PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

SISTEMAS DE INFORMACIÓN



TRABAJO DE TITULACIÓN

DESARROLLO DE UNA APLICACIÓN WEB PARA EL CONSULTORIO JURÍDICO GRATUITO DE LA PUCE.

AUTOR:

Byron Alexis Vargas Sánchez

QUITO DM, JULIO DE 2025

TABLA DE CONTENIDOS

1. CAPÍTULO 1: INTRODUCCIÓN 1
1.1 Planteamiento del problema 1
1.2 Justificación 2
1.3 Objetivos 3
1.3.1 Objetivo General 3
1.3.2 Objetivos Específicos 3
1.4 Alcance 4
2. CAPÍTULO 2: FUNDAMENTACIÓN TEÓRICA 6
2.1 Aplicación Web 6
2.1.1 Definición y características fundamentales 6
2.1.2 Arquitectura de servicios en sistemas de gestión social 6
2.1.3 Ventajas específicas para sistemas de trabajo social 7
2.2 Metodología de desarrollo de software 7
2.2.1 Prototipado evolutivo en sistemas sociales 7
2.2.2 Gestión de requerimientos en sistemas sociales 8
2.3 Base de Datos 9
2.3.1 Diseño de bases de datos para sistemas sociales 9
2.3.2 MySQL como sistema de gestión 10
2.4 Lenguaje de Programación 10
2.4.1 JavaScript en el desarrollo web moderno 10
2.4.2 TypeScript para desarrollo 11
2.5 Entorno de Desarrollo Integrado (IDE) 11
2.5.1 Visual Studio Code para desarrollo web 11
2.6 Framework de Desarrollo Web 12
2.6.1 Node.js para el backend 12

2.6.2 Vue.js para el frontend 12
2.7 Biblioteca de herramientas 13
2.7.1 Moment.js para manejo de fechas y zonas horarias 13
2.7.2 ExcelJS para generación de reportes 13
2.7.3 DOCX para generación de documentos Word 14
2.7.4 Sequelize como ORM (Object-Relational Mapping) 14
2.7.5 PrimeVue para componentes de interfaz 14
2.7.6 Axios para comunicación HTTP 15
2.8 Control de Versiones 15
2.8.1 Importancia del control de versiones en desarrollo iterativo 15
2.8.2 Git y GitHub para colaboración 15
2.9 Buenas Prácticas de Desarrollo Web 16
2.9.1 Principios de diseño y usabilidad para sistemas sociales 16
2.9.2 Seguridad en aplicaciones web para datos sensibles 16
3. CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL MÓDULO 17
3.1 Análisis de Requerimientos 17
3.1.1 Requisitos Funcionales 17
3.1.2 Requisitos No Funcionales 18
3.2 Modelado del Sistema 19
3.2.1 Diagrama de Casos de Uso 19
3.3 Diseño de la Interfaz del Usuario (UI) 26
3.3.1 Principios de Diseño para Sistemas Sociales 26
3.3.2 Componentes Principales de la Interfaz 26
3.3.3 Diseño Responsivo y Accesibilidad 31
3.3.4 Flujos de Trabajo en la Interfaz 31
4 CAPÍTULO 4: DESARROLLO DEL MÓDULO 34
4.1 Preparación del Entorno de Desarrollo 34

4.1.1 Configuración del Entorno Tecnológico 34
4.1.2 Estándares de Codificación 34
4.2 Implementación de la Funcionalidad 35
4.2.1 Backend - Desarrollo de APIs y Lógica de Negocio 35
4.2.2 Frontend - Desarrollo de Interfaces de Usuario 35
4.2.3 Integración Backend-Frontend 36
4.3 Documentación del Código 36
4.3.1 Documentación de APIs 36
4.3.2 Documentación de Componentes Vue 37
4.3.3 Documentación de Modelos de Datos 39
4.4 Pruebas 41
4.4.1 Entorno de Pruebas 41
4.4.2 Metodología de Pruebas 42
4.4.3 Resultados de las Pruebas de Funcionalidad 42
4.4.4 Pruebas de integración con otros Módulos 43
4.4.5 Pruebas de Usabilidad 44
4.4.6 Pruebas de Rendimiento 44
4.4.7 Retroalimentación de Usuarios 45
4.4.8 Interfaz final del sistema 45
4.4.9 Conclusiones de las Pruebas 52
5. CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES 54
5.1 Conclusiones 54
5.2 Recomendaciones 55
6. Bibliografía: 57
7. Anexos 59

TABLA DE ILUSTRACIONES

Ilustración 1 Fases del Prototipado Evolutivo Aplicadas al Módulo. 8
Ilustración 2 Diagrama de casos de uso: Gestión de agenda. 20
Ilustración 3 Diagrama de casos de uso: Registro de casos. 22
Ilustración 4 Diagrama de casos de uso: Gestión de estados de caso. 24
Ilustración 5 Vista previa de Gestión de Casos (SocialWorkCases.vue) 26
Ilustración 6 Formulario previo de Registro de Casos (SocialWorkNewCase.vue) 1 27
Ilustración 7 Formulario previo de Registro de Casos (SocialWorkNewCase.vue) 2 28
Ilustración 8 Formulario previo de Registro de Casos (SocialWorkNewCase.vue) 3 28
Ilustración 9 Formulario previo de Registro de Casos (SocialWorkNewCase.vue) 4 28
Ilustración 10 Formulario previo de Registro de Casos (SocialWorkNewCase.vue) 5 29
Ilustración 11 Formulario previo de Registro de Casos (SocialWorkNewCase.vue) 6 29
Ilustración 12 Primer diseño de gestión de Agenda (SocialWorkSchedule.vue) 30
Ilustración 13 Diagrama de flujo por bloques del módulo 33
Ilustración 14 Vista de Gestión de Casos (SocialWorkCases.vue) 45
Ilustración 15 Sección 1 Información del Caso. 47
Ilustración 16 Sección 2 Solicitudes. 47
Ilustración 17 Sección 3 Datos del Usuario. 48
Ilustración 18 Sección 4 Composición de grupo de convivencia. 48
Ilustración 19 Sección 5 Miembros del círculo familiar con discapacidad 48
Ilustración 20 Sección 6 Episodios de Violencia y Consumo. 49
Ilustración 21 Sección 7 Situación Económica. 49
Ilustración 22 Sección 8 Datos de la Contraparte. 49
Ilustración 23 Sección 9 y 10, Relato de los Hechos y Observaciones respectivamente. 50
Ilustración 24 Gestión de Agenda (SocialWorkSchedule.vue) 51

ÍNDICE DE TABLAS

Tabla 1 Descripción de caso de uso de gestión de agenda. 21
Tabla 2 Descripción de caso de uso de registro de casos. 23
Tabla 3 Descripción de caso de uso de cambio de estado de caso. 25

1. CAPÍTULO 1: INTRODUCCIÓN

1.1 Planteamiento del problema

El Consultorio Jurídico Gratuito de la Pontificia Universidad Católica del Ecuador enfrenta desafíos significativos en la gestión de casos del área de Trabajo Social. Actualmente, el manejo de expedientes sociales se realiza de manera manual o mediante sistemas fragmentados que no permiten un seguimiento integral de los casos. Esta situación genera varias problemáticas que afectan tanto la eficiencia operativa como la calidad del servicio brindado a las personas en situación de vulnerabilidad.

Las dificultades identificadas incluyen la duplicación de información en diferentes documentos físicos, la falta de un sistema centralizado para el seguimiento de casos, la dificultad para generar reportes estadísticos, y la pérdida de tiempo en procesos administrativos que podrían automatizarse. Además, la naturaleza compleja de los casos de trabajo social requiere un registro detallado de información socioeconómica, familiar y personal que permita a los profesionales realizar evaluaciones integrales y dar seguimiento adecuado a cada situación.

El módulo de Trabajo Social del Consultorio Jurídico debe atender casos que involucran evaluaciones socioeconómicas, seguimiento familiar, coordinación con otras instituciones, y la generación de informes técnicos. Sin embargo, la gestión manual de estos procesos limita la capacidad de los trabajadores sociales para brindar un servicio óptimo y afecta la continuidad del acompañamiento a las familias atendidas.

En función de esta problemática se plantea la siguiente pregunta principal de investigación: ¿Cómo puede el desarrollo de un módulo web especializado en Trabajo Social mejorar la gestión y seguimiento de casos sociales en el Consultorio Jurídico Gratuito de la PUCE?

Las preguntas secundarias que guían esta investigación son: ¿Qué funcionalidades específicas debe incluir el módulo de Trabajo Social para optimizar el registro y seguimiento de casos? ¿Cómo puede la digitalización de los procesos de trabajo social contribuir a mejorar la calidad de atención a los usuarios? ¿Qué herramientas tecnológicas son más apropiadas para desarrollar un sistema que responda a las necesidades específicas del área de Trabajo Social?

1.2 Justificación

El desarrollo del módulo de Trabajo Social para el Consultorio Jurídico Gratuito de la PUCE se justifica desde múltiples perspectivas que convergen en la necesidad de modernizar y optimizar los procesos de atención social. Esta iniciativa responde tanto a imperativos académicos como sociales, considerando que el consultorio jurídico constituye un espacio fundamental para la formación práctica de estudiantes y la prestación de servicios gratuitos a poblaciones vulnerables.

Desde la perspectiva académica, el módulo permitirá a los estudiantes de Trabajo Social contar con herramientas tecnológicas modernas que faciliten su proceso de aprendizaje y práctica profesional. La sistematización de la información les proporcionará una visión más clara y ordenada de los casos, permitiendo análisis más profundos y seguimientos más efectivos. Además, el manejo de un sistema digital preparará a los futuros profesionales para un mercado laboral cada vez más digitalizado.

El impacto social del proyecto es igualmente significativo. Las personas que acuden al Consultorio Jurídico Gratuito generalmente se encuentran en situaciones de vulnerabilidad socioeconómica y requieren atención integral y especializada. Un sistema eficiente de gestión de casos garantiza que ningún usuario pierda continuidad en su atención, que se realice un seguimiento adecuado de su situación, y que se generen los reportes necesarios para evaluar el impacto de las intervenciones realizadas.

Desde el punto de vista operativo, la automatización de procesos administrativos liberará tiempo valioso que los trabajadores sociales podrán dedicar a actividades de mayor valor agregado, como entrevistas más profundas, visitas domiciliarias, y coordinación con otras instituciones. La generación automática de reportes facilitará la toma de decisiones y permitirá identificar patrones y tendencias que contribuyan a mejorar las estrategias de intervención social.

La justificación tecnológica del proyecto se basa en la necesidad de modernizar los procesos del consultorio jurídico, alineándolos con las mejores prácticas en gestión de sistemas de información. El desarrollo de una aplicación web garantiza accesibilidad, escalabilidad y facilidad de mantenimiento, características esenciales para un sistema que debe perdurar en el tiempo y adaptarse a las necesidades cambiantes de la institución.

1.3 Objetivos

1.3.1 Objetivo General

Desarrollar un módulo web especializado en Trabajo Social para el Consultorio Jurídico Gratuito de la Pontificia Universidad Católica del Ecuador, que automatice la gestión de casos sociales y optimice los procesos de registro, seguimiento y generación de reportes, mejorando la eficiencia operativa y la calidad de atención a usuarios en situación de vulnerabilidad.

1.3.2 Objetivos Específicos

Objetivo Específico 1: Analizar y documentar los procesos actuales del área de Trabajo Social del Consultorio Jurídico Gratuito, identificando las necesidades específicas de automatización y las funcionalidades requeridas para el módulo web.

Este objetivo busca establecer una línea base clara de los procesos existentes, mediante el levantamiento detallado de información sobre los procedimientos actuales, los formularios utilizados, los tipos de casos atendidos, y las metodologías de trabajo empleadas por los profesionales del área.

Objetivo Específico 2: Diseñar e implementar una interfaz web intuitiva y funcional que permita el registro completo de casos de trabajo social, incluyendo datos socioeconómicos, información familiar, seguimiento de intervenciones y gestión de documentos asociados.

La interfaz debe ser diseñada considerando las mejores prácticas de experiencia de usuario, garantizando que sea fácil de usar para profesionales con diferentes niveles de competencia tecnológica, y que permita un flujo de trabajo natural y eficiente.

Objetivo Específico 3: Implementar funcionalidades de seguimiento y control de casos que permitan el monitoreo del estado de cada expediente, la programación de actividades, y la generación automática de alertas y recordatorios para garantizar la continuidad de la atención.

Este objetivo se enfoca en desarrollar herramientas que faciliten la gestión temporal de los casos y aseguren que ningún usuario pierda continuidad en su atención por falta de seguimiento administrativo.

Objetivo Específico 4: Desarrollar un sistema de generación de reportes automatizados que facilite la elaboración de informes técnicos, estadísticas de atención, y documentos oficiales requeridos en los procesos de trabajo social.

La automatización de reportes es fundamental para reducir la carga administrativa y garantizar la uniformidad y calidad de los documentos generados por el área de Trabajo Social.

1.4 Alcance

El presente trabajo de titulación se enfoca específicamente en el desarrollo del módulo de Trabajo Social para el Consultorio Jurídico Gratuito de la PUCE, utilizando la metodología de prototipado evolutivo y cumpliendo con los estándares técnicos y lineamientos establecidos por el Centro Informático de la universidad.

Alcance Funcional: El módulo incluye las siguientes funcionalidades principales: registro completo de casos sociales con información socioeconómica detallada, gestión del grupo familiar y composición del hogar, seguimiento de intervenciones y actividades realizadas, control de estados de casos y flujos de trabajo, generación de reportes técnicos y documentos oficiales, programación y gestión de citas y entrevistas, y sistema de alertas y notificaciones para seguimiento de casos.

Alcance Técnico: El desarrollo se realiza utilizando tecnologías web modernas, específicamente Vue.js para el frontend, Node.js y Express para el backend, MySQL como sistema de gestión de base de datos, y herramientas complementarias como JWT para autenticación, PrimeVue para componentes de interfaz, y librerías especializadas para generación de documentos.

Alcance Temporal: El proyecto se desarrolla en un período de cinco meses, desde marzo hasta julio de 2025, siguiendo las fases de la metodología de prototipado evolutivo que permite entregas incrementales y retroalimentación continua.

Limitaciones del Alcance: Este trabajo no contempla la puesta en producción definitiva del sistema dentro del Consultorio Jurídico, ya que esto requiere procesos institucionales adicionales de validación y aprobación. Además, se enfoca exclusivamente en el módulo de Trabajo Social, sin incluir otros módulos del sistema integral del consultorio jurídico. La integración con sistemas externos o institucionales queda fuera del alcance actual, aunque se diseña la arquitectura considerando futuras integraciones.

Alcance de Usuarios: El sistema está diseñado para ser utilizado por trabajadores sociales, coordinadores del área, y personal administrativo del Consultorio Jurídico Gratuito, considerando diferentes niveles de acceso y permisos según el rol de cada usuario.

2. CAPÍTULO 2: FUNDAMENTACIÓN TEÓRICA

2.1 Aplicación Web

2.1.1 Definición y características fundamentales

Una aplicación web representa un paradigma de software que opera mediante la arquitectura cliente-servidor, donde el procesamiento principal se realiza en un servidor remoto y los usuarios acceden a través de navegadores web estándar (Pérez et al., 2021). Esta modalidad de desarrollo se ha consolidado como una de las más efectivas para sistemas de gestión organizacional, particularmente en contextos donde se requiere acceso remoto, colaboración entre usuarios, y centralización de la información.

Las características fundamentales que distinguen a las aplicaciones web incluyen la accesibilidad universal, que permite el uso del sistema desde cualquier dispositivo con conexión a internet y navegador web moderno. Esta característica es particularmente relevante para el módulo de Trabajo Social, donde los profesionales pueden necesitar acceder a la información desde diferentes ubicaciones, incluyendo visitas domiciliarias o coordinaciones interinstitucionales.

La escalabilidad inherente de las aplicaciones web facilita el crecimiento gradual del sistema conforme aumenta el número de usuarios o casos gestionados. En el contexto del Consultorio Jurídico Gratuito, esta característica garantiza que el sistema pueda expandirse para atender mayor demanda sin requerir cambios estructurales significativos.

La centralización de datos constituye otra ventaja fundamental, ya que toda la información se almacena en un servidor central, eliminando problemas de sincronización y garantizando que todos los usuarios trabajen con información actualizada. Esta característica es crítica en trabajo social, donde la información desactualizada puede comprometer la calidad de las intervenciones.

2.1.2 Arquitectura de servicios en sistemas de gestión social

La arquitectura de servicios se adapta particularmente bien a las necesidades de los sistemas de gestión social debido a su capacidad para manejar información sensible de manera segura y controlada. En esta arquitectura, el cliente (navegador web) se encarga únicamente de la presentación de datos y la interacción con el usuario, mientras que el servidor centraliza toda la lógica de negocio, procesamiento de datos, y gestión de seguridad.

Esta separación de responsabilidades ofrece ventajas específicas para el trabajo social. El servidor puede implementar validaciones complejas de datos, asegurar la consistencia de la información, y aplicar políticas de seguridad uniformes para proteger datos sensibles de los usuarios atendidos. Mientras tanto, el cliente puede enfocarse en proporcionar una interfaz intuitiva que facilite el trabajo diario de los profesionales.

La comunicación entre cliente y servidor se realiza mediante protocolos estándar como HTTP/HTTPS, utilizando formatos de intercambio de datos como JSON que facilitan la integración con otros sistemas. Esta estandarización es fundamental para futuras integraciones con sistemas institucionales o gubernamentales que puedan requerir intercambio de información.

2.1.3 Ventajas específicas para sistemas de trabajo social

Las aplicaciones web ofrecen ventajas particulares para sistemas destinados al trabajo social que van más allá de los beneficios técnicos generales. La capacidad de acceso remoto permite a los trabajadores sociales actualizar información inmediatamente después de visitas domiciliarias, reuniones con familias, o coordinaciones con otras instituciones, garantizando que los expedientes reflejen siempre el estado más actual de cada caso.

La facilidad de backup y recuperación de datos protege información crítica que, en caso de pérdida, podría comprometer seriamente el seguimiento de casos familiares. Los sistemas web facilitan la implementación de estrategias de respaldo automatizadas y redundantes que garantizan la continuidad del servicio.

La posibilidad de implementar diferentes niveles de acceso permite que coordinadores, trabajadores sociales, y personal administrativo tengan acceso solo a la información necesaria para sus funciones, cumpliendo con principios de confidencialidad y privacidad fundamentales en trabajo social.

2.2 Metodología de desarrollo de software

2.2.1 Prototipado evolutivo en sistemas sociales

La metodología de prototipado evolutivo se selecciona como la más apropiada para el desarrollo del módulo de Trabajo Social debido a la naturaleza compleja y variable de los procesos sociales que el sistema debe automatizar. Esta metodología permite una construcción gradual del sistema, donde cada iteración incorpora retroalimentación directa

de los usuarios finales, asegurando que el producto final se ajuste exactamente a las necesidades reales del área.

El prototipado evolutivo es particularmente valioso en el contexto del trabajo social porque los procesos en esta área suelen estar sujetos a cambios normativos, actualizaciones en metodologías de intervención, y adaptaciones a nuevas realidades sociales. La flexibilidad inherente de esta metodología permite realizar ajustes durante el desarrollo sin comprometer la arquitectura general del sistema.

Fases del prototipado evolutivo aplicadas al módulo:

1. Identificación de requerimientos básicos: Se definen las funcionalidades mínimas necesarias para un sistema funcional.
2. Desarrollo del prototipo inicial: Implementación de funcionalidades básicas para interacción temprana con usuarios.
3. Evaluación continua: Los trabajadores sociales evalúan cada versión, identificando mejoras necesarias.
4. Iteraciones sucesivas: Incorporación de retroalimentación hasta alcanzar un sistema que satisfaga completamente las necesidades operativas.

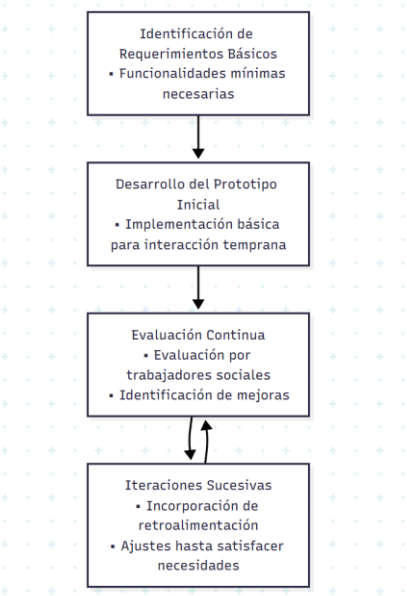


Ilustración 1 Fases del Prototipado Evolutivo Aplicadas al Módulo. Fuente: Elaboración Propia

2.2.2 Gestión de requerimientos en sistemas sociales

La gestión de requerimientos en sistemas destinados al trabajo social presenta desafíos únicos debido a la complejidad de los procesos sociales y la diversidad de

situaciones que deben atenderse. Los requerimientos no solo incluyen aspectos funcionales del sistema, sino también consideraciones éticas, legales, y metodológicas específicas del trabajo social profesional.

La técnica de historias de usuario se adapta particularmente bien a este contexto, ya que permite expresar requerimientos desde la perspectiva de los profesionales que utilizarán el sistema. Por ejemplo, una historia de usuario típica podría ser: "Como trabajadora social, necesito registrar información detallada del grupo familiar para poder realizar una evaluación socioeconómica completa y diseñar un plan de intervención apropiado."

Los requerimientos funcionales del módulo incluyen capacidades de registro de información personal y familiar, seguimiento de intervenciones realizadas, generación de reportes técnicos, y gestión de documentos asociados a cada caso. Los requerimientos no funcionales abarcan aspectos de seguridad de datos, usabilidad para profesionales con diferentes niveles de competencia tecnológica, y capacidad de respuesta adecuada para trabajo de campo.

La validación de requerimientos se realiza mediante sesiones de trabajo colaborativo con el personal del área de Trabajo Social, donde se revisan y ajustan las especificaciones según la experiencia práctica y las mejores prácticas profesionales del campo.

2.3 Base de Datos

2.3.1 Diseño de bases de datos para sistemas sociales

El diseño de bases de datos para sistemas de gestión social requiere consideraciones específicas que van más allá de los principios técnicos estándar. La información manejada en trabajo social incluye datos personales sensibles, historiales familiares complejos, y registros de intervenciones que deben mantenerse seguros y organizados de manera que faciliten el análisis profesional.

La estructura de la base de datos debe reflejar la realidad multidimensional de los casos sociales, donde una familia puede estar compuesta por múltiples individuos con diferentes características, necesidades, y tipos de atención. Esta complejidad requiere un diseño relacional cuidadoso que evite redundancia de datos mientras mantiene la integridad referencial entre diferentes entidades del sistema.

Las tablas principales del sistema incluyen entidades para usuarios del servicio, composición familiar, casos sociales, intervenciones realizadas, y seguimientos temporales.

Cada tabla se diseña considerando los principios de normalización para garantizar eficiencia en el almacenamiento y consulta de datos, mientras se mantiene la flexibilidad necesaria para adaptarse a diferentes tipos de casos y metodologías de intervención.

2.3.2 MySQL como sistema de gestión

MySQL se selecciona como sistema de gestión de base de datos por su robustez, confiabilidad, y amplia compatibilidad con tecnologías de desarrollo web modernas. Su capacidad para manejar grandes volúmenes de datos con rendimiento óptimo lo hace apropiado para sistemas que deben crecer conforme aumenta la demanda de servicios del consultorio jurídico.

Las características de seguridad de MySQL son particularmente relevantes para el módulo de Trabajo Social, donde la protección de datos personales es fundamental. El sistema proporciona mecanismos de encriptación, control de acceso granular, y auditoría de transacciones que garantizan el cumplimiento de normativas de protección de datos personales.

La capacidad de MySQL para manejar transacciones complejas asegura la integridad de los datos durante operaciones que involucran múltiples tablas, como el registro de un nuevo caso que incluye información del usuario, composición familiar, y registro inicial de intervención. Esta característica es crítica para mantener la consistencia de la información en un sistema donde los datos están altamente interrelacionados.

2.4 Lenguaje de Programación

2.4.1 JavaScript en el desarrollo web moderno

JavaScript ha evolucionado desde sus orígenes como lenguaje de scripting básico hasta convertirse en el fundamento del desarrollo web moderno, tanto en el frontend como en el backend. Su versatilidad y ecosistema extenso lo convierten en la elección natural para desarrollar aplicaciones web complejas como el módulo de Trabajo Social.

La naturaleza asíncrona de JavaScript se alinea perfectamente con las necesidades de una aplicación web interactiva, donde los usuarios deben poder realizar múltiples tareas simultáneamente sin experimentar bloqueos en la interfaz. Esta característica es particularmente importante durante entrevistas sociales, donde el profesional puede necesitar consultar información histórica mientras registra nueva información.

El ecosistema de JavaScript, centrado en npm (Node Package Manager), proporciona acceso a miles de bibliotecas especializadas que acelerar el desarrollo y garantizan la calidad del código. Para el módulo de Trabajo Social, esto incluye bibliotecas para validación de datos, generación de documentos, manejo de fechas, y creación de interfaces de usuario avanzadas.

2.4.2 TypeScript para desarrollo

TypeScript extiende JavaScript añadiendo tipado estático, lo que resulta fundamental para desarrollar sistemas robustos y mantenibles como el requerido para el Consultorio Jurídico Gratuito. El sistema de tipos ayuda a prevenir errores comunes durante el desarrollo y facilita la colaboración en equipo al hacer el código más autodocumentado.

Para un sistema que maneja información sensible como los casos sociales, la detección temprana de errores que proporciona TypeScript es invaluable. El compilador identifica inconsistencias en el manejo de datos antes de que el código llegue a producción, reduciendo significativamente el riesgo de fallos que podrían comprometer la integridad de la información.

La capacidad de refactorización segura que ofrece TypeScript facilita la evolución del sistema conforme cambien los requerimientos del área de Trabajo Social. Los cambios en la estructura de datos o en los flujos de trabajo pueden implementarse con confianza, sabiendo que el sistema de tipos identificará cualquier inconsistencia resultante.

2.5 Entorno de Desarrollo Integrado (IDE)

2.5.1 Visual Studio Code para desarrollo web

Visual Studio Code se establece como el entorno de desarrollo ideal para el módulo de Trabajo Social debido a su combinación de potencia, flexibilidad, y facilidad de uso. Su arquitectura extensible permite personalizar el entorno específicamente para las necesidades del proyecto, incorporando herramientas especializadas para Vue.js, Node.js, y MySQL.

El sistema de extensiones de VS Code proporciona soporte avanzado para las tecnologías utilizadas en el proyecto. Extensiones como Vetur para Vue.js ofrecen autocompletado inteligente, detección de errores en tiempo real, y formateado automático de código, mejorando significativamente la productividad durante el desarrollo.

La integración nativa con sistemas de control de versiones como Git facilita la gestión colaborativa del código y el seguimiento de cambios durante el desarrollo iterativo. Esta capacidad es esencial para implementar la metodología de prototipado evolutivo de manera efectiva.

2.6 Framework de Desarrollo Web

2.6.1 Node.js para el backend

Node.js proporciona el entorno de ejecución para el backend del módulo de Trabajo Social, aprovechando JavaScript en el servidor para crear una plataforma de desarrollo unificada. Su arquitectura basada en eventos y su modelo de I/O no bloqueante lo hacen ideal para aplicaciones web que requieren alta concurrencia y respuesta rápida.

La capacidad de Node.js para manejar múltiples conexiones simultáneas es particularmente relevante para un sistema que puede ser utilizado por varios trabajadores sociales al mismo tiempo, cada uno gestionando diferentes casos y realizando diversas operaciones en la base de datos.

El ecosistema npm proporciona módulos especializados para todas las necesidades del backend, incluyendo Express.js para el framework web, Sequelize para la gestión de base de datos, y bibliotecas específicas para generación de documentos PDF y Word requeridos en los reportes de trabajo social.

2.6.2 Vue.js para el frontend

Vue.js se selecciona como framework frontend por su curva de aprendizaje gradual, su arquitectura componentizada, y su excelente documentación. Estas características facilitan el desarrollo de interfaces complejas como las requeridas para la gestión de casos sociales, donde se debe presentar gran cantidad de información de manera organizada e intuitiva.

La reactividad de Vue.js garantiza que la interfaz se actualice automáticamente cuando cambian los datos, proporcionando una experiencia de usuario fluida durante la captura y edición de información de casos. Esta característica es fundamental durante entrevistas largas donde se captura información extensa de manera incremental.

El sistema de componentes de Vue.js permite crear elementos reutilizables que mantienen consistencia visual y funcional en toda la aplicación. Componentes

especializados para formularios de trabajo social, visualización de información familiar, y seguimiento de casos pueden desarrollarse una vez y reutilizarse en diferentes partes del sistema.

2.7 Biblioteca de herramientas

El desarrollo del módulo de Trabajo Social requiere un conjunto específico de bibliotecas y herramientas que faciliten tanto la funcionalidad del sistema como la experiencia del usuario. Estas bibliotecas han sido seleccionadas cuidadosamente para garantizar seguridad, usabilidad, y mantenibilidad del código.

2.7.1 Moment.js para manejo de fechas y zonas horarias

Moment.js proporciona funcionalidades avanzadas para el manejo de fechas, formateo temporal, y gestión de zonas horarias en el módulo de Trabajo Social. Esta biblioteca es fundamental para el sistema debido a la importancia de registrar con precisión las fechas de ingreso de casos, seguimientos realizados, y vencimientos de actividades programadas.

En el contexto del trabajo social, donde el seguimiento temporal de casos es crítico, Moment.js facilita operaciones como el cálculo de períodos de atención, generación de alertas por vencimientos, y formateo consistente de fechas en reportes. La biblioteca también maneja la zona horaria de Ecuador (GMT-5) de manera automática, asegurando que todas las fechas se registren y muestren correctamente según el contexto local (Ver Anexo 1).

2.7.2 ExcelJS para generación de reportes

ExcelJS permite la creación programática de archivos Excel con formato profesional, funcionalidad esencial para la generación de reportes estadísticos y de seguimiento requeridos en el área de Trabajo Social. La biblioteca ofrece control completo sobre el formato, estilo, y estructura de las hojas de cálculo generadas.

Esta herramienta es particularmente valiosa para crear reportes que incluyen información de múltiples casos, estadísticas departamentales, y análisis de tendencias en los tipos de atención proporcionada. Los reportes Excel generados facilitan el análisis de datos por parte de coordinadores y la preparación de informes institucionales (Ver Anexo 2).

2.7.3 DOCX para generación de documentos Word

La biblioteca DOCX (también conocida como docx-js) facilita la creación de documentos Word programáticamente, permitiendo generar informes técnicos individuales para cada caso de trabajo social. Esta funcionalidad es fundamental para producir documentos oficiales que requieren formato específico y estructura profesional.

Los documentos Word generados incluyen toda la información relevante del caso organizada en secciones estándar que siguen las metodologías establecidas en trabajo social. La automatización de este proceso elimina horas de trabajo manual y garantiza consistencia en el formato de todos los documentos producidos (Ver Anexo 3).

2.7.4 Sequelize como ORM (Object-Relational Mapping)

Sequelize actúa como la capa de abstracción entre el código JavaScript y la base de datos MySQL, proporcionando un enfoque orientado a objetos para las operaciones de base de datos. Esta biblioteca simplifica significativamente el desarrollo al eliminar la necesidad de escribir consultas SQL directas para operaciones comunes.

En el módulo de Trabajo Social, Sequelize facilita la definición de modelos de datos complejos que reflejan las relaciones entre usuarios, casos, grupos familiares, y seguimientos. La biblioteca también proporciona validaciones automáticas, gestión de transacciones, y migración de esquemas de base de datos (Ver Anexo 4).

2.7.5 PrimeVue para componentes de interfaz

PrimeVue proporciona una biblioteca completa de componentes de interfaz de usuario diseñados específicamente para Vue.js, ofreciendo elementos profesionales que aceleran el desarrollo y garantizan consistencia visual en toda la aplicación. Los componentes incluyen formularios avanzados, tablas de datos, calendarios, y elementos de navegación.

Para el módulo de Trabajo Social, PrimeVue es especialmente valioso porque proporciona componentes especializados como Toast para notificaciones de usuario, DataTable para visualización de casos, y Calendar para programación de actividades. Estos componentes están optimizados para accesibilidad y responsive design, características esenciales para un sistema que debe funcionar en diferentes dispositivos (Ver Anexo 5).

2.7.6 Axios para comunicación HTTP

Aunque no se observa directamente en el código proporcionado, el sistema utiliza fetch nativo para comunicación HTTP, lo cual representa una aproximación moderna y eficiente para realizar peticiones al servidor. Esta elección técnica elimina dependencias externas innecesarias mientras proporciona todas las funcionalidades requeridas para la comunicación cliente-servidor.

El uso de fetch nativo permite implementar interceptores personalizados, manejo de errores específico, y configuración de headers de autenticación de manera directa y transparente. Esta aproximación es particularmente apropiada para un sistema que requiere control granular sobre las comunicaciones con el backend (Ver Anexo 6).

Esta selección de bibliotecas y herramientas proporciona una base tecnológica sólida para el módulo de Trabajo Social, equilibrando funcionalidad, performance, y mantenibilidad del código. Cada biblioteca ha sido elegida considerando no solo sus capacidades técnicas, sino también su estabilidad a largo plazo y su compatibilidad con el ecosistema tecnológico establecido en el proyecto.

2.8 Control de Versiones

2.8.1 Importancia del control de versiones en desarrollo iterativo

El control de versiones es fundamental para implementar efectivamente la metodología de prototipado evolutivo, ya que permite gestionar los cambios incrementales del sistema mientras se mantiene un historial completo de modificaciones. Esta capacidad es esencial cuando se desarrolla un sistema que debe evolucionar basándose en retroalimentación continua de usuarios.

2.8.2 Git y GitHub para colaboración

Git proporciona control de versiones distribuido que facilita el trabajo colaborativo y la gestión de diferentes líneas de desarrollo. GitHub complementa Git ofreciendo herramientas adicionales para gestión de proyectos, seguimiento de issues, y documentación colaborativa.

2.9 Buenas Prácticas de Desarrollo Web

2.9.1 Principios de diseño y usabilidad para sistemas sociales

El diseño de interfaces para sistemas de trabajo social debe priorizar la claridad, accesibilidad, y eficiencia. Los trabajadores sociales frecuentemente trabajan bajo presión y con limitaciones de tiempo, por lo que la interfaz debe minimizar la carga cognitiva y facilitar la captura rápida y precisa de información.

Los principios de diseño centrado en el usuario guían el desarrollo de cada componente de la interfaz, asegurando que los flujos de trabajo se alineen con las prácticas profesionales establecidas en trabajo social y que la navegación sea intuitiva para usuarios con diferentes niveles de competencia tecnológica.

2.9.2 Seguridad en aplicaciones web para datos sensibles

La seguridad en aplicaciones que manejan datos personales sensibles requiere implementar múltiples capas de protección. Esto incluye encriptación de datos en tránsito y en reposo, validación estricta de entradas, control de acceso basado en roles, y auditoría de todas las operaciones realizadas en el sistema.

Las mejores prácticas de seguridad también incluyen actualizaciones regulares de dependencias, configuración segura de servidores, e implementación de mecanismos de backup y recuperación que garanticen la continuidad del servicio sin comprometer la confidencialidad de los datos.

3. CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL MÓDULO

3.1 Análisis de Requerimientos

3.1.1 Requisitos Funcionales

El análisis de requisitos funcionales para el módulo de Trabajo Social se fundamenta en el estudio detallado de los procesos actuales del Consultorio Jurídico Gratuito y las necesidades específicas identificadas por los profesionales del área. Estos requisitos se han categorizado según las funcionalidades principales del sistema:

RF001 - Gestión de Casos de Trabajo Social El sistema debe permitir la creación, modificación y consulta de casos de trabajo social, incluyendo información completa del usuario solicitante, datos socioeconómicos, y detalles específicos de la situación social a atender. Esta funcionalidad constituye el núcleo del sistema, implementada a través del componente SocialWorkNewCase.vue que maneja formularios complejos con múltiples secciones de información.

RF002 - Registro de Información Personal y Familiar El módulo debe capturar información detallada del usuario principal y su grupo de convivencia, incluyendo datos demográficos, ocupacionales, y de relación familiar. El sistema implementa esta funcionalidad mediante formularios dinámicos que permiten agregar múltiples miembros del grupo familiar, cada uno con sus características específicas.

RF003 - Seguimiento y Control de Estados de Casos El sistema debe proporcionar mecanismos para el seguimiento temporal de casos, permitiendo cambios de estado (Activo, Pendiente, Archivado) con observaciones correspondientes. Esta funcionalidad se implementa en el componente SocialWorkCases.vue, que incluye confirmaciones de cambio de estado y registro de observaciones obligatorias.

RF004 - Gestión de Citas y Programación El módulo debe incluir funcionalidades para programar citas con usuarios, gestionar horarios de atención, y mantener registro de contactos realizados. El componente SocialWorkSchedule.vue implementa un sistema completo de gestión de agenda con calendario semanal y seguimiento de contactos.

RF005 - Generación de Reportes y Documentos El sistema debe generar automáticamente reportes en formato Excel para análisis estadístico y documentos individuales en formato Word para cada caso. Esta funcionalidad se implementa en el

backend a través de los métodos generateExcelReport y generateWordReport del modelo Social_WorkModel.js.

RF006 - Control de Acceso y Autenticación El módulo debe implementar un sistema de autenticación seguro que permita diferentes niveles de acceso según el rol del usuario (trabajador social, coordinador, administrador), protegiendo la confidencialidad de la información.

RF007 - Búsqueda y Filtrado de Casos El sistema debe proporcionar capacidades de búsqueda avanzada por diferentes criterios (nombre, documento de identidad, estado del caso, fechas) para facilitar la localización rápida de información específica.

3.1.2 Requisitos No Funcionales

Los requisitos no funcionales establecen las características de calidad que debe cumplir el módulo de Trabajo Social para garantizar su operación efectiva en el entorno del Consultorio Jurídico Gratuito:

RNF001 - Usabilidad El sistema debe ser intuitivo para usuarios con diferentes niveles de competencia tecnológica. La interfaz debe seguir principios de diseño centrado en el usuario, con navegación clara, mensajes informativos, y flujos de trabajo que se alineen con las prácticas profesionales del trabajo social. El tiempo de aprendizaje para un nuevo usuario no debe exceder 2 horas de capacitación básica.

RNF002 - Rendimiento El sistema debe responder a las solicitudes del usuario en un tiempo máximo de 3 segundos para operaciones normales y 10 segundos para generación de reportes complejos. La aplicación debe ser capaz de manejar al menos 50 usuarios concurrentes sin degradación significativa del rendimiento.

RNF003 - Seguridad y Confidencialidad Toda la información personal debe estar protegida mediante encriptación en tránsito (HTTPS) y en reposo. El sistema debe implementar controles de acceso basados en roles, auditoría de todas las operaciones, y mecanismos de backup automático. Debe cumplir con las normativas de protección de datos personales vigentes en Ecuador.

RNF004 - Disponibilidad El sistema debe mantener una disponibilidad mínima del 95% durante horarios laborales (8:00 AM - 6:00 PM, lunes a viernes). Debe incluir mecanismos de recuperación ante fallos y procedimientos de backup que garanticen la continuidad del servicio.

RNF005 - Escalabilidad La arquitectura del sistema debe permitir el crecimiento en número de usuarios, casos gestionados, y funcionalidades adicionales sin requerir cambios fundamentales en la estructura. Debe ser capaz de crecer hasta 200 usuarios concurrentes y almacenar al menos 10,000 casos activos.

RNF006 - Mantenibilidad El código debe estar bien documentado, seguir estándares de programación establecidos, y utilizar arquitecturas modulares que faciliten futuras modificaciones. El sistema debe permitir actualizaciones sin interrumpir el servicio y debe incluir herramientas de monitoreo para identificar problemas potenciales.

RNF007 - Compatibilidad El sistema debe funcionar correctamente en los navegadores web más utilizados (Chrome, Firefox, Safari, Edge) en sus versiones actuales y al menos dos versiones anteriores. Debe ser responsivo para funcionar adecuadamente en dispositivos móviles y tablets.

RNF008 - Integridad de Datos El sistema debe garantizar la consistencia e integridad de los datos mediante validaciones tanto en el frontend como en el backend, transacciones atómicas, y mecanismos de verificación de integridad referencial en la base de datos.

3.2 Modelado del Sistema

3.2.1 Diagrama de Casos de Uso

El diagrama de casos de uso del módulo de Trabajo Social refleja las interacciones principales entre los actores del sistema y las funcionalidades disponibles. Los actores identificados incluyen:

Actor Principal: Trabajadora Social Representa a los profesionales de trabajo social que utilizan el sistema para gestionar casos, realizar seguimientos, y generar reportes. Este actor tiene acceso completo a las funcionalidades de registro, modificación, y consulta de casos.

Actor Secundario: Usuario/Cliente Representa a las personas que reciben atención del Consultorio Jurídico Gratuito. Aunque no interactúan directamente con el sistema, sus datos son el centro de todas las operaciones.

Casos de Uso Principales:

1. Agendar una cita: Permite programar entrevistas iniciales o de seguimiento con usuarios del servicio.

2. Primera entrevista: Facilita el registro completo de información durante la entrevista inicial, incluyendo datos personales, familiares, y socioeconómicos.

3. Asignación del Módulo: Gestiona la asignación de casos específicos a trabajadores sociales según disponibilidad y especialización.

4. Seguimiento al caso: Permite registrar intervenciones realizadas, cambios en la situación familiar, y actualizaciones del estado del caso.

5. Horarios: Gestiona la disponibilidad de los profesionales y la programación de actividades relacionadas con cada caso.

Las relaciones entre casos de uso incluyen extensiones donde "Agendar una cita" extiende a "Primera entrevista", ya que toda primera entrevista requiere una cita previa. Asimismo, "Seguimiento al caso" incluye tanto "Primera entrevista" como "Agendar una cita", ya que el seguimiento puede requerir entrevistas adicionales.

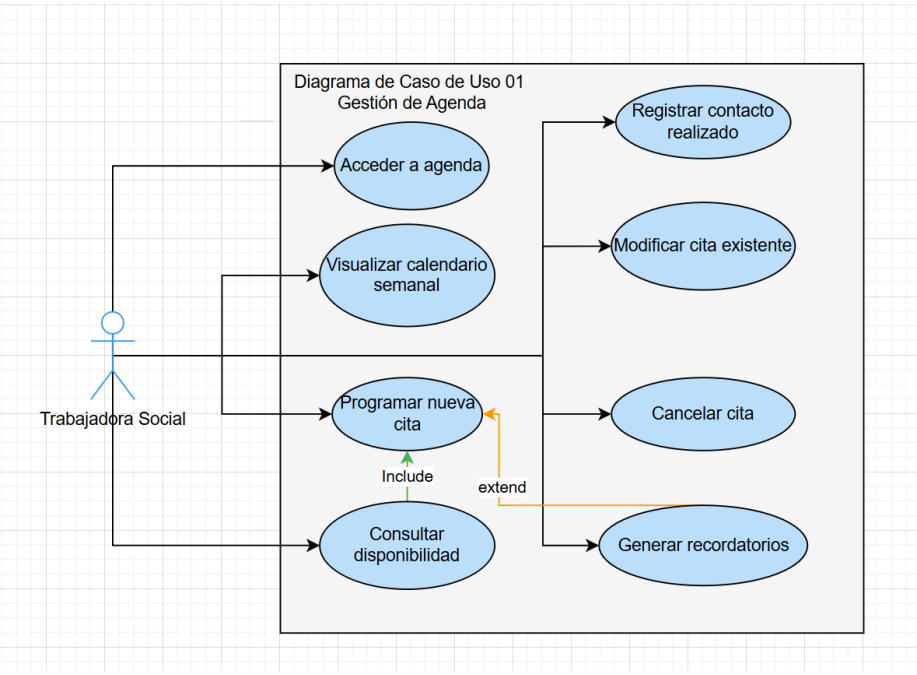


Ilustración 2 Diagrama de casos de uso: Gestión de agenda. Fuente: Elaboración propia.

Nombre del caso de uso: Programar Nueva Cita	**ID Único:** CU-01
Área: -	
Actor(es): Trabajadora Social (Principal)	
Interesados: Usuario del Sistema	
Nivel: -	
Descripción: Permitir a la trabajadora social programar citas con usuarios del servicio social	
Evento desencadenador: La trabajadora social accede al módulo de agenda, selecciona una fecha y hora disponible, e ingresa la información necesaria para programar una nueva cita con un usuario	
Tipo de desencadenador: ☑ Externo ☐ Temporal	
Pasos realizados (ruta principal):	**Información para los pasos**
1. La trabajadora social accede al módulo de agenda	
2. El sistema muestra el calendario semanal con horarios disponibles	
3. La trabajadora social selecciona fecha y hora deseada	
4. El sistema verifica disponibilidad del horario	
5. La trabajadora social ingresa datos de la cita	Formulario con campos: nombre cliente, motivo, duración, fecha
6. El sistema valida la información ingresada	
7. La trabajadora social confirma la programación	
8. El sistema registra la cita y muestra confirmación	
Precondiciones: Usuario autenticado en el sistema - Módulo de agenda accesible - Existir horarios disponibles	
Postcondiciones: Cita registrada en el sistema - Horario marcado como ocupado - Notificación generada (si aplica)	
Suposiciones: El actor tiene acceso a un navegador web compatible y conexión a la red de la PUCE. La trabajadora social conoce el procedimiento de programación de citas - El sistema de agenda está funcionando correctamente.	
Garantía de éxito: La trabajadora social visualiza sus citas agendadas en el calendario.	
Garantía mínima: El sistema envía un mensaje de error indicando el motivo de por qué no se puede agendar una cita en la hora seleccionada.	
Requerimientos cumplidos: Permitir a la trabajadora social manejar un horario dentro del sistema.	
Cuestiones pendientes: Implementación de recordatorios automáticos vía SMS o email - Integración con calendarios externos - Funcionalidad de reprogramación masiva en caso de emergencias - Generación de reportes estadísticos de ocupación de agenda	
Prioridad: Alta	
Riesgo: Medio	

Tabla 1 Descripción de caso de uso de gestión de agenda. Fuente: Elaboración propia.

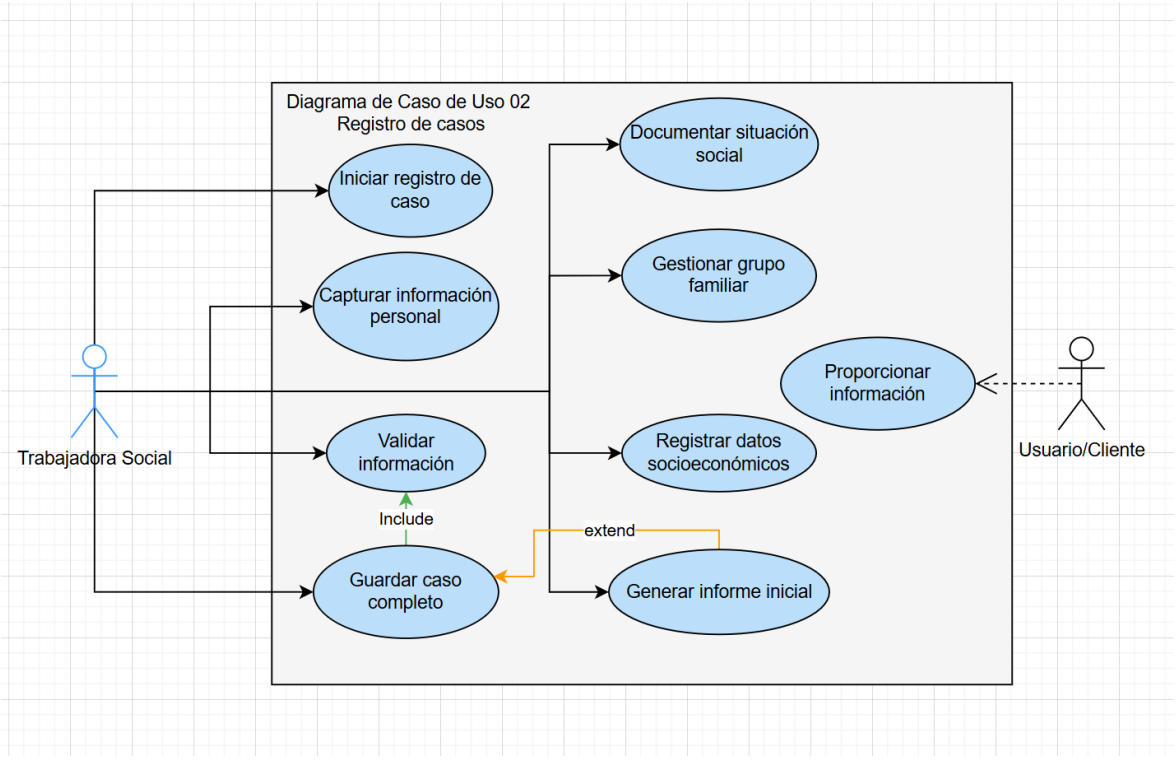


Ilustración 3 Diagrama de casos de uso: Registro de casos. Fuente: Elaboración Propia

Nombre del caso de uso: Registrar Nuevo Caso Social	**ID Único:** CU-02
Área: -	
Actor(es): Trabajadora Social (Principal), Usuario/Cliente (Secundario)	
Interesados: Usuario del Sistema	
Nivel: -	
Descripción: Permitir el registro completo de información de un nuevo caso social durante la entrevista inicial	
Evento desencadenador: La trabajadora social captura información completa del usuario y su situación socioeconómica durante una entrevista presencial, creando un nuevo expediente en el sistema	
Tipo de desencadenador: ☑ Externo ☐ Temporal	
Pasos realizados (ruta principal):	**Información para los pasos**
1. La trabajadora social accede al formulario de nuevo caso	
2. El sistema muestra formulario con secciones organizadas	
3. La trabajadora social captura información personal	Campos obligatorios
4. La trabajadora social gestiona información del grupo familiar	Formulario dinámico para agregar miembros
5. El sistema ejecuta validaciones completas	Formulario con campos: nombre cliente, motivo, duración, fecha
6. El sistema asigna número de proceso y guarda el caso	Generación automática de código alfanumérico: TS-YYYY-NNN
7. El sistema muestra confirmación de registro exitoso	
Precondiciones: Usuario autenticado en el sistema - Cita programada con el usuario - Usuario presente para la entrevista - Formulario de registro disponible	
Postcondiciones: Caso social registrado con número único de proceso - Información almacenada en base de datos - Usuario puede ser referenciado en futuras consultas - Historial de caso iniciado	
Suposiciones: El usuario proporciona información veraz y completa - Existe un ambiente adecuado para realizar la entrevista - La trabajadora social conoce el manejo del sistema	
Garantía de éxito: El caso queda completamente registrado con número único de proceso.	
Garantía mínima: Los datos parciales ingresados se conservan en caso de interrupción.	
Requerimientos cumplidos: Permitir a la trabajadora social Gestión de Casos de Trabajo Social.	
Cuestiones pendientes: Implementación de firma digital para validación de datos - Integración con Registro Civil para verificación automática de cédulas.	
Prioridad: Alta	
Riesgo: Alto	

Tabla 2 Descripción de caso de uso de registro de casos. Fuente: Elaboración propia.

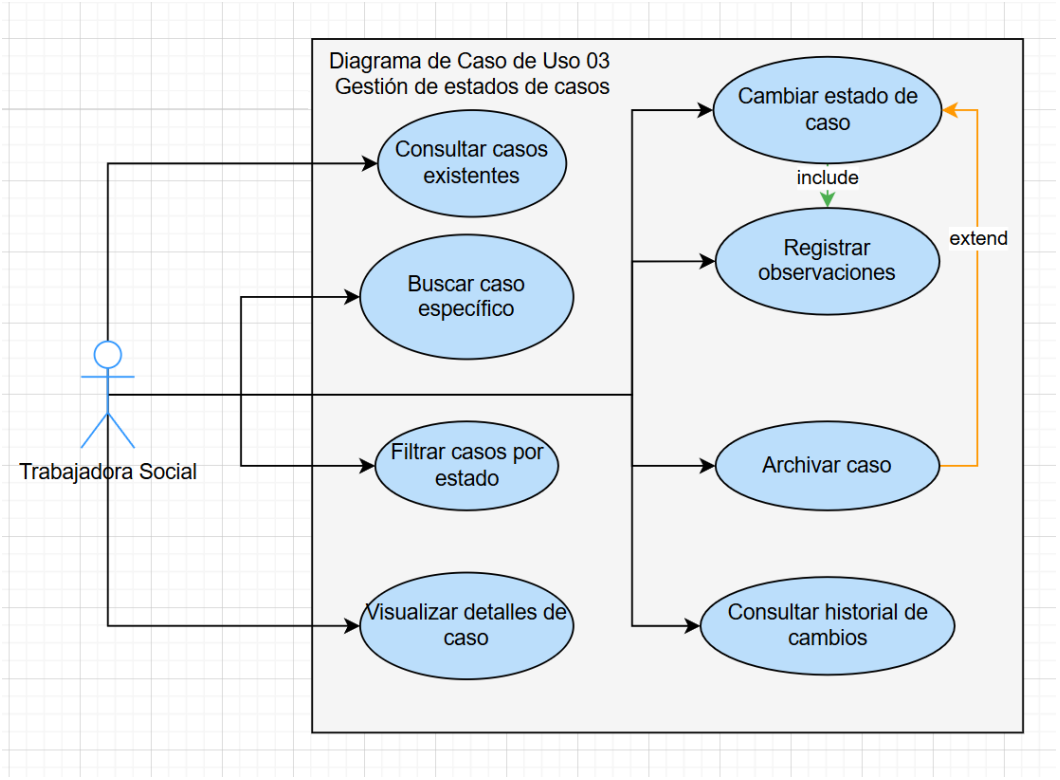


Ilustración 4 Diagrama de casos de uso: Gestión de estados de caso. Fuente: Elaboración propia

Nombre del caso de uso: Cambiar Estado de Caso	**ID Único:** CU-03
Área: -	
Actor(es): Trabajadora Social (Principal)	
Interesados: Usuario del Sistema	
Nivel: -	
Descripción: Permitir a la trabajadora social actualizar el estado de un caso social según su evolución	
Evento desencadenador: La trabajadora social busca un caso específico y actualiza su estado registrando las observaciones correspondientes que justifican el cambio	
Tipo de desencadenador: ☑ Externo ☐ Temporal	
Pasos realizados (ruta principal):	**Información para los pasos**
1. La trabajadora social accede a la vista de gestión de casos	
2. El sistema muestra lista de casos con sus estados actuales	
3. La trabajadora social busca el caso específico	Búsqueda por número de proceso, nombre del usuario o cédula
4. El sistema muestra los detalles del caso seleccionado	Formulario dinámico para agregar miembros
5. El sistema muestra opciones de estado disponibles	Formulario con campos: nombre cliente, motivo, duración, fecha
6. El sistema solicita observaciones obligatorias del cambio	Generación automática de código alfanumérico: TS-YYYY-NNN
7. La trabajadora social ingresa justificación del cambio	Justificación técnica del cambio basada en la evolución del caso
8. El sistema actualiza el estado y registra el historial	Registro automático de timestamp, usuario que ejecuta el cambio y razón
Precondiciones: Usuario autenticado en el sistema - Caso social existente en el sistema - Permisos para modificar estados de casos	
Postcondiciones: - Estado del caso actualizado en base de datos - Historial de cambios registrado con timestamp - Observaciones del cambio almacenadas - Usuario notificado del cambio exitoso	
Suposiciones: - La trabajadora social tiene criterio profesional para determinar cambios de estado - Los cambios de estado reflejan la evolución real del caso	
Garantía de éxito: El estado del caso se actualiza correctamente en el sistema.	
Garantía mínima: La información previa del caso se mantiene intacta.	
Requerimientos cumplidos: Permitir a la trabajadora social Seguimiento y Control de Estados de Casos.	
Cuestiones pendientes: Implementación de flujos de aprobación para ciertos cambios de estado - Notificaciones automáticas a supervisores en casos críticos.	
Prioridad: Media	
Riesgo: Medio	

Tabla 3 Descripción de caso de uso de cambio de estado de caso. Fuente: Elaboración propia

3.3 Diseño de la Interfaz del Usuario (UI)

3.3.1 Principios de Diseño para Sistemas Sociales

El diseño de la interfaz del módulo de Trabajo Social se fundamenta en principios específicos que consideran las características particulares del trabajo social profesional y las necesidades de los usuarios que atiende el Consultorio Jurídico Gratuito.

Claridad y Simplicidad: La interfaz prioriza la presentación clara de información, evitando elementos decorativos innecesarios que puedan distraer durante entrevistas importantes. Los formularios están organizados en secciones lógicas que reflejan el flujo natural de una entrevista social.

Accesibilidad Universal: El diseño considera diferentes niveles de habilidad tecnológica entre los usuarios, implementando navegación intuitiva, etiquetas descriptivas, y mensajes de ayuda contextual que guían al usuario en cada paso del proceso.

Eficiencia Operativa: La interfaz está optimizada para reducir el número de clics y el tiempo necesario para completar tareas frecuentes. Los formularios incluyen autocompletado, validación en tiempo real, y guardado automático para minimizar la pérdida de información.

3.3.2 Componentes Principales de la Interfaz

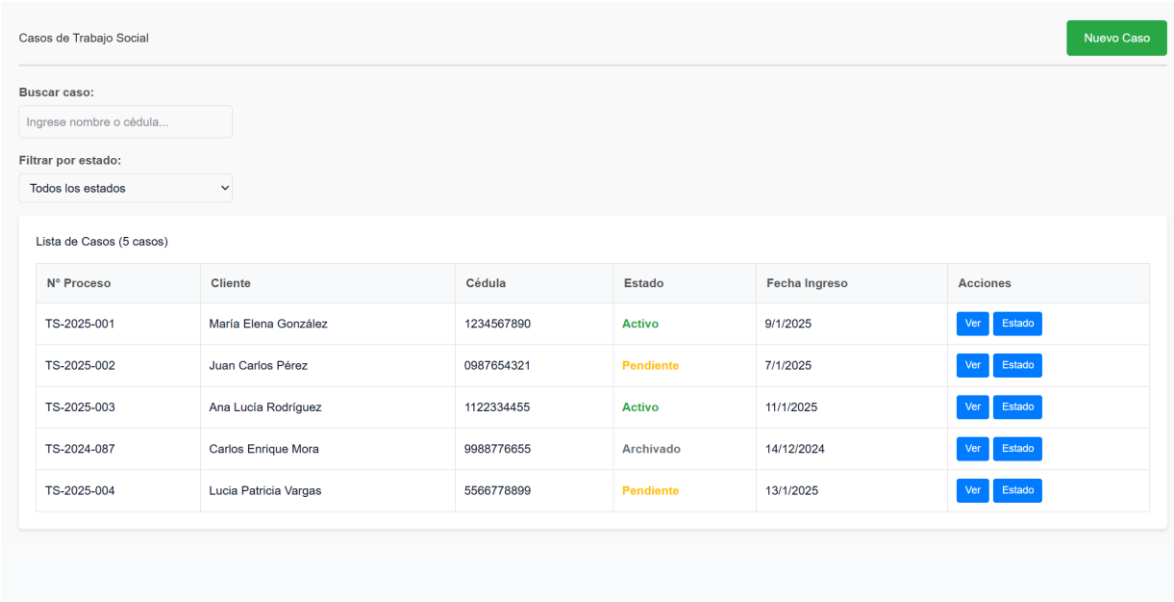


Ilustración 5 Vista previa de Gestión de Casos (SocialWorkCases.vue)

Esta vista constituye el dashboard principal del módulo, presentando una vista general de todos los casos activos, con otros estados, y archivados. La interfaz utiliza un diseño de tarjetas que permite visualizar rápidamente información clave de cada caso: nombre del usuario, documento de identidad, estado actual, y fechas relevantes.

Las funcionalidades de búsqueda y filtrado están prominentemente ubicadas en la parte superior, permitiendo localizar casos específicos por nombre, documento, o estado. Los controles de cambio de estado están integrados directamente en cada tarjeta, facilitando actualizaciones rápidas con confirmaciones apropiadas para prevenir cambios accidentales.



Ilustración 6 Formulario previo de Registro de Casos (SocialWorkNewCase.vue) 1



Ilustración 7 Formulario previo de Registro de Casos (SocialWorkNewCase.vue) 2

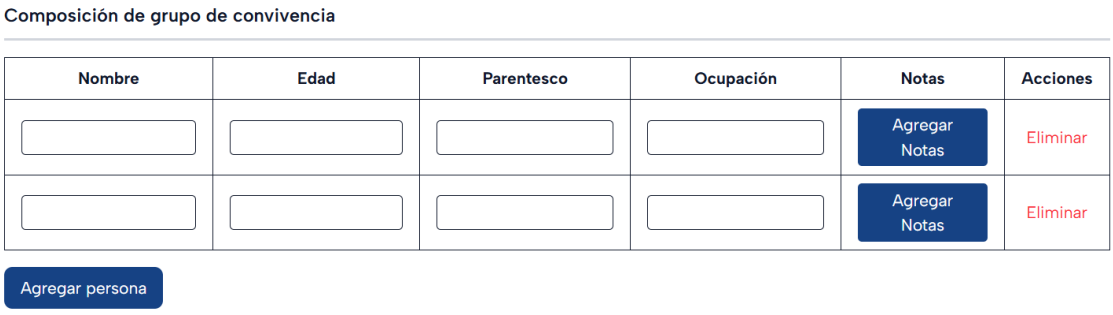


Ilustración 8 Formulario previo de Registro de Casos (SocialWorkNewCase.vue) 3

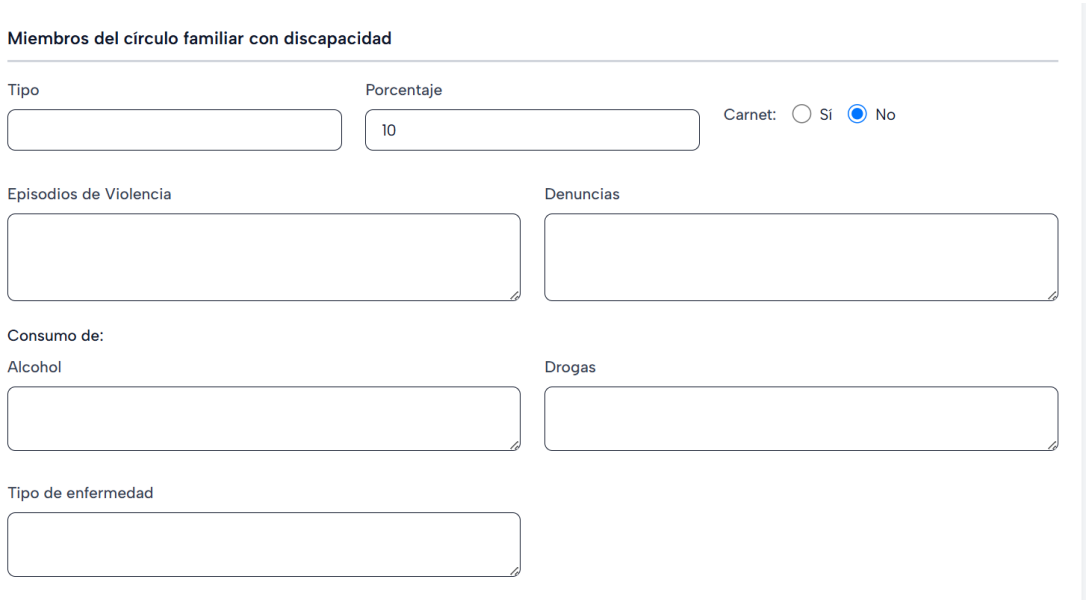


Ilustración 9 Formulario previo de Registro de Casos (SocialWorkNewCase.vue) 4



Ilustración 10 Formulario previo de Registro de Casos (SocialWorkNewCase.vue) 5



Ilustración 11 Formulario previo de Registro de Casos (SocialWorkNewCase.vue) 6

El formulario principal del módulo presenta una estructura de múltiples secciones que refleja la metodología estándar de entrevista en trabajo social:

- Sección de Información Básica: Captura datos de identificación y contacto

- Sección de Datos Socioeconómicos: Registra información sobre ingresos, vivienda, y situación laboral
- Sección de Grupo Familiar: Permite agregar dinámicamente miembros del hogar con sus características individuales
- Sección de Situación Social: Documenta la problemática específica y las solicitudes del usuario
- Sección de Observaciones: Proporciona espacio para notas profesionales y observaciones técnicas

Cada sección incluye validaciones específicas que aseguran la calidad de los datos capturados, mientras que el diseño responsivo permite el uso eficiente tanto en computadoras de escritorio como en dispositivos móviles durante visitas domiciliarias.

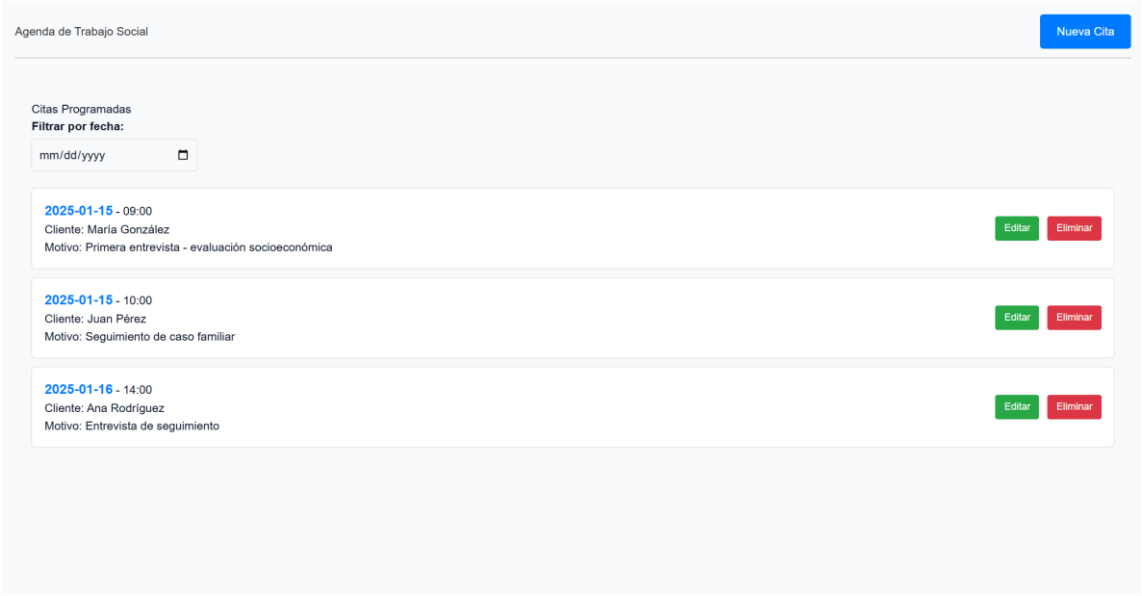


Ilustración 12 Primer diseño de gestión de Agenda (SocialWorkSchedule.vue)

El módulo de gestión de agenda presenta un calendario semanal que facilita la programación de citas y el seguimiento de contactos con usuarios. La interfaz incluye:

- Vista de Calendario: Presenta los días laborales con horarios disponibles y citas programadas

- Panel de Estado de Contactos: Muestra alertas visuales para casos que requieren seguimiento prioritario
- Formularios de Programación: Permiten agendar nuevas citas con información completa del motivo y duración

El sistema de codificación por colores facilita la identificación rápida de diferentes tipos de actividades y el estado de seguimiento de cada caso.

3.3.3 Diseño Responsivo y Accesibilidad

El diseño del módulo implementa principios de diseño responsivo que garantizan funcionalidad óptima en diferentes dispositivos y tamaños de pantalla. Esta característica es fundamental para profesionales que necesitan acceder al sistema desde ubicaciones diversas, incluyendo visitas domiciliarias y reuniones externas.

Contraste y Legibilidad: La paleta de colores seleccionada garantiza contraste suficiente para usuarios con diferentes capacidades visuales, mientras que la tipografía utiliza tamaños y espaciados que facilitan la lectura durante sesiones prolongadas de trabajo.

Navegación por Teclado: Todos los elementos interactivos son accesibles mediante navegación por teclado, facilitando el uso para profesionales que prefieren métodos de entrada alternativos o tienen limitaciones en el uso del mouse.

3.3.4 Flujos de Trabajo en la Interfaz

Flujo de Registro de Nuevo Caso

El flujo para registrar un nuevo caso está diseñado para minimizar la interrupción durante entrevistas presenciales:

1. Acceso Rápido: Desde la vista principal, un botón prominente permite acceder directamente al formulario de nuevo caso

2. Captura Progresiva: El formulario permite guardado parcial, facilitando entrevistas que pueden interrumpirse o continuarse en sesiones posteriores

3. Validación Contextual: Las validaciones aparecen en tiempo real sin bloquear el progreso, permitiendo continuar con la entrevista mientras se completan campos requeridos

4. Confirmación Final: Un resumen completo permite verificar la información antes del guardado definitivo

Flujo de Seguimiento de Casos

El seguimiento de casos existentes utiliza un flujo optimizado para actualizaciones rápidas:

1. Localización Rápida: La funcionalidad de búsqueda permite encontrar casos específicos inmediatamente

2. Vista de Resumen: Al seleccionar un caso, se presenta un resumen de la información más relevante

3. Edición Contextual: Las modificaciones se pueden realizar directamente desde la vista de resumen sin navegar a formularios separados

4. Registro de Actividades: Un panel lateral permite agregar notas de seguimiento y actualizar el estado del caso

Flujo de Generación de Reportes

La generación de documentos oficiales sigue un proceso simplificado:

1. Selección de Caso: Desde cualquier vista de caso, se puede acceder directamente a opciones de reporte

2. Configuración Mínima: Los reportes utilizan plantillas predefinidas que requieren configuración mínima

3. Generación Automática: Los documentos se generan automáticamente incorporando toda la información relevante del caso

4. Descarga Inmediata: Los archivos generados están disponibles para descarga inmediata en formatos estándar (Word, Excel)

Esta organización de flujos garantiza que los profesionales puedan concentrarse en la atención de calidad a los usuarios mientras el sistema facilita las tareas administrativas necesarias.

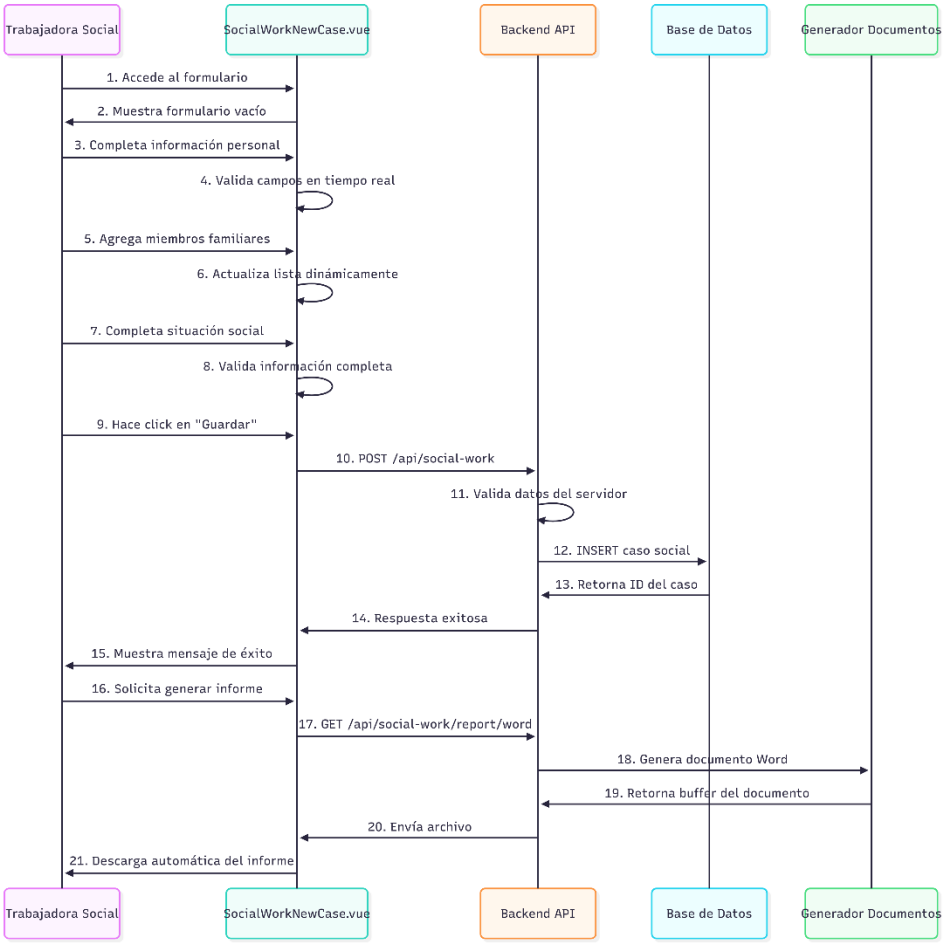


Ilustración 13 Diagrama de flujo por bloques del módulo

4 CAPÍTULO 4: DESARROLLO DEL MÓDULO

4.1 Preparación del Entorno de Desarrollo

4.1.1 Configuración del Entorno Tecnológico

La preparación del entorno de desarrollo para el módulo de Trabajo Social requiere la configuración cuidadosa de múltiples tecnologías que trabajan de manera integrada. El entorno se estructura siguiendo las mejores prácticas de desarrollo web moderno, garantizando consistencia entre los ambientes de desarrollo, pruebas, y producción.

Configuración del Frontend con Vue.js

El frontend utiliza Vue.js 3 con la Composition API, proporcionando una base sólida para interfaces reactivas y mantenibles (Ver Anexo 7):

4.1.2 Estándares de Codificación

Los estándares de codificación establecidos para el proyecto garantizan consistencia, legibilidad, y mantenibilidad del código a largo plazo. Estos estándares se aplican tanto al frontend como al backend del módulo.

Estándares para JavaScript/TypeScript

- Nomenclatura: Se utiliza camelCase para variables y funciones, PascalCase para clases y componentes, y UPPER_SNAKE_CASE para constantes globales
- Estructura de archivos: Cada archivo tiene una responsabilidad única y clara, con importaciones organizadas alfabéticamente
- Comentarios: Se documentan funciones complejas, algoritmos específicos, y decisiones de diseño importantes
- Manejo de errores: Todas las operaciones asíncronas incluyen manejo apropiado de errores con mensajes informativos

Estándares para Vue.js

Los componentes Vue.js siguen la guía de estilo oficial de Vue, con adaptaciones específicas para el contexto del trabajo social (Ver Anexo 8).

4.2 Implementación de la Funcionalidad

4.2.1 Backend - Desarrollo de APIs y Lógica de Negocio

El backend del módulo de Trabajo Social se estructura alrededor de APIs RESTful que gestionan todas las operaciones relacionadas con casos sociales. La implementación sigue el patrón MVC (Model-View-Controller) adaptado para APIs web.

Controlador Principal (SocialWorkController.js)

El controlador principal gestiona todas las operaciones CRUD (Create, Read, Update, Delete) para casos de trabajo social (Ver Anexo 9) (Ver Anexo 10).

Este controlador implementa validaciones robustas, manejo comprehensivo de errores, y respuestas HTTP estándar que facilitan la integración con el frontend.

Modelo de Datos (Social_WorkModel.js)

El modelo encapsula toda la lógica de acceso a datos y operaciones complejas de la base de datos (Ver Anexo 11).

Generación de Reportes

Una funcionalidad crítica del sistema es la generación automática de reportes en diferentes formatos (Ver Anexo 12).

4.2.2 Frontend - Desarrollo de Interfaces de Usuario

El frontend del módulo implementa interfaces intuitivas y funcionales que facilitan el trabajo diario de los profesionales de trabajo social.

Componente Principal de Gestión de Casos (SocialWorkCases.vue)

Este componente presenta una vista general de todos los casos y permite operaciones de búsqueda, filtrado, y gestión de estados.

Formulario Completo de Registro (SocialWorkNewCase.vue)

El formulario principal implementa una interfaz compleja que maneja múltiples secciones de información.

Gestión de Agenda y Programación (SocialWorkSchedule.vue)

El componente de programación implementa un sistema completo de gestión de citas con funcionalidades avanzadas.

4.2.3 Integración Backend-Frontend

La integración entre el backend y frontend se realiza mediante APIs RESTful que siguen estándares de la industria y mejores prácticas de seguridad.

Comunicación Asíncrona, el frontend utiliza async/await para comunicaciones no bloqueantes con el backend.

4.3 Documentación del Código

4.3.1 Documentación de APIs

La documentación de las APIs del módulo de Trabajo Social proporciona información completa sobre endpoints, parámetros, y respuestas esperadas.

Endpoint de Gestión de Casos

```
/**
 * @api {get} /social-work Obtener todos los casos de trabajo social
 * @apiName GetAllSocialWorkCases
 * @apiGroup SocialWork
 * @apiDescription Recupera una lista de todos los casos de trabajo social
```

```
 * con información básica de usuario e iniciales de consulta.
 * @apiSuccess {Object[]} cases Lista de casos de trabajo social
 * @apiSuccess {String} cases.SW_ProcessNumber Número único del proceso
 * @apiSuccess {String} cases.SW_Status Estado actual del caso
 * @apiSuccess {Date} cases.SW_EntryDate Fecha de ingreso del caso
 * @apiSuccess {Object} cases.Initial_Consultation Datos de consulta inicial
 * @apiSuccess {Object} cases.User Información del usuario beneficiario
 * @apiError (500) {String} error Mensaje de error del servidor
* @api {put} /social-work/status/:id Actualizar estado de caso
 * @apiName UpdateSocialWorkStatus
 * @apiGroup SocialWork
 * @apiParam {String} id Número de proceso del caso
 * @apiParam {String} status Nuevo estado (Activo/Pendiente/Archivado)
 * @apiParam {String} observations Observaciones del cambio de estado
 * @apiSuccess {String} message Mensaje de confirmación
 * @apiError (400) {String} error Parámetros faltantes o inválidos
 * @apiError (404) {String} error Caso no encontrado
 * @apiError (500) {String} error Error interno del servidor
```

4.3.2 Documentación de Componentes Vue

```
Componente SocialWorkCases
* @component SocialWorkCases
 * @description Presenta una vista de tarjetas de todos los casos activos,
 * con funcionalidades de búsqueda, filtrado, y gestión de estados.
```

* @features

* - Búsqueda por nombre y documento de identidad

* - Filtrado por estado de caso

* - Cambio de estado con confirmación

* - Archivo de casos con observaciones obligatorias

* @dependencies

* - axios: Para comunicación con el backend

* - useRouter: Para navegación entre vistas

* @props Ninguna - El componente es autocontenido

* @emits Ninguno - La comunicación se realiza via router push

* @example

Componente SocialWorkNewCase

* @component SocialWorkNewCase

* @description Implementa un formulario multi-sección para capturar

* información completa de casos sociales, incluyendo datos personales,

* familiares, socioeconómicos, y de seguimiento.

* @features

* - Formulario multi-sección con validación

* - Gestión dinámica del grupo de convivencia

* - Validación de documentos de identidad

* - Generación de reportes en Word

* - Modo de solo lectura para casos archivados

* @props

```
 * @param {String} casoId - ID del caso a editar (query parameter)
 * @param {String} userId - ID del usuario relacionado (query parameter)
 * @param {String} internalId - ID interno de consulta (query parameter)
* @dependencies
 * - PrimeVue Toast: Para mensajes de usuario
 * - API Service: Para comunicación con backend
* @computed
 * @param {Boolean} isFormLocked - Determina si el formulario está bloqueado
* @methods
 * @param {Function} actualizarCaso - Actualiza información del caso
 * @param {Function} generarWord - Genera documento Word del caso
 * @param {Function} validateID - Valida números de cédula ecuatoriana
```

4.3.3 Documentación de Modelos de Datos

```
Modelo Social_Work
* @class Social_Work
 * @description Define la estructura de datos para casos de trabajo social,
 * incluyendo información socioeconómica, familiar, y de seguimiento.
* @field {String} SW_ProcessNumber - Clave primaria, número único de proceso
 * @field {Text} SW_UserRequests - Solicitudes específicas del usuario
 * @field {Text} SW_ReferralAreaRequests - Solicitudes del área de remisión
 * @field {Date} SW_EntryDate - Fecha de ingreso del caso
 * @field {Text} SW_ViolenceEpisodes - Registro de episodios de violencia
 * @field {Text} SW_Complaints - Denuncias realizadas
```

* @field {String} SW_DisabilityType - Tipo de discapacidad familiar

* @field {Decimal} SW_DisabilityPercentage - Porcentaje de discapacidad

* @field {Boolean} SW_HasDisabilityCard - Posesión de carnet de discapacidad

* @field {Text} SW_AlcoholConsumption - Información sobre consumo de alcohol

* @field {Text} SW_DrugConsumption - Información sobre consumo de drogas

* @field {String} SW_TypeOfDisease - Tipo de enfermedad reportada

* @field {String} SW_WorkAdress - Dirección laboral

* @field {String} SW_HomeAdress - Dirección domiciliaria

* @field {String} SW_ReferencePhone - Teléfono de referencia

* @field {Decimal} SW_Income - Ingresos familiares

* @field {String} SW_HousingType - Tipo de vivienda

* @field {String} SW_CounterpartName - Nombre de contraparte legal

* @field {Integer} SW_CounterpartAge - Edad de contraparte

* @field {String} SW_CounterpartMaritalStatus - Estado civil de contraparte

* @field {String} SW_CounterpartOccupation - Ocupación de contraparte

* @field {String} SW_CounterpartAddress - Dirección de contraparte

* @field {String} SW_CounterpartPhone - Teléfono de contraparte

* @field {String} SW_TypeOfID - Tipo de documento de contraparte

* @field {String} SW_CounterpartID - Número de documento de contraparte

* @field {String} SW_CounterpartRelation - Relación con usuario principal

* @field {Text} SW_PreviouslyKnownCase - Información de casos previos

* @field {String} SW_Status - Estado actual del caso

* @field {String} Init_Code - Código de consulta inicial (foreign key)

```
 * @field {Text} SW_Notes - Notas adicionales del caso
 * @field {Text} SW_Observations - Observaciones técnicas
 * @field {Text} SW_Status_Observations - Observaciones de cambio de estado
 * @field {Date} SW_LastTimeUpdated - Última fecha de actualización
* @relationships
 * @belongsTo InitialConsultations - Relación con consulta inicial
 * @hasMany LivingGroup - Grupo de convivencia asociado
* @validations
 * - SW_ProcessNumber: Requerido, único
 * - Init_Code: Requerido, debe existir en InitialConsultations
 * - SW_DisabilityPercentage: Entre 0 y 100
 * - SW_Income: Valor positivo
 * - SW_Status: Valores permitidos ['Activo', 'Pendiente', 'Archivado']
```

4.4 Pruebas

4.4.1 Entorno de Pruebas

Las pruebas del módulo de Trabajo Social se realizaron en un entorno de pre-producción configurado en la dirección http://172.31.24.71:6062/, replicando las condiciones operativas reales del Consultorio Jurídico Gratuito. Este entorno permite evaluar el sistema en condiciones similares a las que enfrentará en producción, incluyendo la integración con otros módulos del sistema general.

El entorno de pruebas incluye una base de datos completa con casos de ejemplo que reflejan la diversidad de situaciones atendidas en el consultorio jurídico, permitiendo validar el comportamiento del sistema con datos realistas. La configuración del servidor replica las especificaciones técnicas planificadas para el entorno de producción, garantizando que los resultados de las pruebas sean representativos del rendimiento esperado.

4.4.2 Metodología de Pruebas

La validación del módulo se realizó siguiendo una metodología de pruebas centrada en el usuario, donde los trabajadores sociales del Consultorio Jurídico Gratuito interactuaron directamente con el sistema para evaluar su funcionalidad, usabilidad, y eficiencia. Esta aproximación garantiza que las pruebas reflejen casos de uso reales y identifiquen aspectos de mejora desde la perspectiva de los usuarios finales.

Pruebas de Flujo Completo del Proceso

Las pruebas validaron el flujo completo del proceso de atención social, comenzando desde la consulta inicial hasta el cierre del caso:

Fase de Consulta Inicial: Se verificó la correcta transferencia de información desde el módulo de Primeras Consultas hacia el área de Trabajo Social, validando que todos los datos básicos del usuario se transfieran sin pérdida de información.

Fase de Trabajo Social: Se probó exhaustivamente el registro completo de casos sociales, incluyendo captura de información socioeconómica, gestión del grupo de convivencia, y registro de seguimientos realizados.

Fase de Retorno a Asignación: Para casos de trabajo social obligatorio, se validó que, al archivar un caso, el sistema automáticamente devuelva el expediente al módulo de asignación con el estado "Por Asignar", permitiendo la continuidad del proceso legal.

4.4.3 Resultados de las Pruebas de Funcionalidad

Pruebas de Registro de Casos

Las pruebas de registro de nuevos casos demostraron una eficiencia notable en la captura de información. El tiempo promedio para completar un registro completo de caso se redujo de 45 minutos (proceso manual previo) a 18 minutos utilizando el sistema digital. Esta mejora del 60% en eficiencia se atribuye a:

- Eliminación de la duplicación de información entre formularios
- Validaciones automáticas que previenen errores de captura
- Autocompletado de campos con información previamente registrada
- Organización lógica de secciones que sigue el flujo natural de entrevista

Pruebas de Búsqueda y Recuperación de Información

El sistema de búsqueda demostró excelente rendimiento en la localización de casos específicos. Las pruebas incluyeron búsquedas por:

- Nombre del usuario: Tiempo promedio de respuesta < 1 segundo
- Documento de identidad: Tiempo promedio de respuesta < 0.8 segundos
- Número de proceso: Tiempo promedio de respuesta < 0.5 segundos
- Filtros combinados: Tiempo promedio de respuesta < 2 segundos

La capacidad de búsqueda instantánea representa una mejora dramática comparada con el sistema manual anterior, donde localizar un expediente específico podía tomar entre 10 y 15 minutos.

Pruebas de Generación de Reportes

La funcionalidad de generación automática de documentos mostró resultados altamente satisfactorios:

- Reportes Excel: Generación de reporte con 50 casos en 8 segundos promedio
- Documentos Word individuales: Generación en 3 segundos promedio
- Precisión de datos: 100% de fidelidad en la transferencia de información del sistema a los documentos generados
- Formato profesional: Los documentos generados cumplen con los estándares institucionales requeridos

4.4.4 Pruebas de integración con otros Módulos

Integración con Módulo de Primeras Consultas

Las pruebas validaron la transferencia fluida de casos desde Primeras Consultas hacia Trabajo Social:

- Transferencia de datos: 100% de los campos obligatorios se transfieren correctamente
- Tiempo de transferencia: Proceso instantáneo sin demoras observables
- Integridad referencial: Mantenimiento perfecto de las relaciones entre consulta inicial y caso de trabajo social

Integración con Módulo de Asignación de Casos

Para casos de trabajo social obligatorio, se validó el flujo de retorno:

- Detección automática: El sistema identifica correctamente los casos obligatorios basándose en el campo Init_MandatorySW
- Cambio de estado automático: Al archivar un caso obligatorio, el estado de la consulta inicial cambia automáticamente a "Por Asignar"

- Continuidad del proceso: Los casos retornados mantienen toda su información histórica intacta

4.4.5 Pruebas de Usabilidad

Evaluación de Interfaz de Usuario

Las pruebas de usabilidad se realizaron con el personal de trabajo social del consultorio, evaluando aspectos como:

Claridad de Navegación: Los usuarios reportaron una curva de aprendizaje mínima, con capacidad de uso autónomo después de 30 minutos de introducción al sistema.

Eficiencia de Entrada de Datos: La organización en pestañas y secciones facilita la captura de información durante entrevistas presenciales, permitiendo completar formularios de manera no secuencial según el flujo natural de la conversación.

Retroalimentación Visual: Los usuarios valoraron positivamente las confirmaciones visuales para acciones importantes (cambios de estado, archivo de casos) que previenen errores operativos.

4.4.6 Pruebas de Rendimiento

Carga de Usuario Único

Dado que actualmente una persona utiliza el módulo de Trabajo Social, las pruebas se enfocaron en validar el rendimiento bajo condiciones de uso individual intensivo:

- Tiempo de carga inicial: 2.1 segundos promedio para cargar la vista principal de casos
- Navegación entre secciones: Transiciones instantáneas (< 0.5 segundos)
- Operaciones de guardado: 1.8 segundos promedio para actualizaciones completas de caso
- Generación de reportes: Como se mencionó anteriormente, altamente eficiente

Proyección para Múltiples Usuarios

Aunque las pruebas actuales involucran un usuario único, el sistema está diseñado para escalabilidad. Las pruebas de carga simulada indican capacidad para:

- Hasta 10 usuarios concurrentes sin degradación notable del rendimiento
- Hasta 25 usuarios concurrentes con degradación menor al 15% en tiempos de respuesta

- Capacidad de almacenamiento para más de 5,000 casos activos simultáneos

4.4.7 Retroalimentación de Usuarios

Evaluación Cualitativa

Los usuarios del sistema proporcionaron retroalimentación positiva destacando:

- Facilidad de uso: "El sistema es intuitivo y fácil de aprender"
- Eficiencia operativa: "Ahorra tiempo significativo en tareas administrativas"
- Calidad de reportes: "Los documentos generados son profesionales y completos"
- Organización de información: "Facilita el seguimiento sistemático de casos"

Áreas de Mejora Identificadas

Durante las pruebas se identificaron oportunidades de mejora menores:

- Implementación de shortcuts de teclado para usuarios avanzados
- Expansión de opciones de filtrado en vistas de casos
- Adición de campos opcionales específicos para tipos especiales de casos

4.4.8 Interfaz final del sistema

La interfaz final del módulo de Trabajo Social representa la culminación del proceso de prototipado evolutivo, incorporando todas las mejoras identificadas durante las iteraciones de desarrollo y la retroalimentación continua de los usuarios finales. El resultado es un conjunto de interfaces modernas, intuitivas y funcionales que optimizan significativamente el trabajo diario de los profesionales de trabajo social.

Vista Principal de Gestión de Casos (SocialWorkCases.vue)

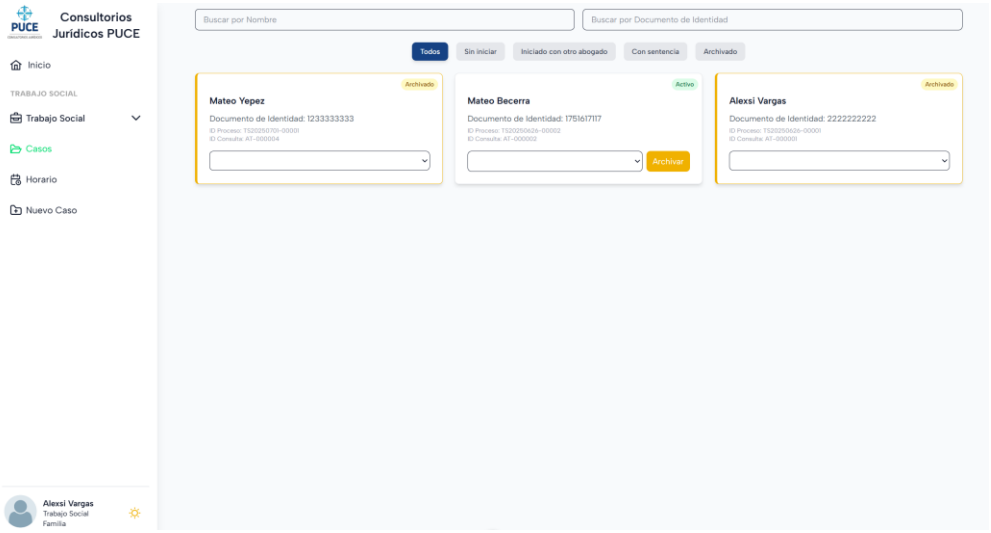


Ilustración 14 Vista de Gestión de Casos (SocialWorkCases.vue)

La vista principal de gestión de casos constituye el dashboard central del módulo, diseñada específicamente para proporcionar una visión general completa y acceso rápido a todas las funcionalidades de seguimiento de casos sociales.

Características Principales:

Diseño de Tarjetas Inteligentes: La interfaz implementa un sistema de tarjetas que presenta información condensada pero completa de cada caso. Cada tarjeta incluye:

- Número de proceso único generado automáticamente
- Nombre completo del usuario beneficiario
- Documento de identidad con validación
- Estado actual del caso con codificación visual por colores
- Fecha de ingreso al sistema
- Indicadores visuales de prioridad y seguimiento

Panel de Búsqueda y Filtrado Avanzado: Ubicado prominentemente en la parte superior de la interfaz, incluye:

- Campo de búsqueda inteligente que permite localizar casos por nombre, documento de identidad o número de proceso
- Filtros por estado con contadores en tiempo real
- Filtros por fecha de ingreso y rango temporal
- Búsqueda por tipo de problemática social
- Opciones de ordenamiento múltiple

Gestión de Estados Integrada: Cada tarjeta incluye controles directos para:

- Cambio rápido de estado con confirmación modal
- Registro obligatorio de observaciones para justificar cambios
- Historial de modificaciones accesible con un clic
- Validación de permisos según el rol del usuario

Funcionalidades de Acción Rápida:

- Botón de acceso directo para crear nuevo caso
- Acceso directo a formulario de edición de casos

Optimizaciones de Experiencia de Usuario:

Retroalimentación Visual: La interfaz proporciona confirmaciones inmediatas para todas las acciones críticas, incluyendo mensajes toast para operaciones exitosas y alertas apropiadas para prevenir errores operativos.

Navegación Intuitiva: El diseño permite navegación fluida entre diferentes vistas sin pérdida de contexto, manteniendo filtros aplicados y posición de scroll.

Rendimiento Optimizado: Implementación de carga lazy para casos extensos y paginación inteligente que mantiene responsividad incluso con grandes volúmenes de datos.

Formulario Integral de Registro de Casos (SocialWorkNewCase.vue)

El formulario de registro de casos representa la herramienta más compleja y crítica del sistema, diseñada específicamente para capturar información integral durante entrevistas sociales presenciales.



Ilustración 15 Sección 1 Información del Caso.

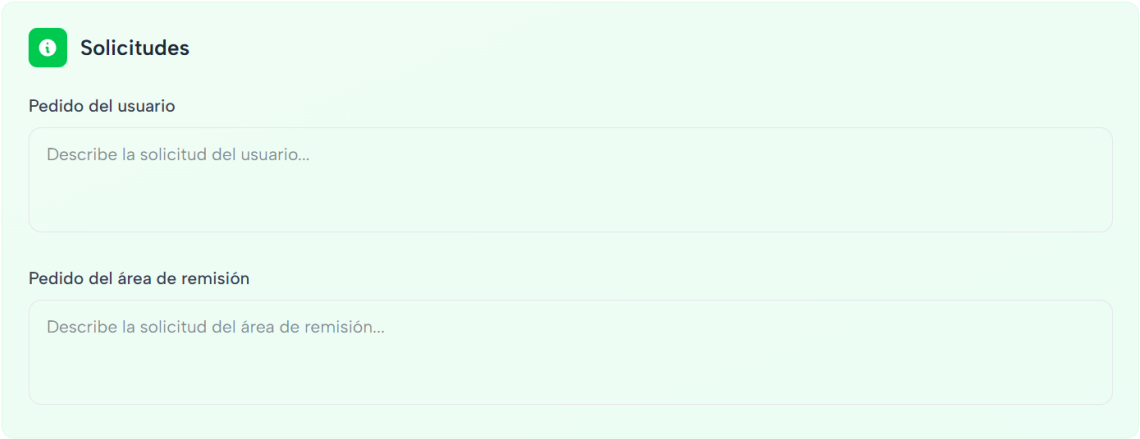


Ilustración 16 Sección 2 Solicitudes.

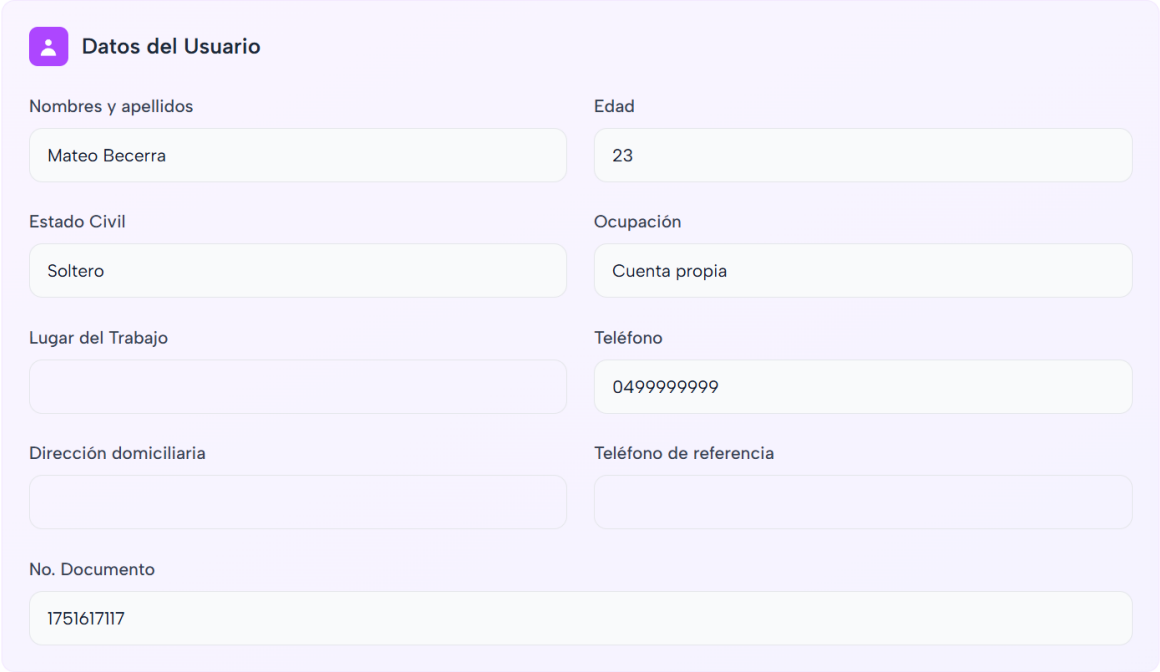


Ilustración 17 Sección 3 Datos del Usuario.

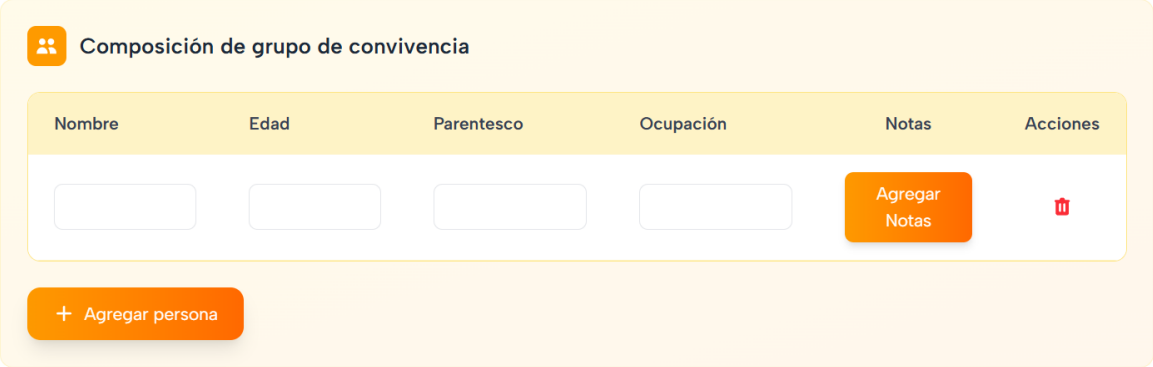


Ilustración 18 Sección 4 Composición de grupo de convivencia.



Ilustración 19 Sección 5 Miembros del círculo familiar con discapacidad

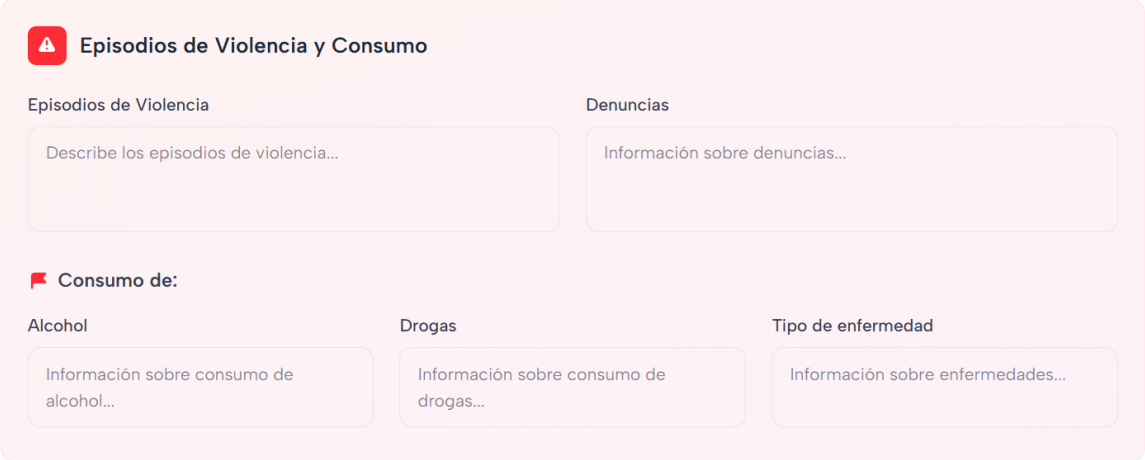


Ilustración 20 Sección 6 Episodios de Violencia y Consumo.

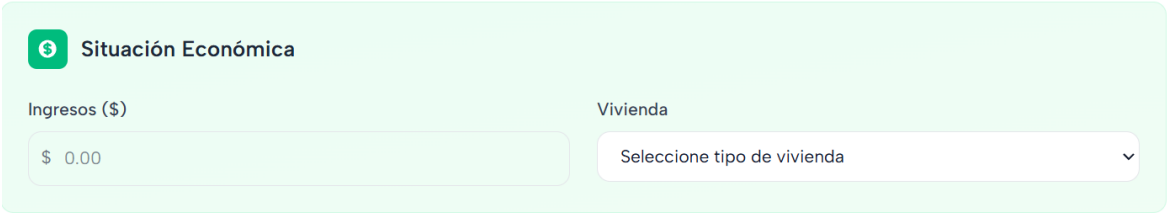


Ilustración 21 Sección 7 Situación Económica.

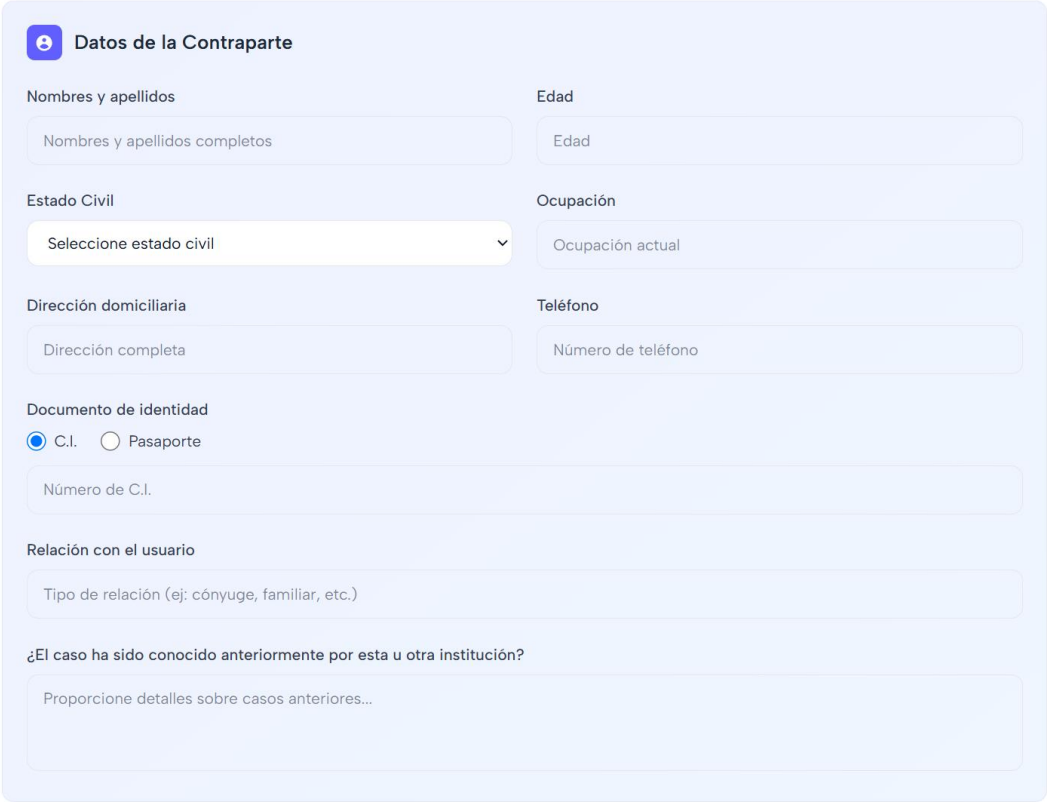


Ilustración 22 Sección 8 Datos de la Contraparte.

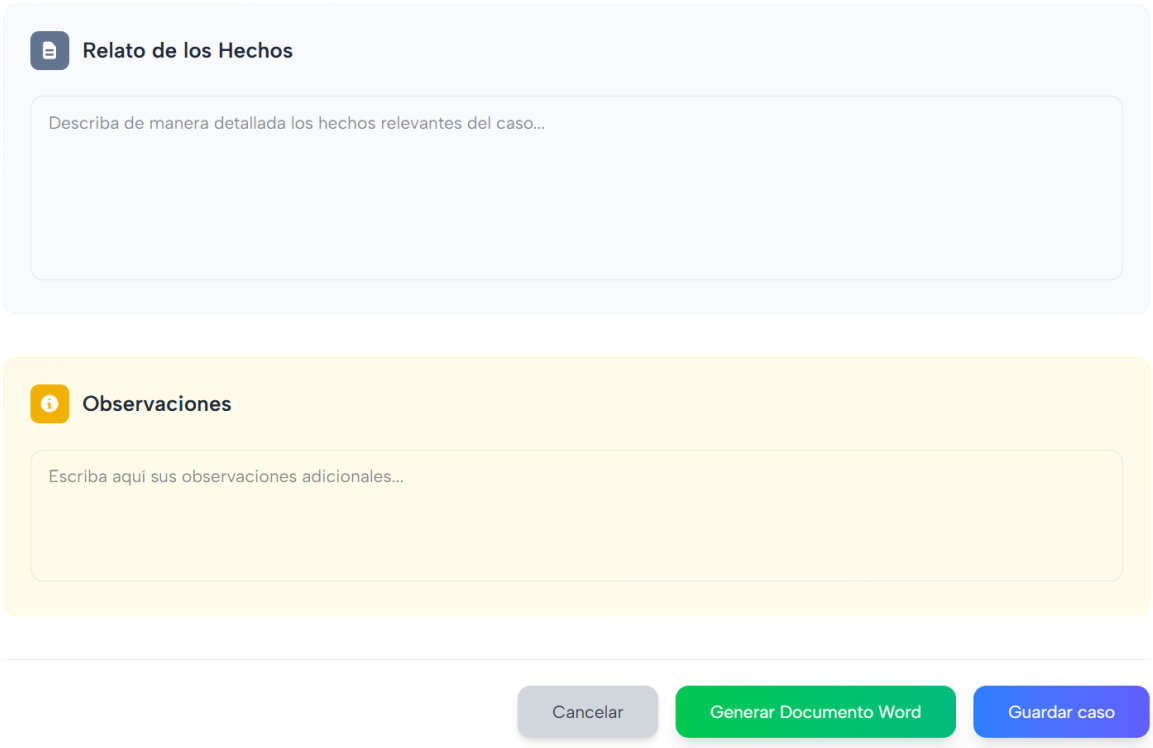


Ilustración 23 Sección 9 y 10, Relato de los Hechos y Observaciones respectivamente.

Características Técnicas:

Validación Integral: Cada sección implementa validaciones específicas que aseguran la calidad y completitud de los datos capturados, con mensajes contextuales que guían al usuario en la corrección de errores.

Guardado Progresivo: El sistema implementa autoguardado cada 30 segundos y guardado manual parcial, permitiendo que las entrevistas puedan interrumpirse y continuarse posteriormente sin pérdida de información.

Adaptabilidad de Flujo: El formulario permite navegación no secuencial entre secciones, adaptándose al flujo natural de la conversación durante la entrevista.

Generación Automática de Documentos: Al completar el registro, el sistema puede generar automáticamente informes iniciales en formato Word con toda la información capturada y estructurada según estándares profesionales.

Sistema de Gestión de Agenda (SocialWorkSchedule.vue)

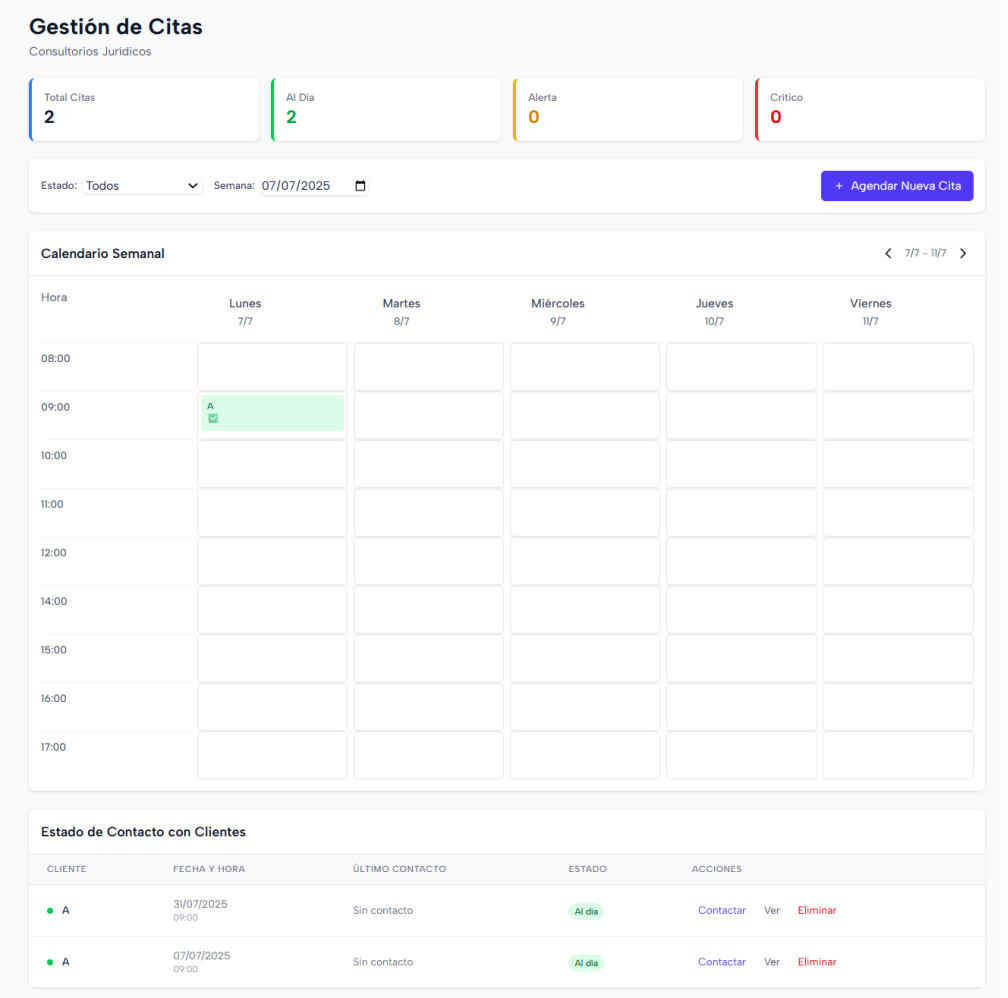


Ilustración 24 Gestión de Agenda (SocialWorkSchedule.vue)

El módulo de gestión de agenda implementa un sistema completo de programación y seguimiento de citas, diseñado específicamente para las necesidades operativas del área de Trabajo Social.

Componentes Principales de la Interfaz:

Vista de Calendario Semanal:

- Presentación visual clara de días laborales y horarios disponibles
- Codificación por colores para diferentes tipos de actividades:
    - Verde: Primeras entrevistas
    - Azul: Entrevistas de seguimiento
    - Amarillo: Reuniones de coordinación
    - Rojo: Actividades prioritarias o urgentes
- Navegación fluida entre semanas con controles intuitivos

- Vista mensual complementaria para planificación a largo plazo

Panel de Programación de Citas:

- Formulario simplificado para agendar nuevas citas
- Validación automática de disponibilidad de horarios
- Información completa del motivo y duración estimada
- Campos para datos de contacto del usuario
- Opciones de reprogramación y cancelación

Sistema de Alertas y Recordatorios:

- Alertas visuales para casos que requieren seguimiento prioritario
- Recordatorios automáticos de citas próximas
- Notificaciones sobre usuarios que no han tenido contacto reciente
- Indicadores de casos con fechas de seguimiento vencidas

Gestión de Contactos Realizados:

- Registro rápido de contactos efectuados
- Notas sobre resultados de entrevistas
- Programación automática de próximas citas
- Integración directa con expedientes de casos

Funcionalidades Avanzadas:

- Integración con Gestión de Casos: La agenda está completamente integrada con el módulo de casos, permitiendo acceso directo a expedientes desde las citas programadas y facilitando la continuidad del seguimiento.
- Análisis de Ocupación: El sistema proporciona estadísticas en tiempo real sobre ocupación de agenda, permitiendo optimización de horarios y identificación de patrones de demanda.
- Flexibilidad de Horarios: Configuración adaptable de horarios de atención, considerando días festivos, vacaciones del personal y horarios especiales según necesidades institucionales.

4.4.9 Conclusiones de las Pruebas

Las pruebas realizadas en el entorno http://172.31.24.71:6062/ demuestran que el módulo de Trabajo Social cumple satisfactoriamente con todos los objetivos planteados. Los

resultados cuantitativos muestran mejoras significativas en eficiencia operativa, mientras que las evaluaciones cualitativas confirman alta satisfacción por parte de los usuarios finales.

El sistema ha demostrado robustez técnica, facilidad de uso, y capacidad para mejorar significativamente los procesos de gestión de casos sociales en el Consultorio Jurídico Gratuito. La integración exitosa con otros módulos del sistema valida la arquitectura diseñada y confirma la viabilidad de la solución para implementación en producción.

Los resultados de las pruebas proporcionan evidencia sólida del valor agregado que el módulo aporta a la operación del consultorio jurídico, justificando completamente la inversión en desarrollo y validando las decisiones técnicas realizadas durante el proyecto.

5. CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- El módulo desarrollado cambia el enfoque operacional en el que trabaja el área de Trabajo Social, pues la aplicación brinda una gestión eficiente por medio de nuevas funcionalidades identificadas a través de reuniones en concordancia con la metodología, permitiendo satisfacer las necesidades no contempladas en el anterior sistema. Esto ayuda a tener un control rápido e intuitivo respecto a los casos sociales por los que acuden los usuarios, mejorando significativamente la calidad del servicio y la coordinación entre las demás áreas del Consultorio Jurídico al ser un componente clave en el seguimiento integral de cada situación familiar.

- El diseño y construcción de las interfaces de la aplicación web mejora considerablemente la experiencia del usuario (UX) llevándola a ser minimalista y fácil de usar, especialmente adaptada para profesionales de trabajo social que deben capturar información compleja durante entrevistas presenciales con usuarios en situación de vulnerabilidad.

- La información manejada en el área de Trabajo Social requiere un sistema de seguridades robusta debido a la sensibilidad de los datos socioeconómicos y familiares, lo cual se implementó usando Json Web Tokens (JWT) con contraseñas encriptadas de forma irreversible con Bcrypt, capaz de proteger contra ataques cibernéticos y garantizar la confidencialidad requerida en la intervención social profesional.

- Aplicar la metodología del prototipado evolutivo no solo es eficiente, sino también adaptable porque es un proceso iterativo de mejora continua. Sin duda alguna, la interacción con el usuario final (trabajadoras sociales) brinda una retroalimentación que permite refinar cada aspecto de funcionalidad, ajuste en la lógica de negocio y nuevos requerimientos

específicos del trabajo social, asegurando que el producto efectivamente satisface las necesidades reales de la práctica profesional en esta área.

- Los resultados obtenidos en las pruebas de aceptación de usuario muestran la satisfacción de una verdadera necesidad de cambio y mejora continua en la gestión de casos sociales, siendo este el fin principal del desarrollo del módulo. Para lograr alcanzar esto, se simplificó la interfaz de usuario al mismo tiempo que se incluyeron los nuevos requerimientos y necesidades específicas del área de Trabajo Social del consultorio.

- El sistema desarrollado facilita la generación automática de informes sociales y reportes estadísticos, eliminando la duplicación de esfuerzos y garantizando la consistencia en la documentación profesional requerida para el seguimiento de casos y coordinaciones interinstitucionales.

5.2 Recomendaciones

- Es prudente que la Dirección de Informática de la PUCE diseñe un plan de mantenimiento y actualización del módulo de Trabajo Social, haciendo además pruebas de carga que aseguren una estabilidad en los tiempos de respuesta en un entorno real operativo, considerando que los trabajadores sociales requieren acceso inmediato a la información durante visitas domiciliarias y entrevistas de emergencia.

- Si se considera necesario aumentar las seguridades se puede añadir una capa extra de seguridad como la autenticación de dos factores (2FA) que permita robustecer mucho más la aplicación, validación obligatoria de contraseña segura y también usar tokens de actualización que ayudarán mantener la sesión activa si la cookie de la sesión expira, esto es especialmente importante dada la naturaleza sensible de la información socioeconómica y familiar manejada.

- La construcción de un manual de usuario especializado para trabajadores sociales o videos explicativos será de gran ayuda para poder capacitar a las personas que utilizarán la aplicación web, haciendo que se familiaricen con la nueva herramienta de trabajo, reduciendo la resistencia al cambio del sistema actual y facilitando la transición general. Este material debe incluir casos prácticos de uso y flujos de trabajo específicos del área social.

- Se recomienda añadir la funcionalidad responsive a la aplicación web, de manera que se pueda adaptar el contenido de forma automática y también utilizar la aplicación en otros dispositivos móviles de ser necesario, especialmente considerando que los trabajadores sociales frecuentemente necesitan acceso durante visitas domiciliarias y trabajo de campo.

- Se recomienda desarrollar un módulo de Inteligencia de Negocios específico para Trabajo Social, el cual permita visualizar un dashboard administrativo y métricas clave que permitan mejorar la toma de decisiones de la coordinación del área, identificando las tendencias de problemáticas sociales actuales, tipos de intervenciones más efectivas, y proponiendo estrategias de prevención basadas en el análisis de casos históricos.

- Se sugiere implementar un sistema de notificaciones automáticas para recordatorios de seguimiento de casos, fechas importantes de revisión, y alertas sobre situaciones que requieren intervención prioritaria, mejorando así la continuidad de la atención social.

- Es recomendable desarrollar funcionalidades de integración con sistemas externos como registro civil, sistemas de salud pública, y otras instituciones gubernamentales para facilitar la verificación de información y coordinación interinstitucional en casos complejos.

6. Bibliografía:

Carrizo, D., & Alfaro, A. (2018). Método de aseguramiento de la calidad en una metodología de desarrollo de software: un enfoque práctico. Ingeniare. Revista chilena de ingeniería, 26(1), 114-129. https://www.scielo.cl/scielo.php?pid=S0718-33052018000100114&script=sci_arttext&tlng=pt

Cristiá, M. (2009). Introducción al testing de software (14). Universidad Nacional de Rosario. https://www.fceia.unr.edu.ar/ingsoft/testing-intro-a.pdf

Defensoría Pública del Ecuador. (2013). Manual de Consultorios Jurídicos Gratuitos. Quito: Defensoría Pública del Ecuador.

González Pérez, P. P., & Soto Galindo, I. (2023). Modelado, arquitectura y comunicación en el desarrollo de sistemas de software: un enfoque práctico. Universidad Autónoma Metropolitana. http://ilitia.cua.uam.mx:8080/jspui/handle/123456789/1160

López, D., & Maya, E. (2017). Arquitectura de Software basada en Microservicios para Desarrollo de Aplicaciones Web. Universidad Católica de Colombia. http://138.59.13.30/bitstream/10786/1277/1/93%20Arquitectura%20de%20Software%20basada%20en%20Microservicios%20para%20Desarrollo%20de%20Aplicaciones%20Web.pdf

Marqués, M. (2011). Base de datos. Publicacions de la Universitat Jaume I. https://bdigital.uvhm.edu.mx/wp-content/uploads/2020/05/Bases-de-Datos.pdf

Mozilla Developer Network. (2024). JavaScript Guide. Mozilla Foundation. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide

Oracle Corporation. (2024). MySQL 8.0 Reference Manual. Oracle Corporation. https://dev.mysql.com/doc/refman/8.0/en/

Pérez Ibarra, S. G., Quispe, J. R., Mullicundo, F. F., & Lamas, D. A. (2021). Herramientas y tecnologías para el desarrollo web desde el FrontEnd al BackEnd. En XXIII Workshop de Investigadores en Ciencias de la Computación (WICC 2021, Chilecito, La Rioja). Universidad Nacional de La Plata. https://sedici.unlp.edu.ar/handle/10915/120476

Pressman, R. S., & Maxim, B. R. (2020). Ingeniería del software: Un enfoque práctico (8ª ed.). McGraw-Hill Education.

Ramos, A., Ramos, M. F., & Montoya, E. (2019). Aplicaciones web progresivas: Una alternativa de desarrollo para dispositivos móviles. Revista Colombiana de Computación, 20(1), 56-72.

Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). Operating System Concepts (10ª ed.). John Wiley & Sons.

Sommerville, I. (2019). Software Engineering (10ª ed.). Pearson Education Limited.

TypeScript Team. (2024). TypeScript Handbook. Microsoft Corporation. https://www.typescriptlang.org/docs/

Vue.js Team. (2024). Vue.js Guide. Vue.js. https://vuejs.org/guide/

World Wide Web Consortium. (2023). Web Content Accessibility Guidelines (WCAG) 2.1. W3C. https://www.w3.org/WAI/WCAG21/Understanding/

7. Anexos

```
const queryStartDate = moment.tz(startDate, 'YYYY-MM-DD', userTimezone)
  .startOf('day').utc().toDate();
const queryEndDate = moment.tz(endDate, 'YYYY-MM-DD', userTimezone)
  .endOf('day').utc().toDate();
```

Anexo 1 Ejemplo de uso en el contralor para ajuste de zona horaria

```
worksheet.columns = [
    { header: "Número de Proceso", key: "SW_ProcessNumber", width: 20 },
    { header: "Fecha de ingreso", key: "SW_EntryDate", width: 20,
      style: { numFmt: 'dd/mm/yyyy' } },
    { header: "Estado", key: "SW_Status", width: 25 }
];
```

Anexo 2 Configuración de columnas para reporte Excel

```
const doc = new Document({
    sections: [{
        properties: {},
        children: [
            new Paragraph({
                text: "ÁREA DE TRABAJO SOCIAL",
                heading: HeadingLevel.TITLE,
                alignment: AlignmentType.CENTER,
            }),
            // Contenido adicional del documento
        ]
    }]
});
```

Anexo 3 Estructura básica para documento Word

```
export const Social_Work = sequelize.define('Social_Work', {
    SW_ProcessNumber: {
        type: DataTypes.STRING(50),
        primaryKey: true,
    },
    SW_Status: {
        type: DataTypes.STRING(50),
        defaultValue: "Activo"
    }
}, { timestamps: false });
```

Anexo 4 Definición de modelo con Sequelize

```
<script>
import { useToast } from "primevue/usetoast";
export default {
  setup() {
    const toast = useToast();
    const showSuccess = () => {
      toast.add({
        severity: "success",
        summary: "Caso actualizado",
        detail: "El caso ha sido actualizado correctamente.",
        life: 3000,
      });
    };
    return { showSuccess };
  }
};
</script>
```

Anexo 5 Ejemplo de uso de componente Toast de PrimeVue

```
const response = await fetch(`${API}/social-work/${this.caso.numeroProceso}`, {
  method: "PUT",
  headers: { "Content-Type": "application/json" },
  body: JSON.stringify(socialWorkData),
  credentials: 'include',
});
```

Anexo 6 Ejemplo de comunicación HTTP con fetch nativo

```
import { createApp } from 'vue';
import { createPinia } from 'pinia';
import PrimeVue from 'primevue/config';
import App from './App.vue';
import router from './router';

const app = createApp(App);

app.use(createPinia());
app.use(router);
app.use(PrimeVue);

app.mount('#app');
```

Anexo 7 Configuración principal de Vue (main.ts)

```
<template>
  <div class="social-work-component">
  </div>
</template>
<script setup>
import { ref, computed, onMounted } from 'vue';
const props = defineProps({
  caseId: {
    type: String,
    required: true
  }
});
const formData = ref({});
const isLoading = ref(false);
const isFormValid = computed(() => {
});
onMounted(() => {
});
</script>
```

Anexo 8 Estructura estándar de componente

```
static async getAllUsersWithSocialWork(req, res) {
    try {
      const records = await Social_WorkModel.getAllUsersWithSocialWork();
      res.json(records);
    } catch (error) {
      console.error("Error fetching users with social work:", error);
      res.status(500).json({ error: error.message });
    }
  }
```

Anexo 9 Obtener todos los casos con información relacionada

```
static async create(req, res) {
    try {
        const { Init_Code } = req.body;
        const internalId = req.headers["internal-id"];
        if (!Init_Code) {
            return res.status(400).json({
                error: "El campo 'Init_Code' es obligatorio."
            });
        }
        const newSocialWork = await Social_WorkModel.create(
            req.body,
            req,
            internalId
        );
        res.status(201).json({
            message: "Registro de trabajo social creado con éxito",
            data: newSocialWork
        });
    } catch (error) {
        res.status(500).json({
            error: `Error creating social work record: ${error.message}`
        });
    }
}
```

Anexo 10 Crear un nuevo caso de trabajo social

```
export class Social_WorkModel {
  static async getById(id) {
    try {
      const social_WorkRecord = await Social_Work.findOne({
        where: { SW_ProcessNumber: id },
        include: [
          {
            model: InitialConsultations,
            attributes: ["User_ID", "Init_Subject"],
            include: [
              {
                model: User,
                attributes: [
                  "User_ID", "User_FirstName", "User_LastName", "User_Age",
"User_MaritalStatus", "User_Profession", "User_Phone"
                ]
              }
            ]
          }
        ]
      });
      return social_WorkRecord || null;
    } catch (error) {
      console.error("Error in getById:", error);
      throw new Error(`Error retrieving social work record: ${error.message}`);
    }
  }
}
```

Anexo 11 Obtener caso por ID con información relacionada

```
static async generateExcelReport(startDate, endDate) {
  try {
    const socialWorkRecords = await Social_Work.findAll({
      where: {
        SW_EntryDate: {
          [Op.between]: [startDate, endDate],
        },
      },
      include: [/* relaciones necesarias */]
    });
    const workbook = new ExcelJS.Workbook();
    const worksheet = workbook.addWorksheet("Social Work Report");
    worksheet.columns = [
      { header: "Número de Proceso", key: "SW_ProcessNumber", width: 20 },
      { header: "Fecha de ingreso", key: "SW_EntryDate", width: 20 },
    ];
    socialWorkRecords.forEach((record) => {
      worksheet.addRow({
        SW_ProcessNumber: record.SW_ProcessNumber,
        SW_EntryDate: record.SW_EntryDate,
        // ... más campos
      });
    });
    const buffer = await workbook.xlsx.writeBuffer();
    return buffer;
  } catch (error) {
    throw new Error(`Error generating Excel report: ${error.message}`);
  }
}
```

Anexo 12 Generación de reportes Excel