



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE CIENCIAS

"Nuevo Módulo de Seguimiento y
Administración de Mecanismos Alternativos de
Solución de Controversias"

**REPORTE DE TRABAJO
PROFESIONAL**

QUE PARA OBTENER EL TÍTULO DE:
**LICENCIADO EN CIENCIAS DE LA
COMPUTACIÓN**

PRESENTA:
OMAR NERIA PÉREZ

TUTOR:

DR. MIGUEL EHÉCATL MORALES TRUJILLO

CIUDAD UNIVERSITARIA, CDMX. 2017





Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

CONTENIDO

- 1. Introducción 5
 - 1.1. Panorama general 5
 - 1.2. Objetivo 6
 - 1.3. Estructura del trabajo..... 6
- 2. Antecedentes 8
 - 2.1. Contexto del Consejo de la Judicatura Federal 8
 - 2.2. Actividades del Consejo de la Judicatura Federal 8
 - 2.3. Dirección General de Tecnologías de la Información..... 9
 - 2.3.1. Actividades de la Coordinación de Sistemas Jurídicos dentro de la DGTI..... 10
 - 2.4. Sistema Integral de Seguimiento de Expedientes 11
 - 2.5. Glosario 11
- 3. Módulo de seguimiento de Mecanismos Alternativos de Solución de Controversias..... 13
 - 3.1. Historias de Usuario 13
 - 3.2. Levantamiento de Requerimientos 14
 - 3.3. Diseño..... 22
 - 3.4. Construcción..... 25
 - 3.4.1. Construcción de base de datos y creación de objetos 25
 - 3.4.2. Construcción de objetos y uso de conceptos de Programación Orientada a Objetos.... 28
 - 3.4.3. Construcción de capa de presentación 30
 - 3.5. Pruebas..... 32
 - 3.6. Liberación 32
- 4. Resultados 34
- 5. Conclusiones..... 36
- Bibliografía 38

1. INTRODUCCIÓN

1.1. Panorama general

En el sistema judicial mexicano, en materia penal, existen diversos criterios con base en los cuales un delito, que se le imputa a un ciudadano, puede ser susceptible de alguna pena que conlleve la privación de su libertad, además de una multa, trabajos comunitarios, etc. En particular, la privación de la libertad es la pena más alta a la que un criminal puede ser acreedor en nuestro sistema judicial (sin considerar el número de años que dicte la sentencia), y representa el punto clave para el nuevo sistema de justicia en materia penal.

En los casos en que la libertad está de por medio, y si el juez otorga su visto bueno, es posible aplicar un mecanismo alternativo que repare el daño causado y que convenga o sea aceptable por el perjudicado. Evitando de esta forma que el conflicto derive en un juicio, y este a su vez en una sentencia que privativa de la libertad. Éstas son el tipo de situaciones contempladas en la nueva Ley Nacional de Mecanismos Alternativos de Solución de Controversias, la cual fue publicada en el Diario Oficial de la Federación el 29 de diciembre de 2014.¹

Un ejemplo concreto podría ser el siguiente: supongamos que el administrador de una empresa es acusado de fraude por desvío de capital. Si la pena que le corresponde, según las leyes y regulaciones aplicables (y así las dictamina el juez), implican algunos años en reclusión, entonces el imputado podría intentar llegar a un acuerdo a través del diálogo, por escrito o en una junta de mediación con el defraudado.

De esta manera el imputado puede comprometerse a pagar una suma considerable de dinero o bien devolver la cantidad de dinero desviada. Si un acuerdo de esta naturaleza satisface a las dos partes, entonces es obligación del imputado cumplir en tiempo y forma con dicho acuerdo, a menos de que el juez mediador permita la postergación de su cumplimiento. En caso de que el imputado incumpla, se procede directamente al inicio del juicio en materia penal que corresponda al conflicto, pudiendo derivar en una sentencia definitiva.

Para dar cumplimiento a esta ley, es necesario contar con una plataforma que permita el registro y consulta de toda la información que constituye a un mecanismo alternativo, así como los detalles que identifiquen tanto los procesos previos como los posteriores a la celebración y cumplimiento del acuerdo reparatorio, así como una funcionalidad que permita consultar el historial de acuerdos reparatorios a los que ha accedido un ciudadano, ya sea en calidad de víctima o de imputado. Por ejemplo, cuando a un ciudadano ya cuenta en su historial con algunos acuerdos reparatorios, queda a la decisión de un juez si el sujeto tendrá la oportunidad de celebrar otro.

¹http://www.dof.gob.mx/nota_detalle.php?codigo=5377563&fecha=29/12/2014

1.2. Objetivo

El desarrollo que se presenta en este reporte muestra la manera en que los conocimientos que adquirí en la Facultad de Ciencias me permiten desempeñarme en el desarrollo de software confiable y robusto que da solución a necesidades actuales y relevantes. Aspectos como el diseño e implementación de bases de datos, el paradigma de programación orientado a objetos, la ingeniería de software y, sobre todo, el nivel de abstracción con que se imparten dichos temas en la Facultad de Ciencias, me han proporcionado la generalidad y amplitud suficiente para adoptar nuevos lenguajes de programación, metodologías de trabajo o tecnologías innovadoras con una curva de aprendizaje bastante aceptable.

Con lo anterior, me ha sido posible participar activamente en el diseño, implementación y desarrollo de los sistemas que proporcionan infraestructura a los procesos dentro del Poder Judicial de la Federación.

En particular, el nuevo módulo de Mecanismos Alternativos de Solución de Controversias (MASC), cuyo ciclo de vida se presenta enteramente en este trabajo. Puntualmente la presentación de este módulo es el objetivo de este trabajo. Se hace una presentación de la forma en que se levantan requerimientos en mi ámbito laboral, la parte de diseño tanto de la base de datos como la arquitectura de la aplicación, así como el proceso de codificación, la fase de pruebas y la etapa de liberación.

1.3. Estructura del trabajo

En este trabajo presento los alcances del desarrollo de software que realicé en la Dirección General de Tecnologías de la Información (DGTI) del Consejo de la Judicatura Federal (CJF). En éste documento el proceso de creación del software a lo largo de las etapas de levantamiento de requerimientos, diseño y creación de base de datos, codificación del sistema, pruebas de los requerimientos desarrollados y solución de incidencias arrojadas en la etapa de pruebas. De la misma forma, describo el proceso de liberación de este desarrollo.

Del capítulo 1 se podrá obtener un contexto jurídico y administrativo en que está envuelto el proyecto, conociendo de las actividades principales del Consejo de la Judicatura Federal dentro del Poder Judicial de la Federación. También se menciona el entorno en el que se publica la Nueva Ley de Mecanismos Alternativos de Solución de Controversias en materia Penal, la cual motiva la implementación de este desarrollo de software.

En el capítulo 2 se profundiza en las actividades de la DGTI a la que estoy adscrito, así como las actividades de la Coordinación de Sistemas Jurídicos en la cual me desempeño. Por último, se presenta un resumen de las múltiples herramientas con las que cuenta el Sistema Integral de Seguimiento de Expedientes (SISE), sistema al que se integra este desarrollo de software y representa un nuevo módulo con un conjunto de funcionalidades propias

Posteriormente, en el capítulo 3 examino propiamente el proceso de creación del nuevo módulo. Partiendo de la etapa de requerimientos, donde se hace uso de historias de usuario como vehículo de especificación de los requerimientos entre el área requiriente y la coordinación de Sistemas

Jurídicos. Continuando con el diseño de una base de datos relacional, así como la arquitectura en tres capas usada para la implementación de la aplicación, la cual está basada en un lenguaje programación orientado a objetos. Así también se muestra el tipo de objetos de base de datos utilizados y de los métodos de la aplicación que proporcionan las funcionalidades más importantes al nuevo módulo.

Por último, se presentan los puntos que se ejecutan en el proceso de liberación de aplicaciones, en mi empleo. Integrado en esta etapa, también se presenta el ambiente de aplicaciones utilizado en las pruebas finales, así como el ambiente final donde se aloja la aplicación y que es accesible desde internet.

El capítulo 4 contiene los puntos importantes derivados de este trabajo y la manera en que este desarrollo facilitará la impartición de justicia alternativa en materia penal. Por último, en el capítulo 5 presento las conclusiones a las que he llegado motivado por esta buena oportunidad de contribuir a la instauración de una justicia pronta y expedita en los conflictos en los que sea posible.

2. ANTECEDENTES

2.1. Contexto del Consejo de la Judicatura Federal

Con la reforma a la constitución política del 31 de diciembre de 1994 y del 26 de mayo de 1995, por decreto publicado por el Congreso de la Unión, se dio origen al Consejo de la Judicatura Federal (CJF), residiendo en él facultades administrativas dentro el Poder Judicial de la Federación. Algunas de éstas le fueron restadas a la Suprema Corte de Justicia de la Nación (SCJN).

El CJF está integrado por 7 consejeros y una presidencia cuya labor recae en el ministro presidente de la SCJN. Este último es también el titular del Poder Judicial de la Federación. El CJF está conformado también por un Pleno, en éste recaen las facultades más relevantes y es donde los consejeros se reúnen para deliberar sobre los temas que les competen. Así también, el CJF está constituido por comisiones que se encargan de distintas tareas de menor índole que el pleno, algunas de ellas son permanentes y es el mismo pleno quien define su permanencia o transitoriedad.

Las decisiones que se toman en el CJF son definitivas e inapelables, salvo las que tienen que ver con la designación o remoción de jueces o magistrados, las cuales pueden impugnarse ante el pleno de la SCJN.

2.2. Actividades del Consejo de la Judicatura Federal

Por el carácter administrativo del CJF, se ha conferido a este órgano el gobierno, administración, vigilancia y disciplina de todos los órganos jurisdiccionales pertenecientes al Poder Judicial de la Federación con excepción de la SCJN y del Tribunal Federal Electoral. También recae sobre el CJF la designación, adscripción y remoción de jueces y magistrados de los Juzgados de Distrito, Tribunales Unitarios, y Tribunales Colegiados de todo el país.

Asimismo, con base en la instrucción constitucional, el CJF tiene la facultad de emitir acuerdos generales que permitan su correcto funcionamiento, así como de elaborar su propio presupuesto. Además, la constitución confiere al CJF la tarea de determinar la división y número de circuitos, competencia territorial, y en caso de ser necesario, la especialización por materia de los tribunales colegiados, tribunales unitarios y juzgados de distrito.

Como se puede observar son muchas las tareas del CJF, razón por la cual los desarrollos de software implementados en la DGTI implican un impacto inmediato a nivel nacional en cuanto entran en operación.

2.3. Dirección General de Tecnologías de la Información

La Dirección General de Tecnologías de la Información (DGTI) se encarga de la administración de todos los aspectos informáticos dentro del Consejo de la Judicatura Federal. También se encarga de vigilar y coordinar las distintas áreas relacionadas como la de infraestructura, la coordinación de Sistemas Administrativos, la de Sistemas Jurídicos y la de Seguridad Informática.

Estas áreas en conjunto proporcionan una red de servicios que utilizan todos los órganos jurisdiccionales del país en sus tareas diarias. De manera ilustrativa en la Figura 1 se presenta el esquema de la DGTI y las áreas que coordina, por otra parte, en la Tabla 1 se describen las actividades de cada área.

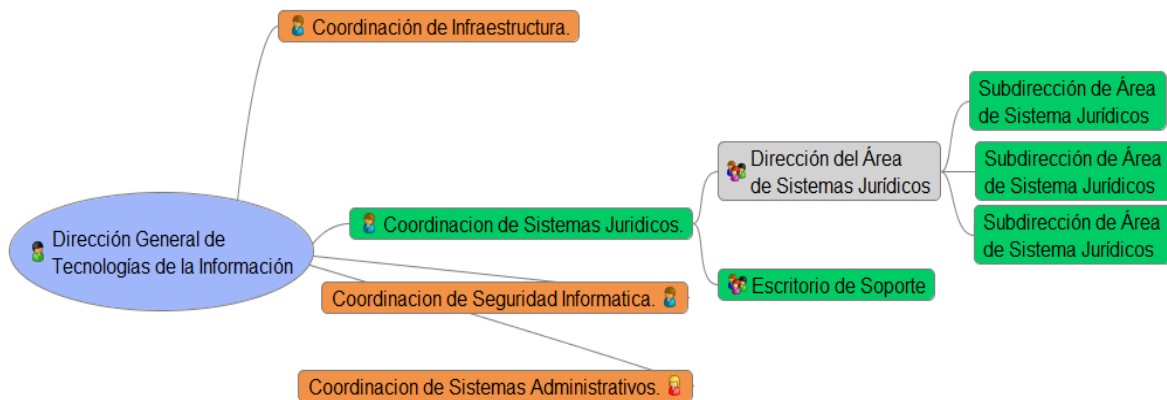


Fig. 1. Estructura de la DGTI.

Tabla 1. Actividades de la DGTI.

Actividades por Coordinación	
Coordinación de Infraestructura	Encargada de proporcionar los servidores necesarios para alojar los sistemas desarrollados por las demás coordinaciones. Así como de vigilar el estado de la red y garantizar su estabilidad y disponibilidad.
Coordinación de Sistemas Jurídicos	Es la encargada de desarrollar los sistemas utilizados por los órganos jurisdiccionales del Poder Judicial de la Federación. Así como los sistemas que permitan la aplicación de justicia de manera pronta y expedita.
Coordinación de Seguridad Informática	Encargada de las estándares y reglas destinados a evitar ataques externos y proporcionar la comunicación segura en la red del CJF.
Coordinación de Sistemas Administrativos	Encargada del desarrollo y mantenimiento de los sistemas que proporcionan servicios y trámites a empleados del CJF.

Como se puede observar cada coordinación tiene bien definido su rol y sus actividades dentro de la DGTI. Esto permite que cada entidad pueda coadyuvar de manera estructurada cuando se realiza un nuevo proyecto, con el fin de concluirlo en tiempo y forma.

Es así como la DGTI puede obtener información clara del desempeño de cada área con la intención de mejorar sus procesos. A su vez, la DGTI es vigilada por los consejeros que constituyen al pleno del CJF.

La coordinación de Sistemas Jurídicos es en donde me desempeño. Valdrá la pena profundizar en las actividades que actualmente realizan en la coordinación y cómo impactan dentro de la DGTI y en las actividades diarias del Poder Judicial de la Federación.

2.3.1. Actividades de la Coordinación de Sistemas Jurídicos dentro de la DGTI

En la Coordinación de Sistemas Jurídicos, se desarrollan sistemas y se proporciona soporte a varios desarrollos de software de gran impacto e influencia en procesos jurídicos del Poder Judicial Federal.

Entre esos sistemas, podemos encontrar al Sistema Integral de Seguimiento de Expedientes (SISE) el cual permite la captura, edición, consulta y explotación de los datos almacenados durante el proceso de vida de un juicio, a través de la generación de reportes con datos estadísticos. Toda esta información se encuentra dentro de un expediente que da identidad a un juicio.

De manera similar funciona el Sistema de Plenos de Circuito, en el cual se registra una propuesta de solución a un asunto y ésta es aprobada o rechazada por un grupo de magistrados al que se le llama Pleno.

Estos son el tipo de sistemas en cuyo desarrollo me veo involucrado cada día. Cabe mencionar que la característica principal de estos desarrollos es el impacto en los procesos jurídicos federales, en particular, aquellos que atañen al CJF.

Particularmente y dentro de la Coordinación de Sistemas Jurídicos, mis actividades consisten en el desarrollo de nuevos módulos para los sistemas ya existentes, así como participar activamente en la totalidad del ciclo de vida del software, desde la etapa de levantamiento de requerimientos de nuevos sistemas hasta el mantenimiento y operación de éstos.

Con base a lo estipulado en la nueva Ley Nacional de Mecanismos Alternativos de Solución de Controversias, es requerido un nuevo módulo que implemente las funcionalidades necesarias para dar cumplimiento a dicha ley. Este es el desarrollo de software que presento en este trabajo.

De manera puntual utilizo programación orientada a objetos, haciendo uso del lenguaje de programación C# y marcos de trabajo de Microsoft como .Net y WPF para aplicaciones de escritorio, o ASP para aplicaciones basadas en páginas web. También, una vez definido el diseño de la base de datos relacional para un nuevo módulo o sistema, utilizo Microsoft SQL Server para su implementación y construcción relacional.

2.4. Sistema Integral de Seguimiento de Expedientes

En la coordinación de Sistemas Jurídicos se proporciona soporte al Sistema Integral de Seguimiento de Expedientes (SISE). Este sistema proporciona un medio para la captura de información, consulta, edición y eliminación relativas a juicios de tipo federal. Sin embargo, con el paso del tiempo se le han agregado más funcionalidades que lo han robustecido.

Actualmente, el SISE es la plataforma de captura y consulta de información de todos los Juzgados de Distrito, Tribunales Unitarios y Tribunales Colegiados del Poder Judicial Federal; así como todos sus órganos auxiliares. El SISE cuenta también con otros módulos que proporcionan reportes estadísticos basados en múltiples criterios de búsqueda.

También es posible llevar a cabo un juicio completamente en línea. Esto último es un requerimiento reciente que permite la aplicación de la justicia de una forma más pronta y expedita, disminuyendo la concurrencia de gente en los juzgados y tribunales; y de paso disminuir la cantidad de papel resguardado en dichos órganos.

Como un paso hacia la implementación del nuevo Sistema de Justicia Penal basado en juicios orales, también se han implementado los esquemas de captura de información para este tipo de juicios. Así como las funcionalidades necesarias para resguardar los videos y audios de los juicios orales celebrados en materia penal, en cada entidad federativa del país. Todos los anteriores son ejemplos de las formas en que el SISE ayuda en las tareas del día a día que se desarrollan dentro del Poder Judicial de la Federación.

A todas estas se añaden las funcionalidades que darán cumplimiento a la Ley de Mecanismos Alternativos de Solución de Controversias. Dichas funcionalidades conforman el nuevo módulo que presento en este trabajo, explicando su ciclo de vida y sus procesos de soporte y mejora.

Esto permitirá al SISE dar cabal seguimiento al cumplimiento del nuevo Sistema Judicial, contemplando tanto a los mecanismos alternativos de solución de controversias, como a la implementación de Juicios Orales.

2.5. Glosario

Para facilitar la lectura de este trabajo, en esta subsección se presenta un glosario de los principales términos utilizados:

- **Mecanismos Alternativos:** La mediación y la conciliación mediante los cuales la víctima u ofendido e imputado buscan, construyen y proponen opciones de solución a la controversia, con el objetivo de atender a las necesidades de la víctima u ofendido, así como las responsabilidades del imputado.
- **Solicitante:** La persona física o moral que acude a algún órgano de justicia alternativa, con la finalidad de buscar la solución a una controversia de materia penal.

- **Requerido:** La persona física o moral que es convocada para solucionar una controversia de materia penal a través de un mecanismo alternativo.
- **Intervinientes:** Las personas que participan en los Mecanismos Alternativos, en calidad de solicitante o requerido, para resolver la controversia.
- **Órgano:** La institución especializada en Mecanismos Alternativos de Solución de Controversias en materia penal perteneciente a la Federación o a alguna entidad federativa.
- **Acuerdo reparatorio:** El acuerdo que repara el daño de manera total o parcial, y que es celebrado entre los intervinientes
- **Facilitador:** El profesional certificado de un órgano que facilita la participación de los intervinientes durante el proceso para llegar a una solución alternativa.

3. MÓDULO DE SEGUIMIENTO DE MECANISMOS ALTERNATIVOS DE SOLUCIÓN DE CONTROVERSIAS

3.1. Historias de Usuario

Las historias de usuario nos permiten aterrizar de manera ágil las necesidades del cliente dentro de una plantilla, haciendo uso de elementos que permitirán ofrecer la información suficiente para iniciar la codificación del sistema y generar rápidamente una versión para el usuario.

Es posible definir un esquema de una historia de usuario de la siguiente manera:

“Como (rol)”
+ “quiero (acción o funcionalidad esperada)”
+ “para (finalidad o resultado de la acción)”.

Con la estructura anterior podemos generar historias de usuario con sentido y funcionalidad, pero aun será necesario asegurarnos de que contengan las siguientes características que nos permitirán iniciar su codificación, de acuerdo con su creador Mike Cohn², las historias de usuario deben ser:

- **Independientes:** No debe existir dependencia de codificación entre cualesquiera dos historias de usuario.
- **Negociables:** Debe existir flexibilidad por parte del cliente en el comportamiento esperado de una aplicación y rendimiento, ya que pueden existir mayores posibilidades de las planteadas o impedimentos dependiendo de la tecnología usada, infraestructura disponible, capacidad adquisitiva del cliente, etc.
- **Valiosas para el cliente o usuarios finales:** Deben describir una funcionalidad de la aplicación que ayude al cliente a cumplir sus expectativas sobre el sistema.
- **Estimable:** Deben ser lo suficientemente claras para que el equipo de desarrollo tenga posibilidad de asignar un tiempo para su codificación y pasar a fase de pruebas.
- **Pequeñas:** Las historias de usuario son cortas, concisas y claras para el cliente y el equipo de desarrollo.
- **Probables:** Deben tener suficiente funcionalidad para ser puestas a prueba, con la finalidad de saber si el desarrollo cumple con el requerimiento o no.

Cuando se logra producir historias de usuario que contemplen la estructura y propiedades anteriores, estamos en posición de comenzar su codificación, para después pasar a una etapa de pruebas y finalmente poder generar un producto entregable al cliente.

²Mike Cohn. 2004. *User Stories Applied: For Agile Software Development*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA.

3.2. Levantamiento de Requerimientos

Esta etapa se realizó a través de reuniones presenciales en las que participó personal del CNI, de la DGEJ y por parte de la DGTI estuvo siempre presente como encargado de la implementación del módulo. En estas reuniones se tocaban los puntos que necesitaba aterrizar el personal de CNI y mi responsabilidad fue proponer las formas en que sus necesidades pudieran ser solventadas a través del sistema.

Dado que el módulo que se presenta en este trabajo contaba con poco tiempo disponible para su desarrollo, pruebas y entrega, se utilizaron un conjunto de herramientas para su desarrollo ágil, como el uso de historias de usuario para el levantamiento de requerimientos y la creación de prototipos de pantallas propuestas durante las reuniones. Como apoyo para la comprensión del flujo completo de las funcionalidades desarrolladas. A continuación, en la Figura 2, se presenta un diagrama general de casos de uso.

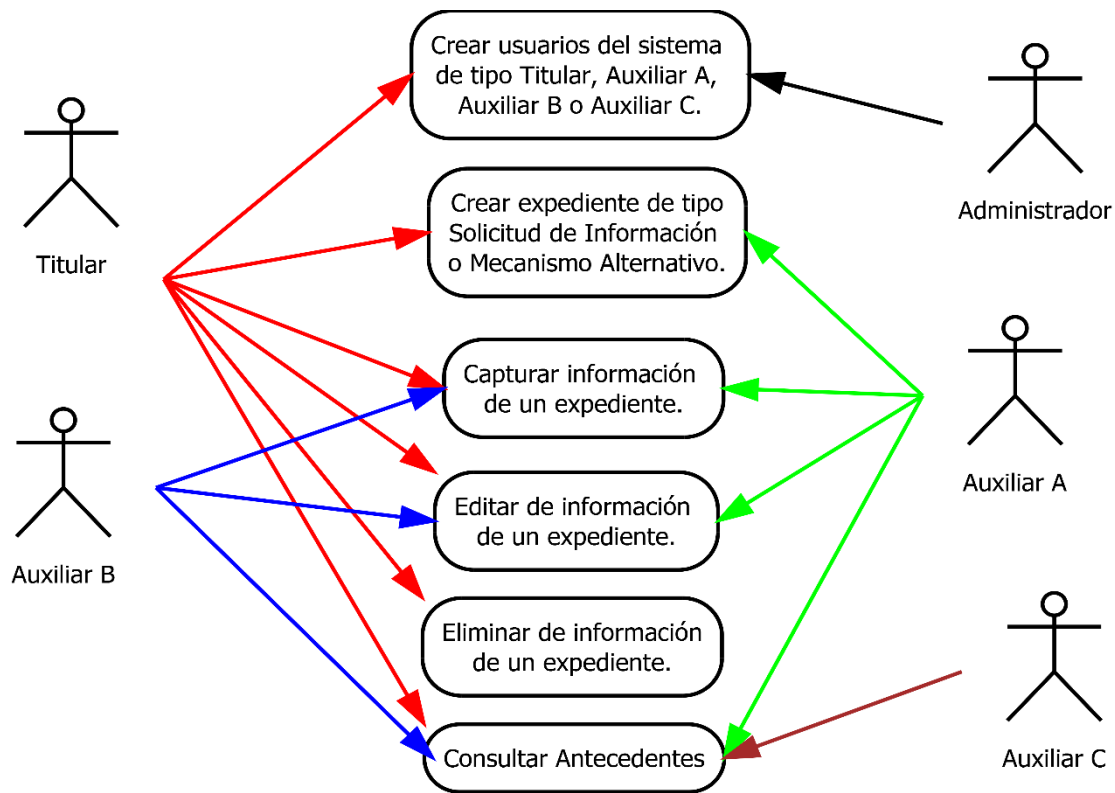


Fig. 2. Diagrama general de casos de uso.

Como se puede ver, existen varios tipos de usuarios del sistema, cada uno tiene acceso a un conjunto diferente de funcionalidades, el usuario de tipo Titular tiene acceso a todas las funcionalidades, mientras que los demás sólo pueden interactuar con algunas de ellas.

La siguiente es una de las historias de usuario que se desprenden del diagrama anterior:

Como usuario, quiero disponer de alguno de los siguientes roles "Titular", "Auxiliar A", "Auxiliar B" y "Auxiliar C" para interactuar con el sistema.

En esta primera historia de usuario podemos observar que se solicitaron los roles de Titular, Auxiliar A, B y C, los cuales han sido creados con una herramienta previamente desarrollada en el sistema. Las siguientes historias de usuario contemplan la etapa previa a un Mecanismo Alternativo de Solución de Controversias (MASC), en la que se genera una Solicitud de Invitación a la contraparte para intentar llegar a un acuerdo que repare el daño causado a la víctima u ofendido. Esta invitación puede ser iniciada por cualquiera de las dos partes involucradas en el conflicto.

Como Titular, Auxiliar A o Auxiliar B quiero capturar una solicitud de invitación, para su posterior consulta.

Tanto en la captura de información de una Solicitud de Invitación como en la captura de un MASC, se requirió la funcionalidad de editar la información capturada de un expediente. Esta edición se realiza accediendo a la pantalla de captura y modificando los datos inicialmente capturados.

Como Titular, Auxiliar A o Auxiliar B quiero editar una solicitud de invitación para disponer de los datos completos y correctos.

También ha sido necesaria la funcionalidad de eliminar un expediente de este tipo. Notemos en el diagrama general de casos de uso, que los permisos a esta funcionalidad están restringidos únicamente a los usuarios de tipo Titular.

Como Titular quiero tener la posibilidad de eliminar un expediente de tipo Solicitud de Invitación para contar sólo con los expedientes necesarios y correctos.

Para implementar estas funcionalidades, ha sido necesario auxiliarnos en un prototipo de la pantalla por desarrollar. Para esto se utilizó la herramienta Pencil, que permite crear prototipos de páginas o formularios.

Cabe aclarar que los campos necesarios, sus tipos, orden y posiciones dentro de la página, así como su estructura, fueron proporcionados por el área requiriente por medio de archivos de hojas de cálculo en los que se especificaron estos atributos, así como las etapas procesales de las que se compone una Solicitud de Invitación y un MASC. En la Figura 3 se puede observar el prototipo del formulario de captura de una Solicitud de Invitación.

Solicitud de invitación

Fecha de ingreso de la solicitud de invitación: 01/01/2016

Tipo de Solicitud: Diferida

Cita

Fecha de cita: 01/01/2016

Hora de cita: 15:30:00

Nombre del facilitador

Nombre: Fernanda

Apellido Paterno: Martínez

Nombre Materno: Chávez

Nombre del solicitante

Carácter: Imputado

Nombre: Pedro

Apellido Paterno: Ugalde

Apellido Materno: Sánchez

Requerido

Carácter: Víctima

Nombre: Martha

Apellido Paterno: Juarez

Apellido Materno: Quiróz

Procedimiento de origen

No. carpeta de investigación/averiguación previa: []

No. causa penal: 2017/FGES/MASC/023

Fig. 3 – Captura de Solicitud de Invitación.

Este formulario contiene los controles que el área requiriente hizo llegar, cumpliendo con el ordenamiento, posición y validaciones de los mismos. Para realizar esta implementación se desarrolló una página ASP de tipo formulario.

De manera similar se implementó el formulario necesario para cumplir con la siguiente HU que pertenece al proceso de un MASC:

Como Titular, Auxiliar A o Auxiliar B quiero capturar una Mecanismo Alternativo de Solución de Controversias, para su posterior consulta.

<input type="button" value="Limpiar información"/> <input type="button" value="Vincular con otro órgano"/> <input type="button" value="Relacionar Expediente"/> <input type="button" value="Finalizar"/> <input type="button" value="Ver Captura"/> <input type="button" value="Normatividad"/>		
Datos de indentificación del expedinete	Entidad federativa de la sede en que se realizó el mecanismo	Jalisco
Nombre del facilitador	Municipio/Delegación de la sede en que realizó el mecanismo	Zapopan
Datos de indentificación de los intervinientes	No. de carpeta de investigación/averiguación previa	2017/PGEJ/C0001
Imputado	No. de Causa penal	
	Fecha de ingreso del expediente	01/06/2017
	Mecanismo aplicado	Acuerdo Reparatorio
	Supuesto de procedencia de conformidad con el artículo 187	Robo con arma blanca
	Estatus de expediente	Proceso
	Fecha de conclusión del asunto	23/07/2017
	Resultado final del asunto	Sanción económica
	Sentido del resultado	Acuerdo Reparatorio
	Tipo de cumplimiento	Inmediato
	Modificación del acuerdo	No
	Acuerdo cumplido	Si
	Causa de conclusión anticipada	Seleccione una opción

Fig. 4 – MASC. Datos de indentificación del expediente.

Como se puede observar en la Figura 4, nuevamente Pencil nos permitió aterrizar el esquema de captura que esperaba el área solicitante. Satisfaciendo sus expectativas y permitiendo clarificar lo que se debía desarrollar. Sin embargo, fueron necesarios otros tres prototipos de pantalla para expedientes de tipo MASC:

<input type="button" value="Limpiar información"/> <input type="button" value="Vincular con otro órgano"/> <input type="button" value="Relacionar expediente"/> <input type="button" value="Finalizar"/> <input type="button" value="Ver Captura"/> <input type="button" value="Normatividad"/>		
Datos de indentificación del expedinete	Nombre	Juan
Nombre del facilitador	Apellido Paterno	Suárez
Datos de indentificación de los intervinientes	Apellido Materno	Vergara
Imputados		

Fig. 5 – MASC. Nombre del facilitador.

Datos de identificación del expediente	Carácter de los intervinientes	Víctima	
Nombre del facilitador	Nombre	Narcizo	
Datos de identificación de los intervinientes	Apellido Paterno	Mendoza	
	Apellido Materno	Martínez	
Imputados	CURP	MEMA880524HDFRRM04	
	RFC	MEMA8805243E4	
	Fecha de nacimiento	24/05/1988	
	Entidad de lugar de nacimiento	Ciudad de México	
	Municipio/Delegación de lugar de nacimiento	Benito Juárez	
	Localidad de lugar de nacimiento	Col. Sta. Cruz Atoyac	
	Estado civil	Casado	
	Número de pasaporte o constancia de calidad migratoria		
	Apodo	Mi estimado	
			<input type="button" value="Agregar"/>

Fig. 6 – MASC. Datos de identificación de los intervinientes.

Datos de identificación del expediente:	Nombre	Mariana	
Nombre del facilitador	Apellido Paterno	Ramírez	
Datos de identificación de los intervinientes	Apellido Materno	Peñaloza	
	CURP (si se tiene)	PERA780212MDFMHM03	
Imputados	RFC (si se tiene)	PERA7802128U3	
	Fecha de nacimiento	12/02/1978	
	Entidad de lugar de nacimiento	Estado de México	
	Municipio/Delegación de lugar de nacimiento	Ecatepec	
	Localidad de lugar de nacimiento	Torres de Satélite	
	Estado civil	Casado	
	Número de pasaporte o constancia de calidad migratoria (si es extranjero)		
	Apodo	Chofis	
	Delito		
	Delito por el que se tramitó	Amenaza de muerte	
	Forma de comisión	Verbal sin arma	
		<input type="button" value="Agregar"/>	
		<input type="button" value="Agregar"/>	

Fig. 7 – MASC. Imputado.

Como podemos notar en las Figuras 5, 6 y 7, la captura de un MASC se realiza en cuatro pantallas de tipo formulario. Cabe resaltar que estos datos también debían estar disponibles para su edición o modificación, ya que el área solicitante también generó la HU correspondiente a esta funcionalidad:

Como Titular, Auxiliar A o Auxiliar B quiero editar el detalle de un MASC celebrado y capturado previamente en el sistema para disponer de los datos completos y correctos.

Para implementar esta funcionalidad, de la misma forma que en la Solicitud de Invitación, se permite el acceso a las pantallas creadas para la captura de información, desde las cuales se permite modificar su captura.

En las dos historias presentadas hasta ahora se hace referencia a las etapas previas a un juicio de tipo penal. Etapas en las que se concilia y acuerda un MASC y de esta manera se llega a un acuerdo reparatorio entre el imputado y la víctima u ofendido.

Las historias de usuario expuestas permiten el registro, edición y eliminación de una Solicitud de Invitación y de un MASC. Si bien este es el objetivo de este nuevo módulo, también fue necesaria la posibilidad de consultar en cuáles acuerdos reparatorios se ha visto involucrado un ciudadano. Así como la posibilidad de generar un archivo en formato pdf con el reporte generado. La HU que permite esta funcionalidad es la siguiente:

Como Titular, Auxiliar A, Auxiliar B o Auxiliar C quiero consultar el historial de acuerdos reparatorios en los que se ha visto involucrado un ciudadano, ya sea como imputado o como víctima, para fines estadísticos e informativos.

Por supuesto esta HU es un tanto larga y poco clara, por lo cual se hizo un taller de retroalimentación en el cual se acordó dividirla en HU's más pequeñas e independientes. Estas HU's muestran funcionalidades más claras e intuitivas, permitiendo comprender la funcionalidad necesaria:

Yo como Titular, Auxiliar A, Auxiliar B o Auxiliar C quiero hacer una búsqueda de ciudadanos involucrados en un MASC, ya sea como imputado o como víctima, a nivel nacional.

Esta es la primera parte del reporte y constituye una búsqueda en la base de datos del SISE, contemplando el carácter de imputado o de víctima del involucrado. Para esto se desarrolló el prototipo de pantalla de la Figura 8:

Constancia de Mecanismos Alternativos de Solución de Controversias

Criterios de búsqueda

Nombre
 Apellido Paterno
 Apellido Materno
 CURP
 RFC
 Apodo

Fig. 8 – Captura de criterios de búsqueda de MASC.

Como podemos ver, la consulta está basada en los datos que puede tener cualquier involucrado, sin considerar su carácter. Cabe resaltar que ningún campo es obligatorio al invocar la consulta, esto debido a que en ocasiones sólo se cuenta con algunos datos, como el nombre y el apodo.

La siguiente parte del reporte consiste en mostrar los resultados de la búsqueda, que como es de esperarse puede consistir en muchas coincidencias con los criterios recibidos. Por ello, se optó por un reporte que contenga los datos de identificación de cada involucrado resultante, junto con un vínculo que nos muestre el detalle de cada uno. Para ilustrar lo antes dicho se muestra el prototipo de la Figura 9.

Constancia de Mecanismos Alternativos de Solución de Controversias							
Constancia	Carácter	Nombre	Apellido Paterno	Apellido Materno	CURP	RFC	Apodo
Generar	Interviniente	José Juan	Fernández	Noroña	CALI880101MHDFBBR89	CALI880101	Tuerto
Generar	Imputado	Jose	Fernández	Nuñez	NEPO880523HDFMRM00		Catarina
Generar	Interviniente	Jose Julian	Fernández	Olvera	MANC880202MDFJN09		Patas

Fig. 9 – Resultados de búsqueda en base de datos de MASC

Este reporte sólo contiene los datos necesarios para identificar a cada involucrado y sus respectivos expedientes. Así como el carácter con el que figura (imputado, víctima o interviniente) y un vínculo para generar un reporte con el detalle de cada registro, el cual también puede obtenerse dentro de

un archivo pdf. Para dicho reporte se hizo el siguiente prototipo dentro de un ejercicio con el área requeriente.

Constancia de Mecanismos Alternativos de Solución de Controversias				
Servidor Público que genera la constancia de consulta				
Nombre	Apellido Paterno	Apellido Materno		
Juan	Zepeda	Martínez		
Institución	Órgano			
CJF	Dirección General de Tecnologías de la Información			
Identificador de Constancia			Fecha de Generación	
5ac5abfc-5c702-4nk2-5dka87sk9aj7ooc			Martha	
Datos Generales				
Nombre	Apellido Paterno	Apellido Materno	CURP/RFC	Carácter
Martha	Cortínez	Chávez	NUGH650988HDFHBM05	Imputado
Acuerdo Reparatorio				
Autoridad que registró el acuerdo				
Alfredo Domínguez Samano				
Autoridad que registró el acuerdo		Fecha de aprobación del cumplimiento del acuerdo		
20/09/2016		20/10/2016		
Resultado final del asunto		Sentido de la Resolución		
Acuerdo Reparatorio		Aprobado		
Sede en que se celebró el acuerdo reparatorio			Entidad Federativa	
Órgano Especializado en Justicia Alternativa			Oaxaca	
Delito	Forma de Comisión		Municipio	
Violencia vecinal	Doloso		Coatecas Altas	
Observaciones				
El imputado ha aceptado culpabilidad y pagará los daños causados, así como trabajos comunitarios.				

Fig. 10 – Detalle de un resultado seleccionado.

En la figura 10 se muestra un reporte con mayor detalle sobre lo capturado del MASC, en el cual está involucrado el ciudadano(a) sobre el que se ejecuta la búsqueda. Si dicho ciudadano(a) ha sido parte de más de un MASC, se presentan los reportes correspondientes a cada uno de dichos expedientes.

3.3. Diseño

En este apartado se explica el diseño de la base de datos, así como la arquitectura de red de los servidores que hospedan la aplicación y a la base de datos. En esta arquitectura se incluye la interacción de un balanceador de cargas para distribuir las solicitudes y respuestas del sistema.

También se presenta el modelo de desarrollo que se utilizó para este módulo, el cual es un modelo de implementación en tres capas: acceso a datos, capa de negocio y capa de presentación.

3.3.1. Arquitectura de red de servidores

En primera instancia, se cuenta con una arquitectura Cliente – Servidor para atender cualquier solicitud entrante desde internet. En caso de requerirse, desde el servidor de aplicaciones se puede invocar un procedimiento almacenado con la información que éste necesite para satisfacer una solicitud. El diseño Cliente – Servidor utilizado se esquematiza en la Figura 11.

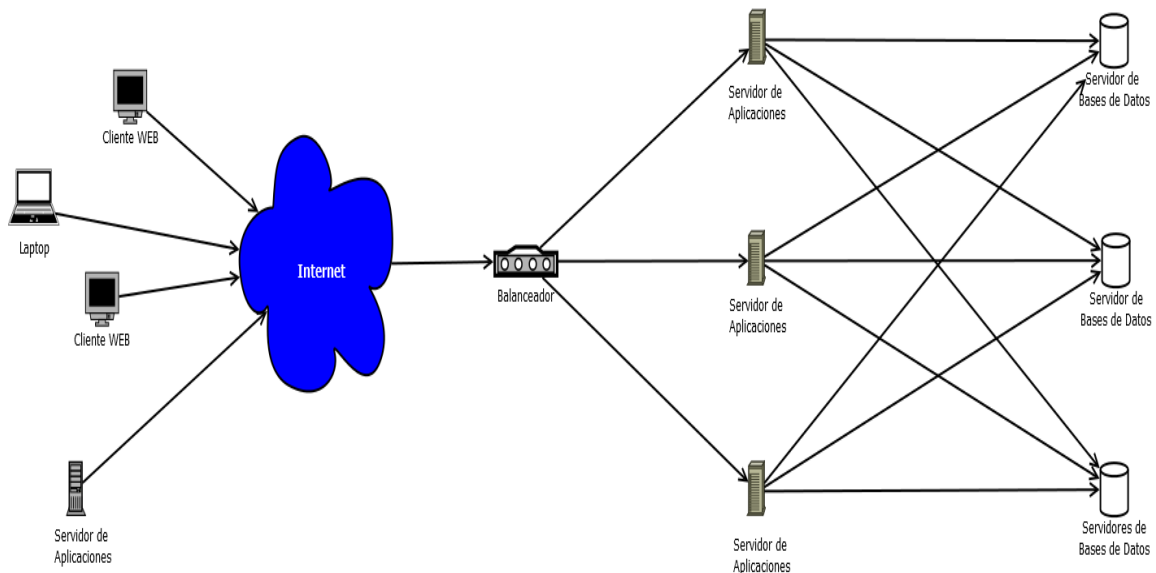


Fig. 11 – Esquema de arquitectura Cliente – Servidor utilizado.

Como se puede observar, cuenta con un balanceador de cargas que expone al sistema y recibe solicitudes a la aplicación. Estas solicitudes son distribuidas hacia el servidor de aplicaciones menos ocupado para que éste se encargue de procesar una respuesta. Estos servidores de aplicación pueden resolver solicitudes de página, consultas a la base de datos, solicitud de archivos, procesos de actualización, entre otras operaciones.

De manera similar pero en la capa adyacente, cada servidor de aplicaciones, de ser necesario, puede consultar una réplica de la base de datos en caso de que el cómputo de una solicitud requiera datos almacenados.

Dada la naturaleza de esta arquitectura, el sistema puede ser utilizado vía web haciendo uso de un explorador de internet. Sin embargo, se ha hecho énfasis en la experiencia de usuario a través de Internet Explorer.

3.3.2. Diseño de la base de datos

El modelo relacional fue el elegido para diseñar la base de datos del módulo. En la Figura 12 se muestra un fragmento de la base de datos, en particular las tablas que contienen catálogos utilizados para la captura de expedientes. Por ser un módulo dentro de un sistema muy grande, cabe mencionar que en esta figura se representan únicamente los objetos de base de datos que inciden directamente en las funcionalidades del módulo.

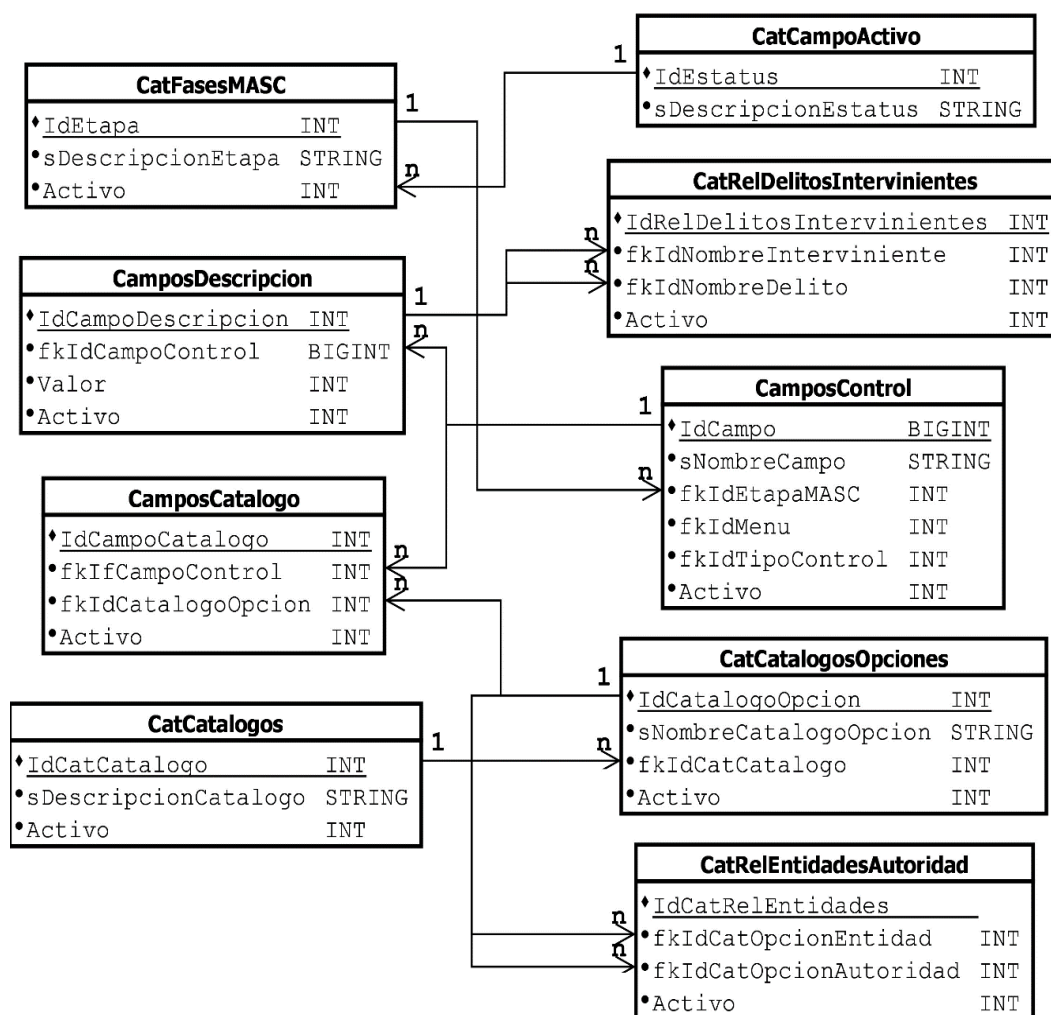


Fig.12 – Esquema de base de datos para el módulo de MASC.

La tabla **CatFaseMASC** contiene las distintas fases de las que se compone un MASC. La tabla **CatCampoActivo** permite distinguir si un registro está 'Activo' o 'Inactivo', para los fines de control

que el sistema requiera aplicar. Como podemos ver cada tabla tiene una llave foránea que referencia a la llave primaria de **CatCampoActivo**.

Por otro lado, **CatCatalogos** proporciona un identificador para cada catálogo usado y **CatCatalogosOpciones** contiene las opciones que pertenecen a cada catálogo y que son visibles en la pantalla de captura de un MASC.

También el catálogo **CamposControl** contiene la definición de los campos utilizados dentro de cada proceso MASC y mientras que las tablas **CamposCatalogo** y **CamposDescripcion** contienen los distintos valores que se han guardado por cada definición de campo.

Cabe mencionar que para cumplir con los requerimientos de este módulo, fue necesaria la creación de algunas tablas como **CatRelEntidadesAutoridad** y **CatRelDelitosIntervinientes**. Donde la primera contiene las fiscalías y procuradurías participantes por cada entidad federativa y la segunda contiene los delitos que cada imputado ha cometido y por los que se celebra un acuerdo reparatorio.

Por su puesto, también se crearon los procedimientos necesarios para registrar, actualizar, eliminar y consultar cada registro dentro de estas dos tablas, que por motivos de seguridad sus nombres no se especifican en este trabajo.

Como se puede observar, la mayoría de las tablas contienen alguna llave foránea de una o más tablas, esto garantiza que no se inserten registros con valores inexistentes en las tablas origen. Vale la pena aclarar que no se ha especificado una acción al borrar o actualizar un registro en alguna tabla origen (SQL Server asigna la opción NO ACTION), por lo que ante estos escenarios también es necesario el borrado o actualización en los registros de las tablas que referencian a la llave primaria.

En cuanto a la integridad de dominio, podemos aclarar que todas las columnas usan dominios predefinidos en SQL como **INT**, **BIGINT**, **BIT**, **VARCHAR** y **NVARCHAR**, de manera que no se permite la inserción de datos o actualizaciones de valores que no pertenezcan al dominio de cada columna, a menos que éstas permitan valores nulos.

Finalmente, en cuanto a la integridad de dominio, siempre se tiene el cuidado de que cada tabla existente o de reciente creación contenga una llave primaria simple o compuesta, de manera que cada registro de cada tabla esté identificado de forma única.

3.3.3. Modelo de desarrollo de la aplicación

En lo que concierne al modelo de implementación del módulo de MASC, se adoptó un modelo de codificación en capas: acceso a datos, negocio y presentación. La capa de acceso a datos implementa métodos que invocan procedimientos almacenados dentro de la base, los cuales ejecutan tareas CRUD (crear, leer, actualizar y borrar). Esto se hace con la intención de prevenir ataques por inyección SQL y ningún método debe ejecutar sentencias SQL puras. Por otro lado, esta capa también permite invocar los procedimientos que obtienen el historial de MASC's que tiene cada ciudadano.

La capa de negocio es el ambiente en el cual se implementan las reglas del negocio con los datos provenientes de la capa de presentación o la capa de datos, aplicando así la lógica para el

funcionamiento esperado por el área requiriente, ya sea para guardar los detalles de un expediente, editarlos, eliminarlos; o bien, generar un reporte de los expedientes existentes y activos, así como generar un archivo PDF a partir de estos. En esta capa también se suele preparar la información que se mostrará en la capa de presentación, con la intención de que esta tercera capa sólo muestre el estado de los datos capturados sin necesidad de darles un trato complejo.

Por último, la capa de presentación nos permite presentar los datos que la capa de control ha manipulado y preparado para su representación visual. También permite que los controles ASP .NET que interactúan en la página ASP, transfieran los datos recibidos del usuario hacia la capa de negocio, como son las opciones de catálogo seleccionadas, fechas, números, cadenas recibidas con y sin formato. Esta capa, nos permite el uso de eventos originados a partir de controles ASP .NET, estos eventos nos permiten identificar el origen de una solicitud para tratarla de la manera necesaria. Por ejemplo, obtener un reporte de MASC, al dar click en el botón "Buscar".

3.4. Construcción

En esta sección se describirán los objetos de base de datos que se utilizaron, las estructuras de datos utilizadas en la capa de negocio, los conceptos de Programación Orientada a Objetos, así como los controles de captura y visualización de datos usados en la capa de presentación del módulo de MASC.

3.4.1. Construcción de base de datos y creación de objetos

Comencemos por detallar los tipos de transacción SQL usados dentro de los procedimientos almacenados. Cabe aclarar que tanto en el esquema de captura de una **Solicitud de Invitación**, como en el de **Mecanismos Alternativos de Solución de Controversias**, se utilizan los mismos procedimientos de captura, lectura, actualización y eliminación de datos, y algunos otros para la obtención de catálogos que son invocados al generarse un evento de botón o de un DropDownList. Por otro lado, tenemos una consulta robusta que se ejecuta al obtener un reporte de involucrados en un MASC, la cual también será detallada en esta sección.

Toda transacción utilizada, debe estar inmersa en una rutina que controla posibles errores, de manera que si algo sale mal la transacción no se compromete y se ejecuta la instrucción **ROLLBACK**. De esta manera los cambios se resarcan y los registros regresan a su estado original. En la Figura 13 se muestra la estructura genérica de una transacción utilizada por el módulo MASC.

```

-- =====
-- Author:      Omar Neria Pérez
-- Create date: 01/01/2017
-- Description: Procedimiento que obtiene valores
--              capturados, en función de la etapa
--              de un MASC.
-- =====

CREATE PROCEDURE ObtenerValoresCaptura
    @IdEtapa INT
AS
BEGIN TRY
    BEGIN TRAN
        -----Transacciones SQL-----

    END TRY
    BEGIN CATCH
        -----En caso de error-----
        ROLLBACK TRANSACTION;

    END CATCH

    -----Transacción completa-----
    COMMIT TRANSACTION;

```

Fig. 13- Estructura genérica de una transacción.

En caso de que la transacción no cause ningún error, esta se compromete con la instrucción **COMMIT TRANSACTION** y finalizan los cambios. Así también, todos los procedimientos almacenados contienen una sección donde se refiere la fecha de creación, al autor, parámetros y descripción de los efectos que produce el procedimiento.

Por otro lado, los procedimientos que permiten la obtención de datos de control de campos, opciones de catálogos o datos previamente capturados de un MASC o Solicitud de Invitación, siempre devuelven las columnas que se requieren y no bloquean las tablas grandes o de uso recurrente mediante la instrucción **WITH (NOLOCK)**. En el caso de las actualizaciones o **UPDATES** se utiliza la instrucción **WITH (ROWLOCK)**, lo cual permite bloquear el registro mientras esté siendo modificado por la instrucción, ver Figura 14.

```

BEGIN TRY
    BEGIN TRAN

    ---- Transacciones SQL ----
    UPDATE CamposDescripcion WITH(ROWLOCK) SET Valor = 'Juan Alberto'
    WHERE FkIdCampoControl = 1;

    SELECT IdCampo, sNombreCampo FROM CamposControl WITH(ROWLOCK)
    WHERE fkIdEtapaMASC = 1;

```

Fig. 14 - Esquema de consulta en procedimientos de lectura de datos.

Una actualización, como la mostrada en la Figura 14, nos permite actualizar el valor de un campo, necesario al finalizar la edición de un expediente. La segunda consulta, permite obtener los campos y sus descripciones del proceso de Solicitud de invitación.

Las instrucciones de guardado consisten básicamente en insertar un registro en la tabla **CamposValor** por cada campo que haya recibido valor durante la captura y su eliminación consiste en dar de baja el registro asignando un **0** al atributo **StatusReg** o eliminando completamente el registro si no es necesario mantenerlo en base de datos, ver Figura 15.

```

BEGIN TRY
    BEGIN TRAN

    --- Transacciones SQL ---
    INSERT INTO CatCatalogos (sDescripcionCatalogo, Activo)
    VALUES('Carácter de los Intervinientes', 1);

    UPDATE CatCatalogos SET Activo = 0
    WHERE IdCatCatalogo = 250;

```

Fig. 15- Inserción y eliminación de valores capturados.

Al hacer la inserción siempre vale la pena especificar los atributos que serán utilizados del nuevo registro, pues no siempre los campos usados están en el orden deseado. Además, es un buen hábito que facilita el entendimiento del procedimiento almacenado.

3.4.2. Construcción de objetos y uso de conceptos de Programación Orientada a Objetos

En esta sección se mencionan y ejemplifican los conceptos propios del paradigma de programación orientada a objetos, de los que se hace uso en la codificación del módulo. Vale la pena recordar que la codificación se implementa en capas (acceso a datos, negocio y presentación). Esto induce el transporte de datos entre capas, según la necesidad o caso de uso; y por lo tanto, el uso del concepto de polimorfismo.

En la capa de acceso a datos encontramos aquellos métodos que invocan procedimientos almacenados, ya sean de lectura, escritura o actualización, ver Figura 16. Estos métodos regularmente reciben objetos definidos ad hoc, y que permiten inferir fácilmente el uso que recibirán dentro del método.

```
public MascResultCadEntity ConsultaCapturaMasc(RegistroMascEntity registro)
{
    MascResultCadEntity res = new MascResultCadEntity();
    try
    {
        ///Conexión a base de datos.
        ///Obtención de datos (dataset).
        ///Se almacenan los datos en una instancia
        ///de MascResultCadEntity, y se transfiere
        ///a la capa de negocio.
    }
    catch (Exception ex)
    {
        ///Si algo sale mal se comunica el error
        ///a través de la instancia de MascResultCadEntity.
    }
    return res;
}
```

Fig.16- Capa de Acceso a Datos.

Se puede notar que este método cuenta con un bloque *try-catch*, lo cual nos permite el manejo de excepciones de la manera más adecuada posible. Estos bloques aparecen en todos los métodos de cada capa de implementación. También se puede ver que el método devuelve una instancia de un objeto que denota el resultado de su ejecución.

Como podemos esperar, el resultado contenido en este objeto puede ser el conjunto de datos de una consulta, el id de un nuevo registro, un indicador booleano de éxito o fracaso de una transacción, o simplemente un conteo de registros afectados; todos estos casos dependen del tipo de transacción requerida.

Si alguna situación no contemplada arroja una excepción en tiempo de ejecución, su propagación en el árbol de ejecución es interrumpida y manejada por la instrucción *try-catch*. Esta previsión, más que una buena práctica, es una construcción indispensable para indagar rápidamente el origen y la causa de un error en el código, ahorrando tiempo y, en situaciones críticas, dinero.

Dentro de la capa de negocio, ver Figura 17, encontraremos aquellos métodos que procesan la lógica del negocio necesaria según el caso de uso, por lo tanto este nivel suele contener la mayor complejidad del código, donde se hace uso exhaustivo de las estructuras de datos más conocidas y las que aportan tecnologías y conceptos como `linq`, variables que infieren su tipo, funciones lambda, entre otras. Todo con el objetivo de cumplir con la funcionalidad esperada en el caso de uso. Por esto se debe ser especialmente cautos en elegir la estructura de datos más adecuada, sin sacrificar rendimiento o versatilidad, así como evitar el uso de iteraciones y recursión excesivo o innecesario.

Puesto que esta capa invoca procedimientos alojados en la capa de acceso a datos, en muchas situaciones esta debe manipular los datos obtenidos y enviarlos a la capa de presentación con la estructura más digerible posible, ya que el objetivo de la capa de presentación es mostrar datos en el formato esperado.

```
public MascResultBusinessEntity ConsultaCapturaMASC(RegistroMascEntity registro)
{
    MascResultBusinessEntity res = new MascResultBusinessEntity();
    try
    {
        ///Se crea una instancia de la capa de acceso a datos
        ///y se invoca el método que obtiene los datos capturados.
        MascDAC MascCapaDatos = new MascDAC();
        var resCAD = MascCapaDatos.ConsultaCapturaMASC(registro);

        ///Se le da el tratamiento necesario y regresa los datos
        ///con la estructura que requiere la capa de presentación.
    }
    catch (Exception ex)
    {
        ///Si algo sale mal se comunica el error
        ///a través de la instancia de MascResultBusinessEntity.
    }
    return res;
}
```

Fig.17- Capa de Negocio que invoca a capa de datos.

Aquí también se observa el uso de una entidad que transporta el resultado de la capa de negocio a la de presentación **MascResultBusinessEntity**, todas las entidades destinadas al transporte de resultados heredan de una clase base **MascResultEntity** que únicamente define una cadena para guardar la descripción de una excepción si ésta se genera.

3.4.3. Construcción de capa de presentación

En esta etapa se describirá la manera en que se comunican la capa de negocio con la capa que genera la presentación de una página de tipo Web Form, este tipo de páginas son de tipo formulario y pueden contener controles ASP .NET diseñados para interactuar directamente con el usuario final.

Estas páginas se dividen en dos archivos, uno con extensión **.aspx**, que contiene únicamente los controles que son visibles al usuario, ya sean controles HTML o ASP. NET y otro archivo de extensión **.aspx.cs** que puede manipular todos los objetos definidos en el archivo **.aspx** para darles formato, extraer o asignar valores, modificar su apariencia, posición dentro de la página, etc. Este segundo está implementado en **C#** y es el medio de comunicación entre la capa de presentación y la capa de negocio.

Existen muchas formas de obtener datos de la base de datos y presentarlos en un *WebForm*, en este caso mostraremos un método que es invocado al inicio de la carga del formulario, se trata de un método ubicado en el archivo **.aspx.cs** o como también se le conoce, el *code behind*, ver Figura 18. Éste puede considerarse el primer caso de uso, donde se entra a la pantalla de captura y se visualiza la captura que hasta ese momento se ha registrado.

Existen otros métodos como la actualización, edición y borrado, los cuales ejecutan el mismo principio: al dispararse un evento en la capa de presentación, ésta instancia un objeto de la capa de negocio e invoca un método que tenga definido. A su vez, la capa de negocio crea una instancia de un objeto de la capa de datos para que éste acceda a la base de datos. Finalmente, el resultado se manipula en cada capa y se le da el tratamiento requerido, según el caso de uso.

```
public void ConsultaCapturaMASC ()
{
    try
    {
        MascCN MascCapaDeNegocio = new MascCN();
        RegistroMascEntity registro = new RegistroMascEntity();

        ///Se manipula el objeto registro para contener
        ///los parámetros para invocar a la capa de negocio.

        ///Se invoca a la capa de negocio a través
        ///del objeto MascCapaDeNegocio
        var resCN = MascCapaDeNegocio.ConsultaCapturaMASC(registro);

        ///Se hace la manipulación necesaria para mostrar
        ///los datos en la capa de presentación.
    }
    catch (Exception ex)
    {
    }
}
}
```

Fig 18- Code Behind en la capa de presentación que invoca a la capa de negocio.

Se han aplicado muchos conceptos importantes durante la codificación de este módulo, así como muchas prácticas recomendadas para generar sistemas confiables y escalables. Hasta este punto,

algunos de estos conceptos ya fueron revisados en el presente trabajo, sin embargo en el siguiente cuadro conceptual se presenta un compendio más detallado de estas propiedades:

Tabla 2. Conceptos utilizados.

Concepto		Utilidad en el módulo
Base de datos	Bloque TRY - CATCH	Manejo de excepciones en tiempo de ejecución dentro de cada procedimiento almacenado.
	Instrucción ROLLBACK TRAN	En caso de que producirse un error, los cambios no se comprometen y se deshacen los cambios.
	Comentarios dentro de procedimientos almacenados	Fecha de creación del procedimiento almacenado.
		Autor del procedimiento.
		Utilidad y módulo donde se utiliza el procedimiento.
	Índices y llaves primarias	Creación de índices en tablas recurrentes de consulta y búsqueda.
Llaves foráneas	Atributos que almacenan la llave primaria de otra tabla permitiendo la definición de relaciones entre tablas.	
Implementación C# y ASP .NET	Capa de Acceso a datos	Proyecto dentro de la solución encargado de acceder a la base de datos e invocar procedimientos que aplican las transacciones requeridas.
	Capa de Negocio	Proyecto dentro de la solución cuya implementación produce el comportamiento definido en la historia de usuario.
	Capa de Presentación	Proyecto dentro de la solución encargado de la interacción con el usuario y mostrar los efectos de ésta.
	Herencia y Polimorfismo	En el transporte de datos entre capas, en la representación de conceptos jurídicos recurrentes como Promociones, Determinaciones, Notificaciones o Juicios. Estos tienen propiedades en común que se pueden abstraer y utilizar usando los conceptos de Herencia o Polimorfismo.
	Entidades para la comunicación entre capas	Entidades que heredan de una clase base para las respuestas de métodos.
		Entidades que heredan de una clase base para el envío de parámetros.
	Validaciones	Sobre los datos que están por guardarse, editarse o eliminarse.
Uso de máscaras o formatos en los datos de entrada del formulario.		

3.5. Pruebas

En este apartado explicaré el proceso de pruebas al que cada funcionalidad del nuevo módulo **MASC** fue sometida. Se trata de un proceso que involucra a dos entidades dentro del CJF: la DGTI y la Dirección General de Estadística Judicial (DGEJ). La primera es el área que desarrolla el software y la segunda es la encargada de validar que el funcionamiento de cada nuevo desarrollo cumpla con el comportamiento esperado (conjuntos de casos de uso o historias de usuario).

Cada nuevo desarrollo puede ser solicitado por la DGEJ o bien alguna otra entidad con la que se tenga un convenio celebrado. En el caso del módulo MASC, se trata de un requisito del CNI (Centro Nacional de Información), por lo tanto son los clientes y los dueños de la información. Dadas las circunstancias, las pruebas realizadas al funcionamiento del módulo estuvieron a cargo tanto de la DGEJ como del CNI.

Para esta fase de pruebas se ha implementado un ambiente de desarrollo constituido por un servidor IIS, con ubicaciones que son accesibles por intranet, con los permisos y herramientas necesarias para ejecutar el sistema. Asimismo, en este ambiente se tiene acceso a una base de datos de pruebas en la que se puede guardar, consultar y explotar datos sin una utilidad real.

Cada historia de usuario fue desarrollada sobre la última versión del SISE y los cambios fueron probados de manera local (en el mismo equipo) como primer paso. Posteriormente, los archivos modificados se integran en un conjunto de cambios (o *shelveset* en Microsoft Visual Studio). Esto permite su integración y su publicación en el ambiente de desarrollo descrito anteriormente. El proceso de pruebas continúa y va por cuenta de personal de la DGEJ, quienes se dedican a hacer dichas pruebas sobre cada requerimiento solicitado.

En caso de que se identifique algún error en el funcionamiento del sistema, éste se comunica al área de Soporte de la DGTI para que se solvete de inmediato. Estas incidencias las resuelve directamente quien desarrolló el sistema o el módulo, posteriormente se genera un nuevo *shelveset* para ser integrado a la solución y reiniciar el proceso de pruebas por parte de la DGEJ y del CNI. Se repite el mismo proceso hasta que se obtiene una versión aceptable del sistema y la entidad solicitante da su aprobación para su publicación.

3.6. Liberación

Una vez obtenido el visto bueno por parte de la DGEJ y del CNI, se publican los cambios en un servidor IIS, en alguna ubicación que sea visible a través de Intranet (o la red local del consejo), pero que no sea visible desde Internet.

Se replican tanto los objetos nuevos como los cambios hechos a los procedimientos almacenados de la base de datos de desarrollo a la de producción (la base con datos reales). También se modifican las conexiones a base de datos del sistema, para que este trabaje con la base de datos productiva.

Para este proceso de liberación, también existe un servidor IIS con las herramientas y servicios necesarios para ejecutar el sistema. Estos cambios se pueden publicar con la herramienta de

publicación del IDE o bien copiando los archivos que se encuentran en el ambiente de desarrollo al ambiente de producción.

Posteriormente, debemos validar que los cambios publicados funcionen correctamente en ambiente de producción. Para esto, dedico un tiempo considerable en verificar que cada nueva funcionalidad trabaje en producción, tal cual lo hace en el ambiente de desarrollo. Por supuesto; con base en las historias de usuario definidas.

En caso de encontrar algún error que provenga del nuevo ambiente de ejecución, se reporta al personal encargado de infraestructura para que éste se encargue de solventar la incidencia; por otro lado, si el error es de programación, este es el momento adecuado para corregirlo.

Este proceso se realiza con el objetivo de obtener cualquier detalle de validación, de presentación o de comportamiento que los nuevos requerimientos puedan arrojar, todo esto, previo a publicar los cambios en la IP de producción (que ya es visible desde Internet).

Una vez que se valida que el nuevo desarrollo funciona correctamente, ya sea haciendo uso de la base de datos de pruebas o haciendo uso de la base de producción, entonces el código de la publicación se transfiere al servidor anfitrión del sistema. Este se encarga de ejecutarlo y servir al usuario final cuando éste lo consume a través de la web. Es hasta este momento que decimos que el sistema se encuentra en producción. Ver Figura 19.

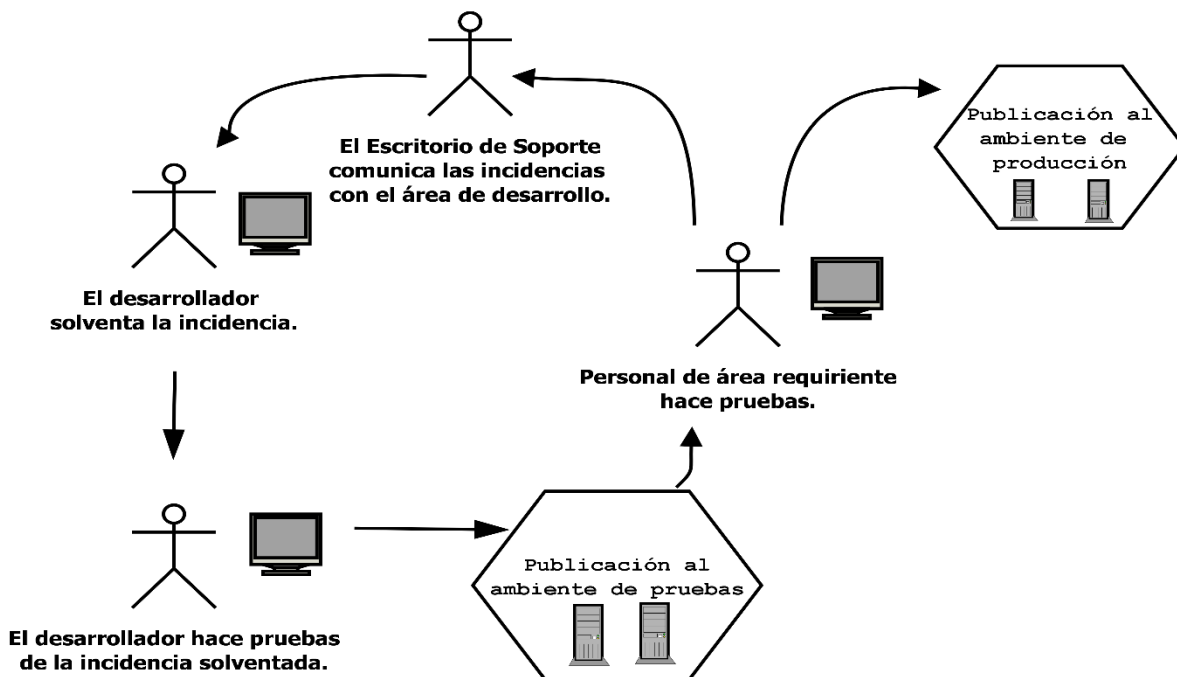


Fig. 19 – Diagrama de pruebas y liberación de funcionalidades.

4. RESULTADOS

Con este nuevo desarrollo los órganos especializados en Mecanismos Alternativos adscritos al Poder Judicial de la Federación contarán con un sistema que les permitirá capturar, editar y consultar los datos generados de cada proceso.

Tomando como origen la información almacenada en la base de datos, una autoridad judicial será capaz de obtener el historial de todos los acuerdos a los que ha llegado un ciudadano; o bien, la constancia de no antecedentes respectiva.

Dada la definición del formulario de captura y de la base de datos, es posible obtener información de gran interés, por ejemplo: ¿qué tipo de delitos suelen ser solucionados por esta vía?, ¿cuántos acuerdos reparatorios se han celebrado en cada estado?, ¿cuántas invitaciones a acuerdos reparatorios son rechazadas?, ¿cuántos imputados incumplen el acuerdo reparatorio que convinieron con el ofendido?, ¿con este nuevo modelo de justicia penal, qué tanto se disminuye el número de juicios en los juzgados y tribunales?, ¿qué tanto se desacelera la población en reclusorios y centros penitenciarios?, ¿La reincidencia delictiva aumenta o disminuye con la aplicación de los Mecanismos Alternativo de Solución de Controversias? Todas estas preguntas pueden ser resueltas haciendo una consulta a la base de datos, ayudando a la generación de estadísticas útiles y en consecuencia a una mejor toma de decisiones.

También, con ayuda de este desarrollo los órganos especializados en Mecanismos Alternativos que ya se encuentran en operación, podrán dejar de ocupar hojas de cálculo como medio de captura de información. Proporcionándoles certidumbre en la integridad y seguridad de los datos que capturan a través del sistema. También, por el diseño del formulario de captura, el personal de estos órganos jurisdiccionales podrá aplicar una captura mucho más ágil del expediente, así como un mayor entendimiento de su información.

Por otra parte, se verán beneficiados al menos sesenta nuevos órganos especializados en justicia alternativa en todo el país, de los cuales algunos ya se encuentran en operación, y otros están en su proceso de constitución y capacitación de personal. Así también, algunas instancias de la Procuraduría General de la República tendrán facultades únicamente de consulta del estatus de los expedientes y generación de constancias. Sin dejar de lado al Centro Nacional de Información, que es la instancia que integra y administra las bases de datos criminalísticas y del personal de seguridad pública.

Todas estas instituciones harán uso de los 4 roles de usuario que se crearon para este nuevo módulo, de esto se desprende que al menos se tendrán 240 empleados usándolo, facilitando las actividades de dicho personal. Sin embargo, los principales beneficiados siempre serán los ciudadanos que hagan uso de la justicia alternativa.

Si bien se obtuvo una primera versión de este desarrollo al cabo de ocho meses de trabajo, existen aún consideraciones de posibles cambios en su funcionamiento. Esto se debe principalmente a que la ley de Mecanismos Alternativos de Solución de Controversias está en una etapa inicial en su aplicación y el ejercicio de esta ley dictará las nuevas funcionalidades que el sistema requiera.

Se estima que este desarrollo entre en funciones en todo el país a más tardar en el mes de julio de 2017, una vez concluidas y aprobadas las pruebas piloto. Estas pruebas consisten en implementar el sistema en órganos que se encuentren en operación para, a efecto de su uso exhaustivo, obtener retroalimentación con propuestas para su funcionamiento. Algunas de estas correcciones también serán integradas. Al cabo de esta integración se habrá aportado un engrane más al nuevo paradigma de justicia penal del país^{3,4}.

³ Comunicado de prensa del CJF, <http://www.cjf.gob.mx/documentos/Comunicados%20Prensa/docsComunicadosPrensa/2017/comunicado22.pdf> (Consultado el 19/08/2017)

⁴ Portal 20 Minutos, <http://www.20minutos.com.mx/noticia/255911/0/poder-judicial-apuesta-a-nuevas-tecnologias-para-dar-certidumbre/> (Consultado el 19/08/2017)

5. CONCLUSIONES

El origen de este desarrollo es la aplicación de un nuevo paradigma de justicia en materia penal, el cual permite solventar y soportar la creciente incidencia de casos en esta materia. Los Mecanismos Alternativos de Solución de Controversias son uno de los puntos clave en este proceso de cambio. En pocas palabras, permitirán conciliar un acuerdo entre las partes y así ofrecer una vía de solución ágil y supervisada por autoridades capacitadas.

Por lo anterior, este sistema será parte importante en la correcta y oportuna aplicación de justicia alternativa en cada entidad federativa del país. Permitirá que la información generada en un expediente de este tipo se almacene en una base de datos que aporte confiabilidad y soporte el estado jurídico de una persona.

También la estructura de las pantallas de captura permite conocer las averiguaciones penales que preceden a cada expediente, los datos que identifican a cada involucrado de cada controversia en materia penal a nivel federal, el proceso inicial por el que una parte invita a la otra a celebrar un acuerdo reparatorio, así como el resultado en que deriva el mismo, si el acuerdo llegó a buen término o no, y si este tuvo complicaciones en el camino. En conjunto permite conocer si una controversia fue solucionada a través de un mecanismo alternativo y cuál fue su ciclo de vida.

Por otro lado, ahora será posible que un órgano jurisdiccional consulte el historial de controversias en que se ha involucrado un individuo dentro del territorio nacional y el ciclo de vida de sus respectivos acuerdos. La existencia o ausencia de dicho historial de acuerdos, posiblemente represente el punto de bifurcación entre la libertad condicionada y su privación a consecuencia del juicio que se derive si el ciudadano ya no es acreedor a un mecanismo alternativo más.

Cabe resaltar que la viabilidad de un acuerdo está sujeta al juicio de la autoridad que fungirá como mediador, y esta autoridad tomará como atenuante el historial de acuerdos arrojado por el sistema. Esta es la razón más importante por la que se debe asegurar su robustez y correcto funcionamiento. El sistema debe ser una herramienta para la correcta aplicación de justicia, no de la injusticia.

Notemos también que una vez que las partes involucradas en el asunto alcanzan un acuerdo reparatorio, si éste no es cumplido por parte del imputado, se procede directamente al proceso de juicio correspondiente, sin embargo este último proceso ya no es parte del alcance del sistema; pertenece propiamente a un juicio en materia penal (que bajo el nuevo sistema de justicia, se trata de un juicio oral).

Vale la pena recordar que este desarrollo está orientado a la implementación y seguimiento de los Mecanismos Alternativos específicamente dentro de la materia penal, siendo que actualmente ya se implementan los Mecanismos Alternativos en otras materias. Esto da una pauta del potencial de crecimiento del módulo, al permitir la generación de nuevos esquemas de captura y ampliando su funcionalidad. Aunque también cabe la posibilidad de que posteriormente se soliciten desarrollos similares enfocados a los Mecanismos Alternativos en materias distintas a la penal.

Si bien el desarrollo de este módulo se concluyó satisfaciendo las expectativas de funcionalidad requeridas, hubo tropiezos, complicaciones y retrasos en el camino. Dado que la aplicación de esta ley implica conocimientos nuevos en la materia penal del derecho, así como una filosofía y procesos

jurídicos poco o nada conocidos en México, fue necesario dedicar un esfuerzo importante a la definición clara de los requerimientos.

Un ejemplo de lo anterior es que en la Ley Nacional de MASC se estipula que un imputado puede llegar a un acuerdo con una víctima y también puede alcanzar un acuerdo reparatorio distinto con las demás víctimas del caso. Esto implica que el sistema debe poder relacionar a un imputado con varias víctimas y generar tantas relaciones de este tipo como sean necesarias. Además, cada acuerdo debe incluir algún delito, y este no se puede repetir en otro acuerdo víctima/imputado. Si bien explicarlo en este momento resulta claro, en la práctica acotar, especificar e implementar este requerimiento fue complejo.

Esta ley se encuentra en sus primeros años de aplicación y existe una variación en como cada entidad la interpreta, este aspecto incrementa la complejidad al momento de definir el comportamiento de un sistema que espera cubrir las expectativas de la mayoría de sus usuarios y las entidades que ya se encuentran aplicando dicha ley.

Por otro lado, puesto que este desarrollo fue realizado en el Consejo de la Judicatura Federal, y siendo el cliente el Centro Nacional de Información, se realizaron varias presentaciones del sistema en muchos eventos. Esto conlleva varios esfuerzos de publicación y ambientación del sistema con el fin de presentar los avances del desarrollo. Por ello fue necesario dedicar tiempo de trabajo en actividades no relacionadas estrictamente con el desarrollo del sistema.

Por supuesto, nada de esto sería una meta alcanzable sin la formación que obtuve en las aulas de la Facultad de Ciencias de la UNAM. En la que, entre otras experiencias, pude conocer los temas más profundos e interesantes de la ciencia del cómputo, así como de las matemáticas que modelan su funcionamiento a través de objetos como la máquina de Turing y la teoría de computabilidad y complejidad a que da origen tan interesante concepto.

Particularmente, el paradigma de programación orientado a objetos, los conceptos involucrados en el diseño de bases de datos y el lenguaje de consulta estructurado SQL, fueron parte de los conocimientos que me permitieron atacar este proyecto exitosamente. Si bien estos conceptos me fueron expuestos usando tecnologías, por lo regular no pertenecientes a Microsoft, si fueron explicados con la generalidad necesaria para entender rápidamente su uso y aplicación a través de otras tecnologías, como lo fueron ASP .Net, y Microsoft SQL Server. Por otro lado, también fue de gran ayuda lo aprendido acerca de lenguajes de programación fuerte y débilmente tipados, esto me permitió entender y usar lenguajes de programación como JavaScript (que es un lenguaje interpretado y débilmente tipado), para modificar el contenido HTML de una página web de manera dinámica.

El anterior entre otros muchos temas más que introducen a cada nueva generación en un profundo mar de conocimiento tan complejo como interesante.

BIBLIOGRAFÍA

1. J.J. Cázarez Ramírez: *Principios que rigen el Proceso Penal Acusatorio y Código Nacional de Procedimientos Penales*, México, Flores, 2016, 186 pp.
2. E. López Betancourt: *Introducción al Derecho Penal*, México, Porrúa 2015, 305 pp.
3. J. De la Fuente Rodríguez y A.F. Mondragón Pedrero: *Los juicios orales Mercantiles*, México, Porrúa, 2015, 293 pp.
4. C. Beccaria: *Tratado de los Delitos y de las Penas*, México, Porrúa, 2014, 254 pp.
5. Instituto de Investigaciones Jurídicas: *Diccionario Jurídico Mexicano*, México, Porrúa, 2011.
6. ISO/IEC Software engineering Lifecycle profiles for Very Small Entities (VSEs) 5-1-2 Management and engineering guide: Generic profile group: Basic profile. ISO-IEC TR 29110-5-1-2 Technical Report.
7. M. Cohn: *User Stories Applied: For Agile Software Development*, Redwood City, CA, EUA, Addison Wesley Longman Publishing Co., Inc.
8. Código Nacional de Procedimientos Penales (Diario Oficial de la Federación 17/06/2016).
9. Ley Nacional de Mecanismos Alternativos de Solución de Controversias en Materia Penal (Diario Oficial de la Federación 29/12/2014).
10. Consejo de la Judicatura Federal, <http://www.cjf.gob.mx/>