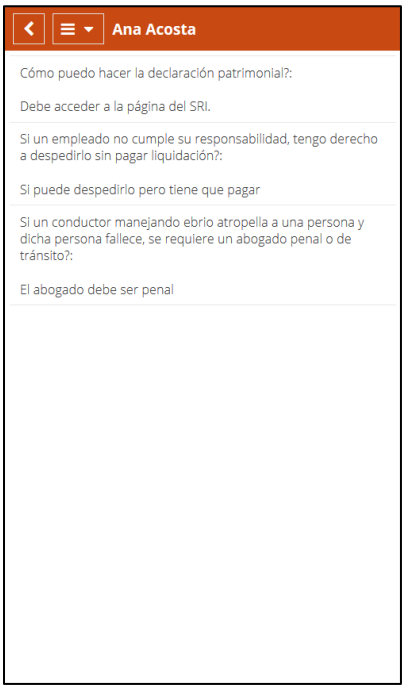

Figura 51 Pantalla de preguntas frecuentes. *Fuente: Elaboración propia*

- Pantalla principal con rol de abogado:

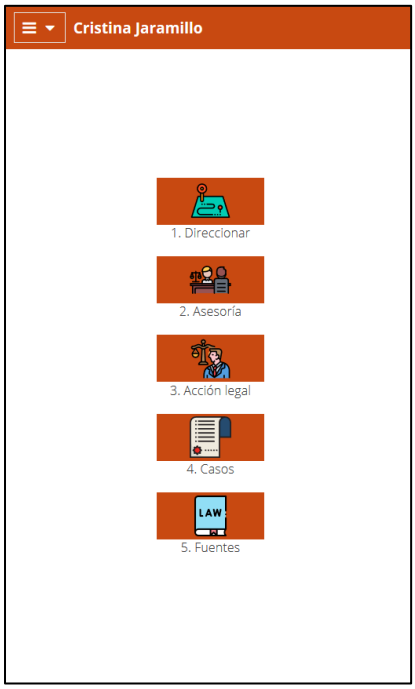

Figura 52 Pantalla principal con rol de abogado. *Fuente: Elaboración propia*

- Pantalla de mensajes por caso:

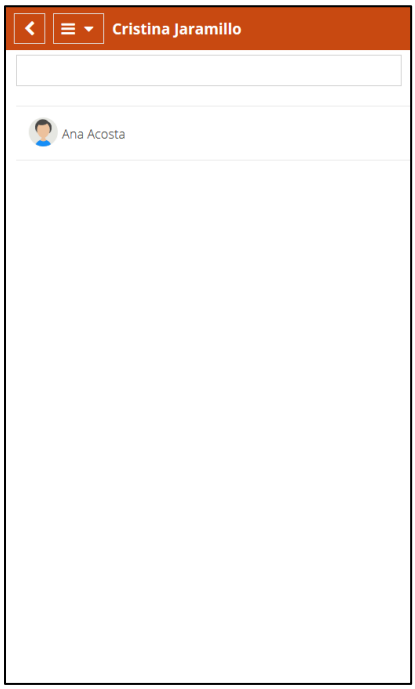

Figura 53 Pantalla de mensajes por caso. *Fuente: Elaboración propia*

- Pantalla de chat:

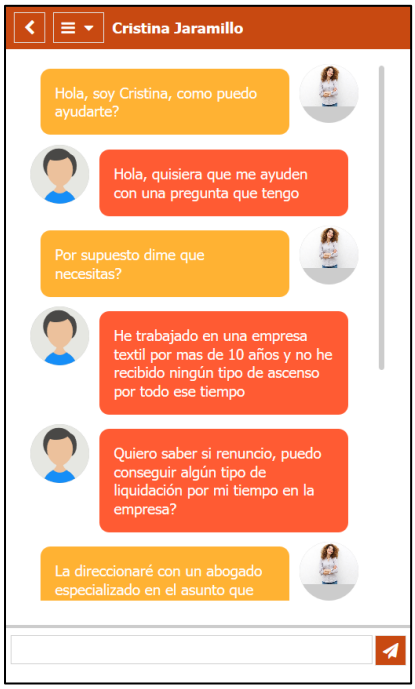

Figura 54 Pantalla de chat. *Fuente: Elaboración propia*

3.2.3. Pruebas de cada requerimiento

Las pruebas deben realizarse a cada uno de los requerimientos listados en el "documento de requerimientos", por lo que es necesario hacer pruebas de aceptación.

Los recursos necesarios para llevar a cabo las pruebas de aceptación serán:

- Documento de requerimientos,
- tabla de evaluación de cada prueba,
- tabla de verificación para las pruebas,
- la autora de la presente disertación, quien realizará las pruebas correspondientes para determinar si el criterio de aceptación fue cumplido o no, de acuerdo con la tabla de verificación.

El estimado de tiempo para hacer las pruebas pertinentes a cada requerimiento es de 5 a 10 minutos si su criterio de aceptación es cumplido satisfactoriamente, de otro modo el tiempo puede ser mayor.

La lista de verificación a seguir para realizar las pruebas es:

Tabla 19

Lista de verificación

Abreviatura	Verificación	Descripción	Principios a considerar
SAT	Satisfactorio	Ha cumplido con el requerimiento completa y satisfactoriamente.	- Completo - Conciso - Factible - Manejable
MEJ	Mejorable	Ha cumplido con el requerimiento completamente, pero es necesario realizar un cambio para mejorarlo.	
PAR	Parcialmente satisfactorio	No se ha cumplido parcialmente con el requerimiento.	
INS	Insatisfactorio	No se ha cumplido con el requerimiento en su totalidad.	

Fuente: Elaboración propia

Capítulo 4. Desarrollo e implementación de la aplicación

Una vez culminadas las fases anteriores, se puede dar paso al desarrollo de la aplicación, la cual está comprendida en 3 subfases: producto, estabilización y pruebas.

4.1. Fase de producto

4.1.1. Implementación de funcionalidades

Para el desarrollo de las funcionalidades, es necesario seguir un estándar de nomenclatura tanto para la base de datos como para las variables, funciones y archivos del backend y frontend.

4.1.1.1. Estándares de nomenclatura

4.1.1.1.1. Nomenclatura base de datos

El nombre de las tablas de la base de datos es en minúsculas, si hay dos o más palabras para el nombre de la tabla se utiliza el guion bajo (_) para unir las palabras.
Ejemplo: nombre_de_la_tabla.
Para la clave primaria de las tablas con una sola palabra se utilizan las tres primeras letras del nombre de la tabla respectiva seguido de un guion bajo y la palabra *id*.
Ejemplo: (tabla: abogado) abo_id
Para la clave primaria de las tablas comprendidas con dos o más palabras se utiliza la primera letra de la primera palabra y las dos primeras letras de la segunda palabra, considerando que, si la segunda palabra solo tiene una letra, se usará la tercera palabra del nombre de la tabla. Seguido de un guion bajo y la palabra *id.*
Ejemplo: (tabla: tipo_identificacion) tid_id, (especialidad_x_abogado) eab_id

4.1.1.1.2. Nomenclatura de variables

Para identificar a los componentes en el frontend se debe utilizar la abreviatura en minúscula de dicho componente seguido del identificador de la ventana que lo contiene, cuya primera letra debe estar en mayúsculas. Seguido de un dígito numérico comenzando en 1, el cual incrementa en 1 por cada componente cuyas características anteriores sean similares.

Tabla 20

Abreviaturas de los componentes visuales

Componente	Abreviatura
Panel	pnl
Button	btn
Form	frm
List	lst
Grid	grd
Display field	dsf
Column	col
Date field	dtf
Text field	txt
Password field	pas
Number field	num
Email field	eml

Fuente: Elaboración propia

Ejemplo: (componente padre que contiene un botón 1: Win001) btnWin0011, (componente padre que contiene un botón 2: Win001) btnWin0012

El identificador de todas las ventanas padre se otorga de acuerdo al nombre del archivo comenzando con la palabra *Win* seguido de 3 cifras numéricas que se colocan en orden.

Ejemplo: Win001, Win002, Win003

4.1.1.1.3. Nomenclatura de funciones

Las funciones en el controlador del backend deben estar en minúsculas, la primera letra de cada palabra a partir de la segunda palabra debe estar en mayúscula.

Ejemplo: obtenerAbogadosPorNombre()

Las funciones en el modelo del backend deben llamarse igual a su respectiva función del controlador, pero anteponiendo un f al inicio.

Ejemplo: fObtenerAbogadosPorNombre()

De modo que las funciones en el frontend se derivan de un componente visual, su nomenclatura depende del identificador de dicho componente y del evento que se realiza.
Ejemplo: (componente: Win001, evento: onAdded) onWin001Added(), (componente: btnWin0011, evento: onTap) onBtnWin0011Tap()

4.1.1.1.4. Nomenclatura de archivos

El nombre de los controladores en el backend deben estar en minúsculas, si hay dos o más palabras para el nombre de la tabla se utiliza el guion bajo (_) para unir las palabras.
Ejemplo: tipo_identificacion.
El nombre de los modelos en el backend debe ser igual que su controlador respectivo añadiendo al final un guion bajo seguido de la palabra *model*.
Ejemplo: tipo_identificacion_model
El nombre de los archivos de modelo del frontend deben estar en minúsculas, comenzando con el nombre de la tabla de la base de datos respectiva, seguido de un punto (.) seguido del mismo nombre de la base de datos, pero sin los guiones bajos, sino todo seguido, considerando que la primera letra de la segunda palabra en adelante debe estar en mayúscula, seguido finalmente de la palabra *Model*.
Ejemplo: (tabla: detalle_pago) detalle_pago.detallePagoModel
El nombre de los archivos del store del frontend deben estar en minúsculas, comenzando con el nombre de la tabla de la base de datos respectiva, seguido de un punto (.) seguido del mismo nombre de la base de datos, pero sin los guiones bajos, sino todo seguido, considerando que la primera letra de la segunda palabra en adelante debe estar en mayúscula, seguido finalmente de la palabra *Store*.
Ejemplo: (tabla: detalle_pago) detalle_pago.detallePagoStore
El nombre de los archivos del view del frontend deben estar en minúsculas, comenzando con el nombre de la tabla de la base de datos respectiva, seguido de un punto (.) seguido de la abreviatura Win y tres dígitos numéricos que incrementan en 1.
Ejemplo: (tabla: detalle_pago) detalle_pago.Win008

4.1.1.2. Backend

4.1.1.2.1. Controlador

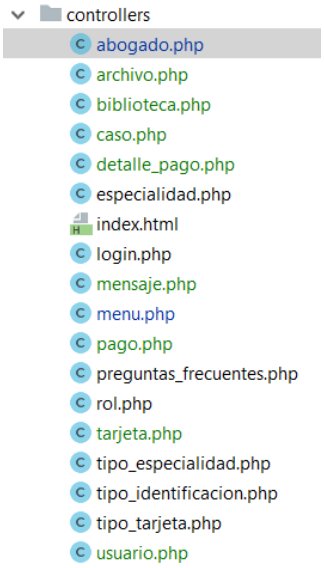

Figura 55 Captura de los archivos del controlador de backend. *Fuente: Elaboración propia*

- Abogado

```php
<?php

class Abogado extends CI_Controller
{
  public function __construct()
  {
    parent::__construct();
    $this->load->model( model: "abogado_model");
    $this->load->model( model: "general_model");
  }

  public function getAbogados(){...}
  public function saveAbogado(){...}
  public function getEstadosAbogado(){...}
}
```

Figura 56 Captura del backend de abogado. *Fuente: Elaboración propia*

- Archivo

```php
<?php

class Archivo extends CI_Controller
{
  public function __construct()
  {
    parent::__construct();
    $this->load->helper( helpers: 'file');
    $this->load->model( model: "abogado_model");
    $this->load->model( model: "general_model");
  }

  public function saveImage(){...}

}
```

Figura 57 Captura del backend de archivo. *Fuente: Elaboración propia*

- Biblioteca

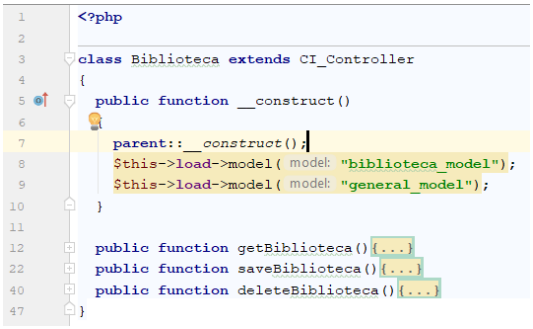

Figura 58 Captura del backend de biblioteca. *Fuente: Elaboración propia*

- Caso

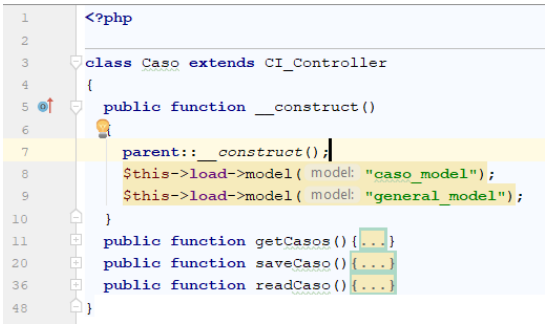

Figura 59 Captura del backend de caso. *Fuente: Elaboración propia*

- Detalle pago

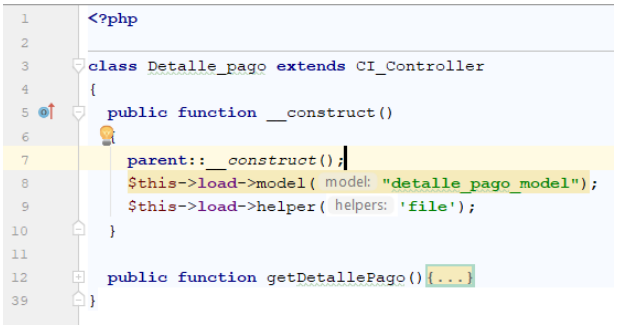

Figura 60 Captura del backend de detalle pago. *Fuente: Elaboración propia*

- Especialidad

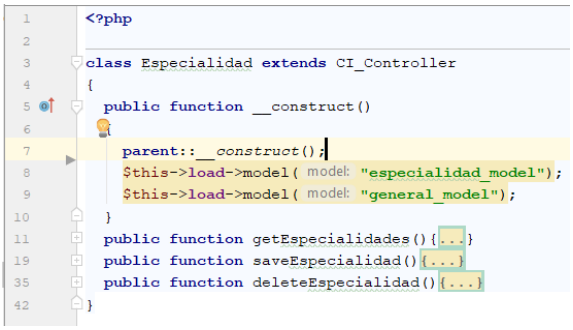

*Figura 61*Captura del backend de especialidad. *Fuente: Elaboración propia*

- Login

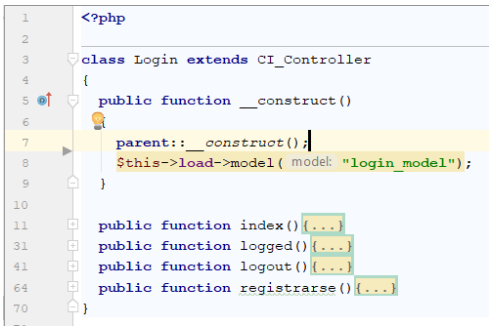

Figura 62 Captura del backend de login. *Fuente: Elaboración propia*

- Mensaje

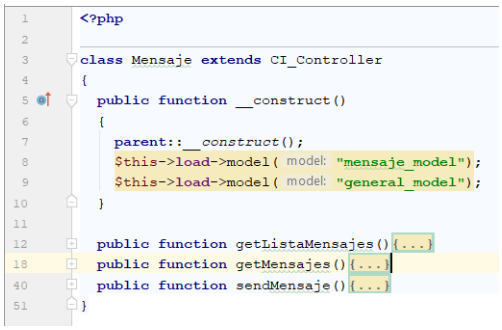

Figura 63 Captura del backend de mensaje. *Fuente: Elaboración propia*

- Pago

```php
<?php

class Pago extends CI_Controller
{
  public function __construct()
  {
    parent::__construct();
    $this->load->model( model: "pago_model");
    $this->load->model( model: "general_model");
  }
  public function habilitaPago(){...}

}
```

Figura 64 Captura del backend de pago. *Fuente: Elaboración propia*

- Preguntas frecuentes

```php
<?php

class preguntas_frecuentes extends CI_Controller
{
  public function __construct()
  {
    parent::__construct();
    $this->load->model( model: "preguntas_frecuentes_model");
    $this->load->model( model: "general_model");
  }
  public function getPreguntasFrecuentes(){...}
  public function savePreguntas(){...}
  public function deletePreguntas(){...}
}
```

Figura 65 Captura del backend de preguntas frecuentes. *Fuente: Elaboración propia*

- Rol

```php
<?php

class Rol extends CI_Controller
{
  public function __construct()
  {
    parent::__construct();
    $this->load->model( model: "rol_model");
    $this->load->model( model: "general_model");
  }
  public function getRoles(){...}
  public function getRolesCombo(){...}
  public function saveRol(){...}
}
```

Figura 66 Captura del backend de rol. *Fuente: Elaboración propia*

- Tarjeta

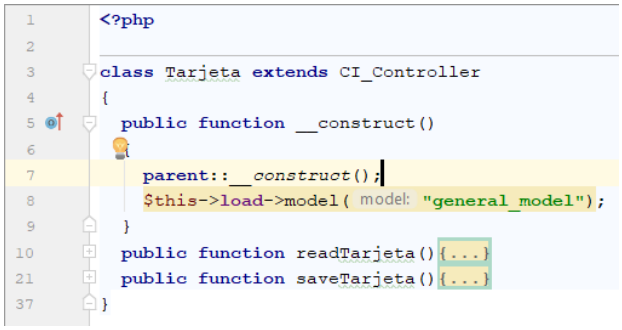

Figura 67 Captura del backend de tarjeta. *Fuente: Elaboración propia*

- Tipo especialidad

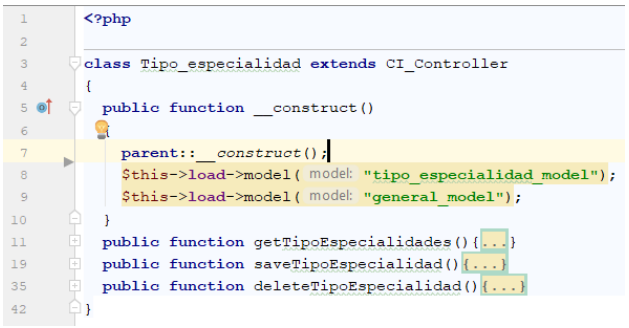

Figura 68 Captura del backend de tipo de especialidad. *Fuente: Elaboración propia*

- Tipo identificación

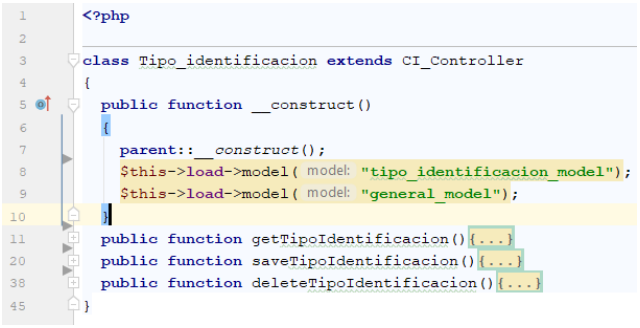

Figura 69 Captura del backend de tipo de identificación. *Fuente: Elaboración propia*

- Tipo tarjeta

```php
<?php

class Tipo_tarjeta extends CI_Controller
{
  public function __construct()
  {
    parent::__construct();
    $this->load->model( model: "tipo_tarjeta_model");
    $this->load->model( model: "general_model");
  }
  public function getTipoTarjeta(){...}
  public function saveTipoTarjeta(){...}
  public function deleteTipoTarjeta(){...}
}
```

Figura 70 Captura del backend de tipo de tarjeta. *Fuente: Elaboración propia*

- Usuario

```php
<?php

class Usuario extends CI_Controller
{
  public function __construct()
  {
    parent::__construct();
    $this->load->model( model: "general_model");
    $this->load->model( model: "usuario_model");
  }

  public function readUsuario(){...}
  public function saveUsuario(){...}
  public function cambiarContrasena(){...}
}
```

Figura 71 Captura del backend de usuario. *Fuente: Elaboración propia*

4.1.1.2.2. Model

Figura 72 Captura de los archivos del modelo de backend. *Fuente: Elaboración propia*

- Abogado

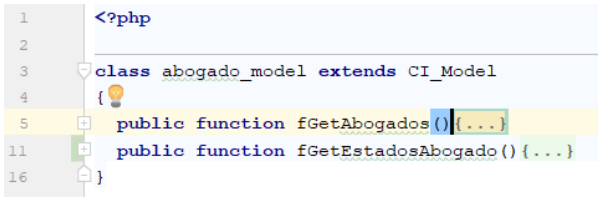

Figura 73 Captura del backend de abogado. *Fuente: Elaboración propia*

- Biblioteca

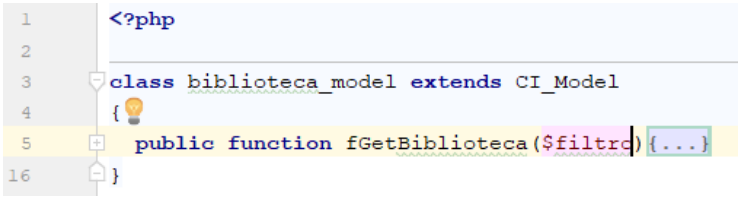

Figura 74 Captura del backend de biblioteca. *Fuente: Elaboración propia*

- Caso

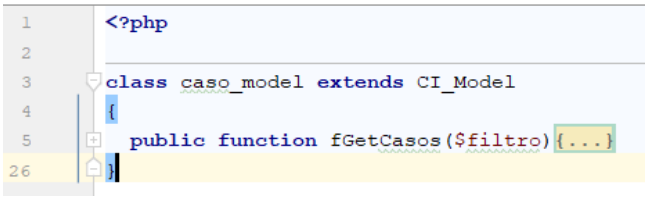

Figura 75 Captura del backend de caso. *Fuente: Elaboración propia*

- Detalle pago

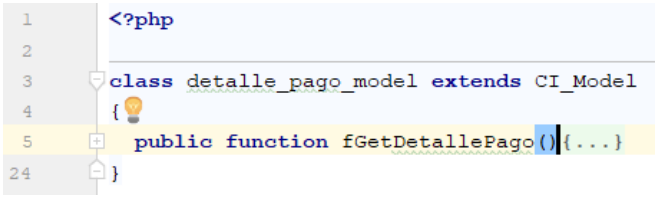

Figura 76 Captura del backend de detalle pago. *Fuente: Elaboración propia*

- Especialidad

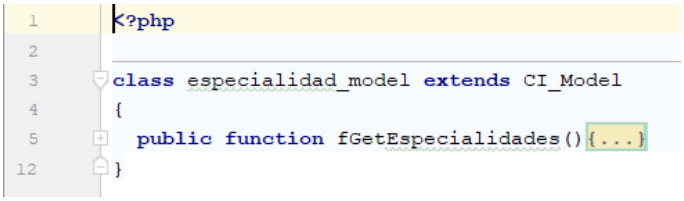

Figura 77 Captura del backend de especialidad. *Fuente: Elaboración propia*

- General

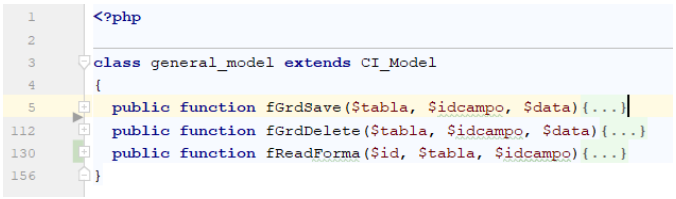

Figura 78 Captura del modelo general en backend. *Fuente: Elaboración propia*

- Login

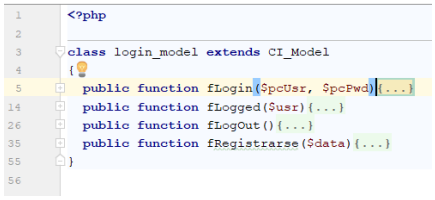

Figura 79 Captura del modelo de login en backend. *Fuente: Elaboración propia*

- Mensaje

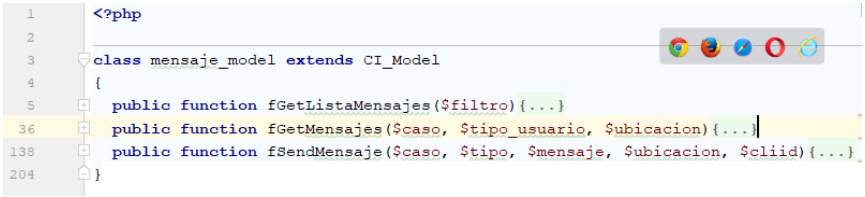

Figura 80 Captura del modelo de mensaje en backend. *Fuente: Elaboración propia*

- Pago

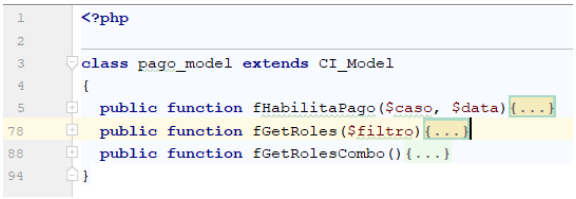

Figura 81 Captura del modelo de pago en backend. *Fuente: Elaboración propia*

- Preguntas frecuentes

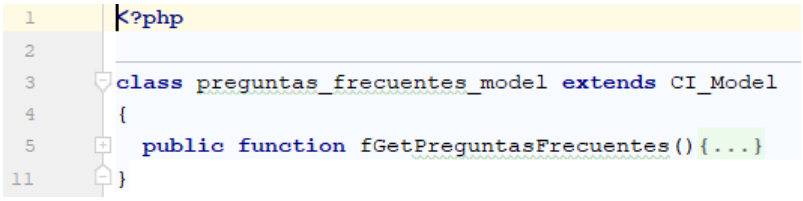

Figura 82 Captura del modelo de preguntas frecuentes en backend. *Fuente: Elaboración propia*

- Rol

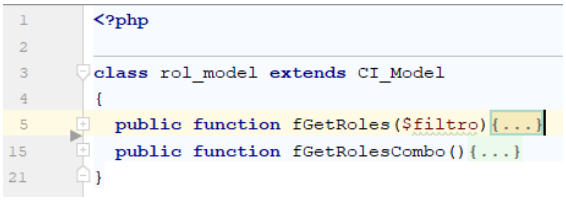

Figura 83 Captura del modelo de rol en backend. *Fuente: Elaboración propia*

- Tipo especialidad

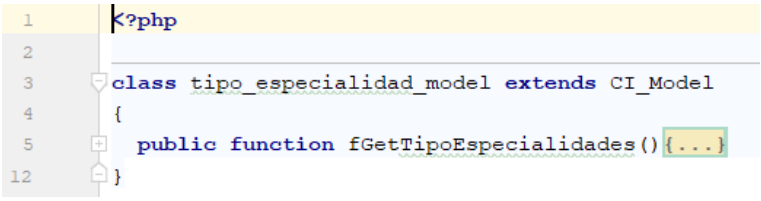

Figura 84 Captura del modelo de tipo de especialidad en backend. *Fuente: Elaboración propia*

- Tipo identificación

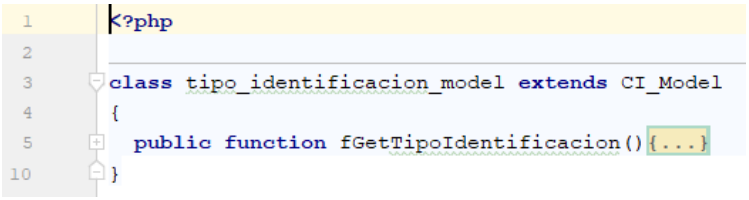

Figura 85 Captura del modelo de tipo de identificación en backend. *Fuente: Elaboración propia*

- Tipo tarjeta

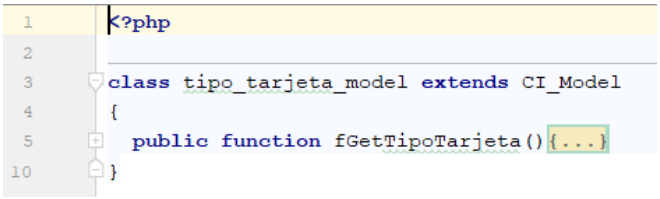

Figura 86 Captura del modelo de tipo de tarjeta en backend. *Fuente: Elaboración propia*

- Usuario

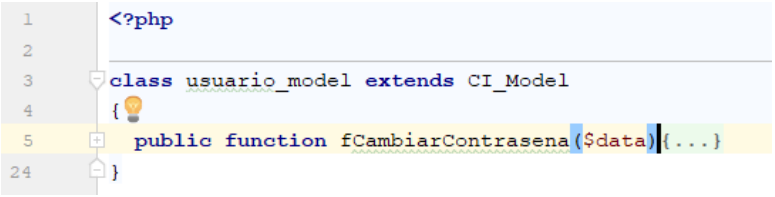

Figura 87 Captura del modelo de usuario en backend. *Fuente: Elaboración propia*

4.1.1.3. Frontend

4.1.1.3.1. Model

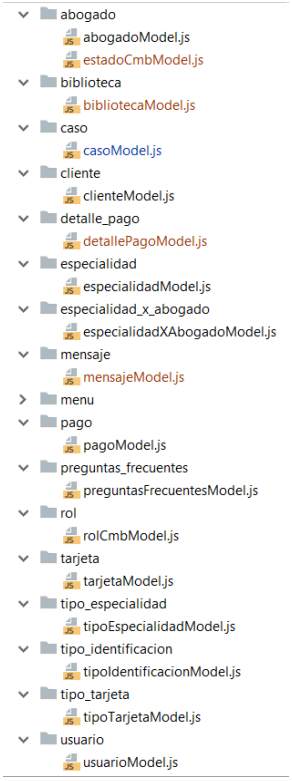

Figura 88 Captura de los archivos de modelo en frontend. *Fuente: Elaboración propia*

4.1.1.3.2. Store

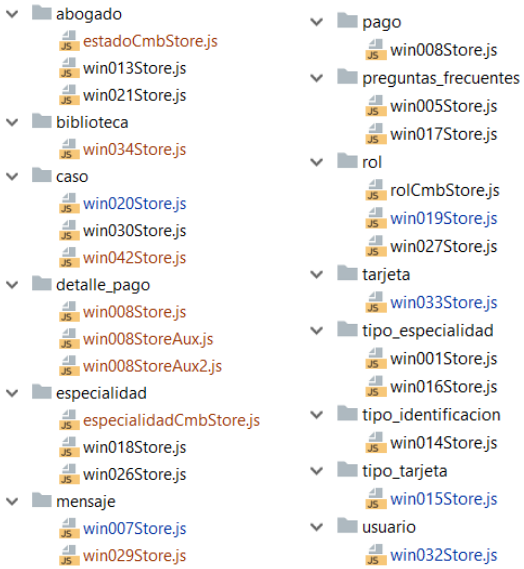

Figura 89 Captura archivos de store en frontend. *Fuente: Elaboración propia*

4.1.1.3.3. View

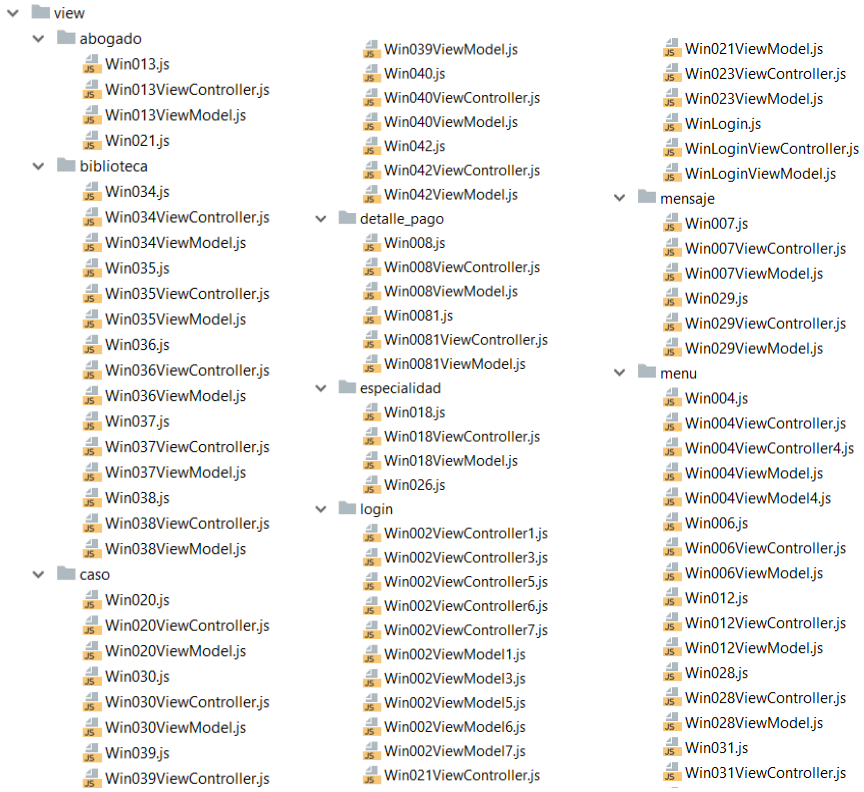

Figura 90 Vistas del frontend parte 1. *Fuente: Elaboración propia*

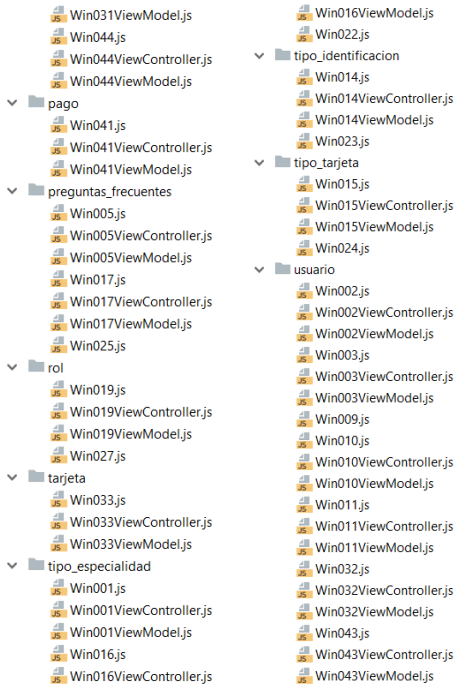

Figura 91 Vistas del frontend parte 2. *Fuente: Elaboración propia*

4.1.2. Pruebas de aceptación

Tabla 21

Pruebas de aceptación iteración 1

Código	Funcionalidad	Criterio de aceptación	Verificación
F1.1	Registrar usuario	En caso de que el usuario desee registrarse en la aplicación, cuando el usuario ingrese la aplicación y haga clic en el botón de registrarse, el sistema desplegará la pantalla con la forma de registro para que el usuario ingrese sus datos y el sistema guarde el usuario.	MEJ
F1.2	Iniciar sesión	En caso de que el usuario desee iniciar sesión en la aplicación, cuando el usuario ingrese sus credenciales y haga clic en el botón de iniciar sesión, el sistema desplegará la pantalla principal de acuerdo con el rol del usuario.	SAT
F1.3	Cerrar sesión	En caso de que el usuario desee cerrar sesión en la aplicación, cuando el usuario haga clic en el botón de cerrar sesión, el sistema desplegará la pantalla inicial.	SAT
F1.4	Restaurar contraseña	En caso de que el usuario desee restaurar su contraseña en caso de haberla olvidado, cuando haga clic en el botón *Olvide mi contraseña,* el sistema desplegará una pantalla para ingresar el correo correspondiente a	INS

		la cuenta, al cual se enviará la contraseña temporal.	
F1.5	Cambiar contraseña	En caso de que el usuario desee cambiar su contraseña, cuando el usuario haga clic en el botón *Cambiar contraseña*, el sistema desplegará la pantalla con el formulario para que el usuario ingrese su contraseña actual y la nueva contraseña, el sistema cambiará la contraseña del usuario al dar clic en cambiar.	SAT
F1.6	Editar usuario	En caso de que el usuario desee cambiar sus datos, cuando el usuario haga clic en el botón *Editar perfil*, el sistema desplegará la pantalla con el formulario para que el usuario edite sus datos, el sistema modificará sus datos al dar clic en guardar.	SAT
F2.1	Ingresar tipo de identificación	En caso de que el administrador desee ingresar un tipo de identificación, cuando el administrador haga clic en el botón de añadir tipo de identificación, el sistema desplegará la pantalla con el formulario de tipo de identificación para que el administrador ingrese los datos y el sistema guarde el tipo de identificación al dar clic en guardar.	SAT
F2.2	Editar tipo de identificación	En caso de que el administrador desee editar un tipo de identificación,	SAT

		cuando el administrador deslice hacia la izquierda el tipo de identificación que desee editar, el sistema desplegará la pantalla con el formulario de ese tipo de identificación para que el administrador modifique los datos y el sistema guarde el tipo de identificación al dar clic en guardar.	
F2.3	Eliminar tipo de identificación	En caso de que el administrador desee eliminar un tipo de identificación, cuando el administrador deslice hacia la derecha el tipo de identificación que desee eliminar, el sistema eliminará dicho registro.	SAT
F2.4	Consultar tipo de identificación	En caso de que el administrador desee visualizar los tipos de identificación, cuando el administrador ingrese en la opción de tipo de identificación, el sistema mostrará la pantalla con la lista de los tipos de identificación almacenados en el sistema.	SAT
F3.1	Ingresar tipo de tarjeta	En caso de que el administrador desee ingresar un tipo de tarjeta, cuando el administrador haga clic en el botón de añadir tipo de tarjeta, el sistema desplegará la pantalla con el formulario de tipo de tarjeta para que el administrador ingrese los datos y el	SAT

		sistema guarde el tipo de tarjeta al dar clic en guardar.	
F3.2	Editar tipo de tarjeta	En caso de que el administrador desee editar un tipo de tarjeta, cuando el administrador deslice hacia la izquierda el tipo de tarjeta que desee editar, el sistema desplegará la pantalla con el formulario de ese tipo de tarjeta para que el administrador modifique los datos y el sistema guarde el tipo de tarjeta al dar clic en guardar.	SAT
F3.3	Eliminar tipo de tarjeta	En caso de que el administrador desee eliminar un tipo de tarjeta, cuando el administrador deslice hacia la derecha el tipo de tarjeta que desee eliminar, el sistema eliminará dicho registro.	SAT
F3.4	Consultar tipo de tarjeta	En caso de que el administrador desee visualizar los tipos de tarjeta, cuando el administrador ingrese en la opción de tipo de tarjeta, el sistema mostrará la pantalla con la lista de los tipos de identificación almacenados en el sistema.	SAT
F4.1	Ingresar tipo de especialidad	En caso de que el administrador desee ingresar un tipo de especialidad, cuando el administrador haga clic en el botón de añadir tipo de especialidad, el sistema desplegará la pantalla con el	SAT

		formulario de tipo de especialidad para que el administrador ingrese los datos y el sistema guarde el tipo de especialidad al dar clic en guardar.	
F4.2	Editar tipo de especialidad	En caso de que el administrador desee editar un tipo de especialidad, cuando el administrador deslice hacia la izquierda el tipo de especialidad que desee editar, el sistema desplegará la pantalla con el formulario de ese tipo de especialidad para que el administrador modifique los datos y el sistema guarde el tipo de especialidad al dar clic en guardar.	SAT
F4.3	Eliminar tipo de especialidad	En caso de que el administrador desee eliminar un tipo de especialidad, cuando el administrador deslice hacia la derecha el tipo de especialidad que desee eliminar, el sistema eliminará dicho registro.	SAT
F4.4	Consultar tipo de especialidad	En caso de que el administrador desee visualizar los tipos de especialidad, cuando el administrador ingrese en la opción de tipo de especialidad, el sistema mostrará la pantalla con la lista de los tipos de identificación almacenados en el sistema.	SAT

F5.1	Ingresar pregunta frecuente	En caso de que el administrador desee ingresar una pregunta frecuente, cuando el administrador haga clic en el botón de añadir pregunta frecuente, el sistema desplegará la pantalla con el formulario de pregunta frecuente para que el administrador ingrese los datos y el sistema guarde la pregunta frecuente al dar clic en guardar.	SAT
F5.2	Editar pregunta frecuente	En caso de que el administrador desee editar una pregunta frecuente cuando el administrador deslice hacia la izquierda la pregunta frecuente que desee editar, el sistema desplegará la pantalla con el formulario de esa pregunta frecuente para que el administrador modifique los datos y el sistema guarde la pregunta frecuente al dar clic en guardar.	SAT
F5.3	Eliminar pregunta frecuente	En caso de que el administrador desee eliminar una pregunta frecuente, cuando el administrador deslice hacia la derecha la pregunta frecuente que desee eliminar, el sistema eliminará dicho registro.	SAT
F5.4	Consultar preguntas frecuentes	En caso de que el administrador desee visualizar las preguntas frecuentes, cuando el administrador ingrese en la opción de preguntas frecuentes, el sistema mostrará la	SAT

		pantalla con la lista de las preguntas frecuentes almacenadas en el sistema.	
F6.1	Ingresar especialidad	En caso de que el administrador desee ingresar una especialidad, cuando el administrador haga clic en el botón de añadir especialidad, el sistema desplegará la pantalla con el formulario de especialidad para que el administrador ingrese los datos y el sistema guarde la especialidad al dar clic en guardar.	SAT
F6.2	Editar especialidad	En caso de que el administrador desee editar una especialidad cuando el administrador deslice hacia la izquierda la especialidad que desee editar, el sistema desplegará la pantalla con el formulario de esa especialidad para que el administrador modifique los datos y el sistema guarde la especialidad al dar clic en guardar.	SAT
F6.3	Eliminar especialidad	En caso de que el administrador desee eliminar una especialidad, cuando el administrador deslice hacia la derecha la especialidad que desee eliminar, el sistema eliminará dicho registro.	SAT
F6.4	Consultar especialidades	En caso de que el administrador desee visualizar las especialidades, cuando el administrador ingrese en la	SAT

		opción de especialidades, el sistema mostrará la pantalla con la lista de las especialidades almacenadas en el sistema.	
F7.1	Ingresar biblioteca	En caso de que el administrador desee ingresar una fuente a la biblioteca, cuando el administrador haga clic en el botón de añadir biblioteca, el sistema desplegará la pantalla con el formulario de biblioteca para que el administrador ingrese los datos y el sistema guarde la fuente al dar clic en guardar.	SAT
F7.2	Editar biblioteca	En caso de que el administrador desee editar una fuente de la biblioteca, cuando el administrador deslice hacia la izquierda la biblioteca que desee editar, el sistema desplegará la pantalla con el formulario de esa biblioteca para que el administrador modifique los datos y el sistema guarde la biblioteca al dar clic en guardar.	SAT
F7.3	Eliminar biblioteca	En caso de que el administrador desee eliminar una fuente de la biblioteca, cuando el administrador deslice hacia la derecha la biblioteca que desee eliminar, el sistema eliminará dicho registro.	SAT
F7.4	Consultar biblioteca	En caso de que el administrador o el abogado desee visualizar una	SAT

		biblioteca, cuando el usuario ingrese en la opción de biblioteca, el sistema mostrará la pantalla con la lista de bibliotecas almacenados en el sistema, al hacer clic en un registro el sistema mostrará la fuente.	
F7.5	Consultar biblioteca por parámetros	En caso de que el administrador o el abogado desee visualizar una biblioteca, cuando el usuario ingrese en la opción de biblioteca, el sistema mostrará la pantalla con la lista de bibliotecas almacenados en el sistema y el usuario puede escribir en la caja de texto, el nombre de la fuente para filtrar la búsqueda y al hacer clic en un registro el sistema mostrará la fuente.	SAT
F8.1	Asignar rol a usuario	En caso de que el administrador desee asignar un rol, cuando el administrador deslice hacia la izquierda el nombre de usuario listado, el sistema desplegará la pantalla con el formulario con el nombre de usuario y el rol que le corresponde, para que el administrador pueda asignar un rol diferente a ese usuario.	SAT
F8.2	Consultar roles	En caso de que el administrador desee visualizar los usuarios del sistema con sus roles, cuando el administrador ingrese en la opción de	SAT

		roles, el sistema mostrará la pantalla con la lista de los usuarios con su rol almacenados en el sistema.	
F8.3	Consultar roles por nombre de usuario	En caso de que el administrador desee visualizar los usuarios del sistema con sus roles, cuando el administrador ingrese en la opción de roles, el sistema mostrará la pantalla con la lista de los usuarios con su rol almacenados en el sistema y el administrador puede ingresar en la caja de texto el nombre del usuario para filtrar la búsqueda.	SAT
F9.1	Ingresar tarjeta	En caso de que el cliente desee ingresar su tarjeta, cuando el cliente haga clic en el botón de datos tarjeta, el sistema desplegará la pantalla con el formulario de tarjeta para que el cliente ingrese sus datos y el sistema guarde la tarjeta al dar clic en guardar.	SAT
F9.2	Editar tarjeta	En caso de que el cliente desee editar su tarjeta cuando el cliente haga clic en el botón de datos tarjeta, el sistema desplegará la pantalla con el formulario de esa tarjeta para que el cliente modifique los datos y el sistema guarde la tarjeta al dar clic en guardar.	SAT
F9.3	Eliminar tarjeta	En caso de que el cliente desee eliminar una tarjeta, cuando el cliente	INS

		haga clic en el botón de datos tarjeta, el sistema desplegará la pantalla con el formulario de esa tarjeta, el cliente hará clic en el botón eliminar datos y el sistema eliminará dicha tarjeta.	
F9.4	Consultar tarjetas	En caso de que el cliente desee visualizar las tarjetas, cuando el cliente haga clic en el botón de datos tarjeta, el sistema desplegará la pantalla con el formulario de esa tarjeta.	SAT
F10.1	Editar abogado	En caso de que el administrador desee editar un abogado cuando el administrador deslice hacia la izquierda el abogado que desee editar, el sistema desplegará la pantalla con el formulario de ese abogado para que el administrador modifique los datos y el sistema guarde el abogado al dar clic en guardar.	SAT
F10.2	Consultar abogados	En caso de que el administrador desee visualizar los abogados, cuando el administrador ingrese en la opción de abogados, el sistema mostrará la pantalla con la lista de los abogados almacenados en el sistema.	SAT
F10.3	Consultar abogados por filtro	En caso de que el administrador desee visualizar los abogados, cuando el administrador ingrese en la opción de abogados, el sistema	SAT

		mostrará la pantalla con la lista de los abogados almacenados en el sistema y el administrador puede ingresar en la caja de texto el nombre del abogado para filtrar la búsqueda.	
F11.1	Ingresar caso	En caso de que el abogado desee ingresar un caso, cuando el abogado deslice a la izquierda el ítem de la lista de mensajes que desee crear el caso, el sistema desplegará la pantalla con el formulario de casos para que el abogado ingrese los datos y el sistema guarde el caso al dar clic en guardar.	SAT
F11.2	Editar caso	En caso de que el abogado desee editar un caso, cuando el abogado deslice a la izquierda el ítem de la lista de mensajes que desee editar el caso, el sistema desplegará la pantalla con el formulario de ese caso para que el abogado edite los datos y el sistema guarde el caso al dar clic en guardar.	SAT
F11.3	Consultar caso con rol de abogado	En caso de que el abogado desee visualizar los casos, cuando el abogado ingrese en la opción de casos, el sistema mostrará la pantalla con la lista de sus casos almacenados en el sistema.	SAT

F11.4	Consultar caso con rol de abogado por parámetros	En caso de que el abogado desee visualizar los casos, cuando el abogado ingrese en la opción de casos, el sistema mostrará la pantalla con la lista de sus casos almacenados en el sistema y el abogado puede escribir en la caja de texto, el nombre del cliente, nombre del caso o número de caso para filtrar la búsqueda.	SAT
F11.5	Consultar caso con rol de administrador	En caso de que el administrador desee visualizar los casos, cuando el administrador ingrese en la opción de casos, el sistema mostrará la pantalla con la lista de los casos almacenados en el sistema.	SAT
F11.6	Consultar caso con rol de administrador por parámetros	En caso de que el administrador desee visualizar los casos, cuando el administrador ingrese en la opción de casos, el sistema mostrará la pantalla con la lista de sus casos almacenados en el sistema y el administrador puede escribir en la caja de texto, el nombre del cliente, nombre del caso o número de caso para filtrar la búsqueda.	SAT
F12.1	Habilitar pago	En caso de que el abogado desee habilitar el pago, cuando al abogado deslice a la derecha el ítem de la lista de casos que desee habilitar el pago, el sistema desplegará la pantalla con	SAT

		el formulario de pago para que el abogado ingrese el valor y el porcentaje para el pago, el sistema guarda el pago al dar clic en guardar.	
F12.2	Realizar pago	En caso de que el cliente desee realizar el pago, cuando el cliente hace clic en realizar pago, se despliega una pantalla con el detalle del pago y se realiza el pago al hacer clic en el botón Realiza pago.	PAR
F13.1	Enviar mensaje	En caso de que el actor desee enviar un mensaje, cuando el actor ingresa en la pantalla de mensajes, puede escribir en la caja de texto y dar clic en el botón de enviar, el sistema enviará el mensaje y se mostrará en pantalla.	SAT
F13.2	Mostrar mensajes	En caso de que el actor desee ver los mensajes, cuando el actor ingresa en la pantalla de mensajes, Se despliega los mensajes correspondientes a su caso y a su etapa en el proceso.	SAT

Fuente: Elaboración propia

4.1.3. Lista de defectos

- Para registrar un usuario, no se verifica que se ingrese un correo válido.
- Para registrar un usuario, no se verifica que se repita la contraseña.
- Para restaurar la contraseña, no se envía el correo con la contraseña temporal.
- Para eliminar la tarjeta, no se muestra el botón de eliminación.
- Para realizar el pago, no se visualizan los datos correctamente en la tabla.

4.2. Fase de estabilización

La presente fase es necesaria para garantizar la calidad de la integración de los módulos y funcionalidades. Para la aplicación LEGALHELP, fue necesario únicamente el desarrollo de un solo autor, por lo que la integración no es requerida.
Si existiesen cambios se debería actualizar el documento del producto. En el caso actual, no hubo cambios por lo que se mantiene lo anterior.

4.3. Fase de pruebas

En esta última fase se lleva a cabo las pruebas faltantes, se documentan los defectos y se corrigen los errores para obtener un producto con calidad.

4.3.1. Documentación de defectos

En la fase de producto se realizó la prueba de aceptación de la iteración 1, de la cual se pudo recopilar una lista de defectos, que fueron corregidos y se realizó una nueva prueba de aceptación correspondiente a la iteración 2, en la cual se mitigaron los defectos en el sistema. A continuación, se detallan los resultados de las pruebas de aceptación:

4.3.1.1. Resultados de pruebas de aceptación

Tabla 22

Resultados de las pruebas de aceptación de iteración 1

	Número de pruebas	Porcentaje
Pruebas satisfactorias	49	92.45%
Pruebas mejorables	1	1.89%
Pruebas parcialmente satisfactorias	1	1.89%
Pruebas insatisfactorias	2	3.77%
Total	53	100%
Pruebas corregidas	0	0%

Fuente: Elaboración propia

Tabla 23

Resultados de las pruebas de aceptación de iteración 2

	Número de pruebas	Porcentaje
Pruebas satisfactorias	53	100%
Pruebas mejorables	0	0.00%
Pruebas parcialmente satisfactorias	0	0.00%
Pruebas insatisfactorias	0	0.00%
Total	53	100%
Pruebas corregidas	4	100%

Fuente: Elaboración propia

4.3.2. Pruebas del sistema

4.3.2.1. Pruebas en SonarCloud

4.3.2.1.1. Primera prueba del backend en sonarcloud.io

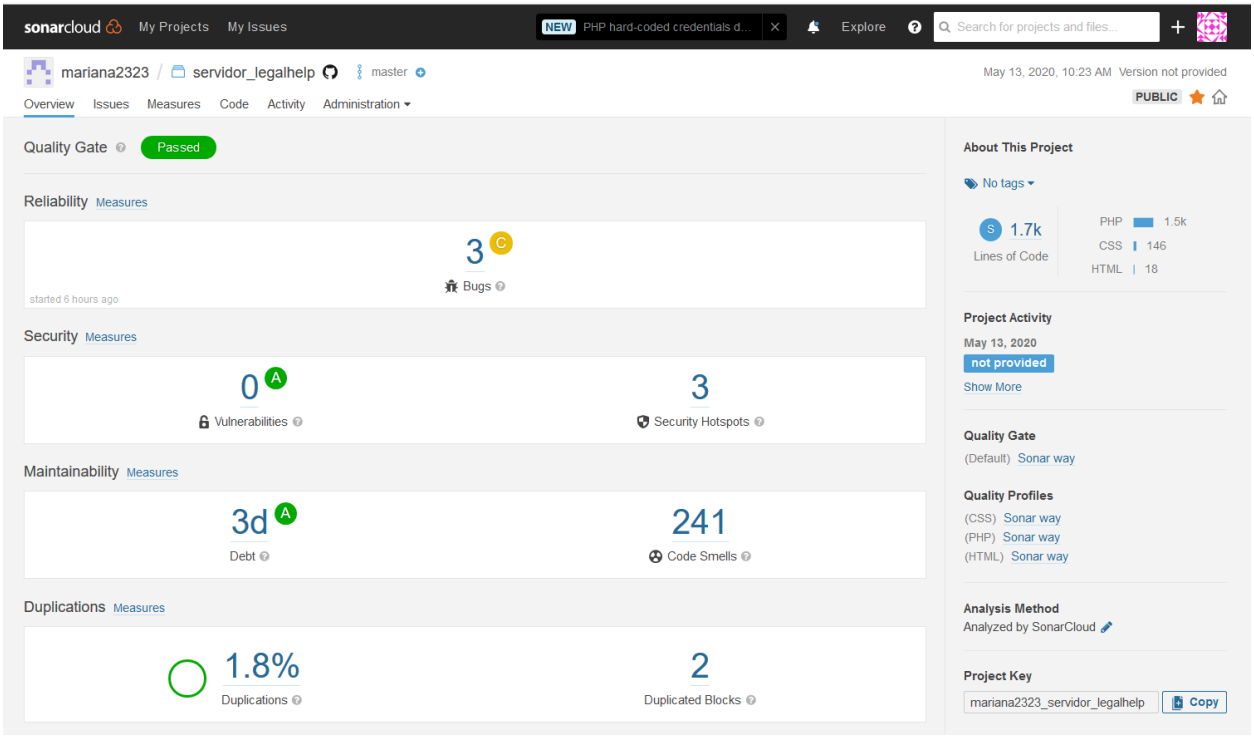

Figura 92 Tablero en SonarCloud del backend (primera prueba). *Fuente: Elaboración propia*

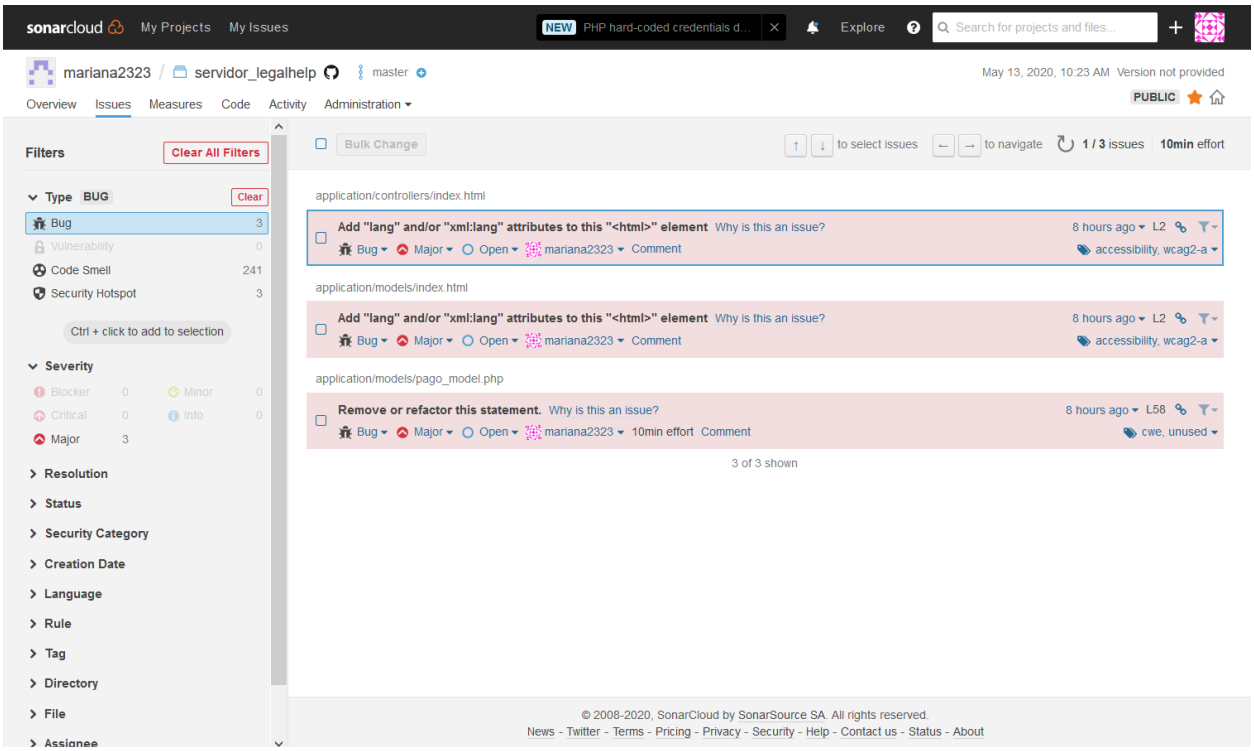

Figura 93 Errores en SonarCloud del backend (primera prueba). *Fuente: Elaboración propia*

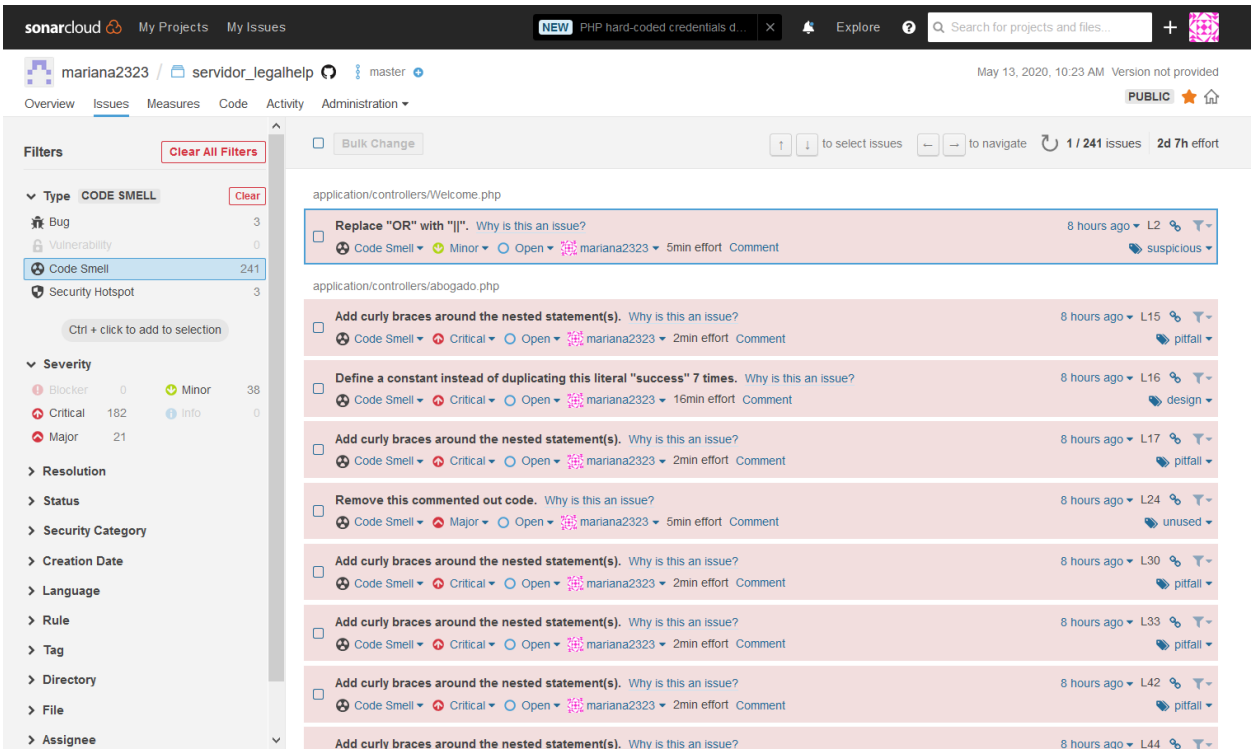

Figura 94 Defectos en SonarCloud del backend (primera prueba). *Fuente: Elaboración propia*

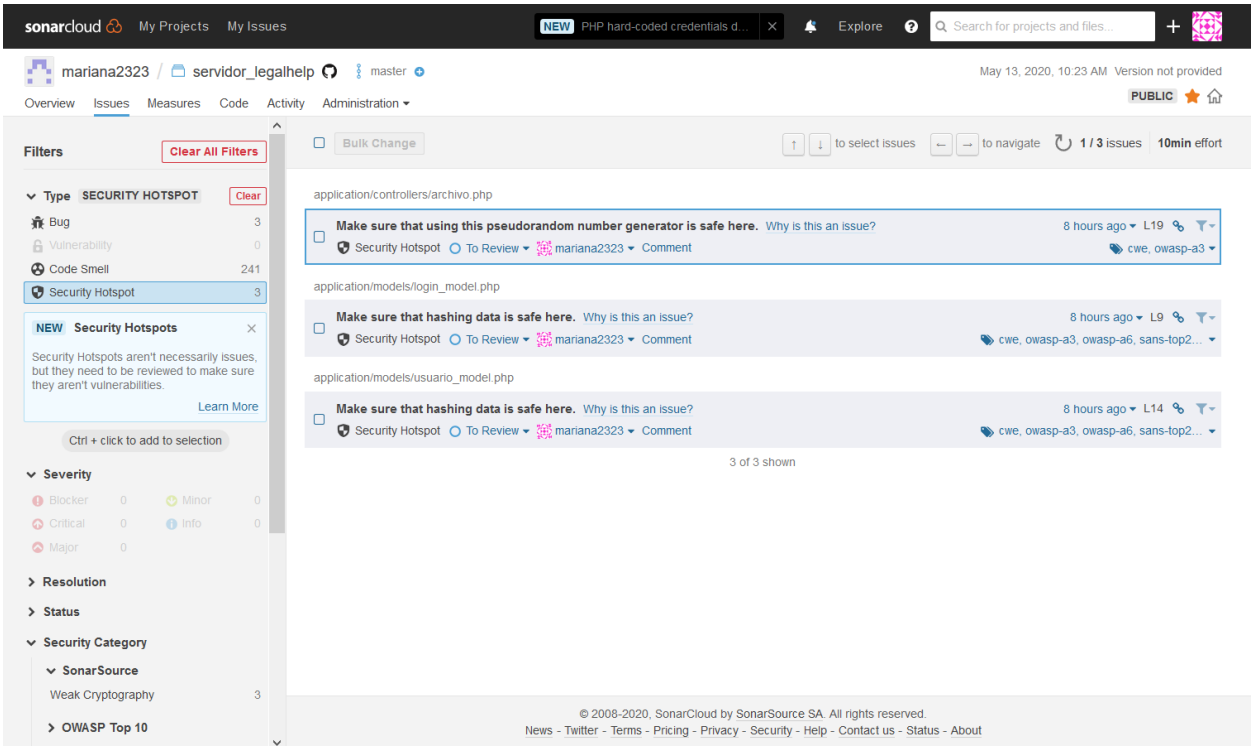

Figura 95 Fallas de seguridad en SonarCloud del backend (primera prueba). *Fuente: Elaboración propia*

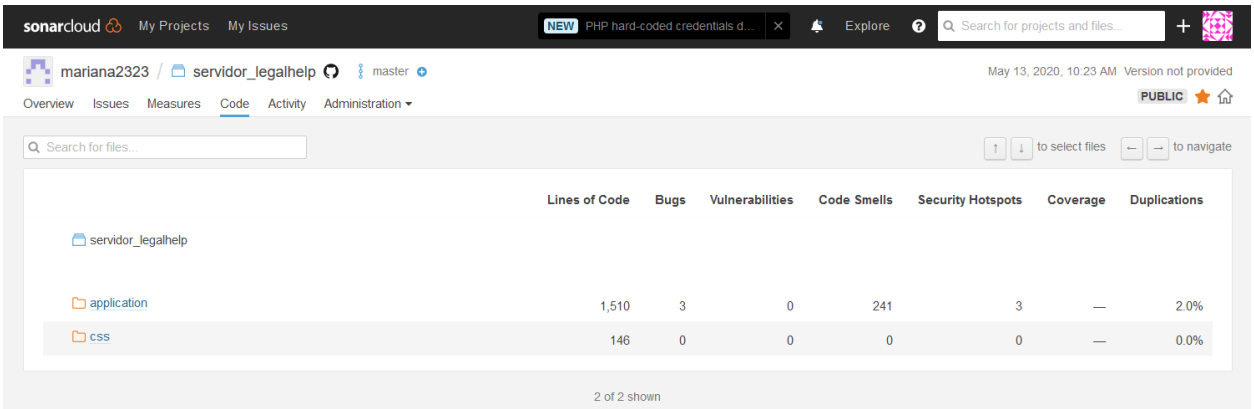

Figura 96 Líneas de código del backend en SonarCloud (primera prueba). *Fuente: Elaboración propia*

4.3.2.1.2. Primera prueba del frontend en sonarcloud.io

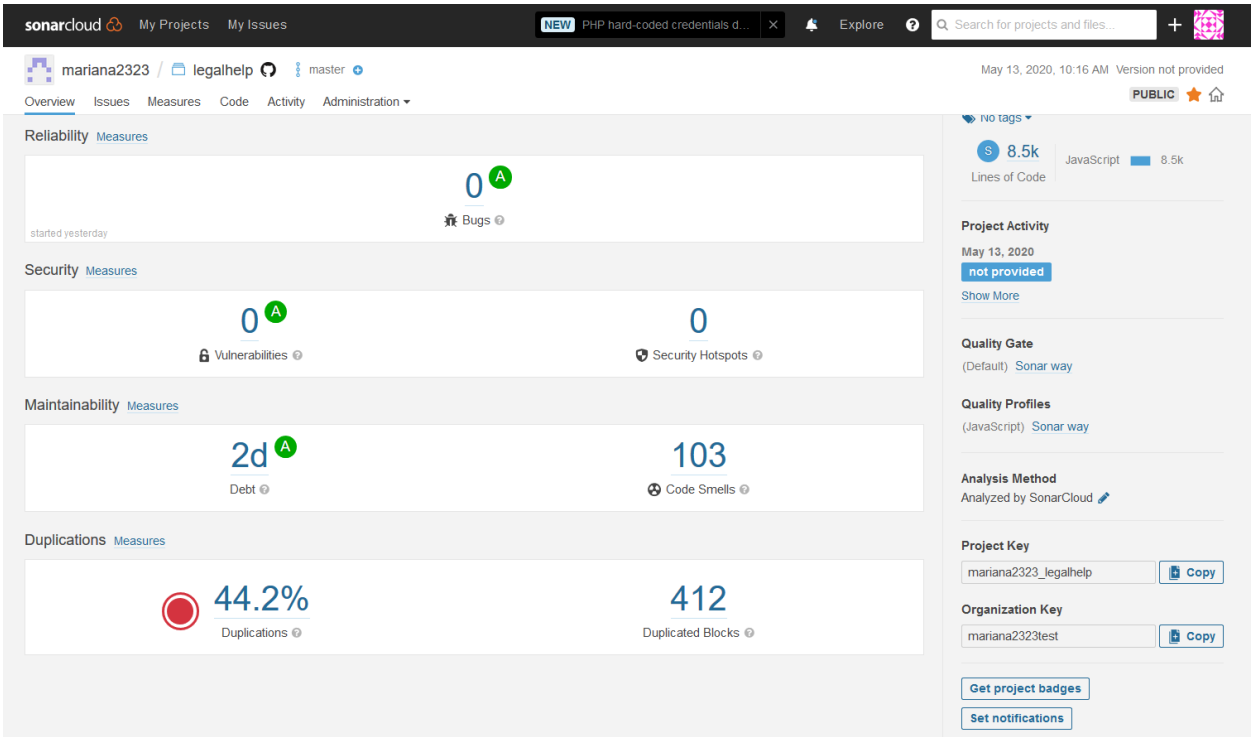

Figura 97 Tablero en SonarCloud del frontend (primera prueba). *Fuente: Elaboración propia*

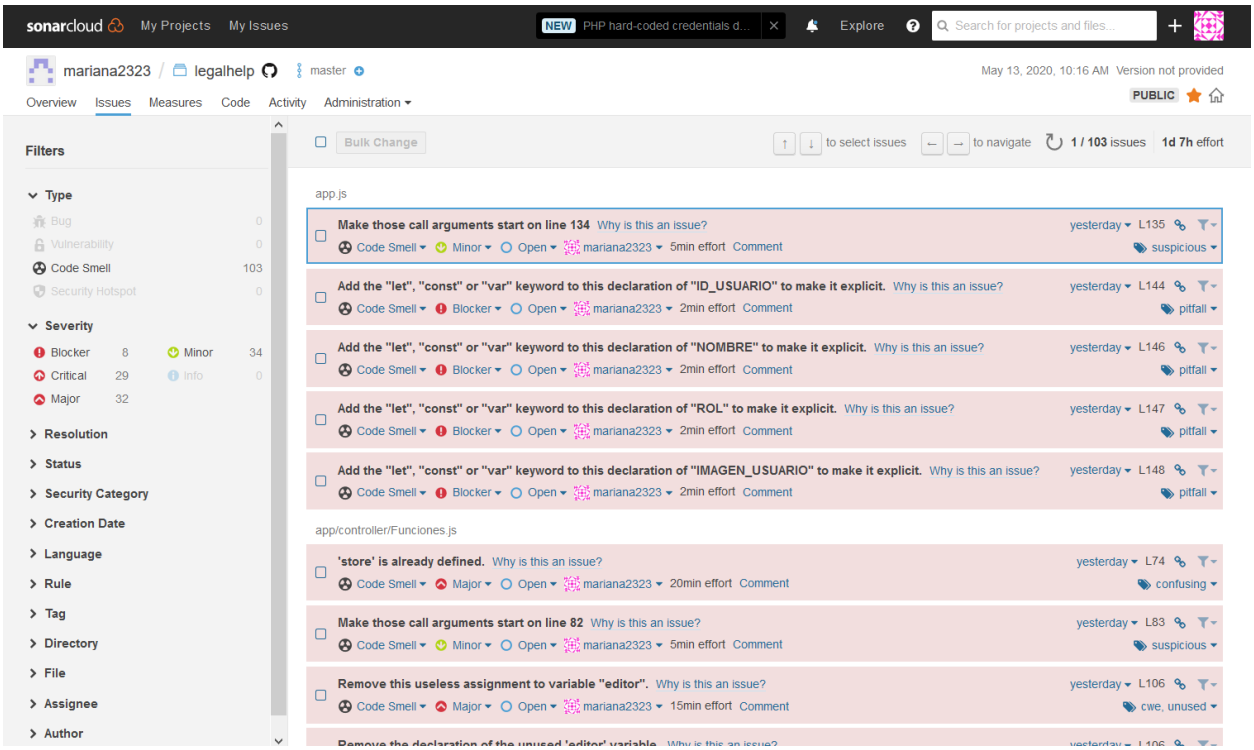

Figura 98 Defectos en SonarCloud del frontend (primera prueba). *Fuente: Elaboración propia*

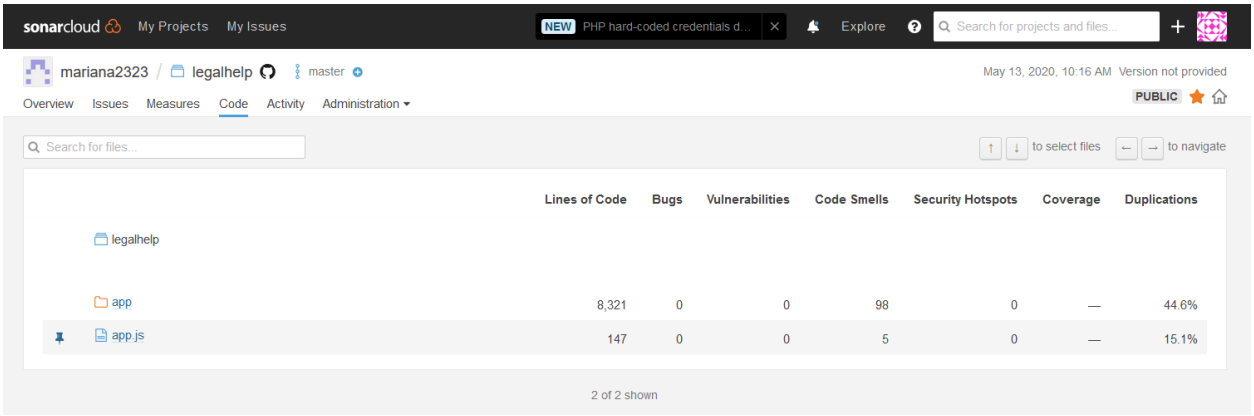

Figura 99 Líneas de código del frontend en SonarCloud (primera prueba). *Fuente: Elaboración propia*

4.3.2.1.3. Segunda prueba del backend en sonarcloud.io

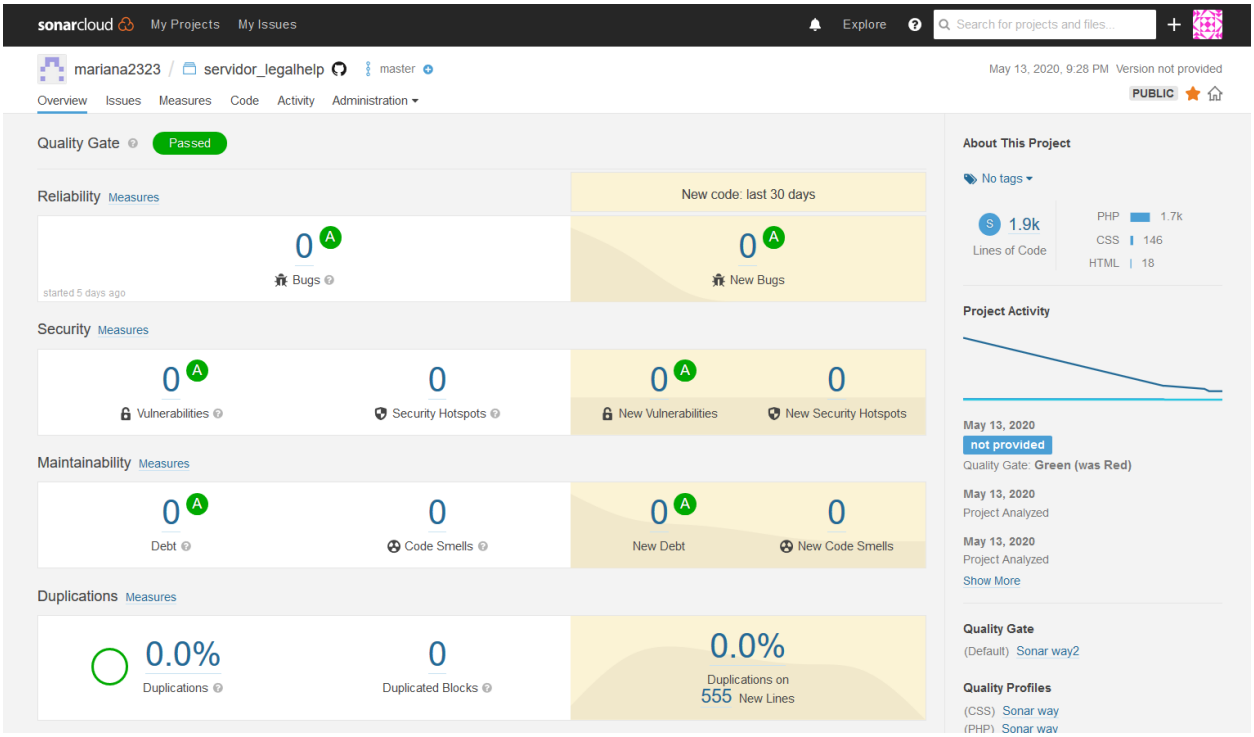

Figura 100 Tablero en SonarCloud de backend (segunda prueba). *Fuente: Elaboración propia*

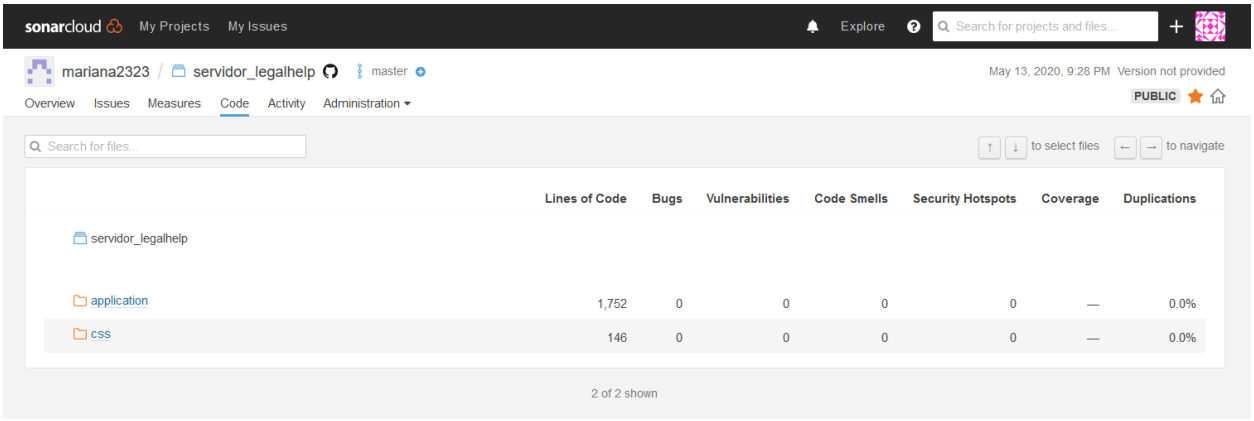

Figura 101 Líneas de código de backend en SonarCloud (segunda prueba). *Fuente: Elaboración propia*

4.3.2.1.4. Segunda prueba del frontend en sonarcloud.io

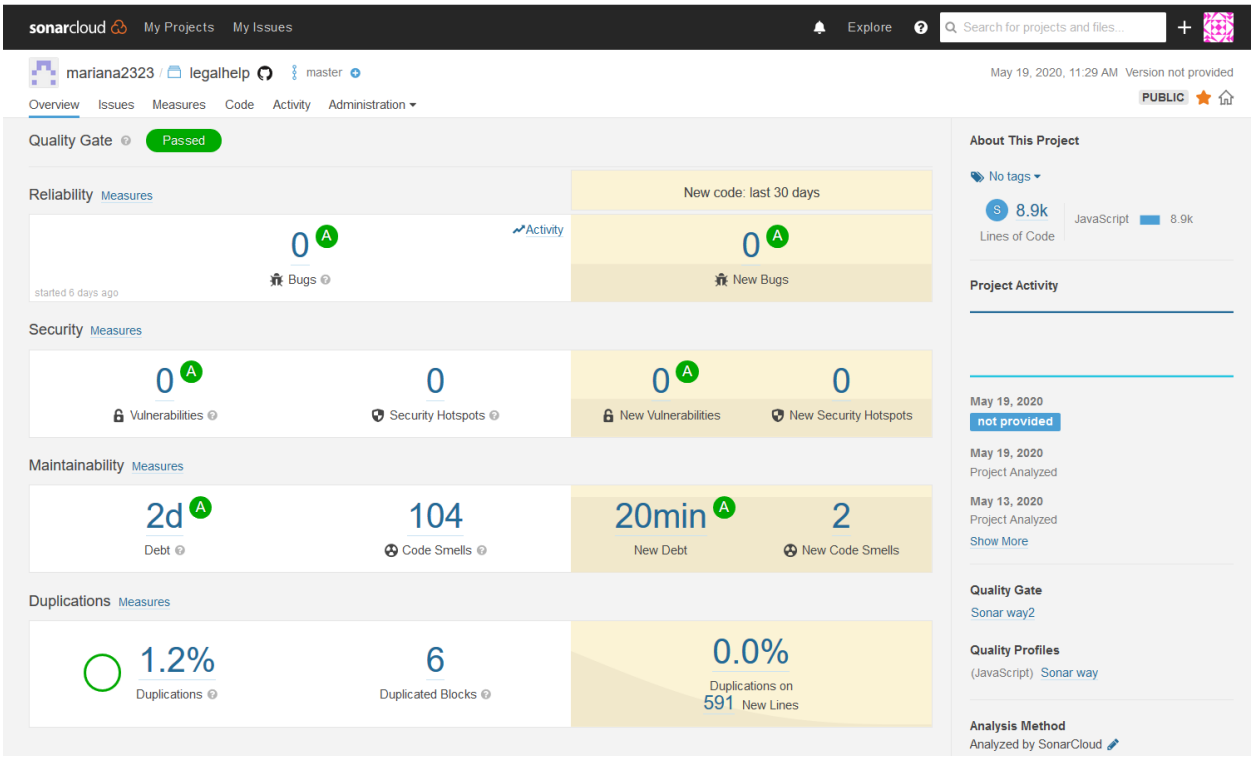

Figura 102 Tablero de SonarCload del frontend (segunda prueba). *Fuente: Elaboración propia*

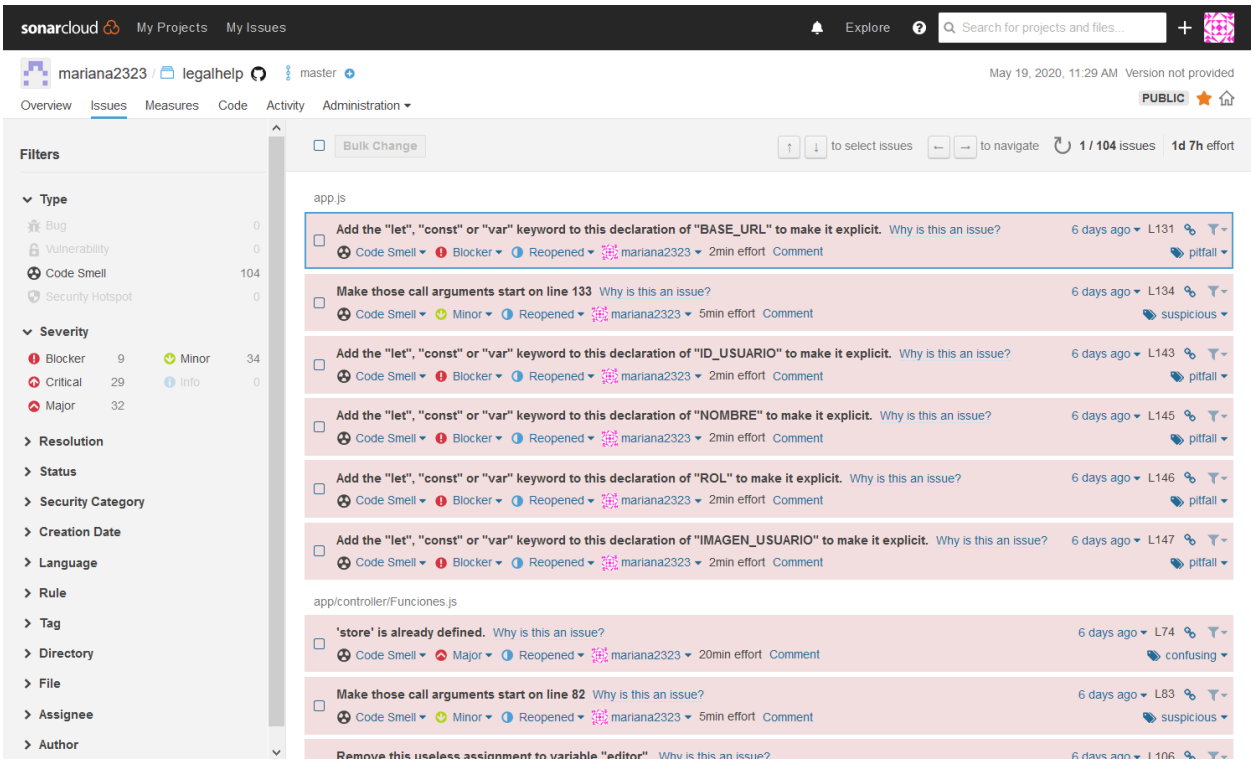

Figura 103 Defectos en SonarCloud del frontend (segunda prueba). *Fuente: Elaboración propia*

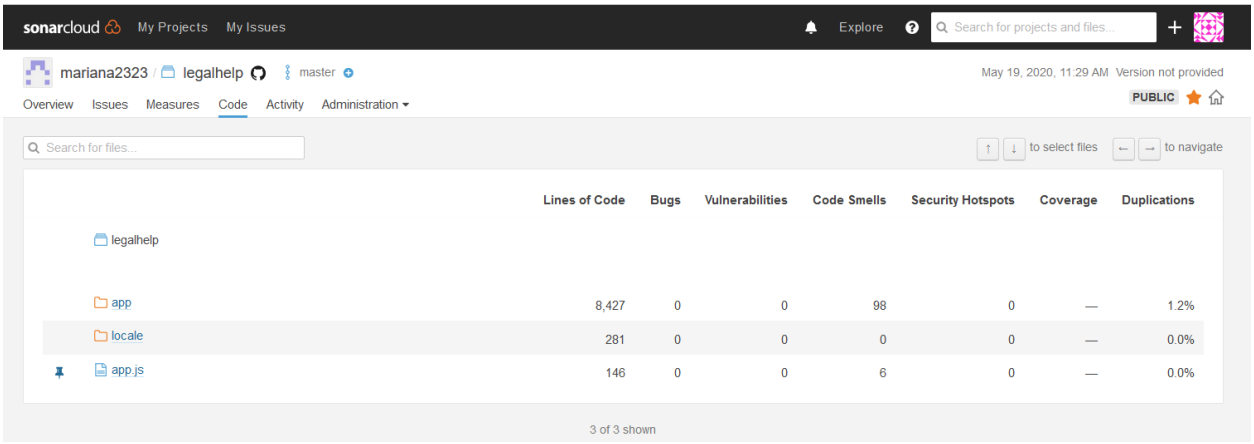

Figura 104 Líneas de código de frontend en SonarCloud (segunda prueba). *Fuente: Elaboración propia*

4.3.2.1.5. Tercera prueba del frontend en sonarcloud.io

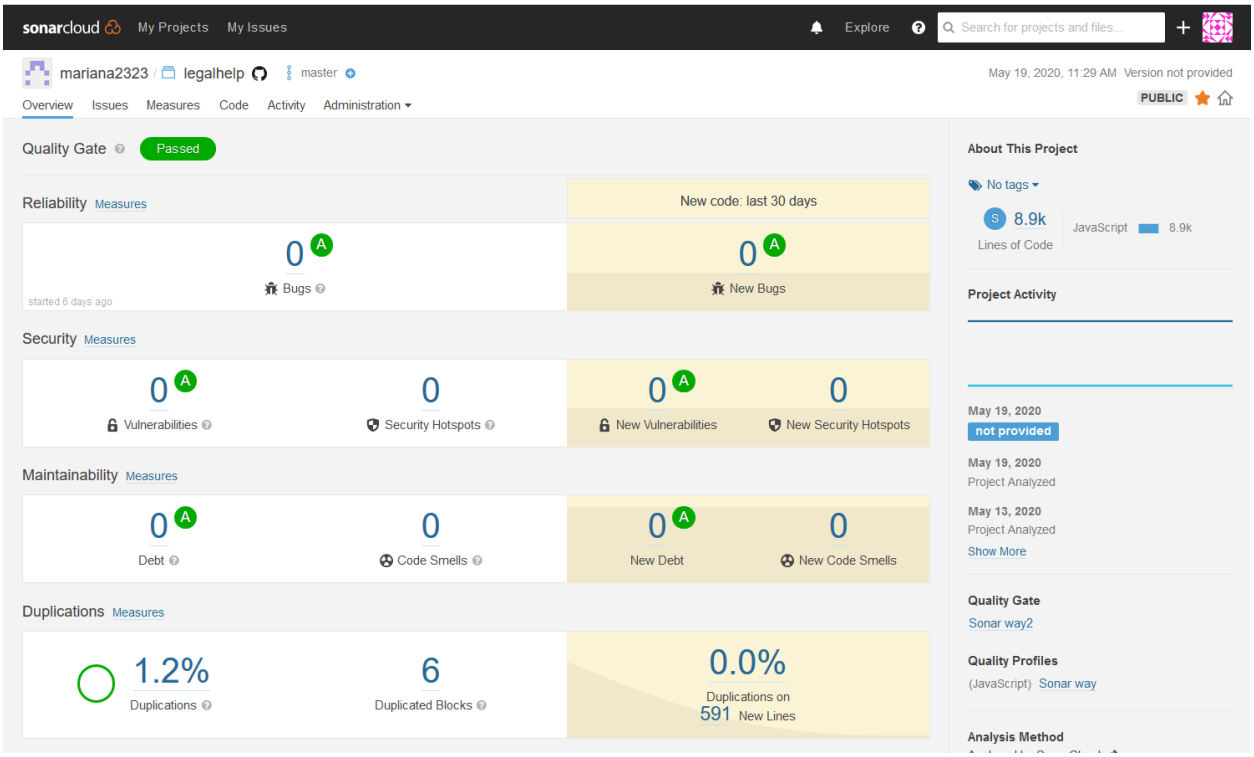

Figura 105 Tablero en SonarCloud de frontend (tercera prueba). *Fuente: Elaboración propia*

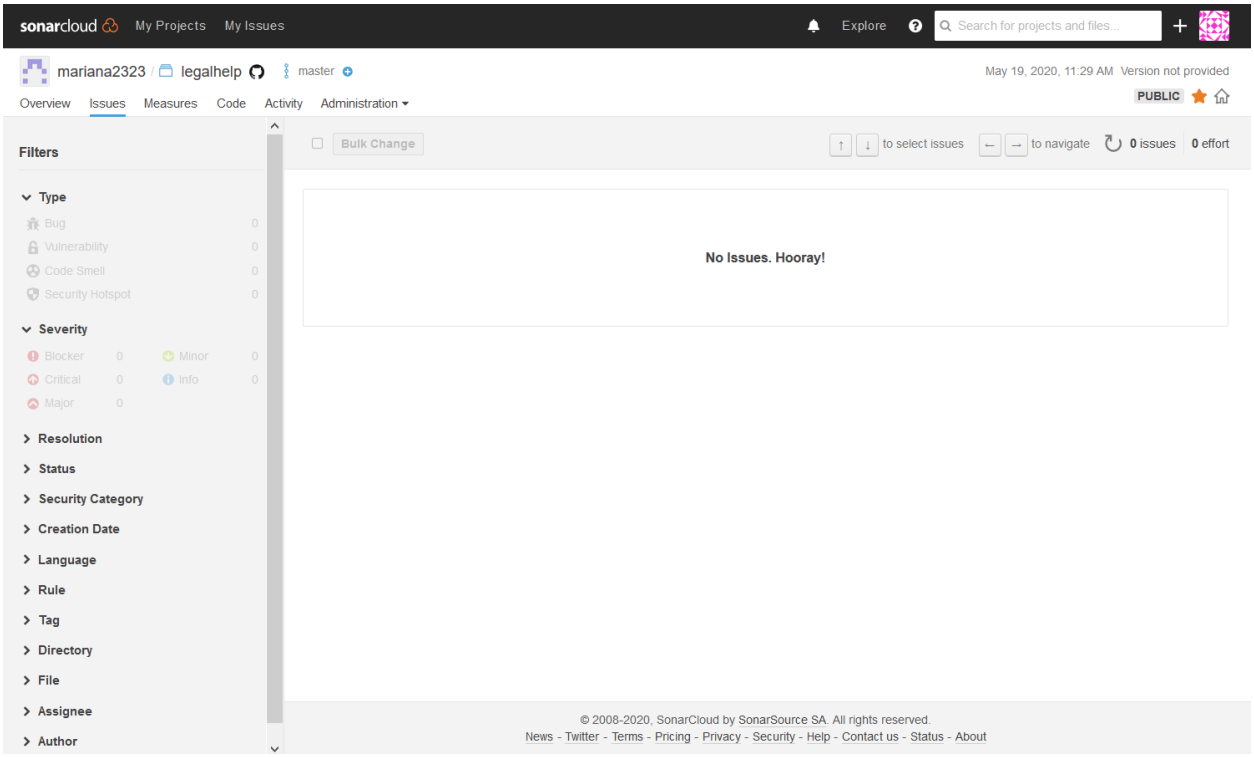

Figura 106 Defectos en SonarCloud de frontend (tercera prueba). *Fuente: Elaboración propia*

Capítulo 5.

5.1. Conclusiones

- Se pudo llegar a identificar los requerimientos de los clientes frente al servicio de consultoría legal en línea, a través de un análisis exhaustivo. Gracias a esto se cumplieron con las funcionalidades descritas y solicitadas por el caso de estudio para la aplicación móvil.
- Se desarrolló una aplicación móvil que permite establecer una comunicación entre el cliente y el abogado consultor, facilitando la prestación del servicio legal proporcionado por el caso de estudio.
- Se concluye que, gracias a la capacidad ágil e incremental de la metodología Mobile-D para el desarrollo de aplicaciones móviles con su estructura cíclica, se pudo desarrollar el sistema con tiempos de entrega reducidos, fácil mantenibilidad y sin inconvenientes con el equipo de desarrollo de una sola persona.
- Se pudo concluir que las prestaciones que brinda la herramienta Sencha Architect, para construir el frontend de una aplicación, permitieron construir un producto de calidad, sin defectos, tal como se pudo observar en la figura 97, 102 y 105, donde el total de defectos y vulnerabilidades se mantienen en cero en cada iteración, por lo cual se recomienda su uso.

5.2. Recomendaciones

- Se recomienda incorporar la funcionalidad al botón de *'Realiza pago'* con un servicio de pagos en línea para mejorar el flujo de pagos para la consultoría en la aplicación móvil.
- Se recomienda aumentar módulos administrativos para la contabilidad y facturación del estudio jurídico en un sistema web.
- Como se menciona en el capítulo 3, la aplicación LEGALHELP es multiplataforma, pero solo se la compiló para celulares con sistema operativo Android, por lo que se recomienda compilar la aplicación móvil para dispositivos con sistema operativo IOS.
- Se recomienda extender la reportería de modo que el administrador del Estudio Jurídico pueda realizar consultas más complejas de los casos con relación a los abogados y sus clientes.

5.3. Glosario

- **AJAX**: “XML y JavaScritp Asíncronos. Es una técnica de programación que permite generar sitios web interactivos” (González Silva, 2007).
- **Backend**: Es una capa que hace referencia a la capa de controlador.
- **Base de datos relacional SQL**: Es un conjunto de datos que se relacionan entre sí, los cuales se manejan con el lenguaje SQL (lenguaje de consulta estructurada).
- **Base de datos**: Es un conjunto de datos.
- **Codeigniter**: Es un framework para el lenguaje PHP.
- **Compilación**: Es la traducción de un lenguaje de alto nivel a un lenguaje de máquina.
- **Controlador**: Es una capa del paradigma MVC que permite al usuario manipular los datos con el propósito cumplir con una funcionalidad específica.
- **Estudio correlacional**: “Estos estudios tienen como finalidad conocer la relación o grado de asociación que existe entre dos o más conceptos, variables, categorías, etc” (Arcos, 2018).
- **Estudio exploratorio**: “Estos se realizan cuando el objeto es examinar un tema o problema de investigación poco estudiado, del cual se tienen muchas dudas o no se ha abordado antes” (Arcos, 2018).
- **Estudios descriptivos**: “Estos buscan especificar las propiedades, las características y los perfiles de personas, grupos, comunidades, procesos, objetos que se someta a un análisis” (Arcos, 2018).
- **Fiabilidad**: “Digno de confianza” (Lexus Editores, 1997, p. 385).
- **Framework**: Es un marco de trabajo desarrollado para minimizar la carga al programar código que podría ser reutilizado. Proporcionan una estructura predefinida de componentes y demás elementos complejos para la construcción de una aplicación web completa, y se orientan bajo modelos de desarrollo como el paradigma MVC que se mencionó antes.
- **Frontend**: Es una capa que hace referencia a la capa de vista, ya que aquí interactúa el usuario con la aplicación mediante interfaces gráficas.
- **GitHub**:
- **GNU**: “Proyecto que trabaja por el desarrollo de software y SO «libre»” (González Silva, 2007).
- **Hardware**: “Soporte físico de un ordenador” (Lexus Editores, 1997, p. 459).

- **HTTP: “**Protocolo de Transferencia de Hipertexto. Es el protocolo de transferencia de archivos de Hipertexto, para ser mostrados mediante un navegador” (González Silva, 2007).
- **IDE**: Significa entorno de desarrollo integrado, el cual es un software que permite diseñar y desarrollar aplicaciones.
- **Interfaz gráfica**: Es el medio visual en el cual el usuario interactúa con el sistema.
- **Javascript**: es un lenguaje de programación potente de archivos de comandos o script que permite añadir unidades funcionales al sistema para simplificar y automatizar la ejecución de las funciones.
- **Línea base**: Son los pilares que ayudarán a que el proyecto se desarrolle apropiadamente, configurando cada capa del paradigma MVC para comenzar el desarrollo respectivo en cada capa.
- **Login**: Iniciar sesión.
- **Mantenibilidad**: “Acción de reparar y conservar en buen estado maquinaria, instalaciones, etc” (Lexus Editores, 1997, p. 585).
- **MariaDB**: Es un motor de base de datos MySQL.
- **Metodología de desarrollo**: Es un marco de trabajo que proporciona una mejor estructuración, diseño, desarrollo y planificación para la construcción de un sistema de software.
- **Modelo**: Es una capa del paradigma MVC que se encarga de la gestión de datos de la aplicación.
- **Motor de base de datos MySQL**: Es un software de gestión y manejo de datos que se relacionan entre sí basados en el lenguaje SQL.
- **Multiplataforma**: “Que puede ser utilizado por distintos sistemas o entornos” (Real Academia Española, 2014).
- **Paradigma**: “Conjunto de formas que sirven de modelo en las respectivas flexiones” (Lexus Editores, 1997, p. 706).
- **PHP**: (Preprocesador de hipertexto), es un lenguaje de programación que está vinculado con el desarrollo de páginas Web y con el lenguaje HTML (Gutiérrez Gallardo, 2004).
- **PhpStorm**: Es un IDE multiplataforma para el lenguaje PHP.

- **Portabilidad**: Consultar definición de multiplataforma.
- **Reportería**: Es una recopilación de información obtenida de la base de datos.
- **Requerimiento**: Es la descripción de lo que se desea que haga el sistema.
- **Robusto**: Es la capacidad de un sistema de dar una respuesta ante la presencia de errores causados generalmente por una gran cantidad de usuarios conectados a la aplicación.
- **Sencha Architect**: Es un IDE multiplataforma para el framework ExtJs.
- **Servidor Apache**: "Servidor HTTP de licencia GNU, muy popular en el mundo" (González Silva, 2007).
- **Sindical**: "Asociación de trabajadores para la defensa de sus intereses" (Lexus Editores, 1997, p. 855).
- **Software**: "Conjunto de programas internos del ordenador que permiten realizar las funciones asignadas por el usuario" (Lexus Editores, 1997, p. 860).
- **SonarCloud**: Es un servicio en la nube basado en SonarQube (Travis CI, 2020).
- **SonarQube**: Es una plataforma para inspeccionar continuamente la calidad del código fuente, detectando defectos, vulnerabilidades y fallas de seguridad, que soporta lenguajes como PHP, Javascript, Java, C#, C, C++, entre otros más (Travis CI, 2020).
- **SQLyog**: Consultar definición de *Motor de base de datos MySQL*.
- **Tecnologías de Información**: "Son el conjunto de tecnologías desarrolladas en la actualidad para una información y comunicación más eficiente" (Chen, 2019).
- **Vista**: Es una capa del paradigma MVC que se encarga de mostrar los datos de la capa de modelo en una interfaz gráfica.

Bibliografía

Arcos, S. (2018). *Metodología de la Investigación para Unidades de Titulación.*

Avilés Pino, E. (n.d.). *Enciclopedia del Ecuador*. Recuperado de Ministerio de Trabajo y Recursos Humanos: http://www.enciclopediadelecuador.com/historia-del-ecuador/ministerio-trabajo-recursos-humanos/

Bobb, N. (2020). *MVC (Modelo Vista Controlador)* [Figura]. Recuperado de https://nicobobb.com/mvc/

Chen, C. (2019, Mayo 21). *Significado de TIC (Tecnologías de la información y la comunicación)*. Recuperado de Significados: https://www.significados.com/tic/

CLE Formación. (2018, Marzo 27). *Tic-tek: Cle Formación*. Recuperado de ExtJS: Caminando hacia las nuevas aplicaciones: http://www.cleformacion.com/tic-tek/-/blogs/extjs-caminando-hacia-las-nuevas-aplicaciones

Corporación de Estudios y Publicaciones. (2019). *Código del Trabajo.* Quito: Talleres de la Corporación de Estudios y Publicaciones.

Fernández, Y., & Díaz, Y. (2012). Patrón Modelo-Vista-Controlador. *Telem@tica, 11*(1), 47-57.

Gómez Medina, J. S., & Hernández Zuleta, D. F. (2016). *Desarrollo de Aplicaciones para Dispositivos Moviles.* Quindio. Recuperado de https://es.slideshare.net/pipehernandez1020/mobile-d-programacion-dispositivos-moviles

González Silva, J. (2007, Noviembre 20). *Softzone*. Recuperado de Diccionario Informático: https://www.softzone.es/glosario/g-h-e-i/

Gutiérrez Gallardo, J. (2004). *Desarrollo Web con PHP 5 y MySQL.* Madrid: Ediciones Anaya Multimedia.

Gutiérrez, J. J. (2014, mayo 12). *¿Qué es un framework web?* Recuperado de http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf

Humprey, W. (2001). *Introducción al Proceso Software Personal.* Madrid: Pearson Education.

Joshi, K. (2020). *Sencha.* Recuperado de Adopting a Unified Component Library Strategy: www.sencha.com

Lexus Editores. (1997). *Lexus Diccionario Enciclopédico.* Barcelona: Ediciones Trébol, S.L.

Páez Benalcázar, A. (2019). *Principios del Derecho del Trabajo.* Quito: Cevallos.

Pérez Ramírez, G. (2014, septiembre 14). El Código del Trabajo ecuatoriano 1938-2014. *El Telégrafo*. Recuperado de https://www.eltelegrafo.com.ec/noticias/columnistas/1/el-codigo-del-trabajo-ecuatoriano-1938-2014

Pressman, R. (2005). *Ingeniería del Software. Un Enfoque Práctico.* Mexico: McGraw-Hill Interamericana.

R2 Data Technology. (2018). *Sencha Ext JS* [Figura]. Recuperado de https://www.r2datatechnology.com/producto/sencha-ext-js/

Real Academia Española. (2014). *Diccionario de la lengua española*, 23.a ed. Recuperado de https://dle.rae.es/multiplataforma

Reyes Mendoza, L. (2012). *Derecho laboral.* Tlalnepantla: Red Tercer Milenio.

Robles Velasco, L. M. (2016). *Introducción bilingüe al derecho español para estudiantes erasmus. Capítulo 1 "DERECHO ROMANO".* Recuperado de https://digibug.ugr.es/bitstream/handle/10481/43413/man%20biling%c3%bce%20INTROD%20DCHO%20ROMANO.pdf?sequence=1&isAllowed=y

Seravo. (2015). *10 reasons to migrate to MariaDB (if still using MySQL)* [Figura]. Recuperado de https://seravo.fi/2015/10-reasons-to-migrate-to-mariadb-if-still-using-mysql

Sommerville, I. (2011). *Ingeniería de Software.* Naucalpan de Juárez: Pearson.

Travis CI. (2020). *Travis CI*. Recuperado de Using SonarCloud with Travis CI: https://docs.travis-ci.com/user/sonarcloud/

Vergara, M. (2016). *Qué es Codeigniter y cuáles son algunas de sus ventajas* [Figura]. Recuperado de https://www.coriaweb.hosting/codeigniter-cuales-algunas-ventajas/

Villanueva, J. E. (2015). *Aplicación móvil con Realidad Aumentada orientada al Marketing y Publicidad para la empresa Boliviamar SRL.* La Paz: Universidad Mayor de San Andrés.

Yáñez Andrade, J. C. (2000). CHILE Y LA ORGANIZACIÓN INTERNACIONAL DEL TRABAJO (1919-1925). HACIA UNA LEGISLACIÓN SOCIAL UNIVERSAL. *Revista de estudios histórico-jurídicos, 22*, 317-332. Recuperado de https://scielo.conicyt.cl/scielo.php?pid=S0716-54552000002200014&script=sci_arttext