



FACULTAD DE INGENIERÍA Y CIENCIAS AGROPECUARIAS

“SISTEMA PARA EL REGISTRO DE CITACIONES A TRAVES DE TELEFONOS MOVILES
CON NAVEGACION WEB (swCITAC)”

Trabajo de Titulación presentado en conformidad con los requisitos establecidos
para optar por el título de Ingeniero en Sistemas.

Profesor Guía:
Ing. Xavier Armendáris

Autor:
Gonzalo Fernando Ibarra Cordovilla

Año:
2013

DECLARACIÓN PROFESOR GUÍA

Declaro haber dirigido este trabajo a través de reuniones periódicas con los estudiantes, orientando sus conocimientos y competencias para un eficiente desarrollo del tema y tomando en cuenta la Guía de Trabajos de Titulación correspondiente.

.....

Ing. Xavier Armendáriz
Ingeniero en Sistemas
C.C: 171133783-0

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

Declaro que este trabajo es original, de mi autoría, que se han citados las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes.

.....

Gonzalo Fernando Ibarra Cordovilla
C.C:1500455843

AGRADECIMIENTO

Primeramente deseo agradecer a Dios por ser fuente de motivación en los momentos de angustia y después de varios esfuerzos, dedicación, aciertos y reveses que caracterizaron el desarrollo de mi formación profesional me supo guiar con su luz divina para no desmayar por este camino que hoy veo realizado.

A la UDLA, mi casa de estudios, a todos mis maestros que aportaron a mi formación.

A mis ex compañeros de la Dirección Provincial del Consejo de la Judicatura que me brindaron el apoyo necesario.

A mi director de tesis, Ingeniero Xavier Armendáris por su confianza y apoyo en la realización del presente proyecto.

DEDICATORIA

A:

Dios, por darme la oportunidad de vivir y estar conmigo en cada paso que doy.

Mi hijo, Juan Sebastián por ser la persona que me dio la motivación que me hacía falta para concluir este proyecto.

Mi esposa, Mónica Mendoza por el apoyo y la comprensión que me supo dar durante la realización del presente proyecto.

A mis padres, porque creyeron en mí y porque me sacaron adelante dándome ejemplos dignos de superación y entrega, porque en gran parte gracias a ustedes, hoy puedo ver alcanzada mi meta, ya que siempre estuvieron impulsándome en los momentos más difíciles de mi carrera.

A mi hermano Juan Carlos por su gran apoyo en todo momento.

A mis ñaños, amigos.

Gracias por haber fomentado en mí el deseo de superación y el anhelo de triunfo en la vida.

RESUMEN

Este proyecto de tesis se realizó con el objetivo de analizar y dar una solución al trabajo diario que realizan los Citadores de la Dirección Provincial del Consejo de la Judicatura Distrito-Guayas.

Se crea un programa denominado SwCITAC, el mismo incorpora el framework ASP.NET MVC 4 dentro de sus principales características. También se estudian los elementos importantes utilizados para el desarrollo, se escoge la metodología a usar conocida como RUP.

Con esta metodología de desarrollo de Software basada en Objetos y UML se realiza el diseño, ingeniería de negocios, implementación y pruebas.

Con un dispositivo móvil (Celular) y un paquete de datos(acceso al internet) para poder navegar, el citador cuenta con el servicio de poder acceder al proyecto por intermedio del browser del navegador de su preferencia para poder realizar sus citaciones en línea, esto debido a que se utiliza ambiente Web para el desarrollo.

Como consecuencia de todo esto se entrega una aplicación para teléfonos móviles con navegación WEB donde el usuario (Citador) puede realizar su trabajo diario en un ambiente amigable y fácil de usar.

ABSTRACT

This project was conducted in order to analyze and provide a solution to the daily work performed by Citadores of the Provincial Council of the Judiciary District-Guayas.

It creates a program called SwCITAC, it incorporates the ASP.NET MVC framework 4 within its main features. It also discusses the important elements used for development, we choose to use the methodology known as RUP.

With this software development methodology based on UML Objects and design is done, business engineering, implementation and testing.

With a mobile device (cell) and a data packet to navigate, the-quoting service has to access the project through the browser the browser of your choice to perform their citations online, because it is used web environment for development.

As a result of all this comes an application for mobile phones with Web browsing where the user (-quoting) can perform their daily work in a friendly and easy to use.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO I	5
1. PROBLEMA A RESOLVER	5
1.1. DETERMINACION DE PROBLEMAS	5
1.1.1 Proceso de Citaciones forma manual	5
1.2 OBJETIVO GENERAL	6
1.3 OBJETIVOS ESPECÍFICOS	6
CAPITULO II	7
2. MARCO TEÓRICO	7
2.1. LEVANTAMIENTO DE INFORMACIÓN	7
2.1.1 A quien se DIRIGE el proyecto	7
2.1.2 obtención de requerimientos	7
2.2. METODOLOGÍA A USARSE	8
2.2.1. UML (lenguaje de modelado unificado)	10
2.2.2. RUP	13
2.3. IEEE 830 ESPECIFICACIONES DE	
REQUISITOS DE SOFTWARE	18
2.4. BENEFICIOS DE DESARROLLAR LA ERS	19
2.5. CARACTERÍSTICAS DEL SOFTWARE	20
2.5.1. Definiciones previas	20
2.5.2. Análisis de las herramientas de software	28
2.5.3. DEFINICIÓN DE HERRAMEINTAS PARA swCITAC	44
CAPITULO III	45
3. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN	45
3.1. ANÁLISIS DE REQUERIMIENTOS	45

3.1.2 ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE	45
3.2. USO DE METODOLOGÍA	47
3.2.1 MODELANDO DEL NEGOCIO	48
3.3. OBTENCION DE REQUISITOS	50
3.3.1. ENTREVISTAS	52
3.3.2. RECOLECCION DE DATOS	52
3.4. ANALISIS Y DISEÑO	53
3.4.1. ARQUITECTURA Y CLASES	53
3.4.2. CLASES Y METODOS DE SwCITAC	57
3.4.3. MODELO ENTIDAD RELACION	59
3.4.4. Identificacion de casos de uso	62
3.4.5. DIGRAMA DE ACTIVIDADES	71
3.4.6. DIAGRAMA DE ITERACIONES	82
3.4.7 DICCIONARIO DE DATOS	90
3.5. IMPLEMENTACIÓN EN LA WEB	93
3.6. PRUEBAS	93
3.6.1. Realización de pruebas	93
3.6.2. RESULTADOS DE PRUEBAS	96
3.6.3. REUNIONES DE ENTREGA Y USO	96
3.7 EVALUACIÓN FINAL	97
CAPITULO IV	98
4. CONCLUSIONES Y RECOMENDACIONES	98
4.1. CONCLUSIONES	98
4.2. RECOMENDACIONES	99
REFERENCIAS	100
ANEXOS	104

INTRODUCCIÓN

Lo expuesto en este documento corresponde al desarrollo de un sistema informático diseñado para la Dirección Provincial del Consejo de la Judicatura distrito Guayas, específicamente al departamento de Citaciones. Esta entidad es una institución no financiera que pertenece al Sector Social del Gobierno Central, la misma que está regulada, por leyes, reglamentos, normas y procedimientos legales; las cuales son emitidas por los organismos rectores (Contraloría General del Estado, Consejo de la Judicatura).

Al tratarse de un sistema dirigido específicamente al Departamento de "Citaciones", se pretende con este facilitar y agilizar el trabajo realizado por el personal de este departamento, dotando de esta manera una solución a uno de los varios problemas inherentes a la entrega de las citaciones.

El presente trabajo se encuentra desarrollado en cuatro capítulos, que se los describen de una forma general a continuación:

En el capítulo I, se define el planteamiento del problema, justificación del tema, los objetivos generales y específicos, la justificación del porqué de este trabajo y la definición conceptual y operacional.

En el capítulo II, se muestra la teoría acerca del desarrollo del proyecto, se describe el marco metodológico, las técnicas e instrumentos de recolección de datos, la selección de las herramientas de desarrollo así como determinación de la metodología.

En el capítulo III, se refiere al desarrollo del proyecto con las fases del desarrollo marcadas por la metodología seleccionada.

En el capítulo IV se presentan las conclusiones y recomendaciones del resultado del desarrollo del proyecto, por último se describe la referencia utilizada con los anexos respectivos producto del desarrollo de este proyecto.

TITULO DEL PROYECTO

“SISTEMA PARA EL REGISTRO DE CITACIONES VÍA WEB ATRAVES DE TELEFONOS MOVILES”

SITUACIÓN ACTUAL

Los procesos para la entrega de citaciones así como el acceso de los usuarios externos para verificar las citaciones de los diferentes juicios, que se realizan en esta institución son muy complicados, esto se deriva del elevado número de procesos que tiene esta institución.

Para que los citados conozcan que tienen un trámite judicial se utiliza personal interno que se encarga de repartir las citaciones en las direcciones domiciliarias o de trabajo que han registrado en los datos de contacto.

No existe un verdadero seguimiento de cuantas citaciones se entrega y cuantas de estas culmina con la presencia de los involucrados a las audiencias, además el control de estas entregas no son registradas en ningún sistema que permita hacer las verificaciones y seguimientos.

Cada una de las citaciones son realizadas de forma manual, esto ocasiona pérdida de tiempo tanto para los empleados de esta judicatura como la desviación de información a los citados.

UBICACIÓN

- Dirección: Av. 9 de Octubre entre Quito y Pedro Moncayo esquina.
- Telefax: 04-232-1600

- Correo electrónico: rr.pp@funcionjudicial-guayas.gob.ec

MISIÓN – DIRECCIÓN PROVINCIAL CONSEJO JUDICATURA DISTRITO GUAYAS

Administrar, vigilar y controlar con calidad, los recursos humanos, financieros, materiales y tecnológicos, para optimizar la administración de justicia y los servicios que ofrece en beneficio de los usuarios del Distrito del Guayas y Galápagos.

Fundamentada en valores esenciales: su gente, liderazgo, calidad, solidaridad y mística de servicio; contribuir a construir una cultura de seguridad vial.

VISIÓN – DIRECCIÓN PROVINCIAL CONSEJO JUDICATURA DISTRITO GUAYAS

Lograr que el usuario de las Provincias del Guayas y Galápagos, nos reconozcan como un referente positivo de la gestión pública, el cual contribuye activamente a la modernización de la Administración de Justicia de este Distrito, de manera transparente e independiente.

VALORES – DIRECCIÓN PROVINCIAL CONSEJO JUDICATURA DISTRITO GUAYAS

- Ética y Transparencia en cada una de nuestras actuaciones.
- Compromiso de buen servicio a la ciudadanía.
- Agilidad y Honestidad.
- Mejoramiento continuo en cada sector de la Administración de Justicia.
- Justicia y Equidad.

ESTRUCTURA – DIRECCIÓN PROVINCIAL CONSEJO JUDICATURAL DISTRITO GUAYAS

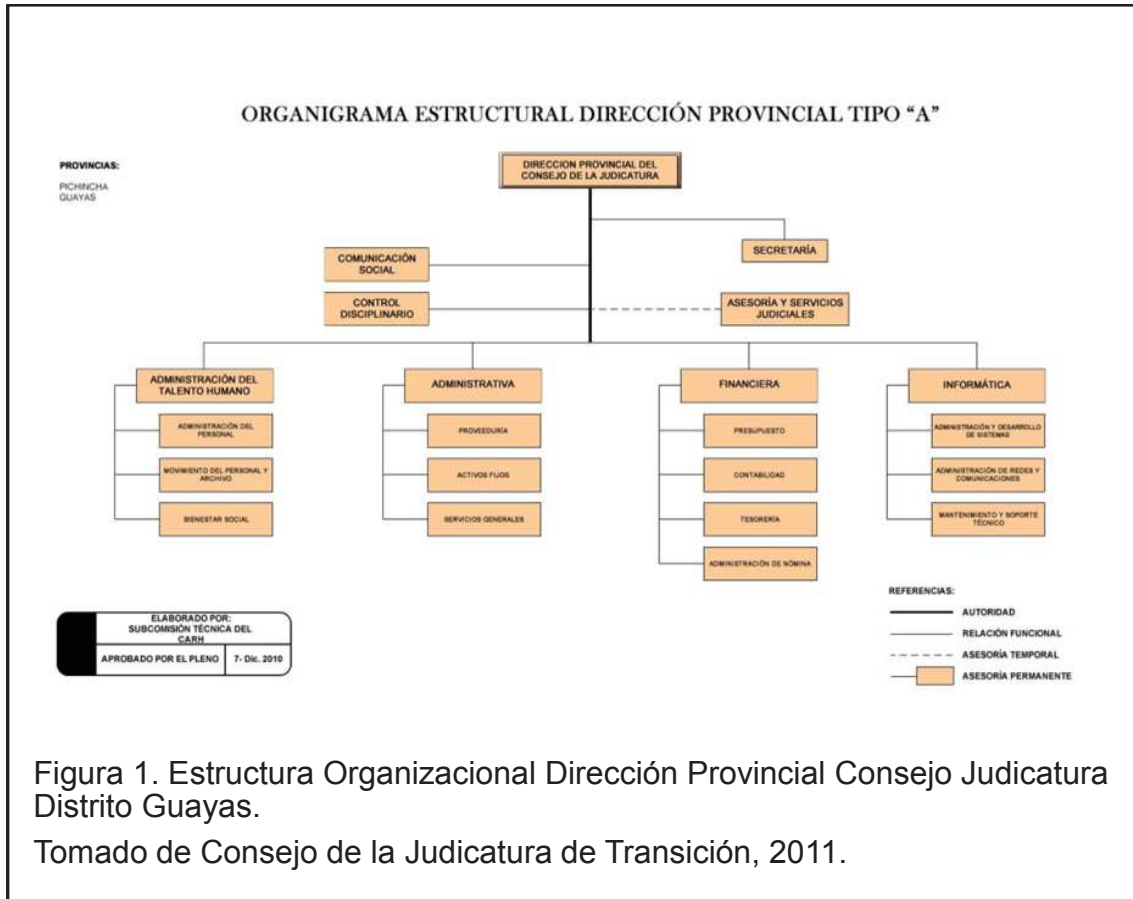


Figura 1. Estructura Organizacional Dirección Provincial Consejo Judicatura Distrito Guayas.

Tomado de Consejo de la Judicatura de Transición, 2011.

CAPITULO I

1. PROBLEMA

1.1. DETERMINACION DEL PROBLEMA

1.1.1. PROCESO DE CITACIONES FORMA MANUAL

El proceso de Citaciones en la Dirección Provincial del Consejo de la Judicatura Distrito Guayas es el principal problema, porque su debilidad es no tener información en línea de la repartición del gran volumen de las citaciones diarias que realiza los diferentes Juzgados de esta Judicatura de los diferentes juicios en procesos, en la actualidad este proceso de citaciones lo realizan en forma diaria y manual, no permitiendo llegar a establecer los registros de entrega en forma oportuna, al contrario deben esperar concluir la jornada de trabajo en el mejor de los casos o la siguiente día para alimentar en el registro la información la misma que tiene un margen de error muy alto, este problema se detectó por múltiples reclamos de los citados y los propios jueces; esto se puede solucionar con la implementación del sistema que funcione en forma automatizada al momento de la entrega de las citaciones, a la vez permitirá el sistema entregar, registrar las citaciones en forma oportuna, confiable, veraz y en tiempo real, en el mismo momento de realizar el proceso de citaciones.

Aún continúa el personal del Departamento de Citaciones, realizando el proceso de éstas en una hoja de trabajo en forma manual y personal por cada proceso, esta hoja es llenada en forma manual al momento de la entrega de la citación a la persona procesada; y, al final de la jornada deben acercarse a las instalaciones de la Institución para que en el respectivo Departamento procedan a iniciar con el registro de las citaciones entregadas a la base de datos descrita anteriormente.

Si las citaciones siguen realizando mediante este proceso manual, aumenta el riesgo de pérdida de tiempo, pérdida de documentos, error de registro, incluso pueden extraviar las hojas de trabajo antes de su registro, lo que seguirá aumentando el número de perjudicados dentro del proceso judicial.

Por lo expuesto anteriormente el proyecto del sistema que se desarrollará permitirá resolver este inconveniente existente, el mismo que facilitara optimizar el tiempo en la entrega de las citaciones, registros de las citaciones en línea; y, disminuyendo casi al 100% el error de subregistros, omisiones de las citaciones y la pérdida de los documentos de las citaciones.

1.2 OBJETIVO GENERAL

Diseñar e implementar un sistema para el registro de citaciones vía web a través de teléfonos móviles (swCITAC).

1.3 OBJETIVOS ESPECÍFICOS

- Determinar el proceso actual del sistema de entrega de citaciones en la Dirección Provincial del Consejo de la Judicatura distrito Guayas.
- Analizar, diseñar e implementar un SISTEMA PARA EL REGISTRO DE CITACIONES A TRAVES DE TELEFONOS MOVILES en ambiente web, considerando aspectos de seguridad de acuerdo al tipo de información.
- Registrar la información para alimentar los reportes existentes en el sistema actual SATJE, los mismos que permitirán a las autoridades realizar un seguimiento del ingreso de citaciones a la base de Datos.

CAPITULO II

2. MARCO TEÓRICO

2.1. LEVANTAMIENTO DE INFORMACIÓN

2.1.1. A QUIEN SE DIRIGE EL PROYECTO

Este proyecto de Titulación se encuentra dirigido a los citadores de la Dirección Provincial del Consejo de la Judicatura distrito Guayas, para que de esta manera estos se sientan satisfechos con su trabajo diario.

El Principal objetivo es brindar al Citador la manera de conectarse con su trabajo diario haciendo uso de la nueva tecnología, desde cualquier lugar donde exista servicio de Internet; Ofreciendo una interfaz de Usuario amigable, sencilla y eficiente.

2.1.2. OBTENCIÓN DE REQUERIMIENTOS

En el desarrollo de un Sistema de Software, la obtención de requerimientos es una etapa primordial y crítica; ya que dependiendo de los resultados los usuarios utilizarán o no el sistema, en otras palabras, se determina si el proyecto es factible. Por tanto, se ha creído correcto realizar entrevistas presenciales, a las personas directamente vinculadas al éxito del mismo.

2.1.2.1. PRINCIPAL PROBLEMA DEL AREA DE CITADORES

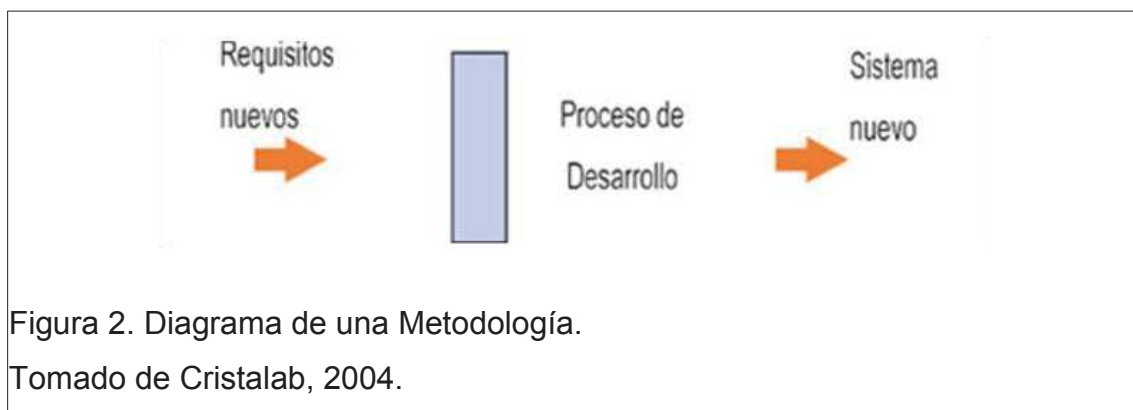
Como se mencionó en la determinación del problema, se puede notar claramente que los citadores realizan doble trabajo, esto causa un malestar laboral ya que luego de entregar las citaciones a los usuarios tienen que retornar a sus puestos de trabajo para ingresar los Datos al Sistema.

Este inconveniente se trata de mermar en su totalidad ya que el objetivo de este nuevo programa es orientado a la satisfacción del Citador, se ha recopilado la información mediante encuestas.

Las entrevistas se realizó al personal de citaciones encontrados en sus puestos de trabajo en la Dirección Provincial del Consejo de la Judicatura Distrito Guayas, en un periodo de 1 semana.

2.2. METODOLOGÍA A USARSE

Una metodología de desarrollo de software permite estructurar, planificar y controlar el proceso de desarrollo en sistemas de información, definiendo Quién debe hacer Qué, Cuándo y Cómo debe hacerlo.



El objetivo del proyecto es proporcionar un programa modular y flexible y, para esto, se debe emplear una metodología de desarrollo de software. RUP (Rational Unified Process), que junto con el Lenguaje Unificado de Modelado UML, forman una metodología estándar la misma que es utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

La metodología RUP ha dado un gran avance para facilitar el desarrollo en la forma de diseño, desarrollo y mantenimiento de software, por lo que se ha decidido tomar esta metodología como referencia por la siguiente razón:

- Brinda una solución al problema que el código no podía ser rehusado ni modificado, o sea la falta de portabilidad de código.

Las condiciones ya evaluadas pueden cambiar a lo largo del tiempo, por condiciones propias de legislatura o del mercado, por lo que el sistema debe ser realizado necesariamente de manera modular; dado que si en un futuro, la Dirección Provincial del Consejo de la Judicatura Distrito Guayas requiere la incorporación de nuevas funcionalidades o módulos, éstos pueden ser incorporados sin necesidad de desechar el programa anterior.

La metodología OO permite realizar un programa totalmente modular y por consiguiente la reutilización y modificación del código.

Por esta razón, se ha escogido la metodología RUP (Rational Unified Process), la cual no solamente usa UML, sino que a su vez, presenta una guía para todo el proceso de desarrollo.

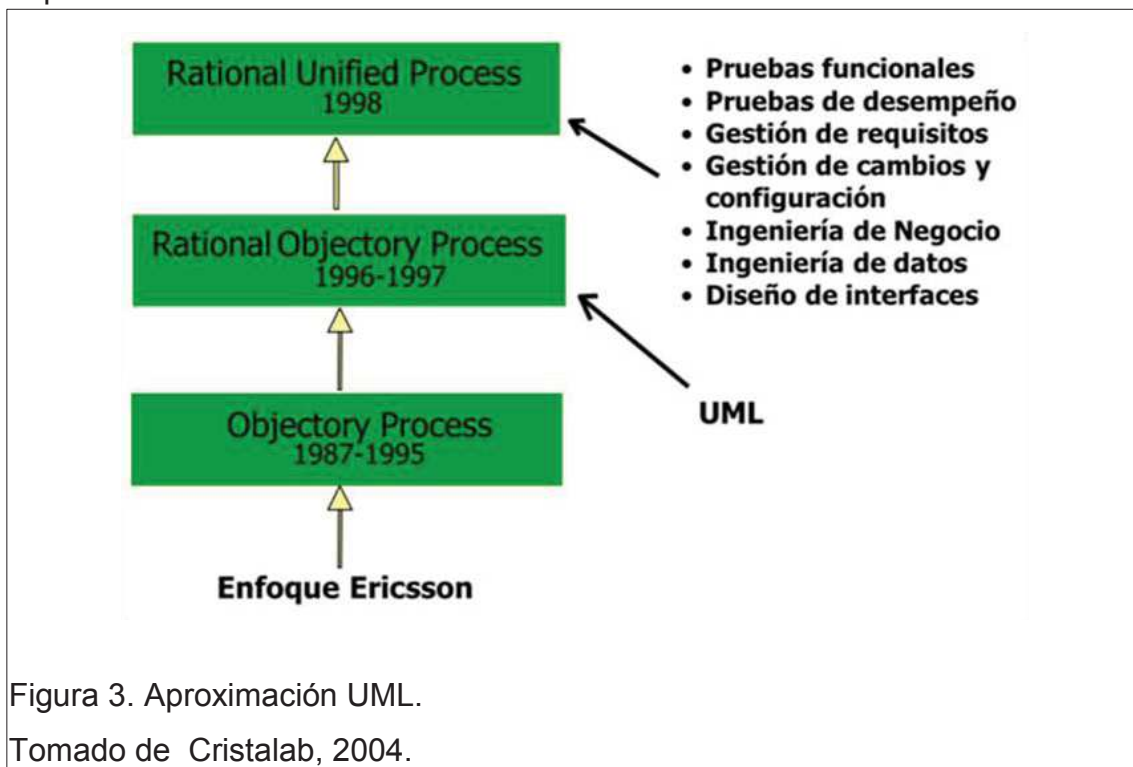


Figura 3. Aproximación UML.

Tomado de Cristalab, 2004.

Este es un proceso de ingeniería de Software muy bien estructurado, que a la vez permite que la aplicación se adapte a las necesidades y características específicas de la empresa.

Ahora, se definirá esta metodología con más de detalle:

2.2.1. UML (LENGUAJE DE MODELADO UNIFICADO)

“En la mayoría de personas, la forma de analizar OO es más natural que las técnicas del análisis y diseño estructurado. Después de todo, el mundo está formado por objetos. Si nos detenemos a pensar un poco, esto tiene sentido.

Desde una etapa muy temprana se categoriza los objetos y se descubre su comportamiento. Los usuarios finales y las personas de las empresas piensan de manera natural en términos de objetos, eventos y mecanismos de activación. Se puede crear diagramas OO que les parezcan familiares, mientras tengan dificultades con los diagramas de relación entre entes, tablas de estructura y diagramas de flujo de datos.” (James Martin, James J. Odell, 1997).

A continuación, se presenta un cuadro detallando las principales ventajas al usar una metodología basada en Objetos, y a su vez, sustentado la elección de esta metodología:

Tabla 1. Ventajas de una Programación Orientada a Objetos

FACTOR	DETALLE
Reutilización de Código	El Código mediante objetos y clases permite la reutilización de código.
Estabilidad	Las clases al ser reutilizadas hacen que el programa se comporte de manera más estable.
Complejidad	Términos como encapsulamiento, herencia, polimorfismo hacen que el diseño de algunas clases resulte más sencillo.
Calidad	Los diseños suelen tener mejor calidad, puesto que se construyen a partir de componentes probados, que han sido verificados y pulidos varias veces.
Sencillez	Los programas se elaboran a partir de piezas

FACTOR	DETALLE
	pequeñas las cuales ayudan en gran magnitud al desarrollo del mismo.
Mantenimiento	Cada clase efectúa sus funciones independientemente de las demás. Por lo cual los errores serán de fácil ubicación. Los objetos al ser independientes, varios programadores pueden corregir errores al mismo tiempo.
Tiempo	Al poder reutilizar el código es un ahorro significativo de tiempo.
Independencia	Las clases están diseñadas para ser independientes del ambiente, plataformas, hardware y software.

Tomado de James Martin, James J. Odell, 1997

UML es un estándar que nace en 1994, el cual permite especificar, visualizar y construir los artefactos de los sistemas de Software. Actualmente esta metodología agrupa varios enfoques correspondientes a distintos desarrolladores lo que se representa en la Figura 4.

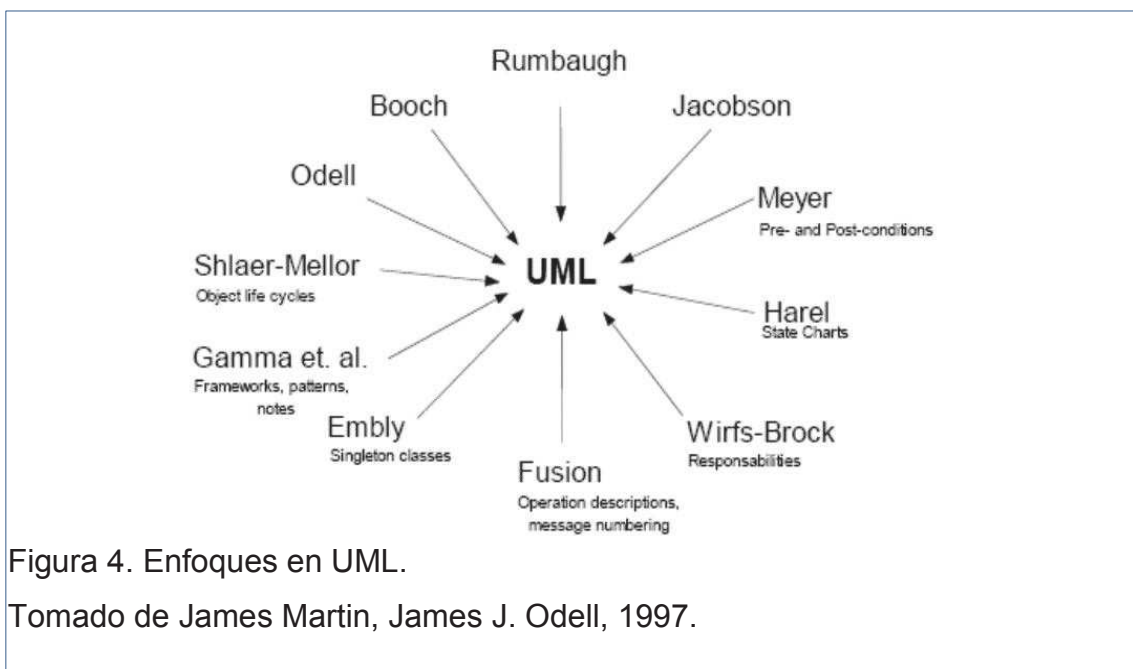


Figura 4. Enfoques en UML.

Tomado de James Martin, James J. Odell, 1997.

2.2.1.1. DEFINICIONES DE UML

- **Modelo:** captura una vista de un sistema del mundo real. Es una abstracción de dicho sistema, considerando un cierto propósito. Así, el modelo describe completamente aquellos aspectos del sistema que son relevantes al propósito del modelo, y a un apropiado nivel de detalle.
- **Diagrama:** una representación gráfica de una colección de elementos de modelado, a menudo dibujada como un gráfico con vértices conectados por arcos.
- **Paquete:** Ofrece un mecanismo general para la organización de los modelos/subsistemas agrupando elementos de modelado. Un paquete puede tener sub-paquetes sin límite de anidamiento.
- **Casos de Uso:** Es una técnica para capturar información de cómo el sistema trabaja, o como desea que trabaje.

Estos términos son de importancia al entender que un sistema está compuesto por un conjunto de modelos mismos que permiten expresar a un producto desde cada una de las respectivas perspectivas de interés.

El código fuente viene a ser el modelo más detallado del sistema, sin embargo es necesario tener otras visiones del sistema.

UML se encuentra conformado de los siguientes diagramas:

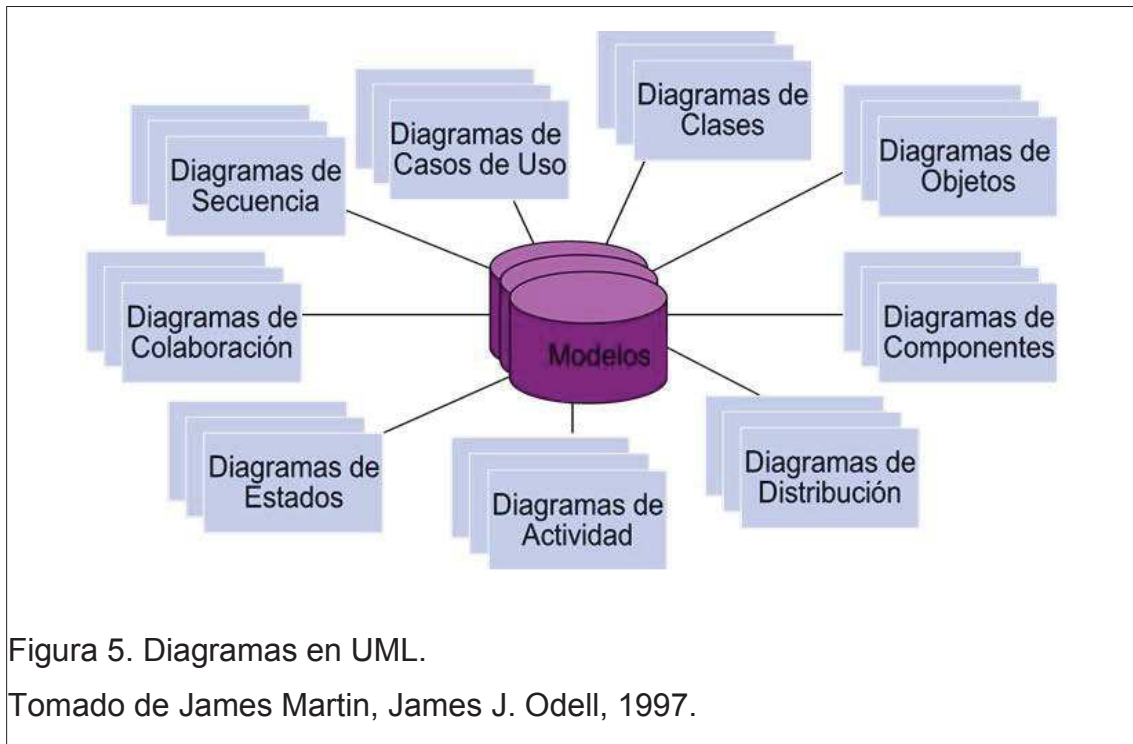


Figura 5. Diagramas en UML.

Tomado de James Martin, James J. Odell, 1997.

“Para modelar un sistema es suficiente utilizar una parte de UML, con un 80% de la mayoría de los inconvenientes pueden modelarse usando alrededor del 20% de UML” adaptado de Grady Boch. Universidad de las Américas Puebla, 2012.

Es por esta razón no se realizarán todos los diagramas que constan en UML, solamente los más importantes.

2.2.2. RUP

2.2.2.1. HISTORIA DEL RUP

A continuación, se presenta la Figura 6, la cual detalla la historia que ha venido siguiendo RUP:

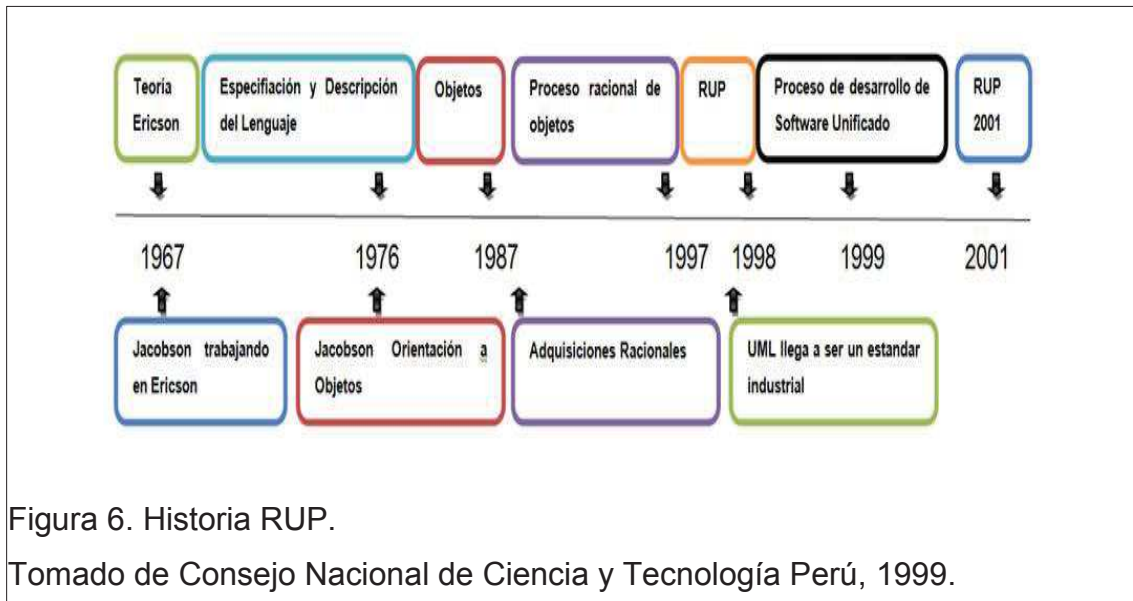


Figura 6. Historia RUP.

Tomado de Consejo Nacional de Ciencia y Tecnología Perú, 1999.

Un acontecimiento importante se da por el año 1967 con la Metodología Ericsson (Ericsson Aproxche), elaborada por Ovar Jacobson, con esta metodología se realizó una aproximación de desarrollo basada en componentes, la cual introdujo el concepto de Caso de Uso.

Jacobson creó la compañía Objektor AB con lo cual difundió el proceso de desarrollo Objektor que es una abreviación del Objeto Factor esto sucedió por los años 1987 a 1995.

En el año 1995, Rational Software Corporación se hace de los derechos de Objektor AB, con esto entre los años 1995 y 1997 desarrollaron Rational Objektor Procesos (ROP), basados en la versión del Objektor 3.8 y del Enfoque Racional (Racional Aproxche), tomando UML como lenguaje de modelado.

Encabezados de Grady Boch, Ovar Jacobson y James Rumbaugh, Rational Software desarrolla e incorpora diversos elementos expandiendo de esta manera la metodología ROP; sobresaliendo fundamentalmente el flujo de trabajo más conocido como modelado del negocio. Dando origen para que en el mes de junio del año 1998 salga al mercado Rational Unified Process.

2.2.2.2. CARACTERISTICAS ESENCIALES DE RUP

2.2.2.2.1. CICLO DE VIDA DE RUP

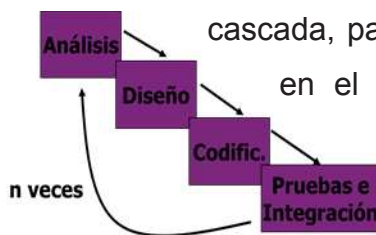
- RUP se divide en cuatro fases, dentro de cada una se realizan varias iteraciones en número variable según el tipo de proyecto que se desea realizar.
 - Iniciación (Puesta en marcha).
 - Elaboración DAD (definición, análisis, diseño).
 - Construcción (Implementación).
 - Transición (Fin del proyecto y puesta en producción).

- Dirigido por Casos de Uso: la gran ventaja que presenta el usar Casos de Uso es que permiten integrar el trabajo.



Los Casos de Uso son la integración del Trabajo; Hace capturas, validaciones y definición de requisitos, para luego de esto, analizar, diseñar, implementar y probar.

- Proceso Iterativo e Incremental: Las actividades se encadenan en una cascada, para luego volverse a repetir con el objetivo de que si en el último escalón se produce un error, se lo corrija siguiendo el mismo proceso.



- Adaptar el Proceso: Las características propias del proyecto o empresa deben ser totalmente cubiertas por el proceso. La magnitud del mismo, así como su naturaleza o las regulaciones que lo estipulen van a influir directamente en su diseño específico.
- Equilibrar Propiedades: Los requerimientos pueden variar dependiendo de la persona que lo solicite, por tanto, es necesario encontrar el equilibrio

adecuado para satisfacer los deseos de todos los involucrados, encontrado el equilibrio se podrá corregir desacuerdos que pueden presentarse en el futuro.

- Nivel de Abstracción: Se deben escoger las herramientas adecuadas, y utilizarlas apropiadamente para que el usuario no perciba nada más que su interfaz gráfica.
- Calidad: Debe asegurarse que cada fase lleve un control de Calidad y no esperar al final, es decir se debe incluir dentro del proceso del desarrollo al aseguramiento de la calidad y no ser tomado como un proceso de un grupo independiente.

MODELO DEL NEGOCIO

En esta parte del modelo se define cuáles son los procesos y procedimientos del porque se va a desarrollar la aplicación. Identificando de esta manera los casos que deben ser automatizados, realizando una relación directa entre el cliente y el software.

Con el modelo del negocio se podrá conocer cómo se encuentran funcionando actualmente los procesos dentro de la organización, además de comprender la estructura y la dinámica de la empresa.

REQUISITOS

También conocido como análisis de requerimientos, se puede decir que esta etapa es la más importante en el desarrollo ya que aquí se determinan las condiciones que debe cumplir el sistema que se necesita diseñar, satisfaciendo de esta manera las necesidades de los usuarios.

Se entiende como requerimiento a las necesidades y deseos que tienen el o los usuarios finales respecto a un producto de software.

ANALISIS Y DISEÑO

En esta etapa se establece que cada uno de los requerimientos especificados en la etapa correspondiente deben ser transformados a diseños del sistema, con el objetivo de alcanzar una arquitectura robusta para el sistema.

IMPLEMENTACION

Esta etapa tiene como propósito implementar clases y objetos que den como resultado un sistema con calidad. Para que esto se cumpla es recomendable planificar el orden de implementación de cada módulo los mismos que deben ser probados para determinar el cumplimiento de los requerimientos.

PRUEBAS

Esta etapa tiene su enfoque en la evaluación y aseguramiento de la calidad del sistema, este será sometido a pruebas con la finalidad de encontrar errores los mismos que deberán ser corregidos a tiempo, esto quiere decir antes de poner en producción el sistema, evitando de esta manera costos elevados al tratar de corregir errores de un sistema en marcha.

DESPLIEGUE

El objetivo de esta etapa es realizar pruebas al sistema antes de ponerlo en producción aquí también se hace la migración de datos, dando paso así a la distribución del sistema a los usuarios finales.

GESTION DE CAMBIO Y CONFIGURACION

Radica en analizar previamente los cambios para tomar la decisión si pueden ser ejecutados manteniendo así la integridad del sistema. Cada cambio realizado es necesario documentarlo.

GESTION DEL PROYECTO

La gestión del proyecto vigila el cumplimiento de los objetivos, los riesgos y las restricciones del sistema. Así se mantiene un conocimiento total de cómo evoluciona el sistema.

ENTORNO

Tiene como finalidad enfocarse a las actividades necesarias para configurar el proceso del sistema, además de determinar el ambiente donde se va a desarrollar. Se establece las herramientas, procesos y métodos que presten el soporte al equipo de desarrollo en la implementación de la aplicación.

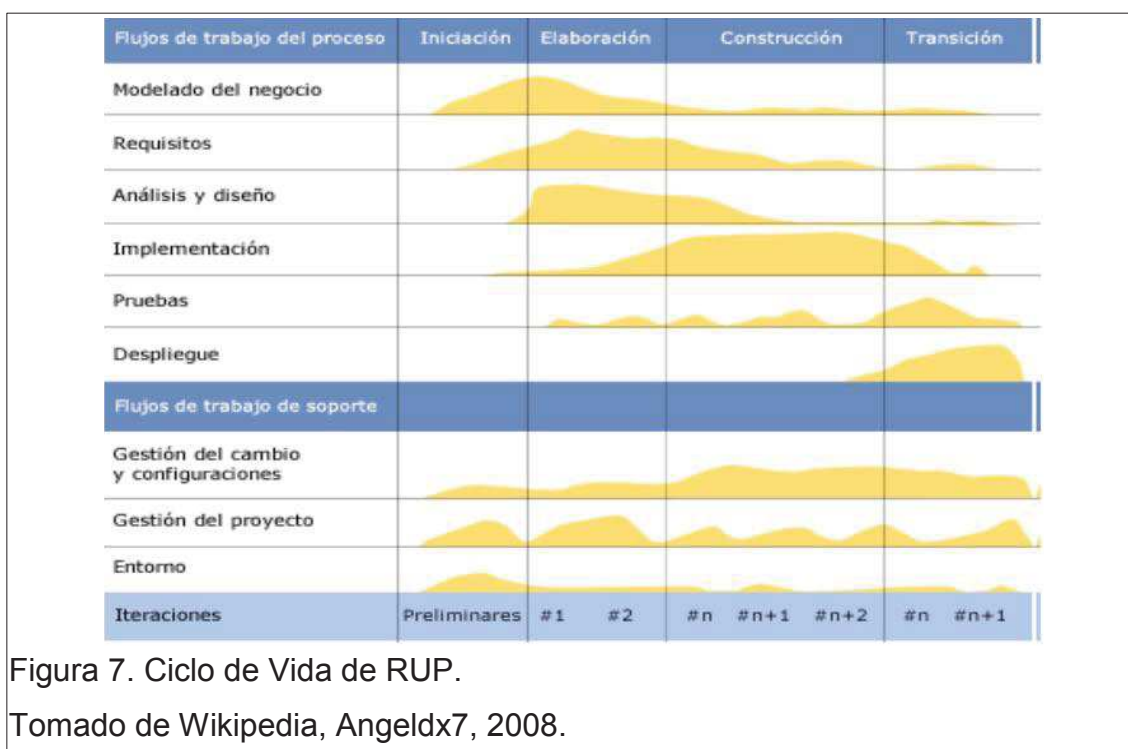


Figura 7. Ciclo de Vida de RUP.

Tomado de Wikipedia, Angeldx7, 2008.

2.3. IEEE 830 ESPECIFICACIONES DE REQUISITOS DE SOFTWARE

Según la página web, es.scribd.com, 2010. Describe la IEEE como “Documento que define de forma completa precisa y verificable, los requisitos, el diseño, el comportamiento u otras características de un sistema o componentes de un sistema”

La determinación de los requerimientos es una de las tareas más primordiales para el ciclo de vida en el desarrollo de software, ya que aquí se determina la nueva estructura de la aplicación.

El no realizar una buena especificación de requerimientos trae consigo que los costes del desarrollo incrementen, esto por los cambios que se pueden dar durante la implementación de la aplicación; el tiempo y el dinero serían unos grandes inconvenientes para las partes involucradas.

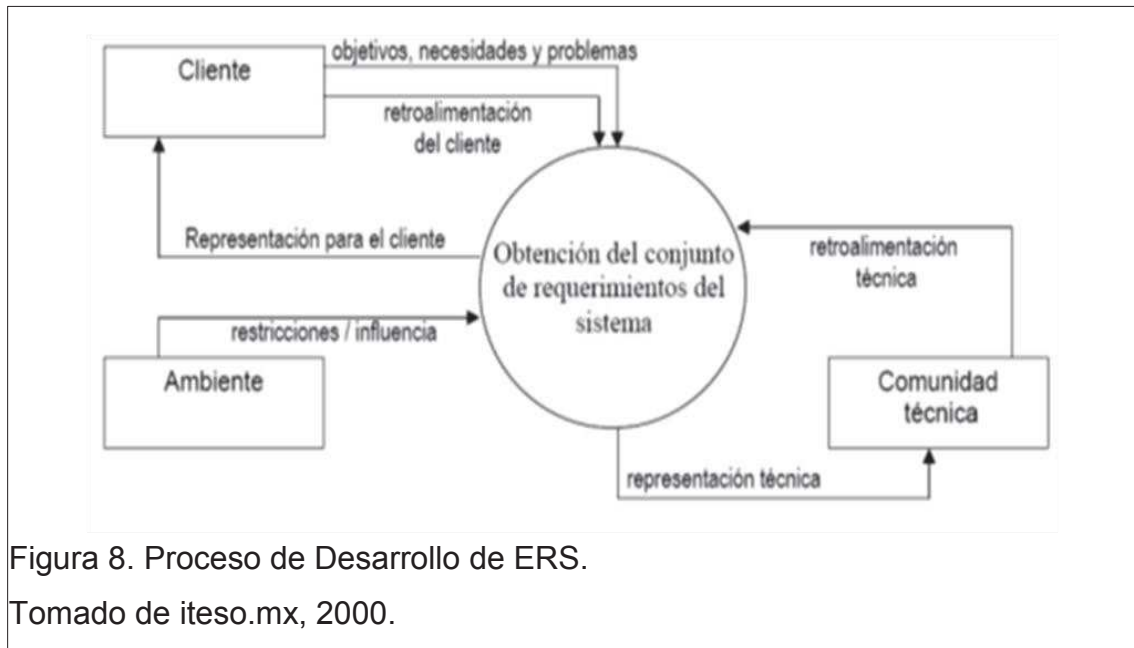
La ERS abarca el contenido de lo que un sistema debe realizar en términos de las interacciones, o las interfaces con su ambiente externo. Es decir la ERS describe las entradas, salidas y las relaciones que deben tener entre las mismas. El cliente es el encargado de proporcionar toda esta necesaria información.

2.4. BENEFICIOS DE DESARROLLAR LA ERS

- Representa una partida para calificar la calidad que tiene el sistema con la verificación de las necesidades del cliente.
- Permiten interactuar al cliente con el desarrollador de forma directa para que determinen las necesidades para desarrollar la aplicación, obteniendo de esta manera una implementación sin mayores contratiempos en un futuro.
- Permite el desarrollo de todas las fases según el ciclo de vida de una forma eficiente.
- Facilita a los desarrolladores y al cliente detectar problemas y corregir, cuidando que los costes sean económicos.

DESCRIPCION DEL PROCESO DE DESARROLLO DE LA ERS

En la Figura 8 se observa el proceso de desarrollo de la Especificación de Requerimientos, se muestra la relación entre el cliente, ambiente y desarrolladores.



2.5. CARACTERÍSTICAS DEL SOFTWARE.

2.5.1. DEFINICIONES PREVIAS

2.5.1.1. NET FRAMEWORK

Un Framework es un soporte definido, en el cual se apoyan otras aplicaciones para poder ser desarrolladas. Generalmente, consta de programas y bibliotecas los mismos que son necesarios para ligar los diferentes componentes de la aplicación.

Dentro de los Sistemas Operativos; Windows Server 2008, Windows Vista / 7; .Net Framework ya está incluido, también es posible instalar de manera manual en Windows Server 2003 y Windows XP, también existe una versión reducida que se encuentra en dispositivos inteligentes en el Sistema Operativo Windows Mobile.

Algunos componentes para .NET Framework son:

- Dentro del entorno común en la ejecución de los lenguajes o CLR, se administra el código en tiempo de ejecución, proporcionando los servicios

básicos (administración de memoria, control de excepciones, control de hilos de ejecución). Este es el verdadero núcleo de .NET Framework.

- Es la herramienta de desarrollo la encargada de compilar el código fuente de cualquier lenguaje soportado por .NET en un código intermedio llamado MSIL (Microsoft Intermediate Language), este código es similar al BYTECODE de Java; donde para generarlo, el compilador se basa en la especificación CLS (Common Language Specification) que es la parte encargada de determinar las reglas necesarias para crear el código MSIL compatible con el CLR. Para ejecutarse se necesita un segundo paso, un compilador JIT (Just-In-Time) el que genera el código máquina real que se ejecuta en la plataforma del cliente.
- Esta es la forma de conseguir con .NET independencia de la plataforma de hardware. La compilación JIT la realiza el CLR a medida que el programa invoca métodos. Al obtener el código ejecutable este se almacena en la memoria caché del ordenador, si el caso es que existe algún cambio en el código fuente por medio de la solicitud podrá ser recompilado nuevamente el código almacenado.
- Quien define un conjunto funcional para que sea soportado por un Sistema Operativo es la Biblioteca de Clases Base o BCL de .NET Framework misma que contienen los tipos básicos como clases de entrada/salida de datos (XML, TCP/IP), seguridad, manejo de dispositivos periféricos, etc.
- La Biblioteca de Clases Base o BCL de .NET Framework incluye dos componentes ADO.NET y ASP.NET:
 - ADO.NET es derivado de ActiveX Data Object mismo que es usado para acceder a la información y servicios de datos en un SGBD (Sistema de gestor de Base de Datos).
- ASP.NET no es más que un Framework disponible para aplicaciones web. La primera aparición de ASP.NET fue junto con la versión de .NET Framework 1.0. Gracias al CLI Infraestructura común de Lenguajes, que es una especificación estandarizada cuya característica esencial es permitir que diferentes lenguajes de alto nivel puedan ejecutarse en múltiples plataformas sin necesidad de recompilar el código fuente, .NET soporta

actualmente alrededor de cincuenta lenguajes de programación, esto da origen a una gran versatilidad en el desarrollo de un programa, sin tener mayores diferencias si se programa con uno u otro lenguaje.

- Entre los lenguajes más usados en el desarrollo .NET están: C#, Visual Basic NET, PowerBuilder, C++, J#, Delphi (Object Pascal), Perl, Python, Fortran, Cobol.

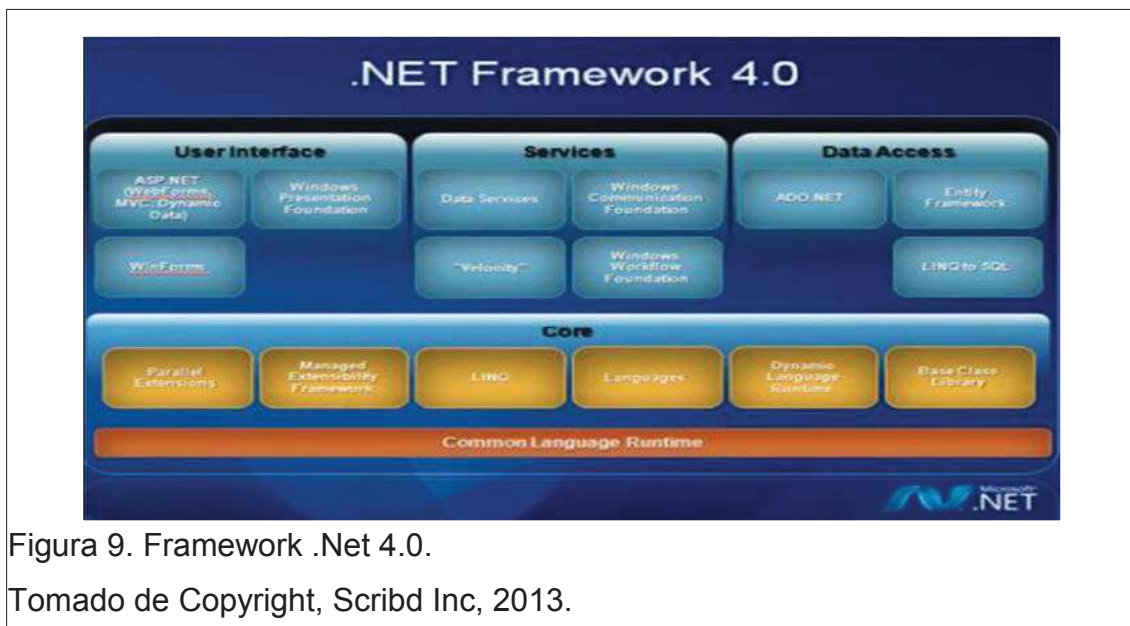


Figura 9. Framework .Net 4.0.

Tomado de Copyright, Scribd Inc, 2013.

.NET Framework 4 es completamente compatible con las aplicaciones que han sido compiladas con versiones anteriores de .NET Framework, no sucede lo mismo con respecto a ciertos cambios que se hicieron para el mejoramiento de la seguridad dando cumplimiento así con las normas, confiabilidad, exactitud y rendimiento.

.NET Framework 4 no utiliza automáticamente su versión de CommonLanguageRuntime para ejecutar aplicaciones que han sido compiladas con las versiones anteriores de .NET Framework. Si se desea ejecutar aplicaciones anteriores con .NET Framework, es necesario compilar la aplicación con la versión de .NET Framework de destino específica en las propiedades del proyecto en Visual Studio, o se puede especificar el runtime compatible mediante el <supportedRuntime> en un archivo de configuración de la aplicación.

A continuación se resume algunas características de esta versión de Framework 4.

Existen mejoras en Security, Parallel Computing, diagnóstico y rendimiento así también en el nuevo DLR (Dynamic Language Runtime) y otros.

Los desarrolladores de este Framework han dedicado mucho esfuerzo en mejorar la seguridad, para lo cual han incluido simplificación y transparencia en las implementaciones.

Esta versión de Framework incluye un nuevo modelo de programación paralela (multi-hilos). Con esta nueva idea lo que se pretende es simplificar el desarrollo de multithread permitiendo la escritura de código multi-hilos en lenguaje natural.

Dentro del CLR se encuentra un nuevo Runtime esto para la ejecución de código dinámico (Dynamic Language Runtime). La idea del DLR es facilitar y simplificar el desarrollo del código dinámico en .NET. El DLR se encarga de ejecutar código en C# o VB en tiempo de ejecución.

En lo concerniente a aplicaciones web este Framework incluyen mejoras en ASP.NET, Dynamic Data, Web Forms, etc.

Dentro de ASP.NET Web Forms podemos citar a continuación algunas mejoras:

- Mejoras con el control del View State.
- Posibilidad de establecer meta tags.
- Mejoras con el renderizado de los controles FormView y ListView.
- Mejoras con la generación de ID's de los controles ASP.NET.

2.5.1.2. ENTORNO INTEGRADO DE DESARROLLO (IDE)

Un IDE es un entorno de programación que a menudo consiste de un editor de código, un constructor de interfaz gráfica y un compilador.

Visual Studio 2010 es un IDE creado por Microsoft®. Éste presenta un conjunto de herramientas destinadas a escribir y modificar código, además a detectar y corregir errores en las diferentes aplicaciones. Microsoft Visual Studio 2010 Ultimate incorpora un completo conjunto de herramientas las cuales permiten la gestión de aplicaciones de ciclo de vida de los equipos garantizando calidad en sus resultados, empezando por el diseño hasta llegar a la implementación, no importa si se está creando nuevas soluciones o si se está mejorando las aplicaciones existentes, Visual Studio 2010 Ultimate le permite llevar su visión a la vida apuntando a un número cada vez mayor de plataformas y tecnologías- como la nube y la computación paralela.



Figura 10. Visual Studio 2010 Ultimate (Logo).

Tomado de Microsoft, Visual Studio editions, 2012.

Es importante destacar que si se instala Microsoft Visual Studio 2005 o posterior (excepto las ediciones Express), no es necesario instalar el SDK de .NET Framework por separado.

Visual Studio 2010 crea aplicaciones enriquecidas para SharePoint y la Web, también se incorpora nuevas y mejoradas características que ayudan a tener, desde el diseño a la implementación en el desarrollo un proceso más sencillo. El área de trabajo de Visual Studio 2010 es compatible con varios monitores esto gracias a la personalización que se puede realizar.

“Los requisitos que debe tener un sistema para un correcto funcionamiento en este IDE son:

- Con respecto al Sistema Operativo se puede instalar sin ningún inconveniente en Windows Server 2003; Windows Server 2008; Windows Vista; Windows XP, y Windows 7.
- Su hardware mínimo para la instalación es de 1.6 GHz CPU, 384 MB RAM, pantalla 1024x768, disco duro 5400 RPM
- Lo recomendado en hardware para un funcionamiento adecuado podría ser de 2.2 GHz o mayor de CPU, 1024 MB o mayor en RAM, pantalla de 1280x1024, 7200 RPM o mayor de Disco Duro.
- Si la instalación se va a realizar sobre Windows Vista o Windows 7 lo recomendable en hardware podría ser 768 MB RAM, 2.4 GHz CPU.”

Adaptado de Ultimate, Microsoft Visual Studio 2010, 2012.

2.5.1.3. INTERFAZ DE PROGRAMACION DE APLICACIONES (API)

Podríamos indicar que la API es un biblioteca, generalmente dinámica o DLL (Dynamic Link Library), utilizada para conseguir abstracción en la programación de las capas inferiores, brindando simplicidad en la programación, o incluso independencia del SO o del lenguaje de programación.

Visto desde código en bajo nivel (lenguaje de máquina), tiene como propósito proporcionar un grupo de funciones y procedimientos de uso general. La gran ventaja radica en que el programador evita el trabajo de programar

funcionalidades ya existentes, en este punto es indispensable indicar que, las APIs de alto nivel generalmente pierden flexibilidad; por tanto, es muy importante al momento de elegir una API hay que saber llegar a un equilibrio entre su potencia, simplicidad y pérdida de flexibilidad.

Entre las APIs más destacadas se encuentran: Microsoft WMI, Microsoft Win32 API, Microsoft Framework .NET, OpenGL, Java EE, entre otros.

2.5.1.4. KIT DE DESARROLLO SDK

Los SDK frecuentemente traen consigo también, ejemplos de códigos así como también notas técnicas para soporte o alguna otra documentación las cuales ayudan a clarificar ciertos puntos de donde se necesite este tipo de consulta.

SDK se presenta como un API incluyendo herramientas, documentación y ejemplos que necesitan los desarrolladores para escribir, generar, probar e implementar aplicaciones de .NET Framework. Es compatible con los SO de las familias Windows Vista, Windows 7, Windows Server 2003, Windows 2000, Windows XP.

Los SDK con frecuencia también llevan consigo códigos de ejemplo, notas técnicas de ayuda y otra documentación, esto con el objetivo de dar al usuario el apoyo necesario en ciertos puntos del material de referencia primario que el desarrollador necesite.

C#

Se trata de un lenguaje de programación orientado a objetos, fue creado por Microsoft e incluida desde un principio para formar parte de la plataforma .NET se podría utilizar para el desarrollo de aplicaciones en otras plataformas, para lo cual se necesitaría tener un compilador específico de C# en la plataforma que se requiera.

Analizando la sintaxis podríamos indicar que es similar a C++ o Java, según la empresa Microsoft C# se diseñó para recolectar en este lenguaje las virtudes

de C++, Visual Basic y Java, dando paso de la programación orientada a componentes a una programación ampliada en el lenguaje orientado a objetos. Siendo un lenguaje muy optimizado para .NET dentro de la plataforma de Microsoft, C# muy personalmente me atrevo a decir que es el lenguaje más adecuado para desarrolladores, por cuanto que, C# es el lenguaje oriundo de la plataforma .NET.

Seguidamente se presentan ciertas características destacadas de C#, pudiendo ser algunas de estas características generales, características de cualquier otro lenguaje incluido dentro de la plataforma .NET.

- Para que las diferentes aplicaciones escritas en el lenguaje C# funcionen adecuadamente, no es necesario la utilización de ficheros adicionales al código fuente propio, como por ejemplo las DLLs.
- La portabilidad del código es posible ya que al ser presentado en un tamaño independiente no depende del tipo de máquina o del sistema operativo que se esté usando.
- Al ver que existen elementos útiles para los desarrolladores en otros lenguajes como el foreach, o el tipo básico decimal, estos han sido incluidos como sentencias indispensables en C#.
- Para dar facilidad en la comprensión del código fuente y evitar confusiones, C# permite la utilización de funciones o variables globales. Es importante dejar indicado que se deben incluir en definiciones de tipos de datos a todas las definiciones de variables y funciones.
- Al tratarse de un lenguaje orientado a objetos, C# si soporta polimorfismo, herencia y encapsulación; C# no soporta herencia múltiple.
- Sabiendo que C# también es un lenguaje orientado a componentes, es posible desde ese punto de vista definir una clase que actué como componente, para lo cual es necesario definir sus propiedades, atributos y eventos de una forma muy simple.

- El acceso al código nativo escrito en funciones no orientada a objetos en C# es posible con la utilización de la sentencia Pinvoke. Normalmente estas funciones se refieren a DLLs y al API de Windows.

2.5.2. ANÁLISIS DE LAS HERRAMIENTAS DE SOFTWARE

2.5.2.1. ASP.NET MVC

Desarrollando aplicaciones web en ASP.NET MVC 4.0 y usando .NET Framework 4.0 y Visual Studio 2010, no es posible utilizar Visual Studio 2008.

Es normal preguntarnos si ASP.NET MVC sustituye a ASP.NET, a dicha pregunta podríamos responder que ASP .NET MVC no sustituye a ASP.NET por las razones que a continuación se detallan:

”

- ASP .NET es la base para la construcción de ASP.NET MVC. Dentro de los aspectos transversales de ASP.NET (autenticación, cache, sesión, roles), en ASP.NET MVC siguen siendo los mismos.
- Podemos indicar que ASP.NET MVC sustituye en algo a Webforms, es decir a las páginas .aspx. Ante esto la empresa Microsoft ya se ha pronunciado que esta posible sustitución no sucederá, los dos frameworks (ASP.NET MVC y Webforms) seguirán sacando actualizaciones. Así que si se ha invertido tiempo en estudiar y dominar Webforms, por el momento hay que estar tranquilos ya que el conocimiento adquirido sigue siendo válido, de todas maneras no está por demás sugerir a las personas interesadas que investiguen a ASP .NET MVC para que luego tomen una decisión a y miren cuál de los dos frameworks les parece el apropiado para el desarrollo de aplicaciones web usando .NET.”

Adaptado de (Desarrolloweb.com, 2011)

ASP.NET MVC 4 sigue el patrón de controlador de vista de modelo (MVC), con lo cual podemos decir que es un marco para desarrollar aplicaciones web, fáciles de probar y mantener.

Soporte para Unión y Minificación es lo que trae consigo ASP .NET MVC 4, esto se logra gracias a la nueva iteración del Framework ASP.NET 4.5. Con estas características es posible construir aplicaciones con un tiempo de carga mucho más rápido puesto que la disminución del tamaño y el tiempo que hacen las páginas a los HTTP es muy notable. Algo novedoso es que se tiene un mejor manejo de archivos por la inclusión de “cache busting”.

En el nuevo Entity Framework 4.3, del ASP.NET MVC 4 se tiene una principal ventaja, la migración de bases de datos para lo cual usa únicamente código preservando de los datos de la base.

Soporte para la creación de “Wep APIs”, incluye ASP.NET MVC 4; un marco de trabajo que permiten crear y consumir servicios HTTP y APIs asequibles a una gran cantidad y variedad de clientes, incluyen exploradores, teléfonos y tabletas. ASP .NET Web API es idóneo para crear servicios que siguen un estilo de arquitectura REST siendo además compatible con patrones RCP.

Soporte para construir aplicaciones y sitios web para móviles incluye ASP.NET MVC 4, esto permite crear aplicaciones de forma mucho más sencillas dirigidas a teléfonos y tabletas.

La Versión del Motor de Visitas que se incluye en ASP.NET MVC 4, ofrece muchas mejoras que facilitan para la creación de plantillas, siendo estas mucho más limpias y consistentes, dando al usuario un modo rápido, sencillo y asequible para combinar código de servidor con HTML.

Es posible combinar ASP.NET MVC 4 con .NET 4.5 y Visual Studio 11, permitiendo de esta manera la utilización de todos los elementos combinados; con el soporte asíncrono se puede construir aplicaciones completamente escalables inclusive usando las mejoras que son incluidas en Visual Basic o C#, estas aplicaciones es posible utilizarles con las nuevas capacidades de los exploradores web.

2.5.2.2. MOTOR DE VISTAS RAZOR

Podemos comenzar indicando que Razor no es en realidad un lenguaje de programación, simplemente es un motor de vistas, que ha tenido una excelente acogida. En MVC la idea principal es pretender separar completamente los datos generados por la aplicación (Modelo) de la forma de presentación (Vista), desuniendo estas dos capas mediante una capa intermedia (Controlador).

Con ASP .NET MVC se llega a cumplir con este principio señalado anteriormente, es así que se puede expresar que modo queremos crear en nuestra vista con esto garantizamos que el 100% de contadores como modelos se pueda reutilizar.

Desde la versión ASP.NET MVC 3 incluye consigo el motor de vistas llamado Razor. Esto no quiere decir que el motor ASPX no se use, se puede seguir usando en ASP.NET MVC 3, pero creo que no hay ninguna razón para hacerlo, salvo en casos de migraciones. Utilizando Razor nos encontraremos con algo más claro, sencillo e intuitivo.

Son muchos objetivos por el cual se creó este motor de vistas pero los más destacados podrían ser:

Se busca reducir la cantidad de código en la creación de vistas, el compilador será el encargado de deducir que tipo de código se está escribiendo.

Es posible utilizarlo en cualquier editor de texto, hay que tomar en cuenta que el soporte de lenguaje es superior si se utiliza la herramienta de Visual Studio, con esto se señala que la última decisión será la nuestra al momento de utilizar el Motor de Vistas Razor.

Sintaxis de Razor.

En Razor se tiene una ventaja que le hace diferente del motor ASPX donde existe el tag que indica el inicio del código en el servidor y su finalización. En

motor Razor en cambio solo existe el tag para indicar cuando comienza el código en el servidor, esto porque Razor sabe cuándo se termina el código en el servidor, es decir no existe necesidad alguna de indicar en Razor la terminación del código.

Ejemplo de una vista usando Razor:

```
@{ var nombre = "Gonzalo"; }  
<html>  
  <body>  
    <h2>Hola/@nombre, son las @DateTime.Now</h2>  
  </body>  
</html>
```

Ilustrado en el ejemplo anterior con un simple símbolo arroba (@) Razor es capaz de interpretar e indicar el inicio del código procedural ejecutable en el servidor en el momento de enviar la página al navegador bajo una solicitud, tal como se comentó anteriormente en Razor no existe símbolo alguno para indicar la terminación del código, ya que el motor Razor es capaz de deducir cuando termina el código basándose en el contexto.

Observamos también según el ejemplo, la forma sencilla que se puede mezclar el código procedural con el código HTML sin necesidad de recurrir a las típicas secuencias <% del ASP. NET clásico, o cualquiera que este sea el carácter de escape utilizado por el lenguaje que uno este acostumbrado a utilizar.

Para este ejemplo vemos en dos ocasiones la aparición del símbolo arroba (@).

- @nombre, en este punto se hace referencia a una variable, la misma que está definida en la parte superior del código siendo esta definición código de C#, para Razor todo código que no es HTML (código procedural) es programable en VISUAL.NET y en C#. En este código procedural se ha

dado un valor con el nombre propio "Gonzalo" a la cadena de texto con la variable nombre.

- `@DateTime.Now`, aquí se hace uso de la propiedad de un objeto con el cual estamos obteniendo la fecha actual.

HTML 5

Algunos entendidos lo consideran simplemente como nuevas etiquetas tales como `<header>` y `<footer>`, sumadas a una gran cantidad de atributos disponibles. Para otros, se dice significar la parte nueva e interesante en la Web, formando parte también las tecnologías aplicadas en un navegador.

Se señala también que HTML5 es la actualización de HTML, HTML5 es un término de marketing en el que se agrupa nuevas tecnologías de desarrollo de aplicativos web (HTML5, nuevas capacidades de Javascript, CSS3).

Según el organismo internacional de normalización conocido como World Wide Web Consortium (W3C), señala que HTML5 lleva con sigla más de 100 especificaciones que se relacionan con la próxima generación de tecnologías de la Web. Poniendo todas las más de 100 especificaciones del HTML5 podríamos argumentar que el W3C simplificó demasiado las cosas. Y si bien es difícil tomar algo tan amplio como HTML5 y definirlo en forma inequívoca, creo que el W3C está tratando de hacer frente a la magnitud de lo que está cambiando en la web introduciendo HTML5 como un concepto unificador de ese cambio.

Sin lugar a duda, HTML5 viene a ser un término genérico para describir una agrupación de especificaciones HTML, CSS y Javascript, fue diseñado para permitir a los desarrolladores crear la siguiente generación de aplicaciones y sitios Web. En estas especificaciones que se menciona existen tres partes que llaman la atención en la definición, en su orden, HTML, CSS y Javascript. Según nos describen se define cómo utilizan los desarrolladores los términos de marcado y mejorado con capacidades de estilo mucho más ricas y nuevas APIs de JavaScript aprovechando al máximo las nuevas características de

desarrollo Web. Por decirlo simplemente aplicando una agregación se tendría que, HTML5 = HTML + CSS + JavaScript.

Por consiguiente entendemos que HTML5 implica cambios en HTML, CSS y Javascript.

No olvidemos también que HTML y HTML5 son muy compatibles entre sí, todo el código que se ha llegado a utilizar en HTML seguirá siendo válido sin ningún inconveniente en HTML5. Si queremos empezar a utilizar HTML5 solo es necesario empezar colocando DOCTYPE antes de la etiqueta <html>, ilustrando las palabras anteriores la etiqueta quedaría: <!DOCTYPE html>.

Con respecto a las principales nuevas etiquetas en HTML5 se puede decir que no tienen una representación especial en la pantalla, ya que todas tienen un comportamiento de un <div> o un con un significado superiormente sistemático a los simples div o span.

2.5.2.3. SISTEMA GESTOR DE BASE DE DATOS (SGBD)

DEFINICION SGBD

Sistema Gestor de base de datos (SGBD) podríamos definirlo como la agrupación de programas, dichos programas permiten mantener y crear una Base de datos, asegurando su integridad, confidencialidad y seguridad. Por lo tanto debe permitir:

- La definición de una base de datos para lo cual se debe especificar bien los tipos, estructuras y restricciones de datos.
- La construcción de la base de datos esto es posible cuando se guardan los datos en algún medio controlado por el mismo SGBD.
- La manipulación de la base de datos para lo cual debe permitir realizar consultas, actualizar, generar informes.

Los ejemplos más comunes de SGBD conocidos desde una apreciación muy personal pueden ser Oracle y SQL Server de Microsoft, MySql, Postgresql.

Se detalla a continuación ciertas características deseables en un Sistema Gestor de base de datos:

- Tener redundancia en los datos trae consigo ciertos efectos negativos ya que el trabajo se puede duplicar al actualizar, se desperdicia espacio en disco, podría provocar inconsistencia de datos esto en ciertas ocasiones es deseable para mejorar el rendimiento, por todo esto se desearía controlar la redundancia.
- Cada usuario es necesario que tenga permisos de acceso y autorización para el ingreso a los datos, en este punto lo deseado sería el control de accesos no autorizados.
- Es necesario que el SGBD ofrezca recursos básicos y necesarios, con esto se estaría definiendo el cumplimiento y garantizando las restricciones de integridad.

Los componentes de una base de datos:

- a) **Hardware:** Son los diferentes mecanismos que se utiliza para el almacenamiento donde va a residir la base de datos, así también los varios equipos periféricos necesarios para su uso.
- b) **Software:** Es el programa que común mente se los conoce como SGBD, este maneja las múltiples peticiones que los usuarios realizan para poder acceder a la base de datos.
- c) **Usuarios:** Al relacionar los usuarios con una Base de datos podemos indicar que existen tres tipos:
 - El que programa las aplicaciones quien es el que crea todos los programas necesarios en la aplicación para ser utilizados en la base de datos.
 - El que puede acceder a la base de datos a base de un medio de lenguaje de consulta o programas de aplicación.
 - El que administra la base de datos común mente llamado DBA (Data Base Administrator), quien es la persona encarga de controlar todo el Sistema de Base de Datos.

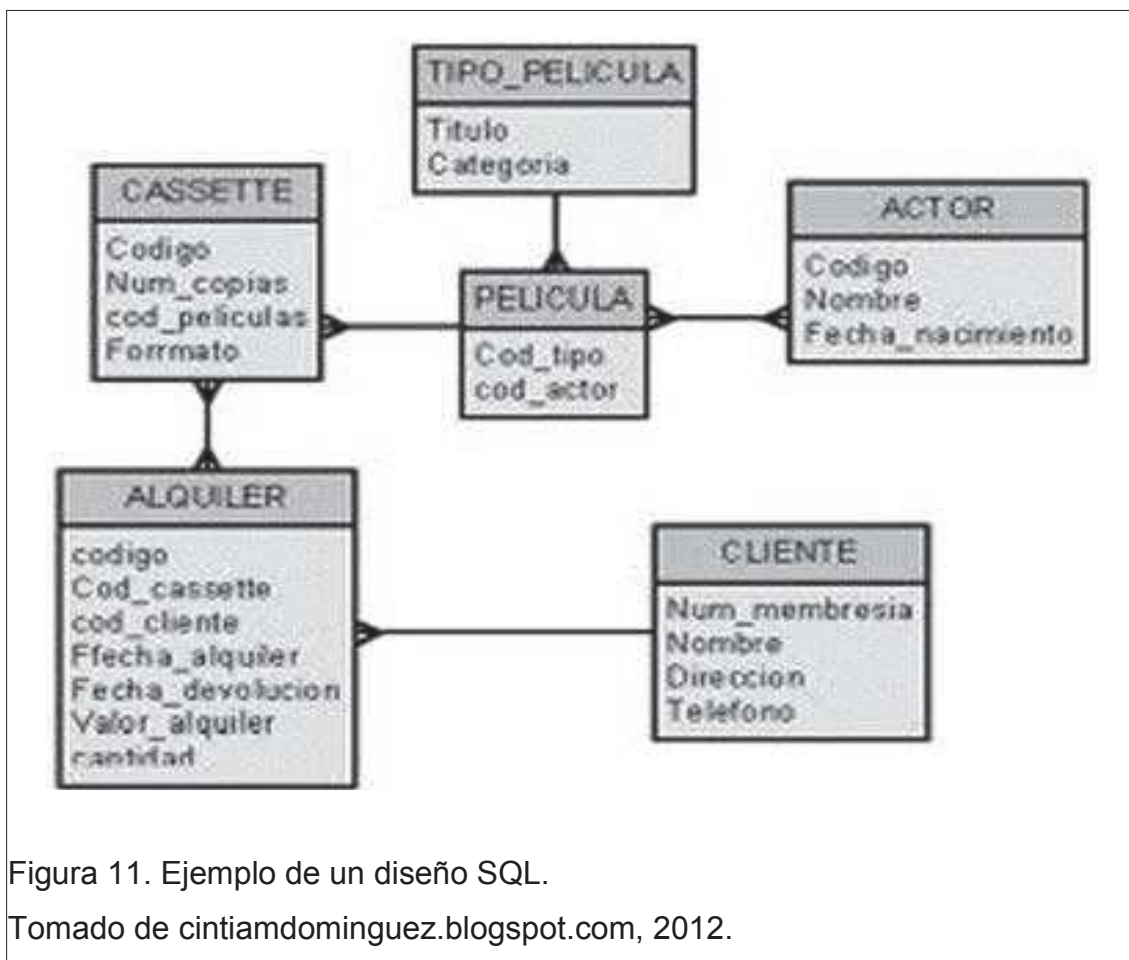
BASE DE DATOS RELACIONAL

Se construye a través de tablas, registros (líneas), campos (columnas); estas tablas se relacionan por medio de un campo en común en dos o más tablas que tienen las mismas características. Es diferente a la base de datos orientada a objetos, en una relacional existen RDBMS (Sistema de Gestión de Base de Datos Relacional), con el mismo objetivo que la anterior.

Las bases de datos relacionales ya han sido implementadas hace muchos años presentando un buen rendimiento para aplicaciones de negocio o sistemas de información. Este tipo de bases de datos relacional siguen por un proceso conocido como normalización de una base de datos, siendo este proceso muy indispensable y necesario para que una base de datos sea utilizada de manera óptima.

Las fortalezas de este tipo de base de datos son los tipos de datos sencillos, lenguajes de consulta potentes, protección elevada; su desventaja sale a la luz cuando sus soluciones no tienen nada que ver con datos de negocio, como: imágenes, multimedia o información geográfica, ya que el modelo relacional no es apropiado por no poseer estructuras de datos para soportar tipos de datos mencionados anteriormente.

En el mercado hoy en día podemos encontrar una inmensa gama de base de datos la mayoría de ellas relacionales, este tipo de base de datos permiten la organización de la información en tablas que se relacionan entre sí por intermedio de campos de relación, con lo cual es indispensable que se ejecuten una serie de reglas que procuran evitar redundancia, asegurando de esta manera la integridad de los datos almacenados en ellas así como la seguridad de los mismos.



BASE DE DATOS DE CODIGO CERRADO

Conocido también como software no libre o privativo de código cerrado. Refiriéndose de esta manera a cualquier sistema informático donde el usuario final tiene ciertas limitaciones para poder utilizar, modificarlo o redistribuirlo; es decir la estructura del código fuente no está disponible o el acceso a éste tipo de SGBD se encuentra restringido por un acuerdo de licencia.

Un ejemplo muy concreto es Microsoft; todos sus códigos fuentes no se encuentran disponibles al público.

Dentro de las más importantes SGBD de código cerrado tenemos a las versiones de SQL Server y Oracle.

SQL SERVER



SQL Server fue desarrollado por la empresa Microsoft, es un lenguaje de base de datos normalizado, utilizado por el motor de base de datos Microsoft Jet.

Esta herramienta se la usa comúnmente en la construcción, administración e implementación de aplicaciones de negocio usando el enfoque relacional.

Una gran ventaja es tener el enlace directo con Microsoft Visual Studio, el Microsoft Office System y un conjunto de nuevas herramientas de desarrollo, incluido el Business Intelligence Development Studio.

Presenta siete diferentes versiones que son:

- Enterprise Edition (Empresarial).
- Standard Edition (Versión Estándar).
- Workgroup Edition (Para Grupos de Trabajo).
- Express Edition (Edición Personal).
- Compact Edition (Para dispositivos móviles).
- Developer Edition (Para desarrolladores de Software).
- 64-Bit Platform Edition (Para plataformas de 64 Bits).

Microsoft SQL Server se encuentra en los primeros lugares en cuanto a SGBD esto es porque muchas empresas optaron por utilizar un producto creado por una empresa pionera como Microsoft y que con el paso del tiempo han seguido confiando en este software que cada vez se intenta aproximar más al usuario y reduciendo sustancialmente la dificultad de las tareas que conlleva la gestión de una base de datos.

A continuación se exponen algunas de sus principales características:

- Soporte de transacciones.
- Escalabilidad, estabilidad y seguridad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Da la facilidad de trabajar en modo cliente-servidor, donde la información y datos son alojados en el servidor y las terminales o clientes de la red solo acceden a la información.
- Da la facilidad de administrar la información de otros servidores de datos.

Tabla 2. Características de las diferentes versiones SQL SERVER

Edición	Beneficio	Tamaño	Características Clave
Express	La forma más rápida de aprender, crear e implementar aplicaciones simples orientadas a datos.	1 CPU 1 gigabyte (GB) RAM 4 GB de tamaño de base de datos	4 GB de tamaño de base de datos Informes simples
Workgroup	Solución de base de datos sumamente accesible y muy	1 o 2 CPU 3 GB RAM	Management Studio Importación/Exportación Organización en clústeres

Edición	Beneficio	Tamaño	Características Clave
	fácil de usar para pequeños departamentos y empresas en crecimiento.		Transmisión de registros de seguridad
Standard	Plataforma completa de administración y análisis de datos para empresas medianas y grandes departamentos.	1 a 4 CPU RAM ilimitada	Reflejo de bases de datos Estándar con Análisis Servicios Informes estándar con Reporting Services Réplica completa y Disponible en ediciones nativas de 32 y 64 bits Admite Itanium 2 y x64
Empresarial	Plataforma totalmente integrada de administración y análisis de datos para aplicaciones empresariales esenciales.	Escala y particionamiento ilimitados	Reflejo de bases de datos avanzados, operaciones completas en línea y paralelas Creación avanzada de informes con informes especiales, de muy alto nivel y personalizados Enrutamiento de datos y capacidades de transformación Disponible en ediciones de 32 y 64 bits Admite Itanium 2 y x64

Adaptado de technet.microsoft.com, 2012.

ORACLE

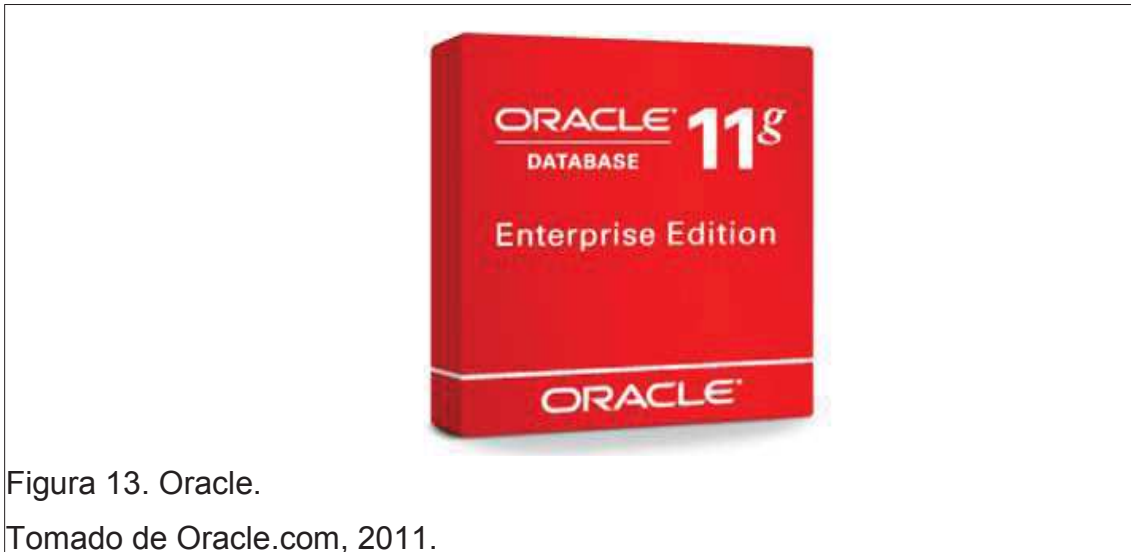


Figura 13. Oracle.

Tomado de Oracle.com, 2011.

Oracle comienza a surgir a finales de los años 70, su tecnología ha abarcado prácticamente todas las industrias alrededor del mundo, y en la actualidad encabeza la lista de los SGBD más usados. Por su elevado precio esta herramienta se ve en empresas muy grandes.

Sus principales características son:

- Su constante soporte en las transacciones.
- Su estabilidad.
- Su escalabilidad.
- Su soporte multiplataforma.

La arquitectura Oracle tiene tres componentes básicos que son:

- Las estructuras para almacenar datos y el código ejecutable.
- Los procesos que corren el sistema de bases de datos y las tareas de cada usuario conectado a la base de datos.
- Los archivos que sirven para el almacenamiento físico, en disco, de la información de la base de datos.

A continuación se ilustra en la Figura 14 su arquitectura.

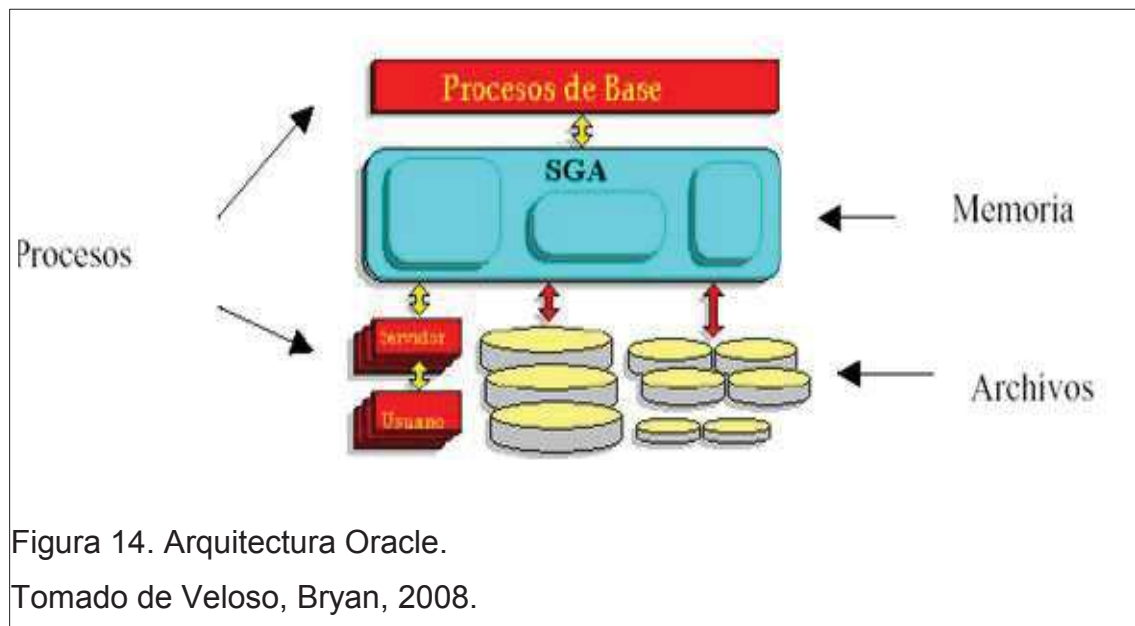


Figura 14. Arquitectura Oracle.

Tomado de Veloso, Bryan, 2008.

Oracle a partir de la versión 10g Release 2 presenta diferentes versiones que son:

- Oracle Database Enterprise Edition (EE).
- Oracle Database Standard Edition (SE).
- Oracle Database Standard Edition One (SE1).
- Oracle Database Express Edition (XE).
- Oracle Database Personal Edition (PE).
- Oracle Database Lite Edition (LE).

COMPARACION ENTRE LOS PRINCIPALES SGBD

Tabla 3. Comparación de SGBD

Parámetro	Mysql	SQL Server	Oracle
Velocidad y Robustez	X	X	X
GNU (Sistema Libre)	X		
Versión Express		X	X
Multiproceso	X	X	X
Varios Sistemas Operativos	X		
Conexión TCP/ IP	X	X	X
Trabajo con Concurrencia	X	X	X
Usa Estándar SQL	X	X	X
Compatibilidad con la autenticación de Windows.		X	
Conexión Directa con Visual Studio		X	
Contabilidad con XML		X	X
Integridad Referencial	X	X	X

Adaptado de GUERRERO GONZALES, PEDRO ENRIQUE, 2011.

Veracruzana, Universidad, 2009. PROAÑO, ING. DIEGO JAVIER BURBANO, 2006.

SGBD utilizado en la Dirección Provincial del Consejo de la Judicatura distrito Guayas.

Actualmente en el distrito guayas que es a donde va enfocado este sistema se utiliza como SGBD SQL Server 2008, con esta herramienta se ha venido trabajando desde la versión SQL Server 2000, la decisión de utilizar SQL Server como SGBD fue porque luego de un análisis se dieron cuenta que cumplía con los requisitos que en ese entonces necesitaban debido a sus bondades que este gestor nos proporciona. En la Tabla 4 se menciona algunas características:

Tabla 4. Características de SQL SERVER

Compatibilidad con .NET	Es compatible con Visual Studio 2010.
Concurrencia	SQL SERVER presenta buena respuesta a concurrencia, cuando se está en la lectura de los datos de una tabla SQL SERVER bloquea con esto se garantiza que nadie más pueda escribir en ella sin embargo es posible leerla.
Respaldos	Actualmente existen una serie de comandos que permite hacer una copia de seguridad de una o múltiples bases de datos. Existen varias formas de realizar un backup en SQL.
Cuentas de Usuario	Es importante tener una buena administración de la base de datos brindando una mayor seguridad de la información.
Integridad de Datos	SQL presenta un gran número de herramientas, esta permiten garantizar de buena manera la integridad de los datos como: claves primarias y foráneas, claves únicas, etc.
Seguridad	SQL provee diversas opciones para proteger contraseñas. Es posible usar SHA1 y MD5. Y aunque se han encontrado formas de romper contraseñas que utilizan estos métodos de encriptación, siempre será mejor tener esta opción a la de contraseñas en texto plano.

Adaptado de gravitar.biz, 2008. Entrevista con los desarrolladores de la base de datos que actualmente funciona en la Dirección Provincial del Consejo de la Judicatura Distrito Guayas.

2.5.3. DEFINICIÓN DE HERRAMIENTAS PARA SWCITAC

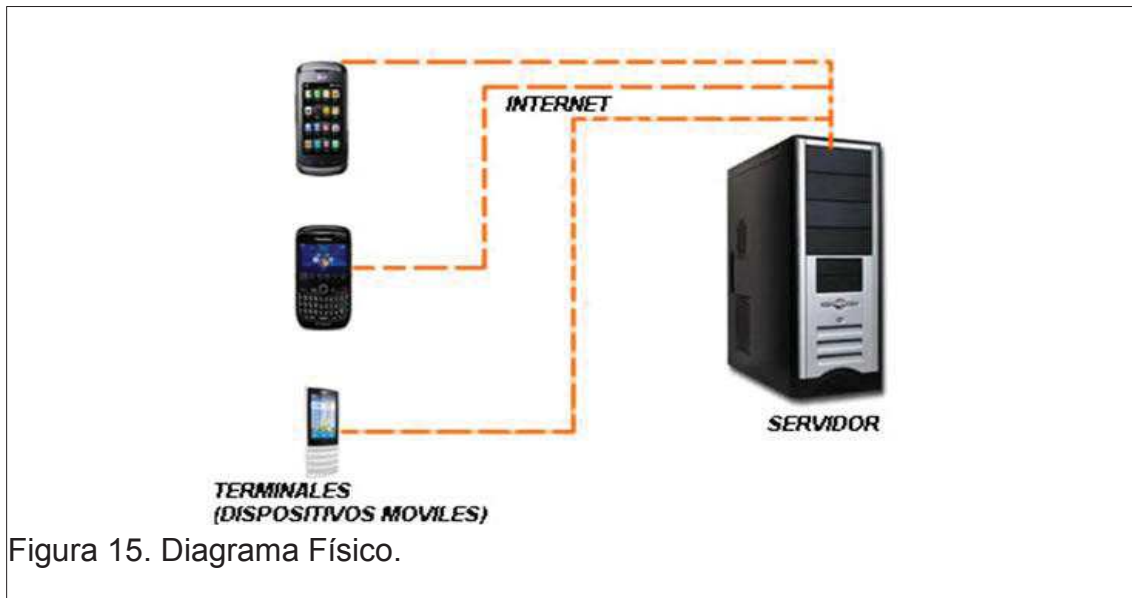


Figura 15. Diagrama Físico.

SwCITAC se basará en las siguientes características:

SwCITAC estará encaminada en una arquitectura Cliente-Servidor, donde cada uno de los citadores accederá al sistema a través del internet con su dispositivo móvil, mientras que el servidor responderá todas sus peticiones. El sistema utiliza el modelo MVC 4. (Microsoft Asp.Net, 2012)

MODELO: Esta realizado en SQL SERVER actualmente migrado a la versión SQL SERVER 2008 donde se utiliza con las bondades que esta versión presenta. Con respecto a la lógica del negocio es utilizada mediante el lenguaje C#.

VISTA: Programado para una interfaz fácil para el usuario y liviana para su comportamiento en los dispositivos móviles, utilizando herramientas tales como Razor para una conexión mediante servicios, con la utilización de C# para su programación de Objetos.

CONTROLADOR: Programado con C#, y se puede indicar que aquí están los eventos de interfaz tales como, CLICK, DOUBLE_CLICK, etc.

CAPITULO III

3. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN

3.1. ANÁLISIS DE REQUERIMIENTOS.

3.1.2 ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE.

3.1.2.1 INTRODUCCION

3.1.2.1.1 JUSTIFICACION

Una vez definida la problemática existente y establecer las causas que ameritan el desarrollo del SwCITAC, es apropiado realizar un estudio de factibilidad para determinar la estructura tecnológica y la capacidad técnica que implica el desarrollo e implementación del sistema.

3.1.2.1.2 FACTIBILIDAD HUMANA

Se cuenta con el recurso humano suficiente para el desarrollo del presente proyecto de grado, involucrando a los siguientes actores: estudiante Gonzalo Ibarra conjuntamente con el personal del departamento de Citaciones de la Dirección Provincial del Consejo de la Judicatura distrito Guayas, quienes brindarán el apoyo necesario con su conocimiento, experiencia teniendo de esta manera las facilidades necesarias para el desarrollo del sistema.

3.1.2.1.3 FACTIBILIDAD ACADEMICA

Se tiene los conocimientos necesarios para desarrollar el sistema SwCITAC, gracias a la experiencia adquirida en las aulas en los años de educación superior obtenidos en la prestigiosa Universidad de las Américas.

3.1.2.1.4 FACTIBILIDAD TECNOLÓGICA

Bajo los enfoques de Hardware y Software podemos señalar que la Dirección Provincial del Consejo de la Judicatura distrito Guayas reúne los requerimientos mínimos establecidos para el desarrollo del sistema SwCITAC. Es así que en Hardware cuenta con un servidor robusto donde actualmente está alojada la base de datos.

En lo que respecta a Software se cuenta con la versión de SQL Server 2008 instalada en el servidor.

3.1.2.1.5 ALCANCE

EL SISTEMA PARA EL REGISTRO DE CITACIONES A TRAVÉS DE TELEFONOS MÓVILES CON NAVEGACIÓN WEB “SwCITAC”, permitirá ingresar la información a través de un celular (Dispositivo móvil) por intermedio de los diferentes usuarios “citadores”, alimentando de esta manera la base de datos en tiempo real, también se podrá visualizar las citaciones ya realizadas al momento de realizar una nueva citación, para lo cual se tiene que ingresar dentro de cada proceso con el uso del dispositivo móvil. El “SwCITAC” guardará información permanente en su base de datos permitiendo a los trabajadores judiciales verificar con el sistema utilizado actualmente si la citación fue realizada o no oportunamente.

El Sistema tendrá un control de usuarios con privacidad exclusiva para cada citador para que puedan ingresar información y procesar, SwCITAC permitirá interactuar, actualizar, modificar, eliminar y registrar las citaciones realizadas, producción en tiempo real.

Este sistema será desarrollado para un ambiente web, el sistema estará a disposición principalmente de los usuarios del proceso de citaciones.

3.1.2.1.6 PERSPECTIVA DEL PRODUCTO

SwCITAC es un sistema diseñado a satisfacer a los usuarios del departamento de Citaciones. Presentará una interfaz amigable al usuario, de fácil aprendizaje, y que proporcionará mecanismos para la culminación rápida y correcta de los procesos.

Actualmente la Dirección Provincial del Consejo de la Judicatura distrito Guayas mantiene una infraestructura basada en Windows por lo que SwCITAC será un producto que trabajara en plataformas Windows.

Para definir los componentes del sistema, se seguirá UML sobre los casos de Uso.

3.2. USO DE METODOLOGÍA

Para el desarrollo del presente proyecto se ha elegido Rational Unified Process (RUP), basado en Unified Modeling Language (UML), que trabajando de manera conjunta constituyen una metodología estándar para el desarrollo de aplicaciones de software.

Tanto RUP como UML está estrechamente relacionado uno al otro, puesto que el primero define quien, como, cuando y que debe hacer el proyecto de acuerdo a las ideas, necesidades de los usuarios hasta llegar a un programa en ejecución, el segundo proporciona la notación grafica necesaria para representar los diferentes modelos obtenidos en los requerimientos.

A continuación describiremos cada una de sus etapas enfocadas en las necesidades del presente proyecto.

3.2.1. MODELANDO DEL NEGOCIO

3.2.1.1. ESTRUCTURA ORGANIZACIONAL

Dentro del Consejo de la Judicatura Distrito Guayas existen 5 niveles de jerarquía organizacional, en este quinto nivel se desglosan subniveles de acuerdo a la jefatura correspondiente siendo el nivel máximo la Dirección Provincial del Consejo de la Judicatura. En la Figura 1 se muestra el orden jerárquico de los niveles de operación de esta Institución Pública.

3.2.1.2. PROCESO DEL NEGOCIO

En la Figura 16 se ilustra el proceso de registro de una citación desde la salida de su puesto de trabajo por parte del usuario judicial hasta su retorno a registrar las hojas de trabajo diario (Citaciones) en el sistema actual SATJE.

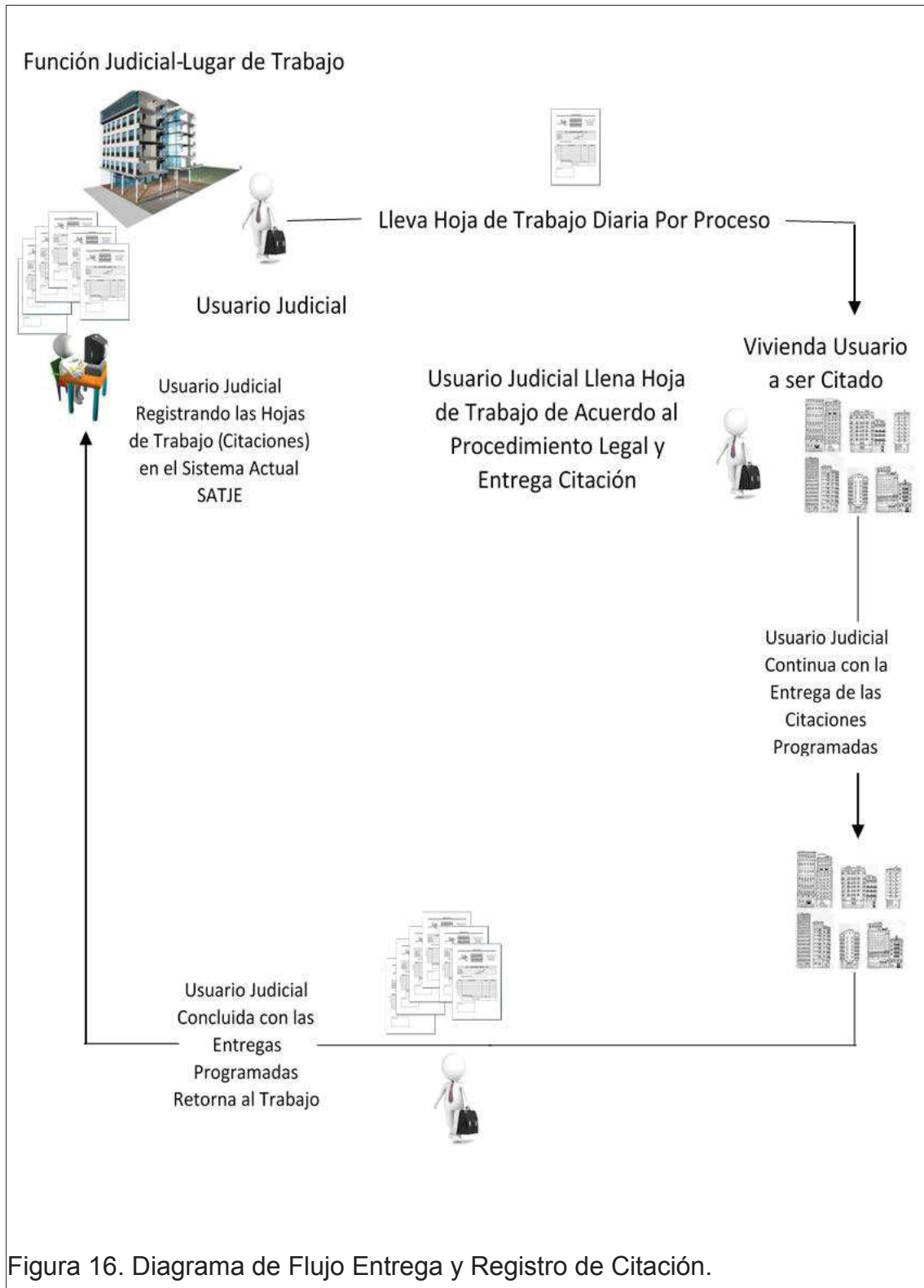


Figura 16. Diagrama de Flujo Entrega y Registro de Citación.

3.2.1.3. DESCRIPCION DEL PROCESO DEL NEGOCIO

Enfocándonos en la actividad principal para la elaboración de nuestro proyecto que es el registro de las Citaciones, el mismo que se desempeña en el Departamento de Citaciones que, según nuestro organigrama está ubicado en Asesoría y Servicios Judiciales. Los Usuarios Judiciales de dicho departamento son los encargados de revisar la programación diaria que se les hace llegar por medio de los diferentes Juzgados para que día a día salgan a realizar la entrega de las citaciones en los diferentes puntos de la ciudad, para que al finalizar la jornada en el mejor de los casos retornen a la Institución a sentarse frente al computador a ingresar las hojas de trabajo (Citaciones) en un módulo del sistema actual SATJE.

3.3. OBTENCION DE REQUISITOS

Antes de empezar el diseño de la aplicación se procede a generar un cronograma de reuniones para realizar el levantamiento de los requisitos.

Tabla 5. Cronograma de Reuniones para Obtener los Requisitos

MODULO	ACTIVIDAD	TIEMPO	jul-12	ago-12		
		RECURSOS	Semana 4	Semana 1	Semana 2	Semana 3
Seguridad	Autenticación	Po, Lp				
Negocio	Consulta Litigantes	Po, Lp				
	Registro de Citación	Po, Lp				
	Editar, Eliminar Registro de Citación	Po, Lp				
	Devolver Citación	Po, Lp				

Nomenclatura de Recursos	
Po	Personal Operativo de la empresa
Lp	Líder de Proyecto

3.3.1. ENTREVISTAS

Con el objetivo de sacar las necesidades del departamento de citaciones y ratificar así la correcta implantación del sistema se entrevistó a los Usuarios Judiciales según el estándar IEEE830-1998, (Ver anexo 4), quienes son los encargados directos de realizar día a día la entrega de las citaciones, se obtuvo los siguientes resultados:

- **Accesibilidad:**

Actualmente la mayoría de Citadores cuenta con un celular el mismo que tiene incorporado servicio de datos y browser para poder navegar; herramientas necesarias para el éxito del sistema.

Los Citadores que no cuentan con un celular con las características mencionadas anteriormente, están dispuestos a adquirir uno o contratar los servicios necesarios para poder utilizar el sistema.

- **Funcionalidad:**

Hoy por hoy, los Citadores ingresan la información de las citaciones realizadas al sistema en el mejor de los casos al terminar su jornada de trabajo diaria, esto cuando avanzan a llegar a la oficina; no tiene una apertura para alimentar el trabajo realizado directamente apenas se entregue la citación al usuario externo.

Esto puede llegar a ocasionar mal interpretaciones por los usuarios externos ya que al Citador se le puede extraviar o confundir las citaciones ocasionando de esta manera un retraso en la alimentación de las citaciones al sistema.

3.3.2. RECOLECCION DE DATOS

Los requisitos del sistema son la base fundamental sobre la que se desarrollara la aplicación.

Una vez finalizado todas las reuniones se procede al desarrollo del acta de levantamiento de requerimientos (Anexo 6)

3.4. ANALISIS Y DISEÑO

Con todos los requerimientos planteados por parte de los Usuarios del Departamento de Citaciones de la Dirección Provincial del Consejo de la Judicatura Distrito Guayas, se procederá a crear un acercamiento a la realidad para entender de mejor manera el sistema a desarrollar.

Por lo expuesto se construirá modelos UML, las partes dinámicas y estáticas del sistema estarán representados por los siguientes diagramas:

- Diagrama de Casos de Uso
- Diagrama de Actividades
- Diagrama de Componentes
- Diagrama de Clases
- Diagrama de Iteraciones

3.4.1. ARQUITECTURA Y CLASES

En la Figura 17 se representa el modelo lógico de la arquitectura del sistema con sus relaciones o referencias en lo que respecta a cómo va a estar comunicado mediante un Diagrama de Componentes.

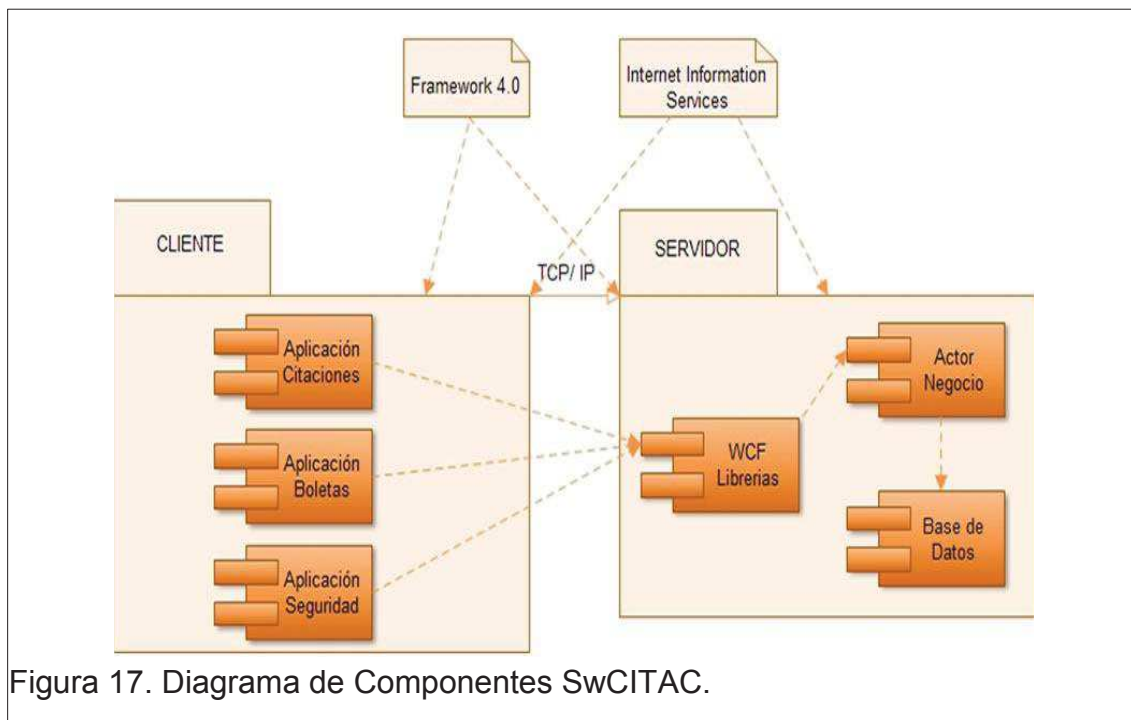


Figura 17. Diagrama de Componentes SwCITAC.

Los diagramas de componentes son utilizados para modelar los componentes de un sistema y mostrar las dependencias entre ellos.

Los componentes pueden existir en tiempo de compilación, de enlace o de ejecución o incluso en varios o todos esos tiempos. Los componentes son equivalentes a tipos, siendo sus instancias las que aparecerán en el diagrama de despliegue. Hay una gran diferencia entre clases y componentes.

Las clases son “componentes lógicos”, mientras que los componentes tienen significado físico. Antes de poder generar código, las clases deben ser mapeadas a componentes (esto es, las clases residen o son implementadas en componentes), y estos componentes serán los que contengan su código fuente.

Describiendo la arquitectura que presenta la aplicación tenemos que:

Desarrollo Proyecto Servidor

La realización de Servidor se ve enfocada en las entidades, acceso a datos, y lógica de negocio que utilizara el cliente para realizar las diferentes peticiones.

Está conformado por los siguientes módulos:

- Seguridad: Dirigido al logueo exclusivo de los usuarios del sistema.
- Citaciones: Dirigido a la consulta e ingreso de todas las citaciones por instancia y judicatura.
- Utilidades: Utilidades comunes usadas por todos los demás módulos.

Desarrollo Proyecto Cliente

La realización del Cliente se ve enfocada en las diferentes pantallas que observará el usuario.

Está conformado por los siguientes módulos:

- Seguridad: Diseñado para manejar el tema de ingreso de usuario y contraseña y mantener en sesión a los usuarios autenticados.
- Citaciones: Diseñado para brindar todo tipo de información requerida por el usuario.
- Común: Son los utilitarios utilizados en los módulos para un correcto funcionamiento.

Esta realizado mediante una interfaz orientada a los dispositivos móviles.

El “SISTEMA PARA EL REGISTRO DE CITACIONES A TRAVES DE TELEFONOS MOVILES CON NAVEGACION WEB (SwCITAC)” se encuentra definido de la siguiente manera:

Tabla 6. Descripción Sistema SwCITAC

PROYECTO	herramienta de desarrollo	Lenguaje de programación	Descripción
SGBD	Sql Server 2008	Sql	Base de Datos Relacional existente badalex1
SERVIDOR SwCITAC	Visual Studio 2010	C#, Entity Framework	Definición de Clases, Objetos, Métodos y Servicios
CLIENTE SwCITAC	Visual Studio 2010	C#, MVC 4.0	Desarrollo Visual de la Aplicación al Usuario.

3.4.2. CLASES Y METODOS DE SWCITAC

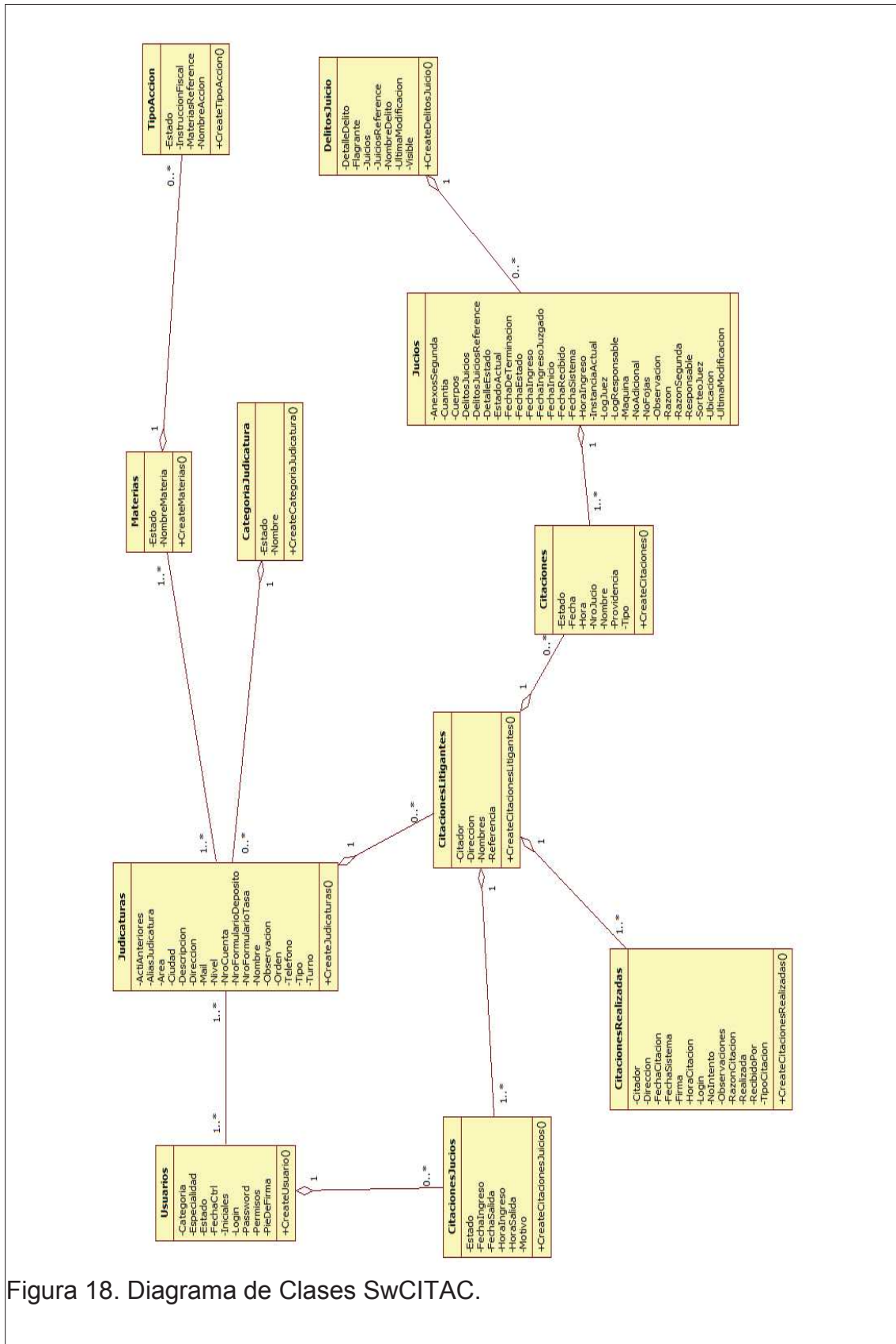


Figura 18. Diagrama de Clases SwCITAC.

En la Figura 18 se representan las clases que serán utilizadas dentro de los sistemas que existen entre ellas. Nos permite apreciar una visualización clara de las clases que están involucradas en el sistema, las mismas pueden caracterizarse por ser asociativas, de herencia, de uso y de convencimiento.

El diagrama principal para el diseño del sistema es el diagrama de clases que está compuesto por los elementos, Clase: atributos, métodos y visibilidad. Relaciones: Herencia, Composición, Agregación, Asociación y Uso.

En la Figura 19 contiene los objetos del sistema que son usados en la base de datos, es importante mencionar que la figura muestra las tablas con las que estamos desarrollando la aplicación por tratarse de información muy delicada y reservada no es procedente dejar plasmada toda la totalidad de la base de datos. Con este modelo estamos representando de manera gráfica la estructura lógica de la base de datos.

Casos de Uso

Es una técnica para recolectar información de la forma como un sistema trabaja actualmente, o como se necesita que trabaje, especificando el comportamiento del mismo.

Todo programa de software brinda a su entorno una serie de opciones en sus servicios. El caso de uso en cambio insinúa de cierta forma la necesidad saber cómo alguien o algo externo hace uso de un sistema. Cuando decimos “alguien o algo” hacemos referencia a que los sistemas pueden ser manipulados no solo por usuarios, sino también es posible ser manipulados por otros sistemas de hardware y software.

En los casos de uso se encuentra una representación gráfica en los denominados diagramas de casos de uso [Boo99]. En estos diagramas, los actores se representan en forma de pequeños monigotes y los casos de uso se representan por elipses contenidas dentro de un rectángulo que representa al sistema. La participación de los actores en los casos de uso se indica por una flecha entre el actor y el caso de uso que apunta en la dirección en la que fluye la información. Cada caso de uso puede estar definido por: texto que lo describe, secuencia de pasos ejecutados dentro del caso de uso, condiciones pre-post para que el caso de uso comience o termine.

Los diagramas de casos de uso sirven para proporcionar una visión global del conjunto de casos de uso de un sistema así como de los actores y los casos de uso en los que éstos intervienen. Las interacciones concretas entre los actores y el sistema no se muestran en este tipo de diagramas.

Hay que tomar en cuenta los tipos de relaciones:

- <<Include>> Se puede incluir una relación entre dos casos de uso de tipo “include” si se desea especificar comportamiento común en dos o más casos de uso, en esta relación se dice que el primer caso de uso base incluye al segundo caso de uso incluido.
- <<Extend>> Se puede incluir una relación entre dos casos de uso de tipo “include” si se desea especificar diferentes variantes del mismo caso de uso. Es decir, esta relación implica que el comportamiento de un caso de uso es diferente dependiendo de ciertas circunstancias. En principio esas variaciones pueden también mostrarse como diferentes descripciones de escenarios asociadas al mismo caso de uso.

Identificación de Actores

- PRINCIPALES: Personas que usan el sistema.
- SECUNDARIOS: Personas que mantienen o administran el sistema.
- MATERIAL EXTERNO: Dispositivos materiales imprescindibles que forman parte del ámbito de la aplicación y deben ser utilizados.
- OTROS SISTEMAS: Sistemas con los que el sistema interactúa.

Tabla 7. Actores UML en SwCITAC

ACTORES EN SwCITAC	
Usuario SwCITAC(Usuario Registrado en el Sistema)	Actor Principal (Usuario Citador).
Administrador	Actor Principal(Programador)
Base de Datos	Sistema. SGBD que forman parte del ámbito de la aplicación y debe ser utilizado.

3.4.4. IDENTIFICACION DE CASOS DE USO

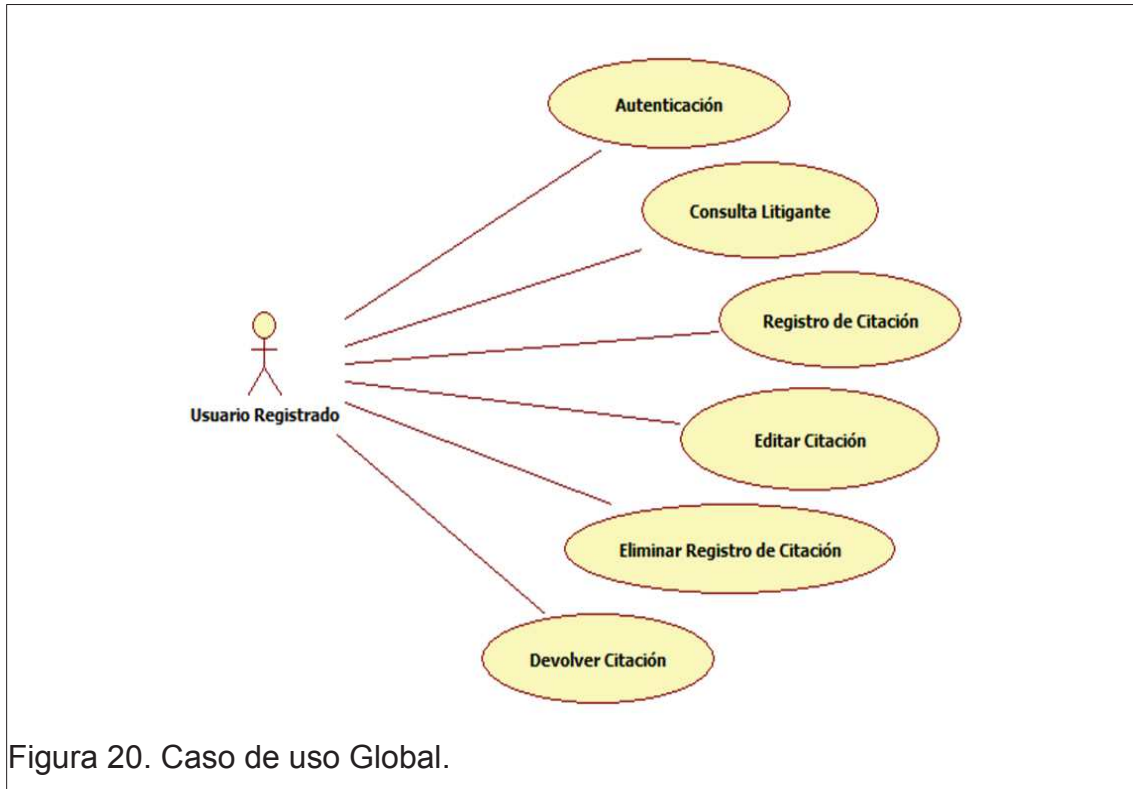


Figura 20. Caso de uso Global.

La Figura 20 representa el Caso de Uso de SwCITAC de la manera global (paquete) con su comportamiento.

CASO DE USO AUTENTICACION

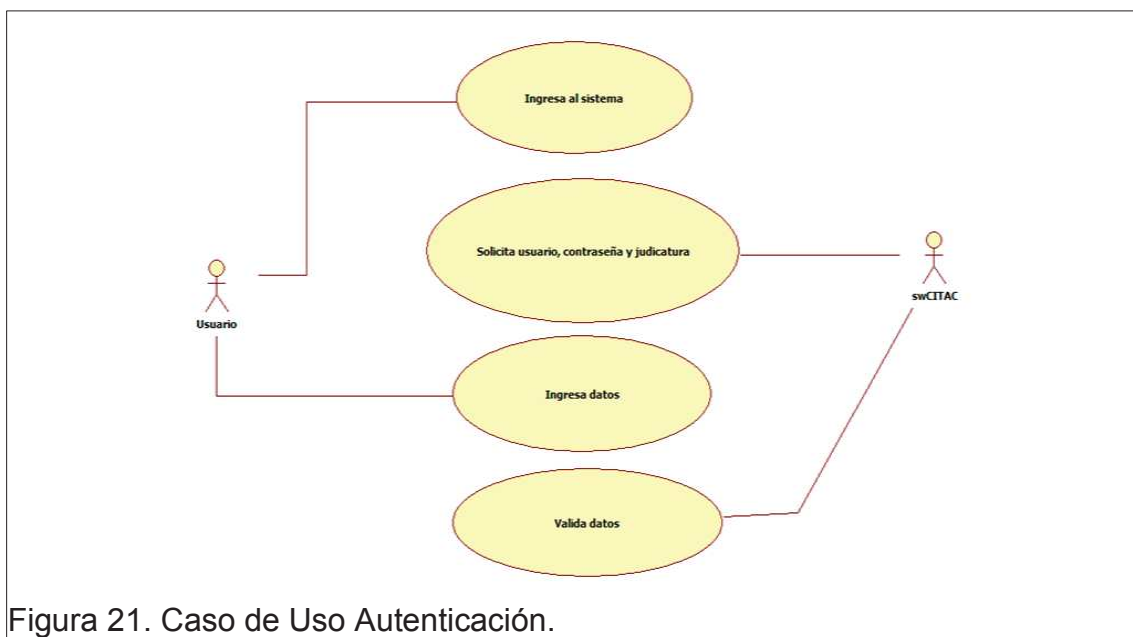


Figura 21. Caso de Uso Autenticación.

Tabla 8. Descripción Caso de Uso Autenticación

DESCRIPCIÓN CASO AUTENTICACIÓN	
ACTORES	Usuario, Sistema (swCITAC).
OBJETIVO	Ingresar al Sistema de SwCITAC V1
DESCRIPCIÓN	El usuario ingresa al sistema. El sistema solicita, usuario, contraseña, y la judicatura. El usuario ingresa los datos solicitados. El sistema valida los datos ingresados.
PRECONDICIÓN	Ninguna.
ALTERNATIVA	Ninguna.
EXTENSIONES	Si no han ocurrido errores, la validación se genera correctamente. Caso contrario, el sistema pedirá verificar los datos ingresados (Posibles errores humanos), enviando el mensaje correspondiente.

CASO DE USO CONSULTA DE LITIGANTES

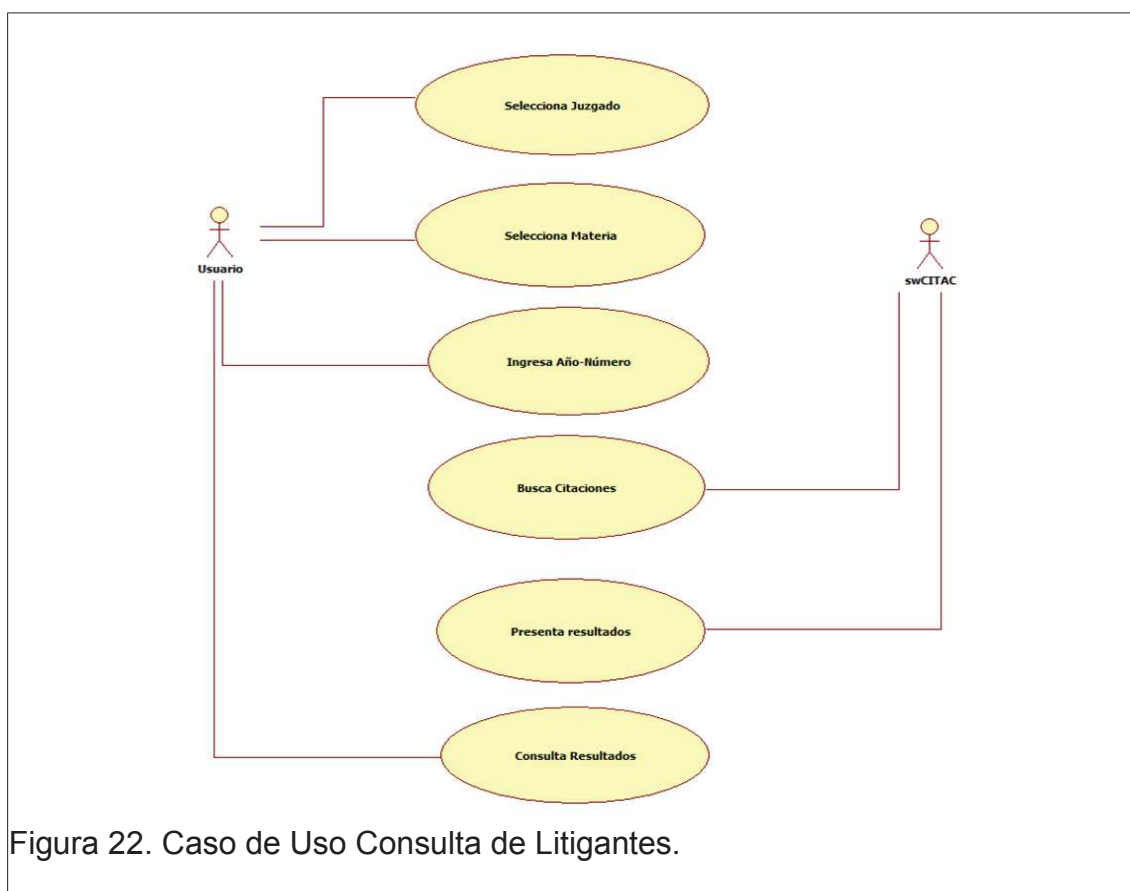


Figura 22. Caso de Uso Consulta de Litigantes.

Tabla 9. Caso de Uso Consulta de Litigantes

DESCRIPCIÓN CASO CONSULTA DE LITIGANTES	
ACTORES	Usuario Registrado, Sistema (swCITAC).
OBJETIVO	Consultar los o el litigante a ser citados. Previamente los juzgados correspondientes ingresan la información del porqué de la citación.
DESCRIPCIÓN	<p>El Usuario Registrado, selecciona la Judicatura.</p> <p>El Usuario Registrado, selecciona la Materia.</p> <p>El Usuario Registrado, ingresa el proceso (formato completo “aaaa+No proceso”).</p> <p>El Sistema busca en la base de datos la información proporcionada.</p> <p>El Sistema presenta resultados encontrados.</p> <p>El Usuario Registrado, consulta los resultados.</p>
PRECONDICIÓN	Previamente Autenticado siendo: USUARIO REGISTRADO.
ALTERNATIVA	Ninguna.
EXTENSIONES	<p>Si el proceso buscado tiene información, la validación se genera correctamente y presenta datos sobre el proceso.</p> <p>Caso contrario, el sistema desplegara un mensaje indicado que el proceso buscado no tiene citaciones pendientes.</p>

CASO DE USO REGISTRO DE CITACION

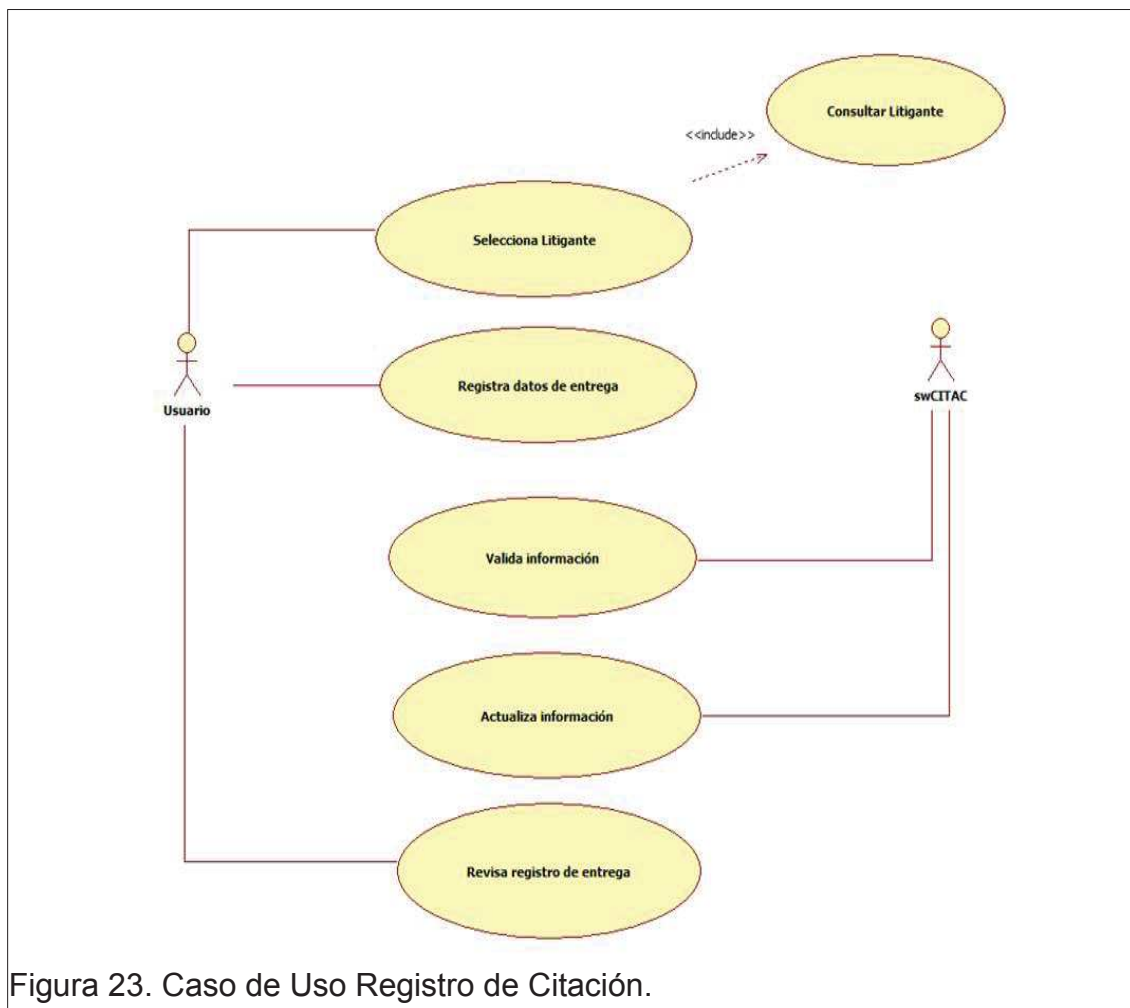


Tabla 10. Descripción Caso de Uso Registro de Citación

DESCRIPCIÓN CASO REGISTRO DE CITACION	
ACTORES	Usuario Registrado, Sistema (swCITAC).
OBJETIVO	Realizar el registro de la citación en el sistema.
DESCRIPCIÓN	<p>El Usuario Registrado, selecciona el Litigante (persona) a ser citada.</p> <p>El Usuario Registrado, registra datos de entrega según el modelo de la hoja de trabajo.</p> <p>El Sistema valida la información proporcionada.</p> <p>El Sistema actualiza información proporcionada en la base de datos.</p>

DESCRIPCIÓN CASO REGISTRO DE CITACION	
	El Usuario Registrado confirma el registro de la citación.
PRECONDICIÓN	Previamente Autenticado siendo: USUARIO REGISTRADO. Previamente debe consultar y seleccionar el proceso.
ALTERNATIVA	Ninguna.
EXTENSIONES	Si el proceso tiene completa la información, este se generara correctamente obteniendo el mensaje de guardado satisfactoriamente. Caso contrario tendremos un mensaje de alerta según el motivo de citación.

CASO DE USO EDITAR CITACION

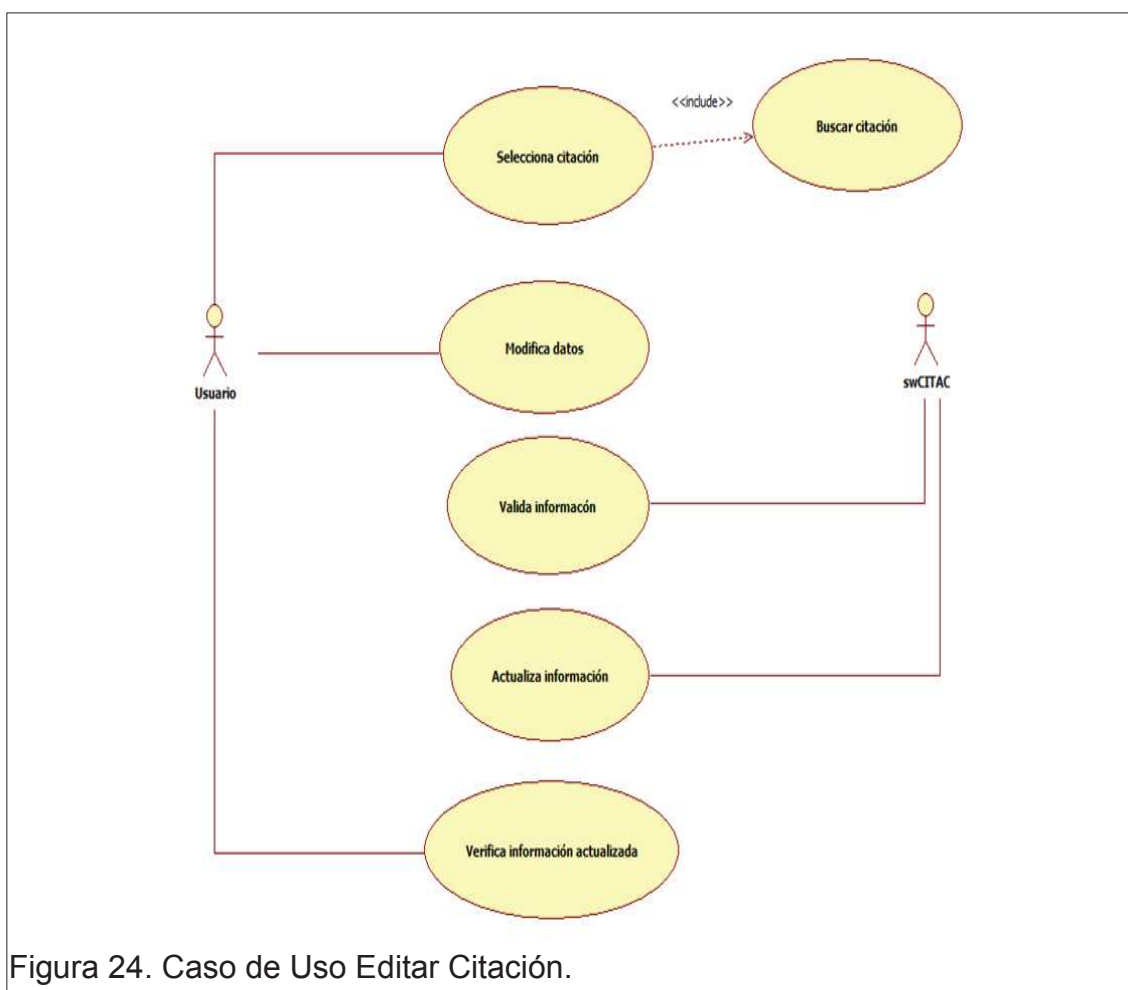


Figura 24. Caso de Uso Editar Citación.

Tabla 11. Descripción Caso de Uso Editar Citación

DESCRIPCIÓN CASO EDITAR CITACION	
ACTORES	Usuario Registrado, Sistema (swCITAC).
OBJETIVO	Editar la citación realizada, esto para corregir algún error involuntario.
DESCRIPCIÓN	<p>El Usuario Registrado, selecciona la citación a ser editada.</p> <p>El Usuario Registrado, modifica la información errónea por información correcta.</p> <p>El Sistema valida la información proporcionada.</p> <p>El Sistema actualiza información proporcionada en la base de datos.</p> <p>El Usuario Registrado confirma el registro de la citación.</p>
PRECONDICIÓN	Previamente Autenticado siendo: USUARIO REGISTRADO. Previamente debe consultar y seleccionar el proceso.
ALTERNATIVA	Ninguna.
EXTENSIONES	<p>Si el proceso tiene completa la información, este se generara correctamente obteniendo el mensaje de guardado satisfactoriamente.</p> <p>Caso contrario tendremos un mensaje de alerta según el motivo de citación.</p>

CASO DE USO ELIMINAR REGISTRO DE CITACION

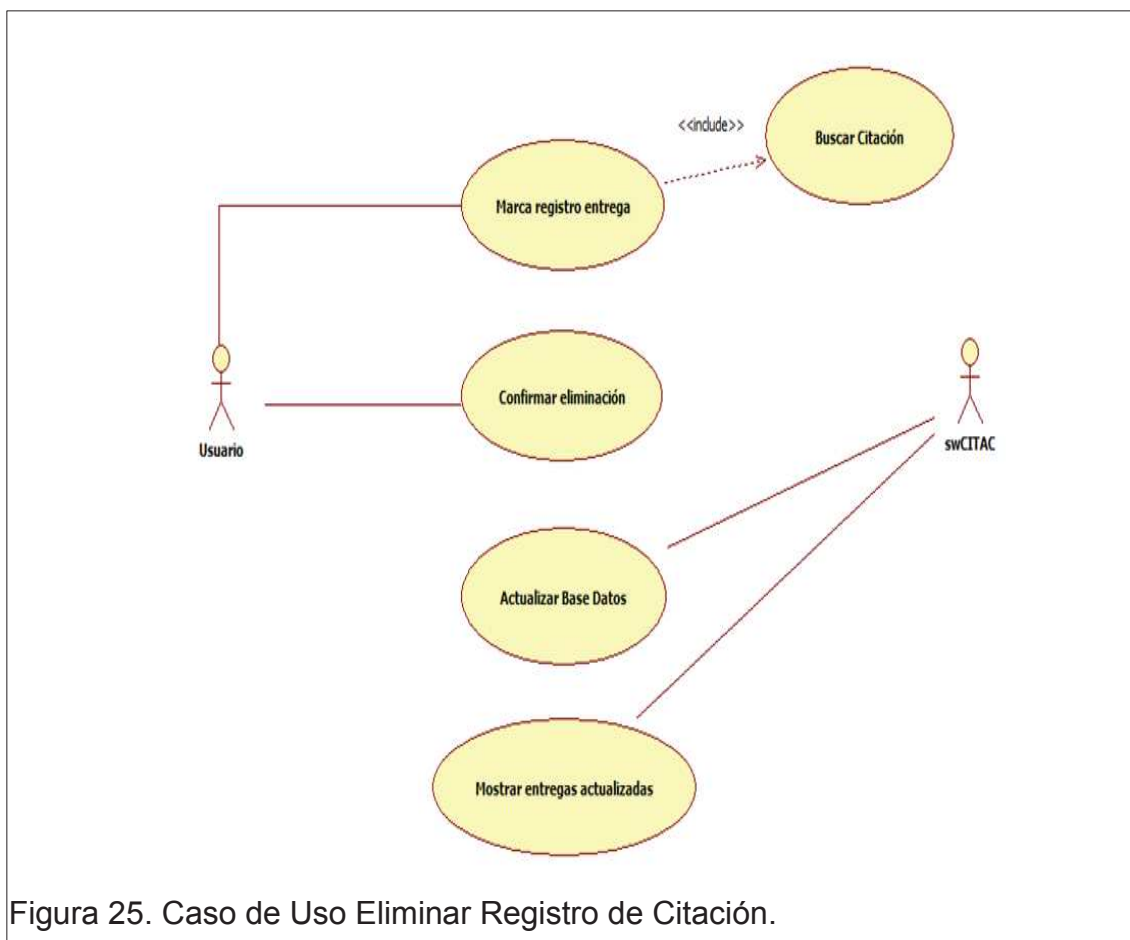


Figura 25. Caso de Uso Eliminar Registro de Citación.

Tabla 12. Descripción Caso de Uso Eliminar Registro de Citación

DESCRIPCIÓN CASO ELIMINAR REGISTRO DE CITACION	
ACTORES	Usuario Registrado, Sistema (swCITAC).
OBJETIVO	Eliminar un registro de la citación realizada.
DESCRIPCIÓN	<p>El Usuario Registrado, selecciona el registro de la citación realizada.</p> <p>El Usuario Registrado, confirma la eliminación del registro.</p> <p>El Sistema actualiza la información en la base de datos.</p> <p>El Sistema despliega los registros restantes de las citaciones realizadas si los hubiere en el proceso</p>

DESCRIPCIÓN CASO ELIMINAR REGISTRO DE CITACION	
	consultado.
PRECONDICIÓN	Previamente Autenticado siendo: USUARIO REGISTRADO. Previamente debe consultar y seleccionar el proceso.
ALTERNATIVA	Ninguna.
EXTENSIONES	Seleccionar el registro a eliminar, una vez confirmada la eliminación se obtendrá un mensaje de eliminación satisfactoria.

CASO DE USO DEVOLVER CITACION

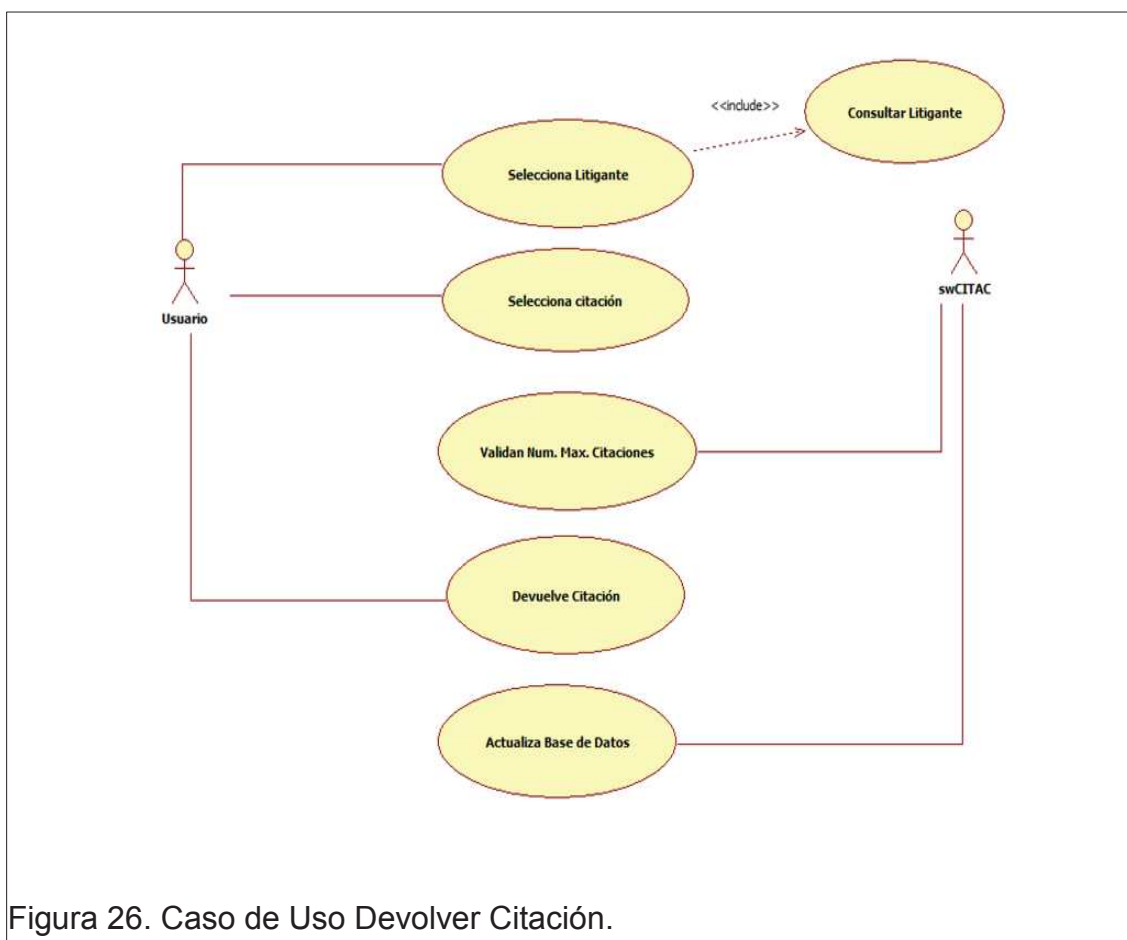


Figura 26. Caso de Uso Devolver Citación.

Tabla 13. Descripción Caso de Uso Devolver Citación

DESCRIPCIÓN CASO ELIMINAR REGISTRO DE CITACION	
ACTORES	Usuario Registrado, Sistema (swCITAC).
OBJETIVO	Devolver la citación al Juzgado una vez terminado el proceso de la citación.
DESCRIPCIÓN	<p>El Usuario Registrado, selecciona a la persona a ser citada.</p> <p>El Usuario Registrado, selecciona la citación.</p> <p>El sistema valida según el reglamento, el número máximo de citaciones permitidas por cualquier motivo existente que se puede presentar.</p> <p>El Usuario Registrado, devuelve en una forma lógica la citación al Juzgado correspondiente.</p> <p>El Sistema actualiza en la base de datos cambiando el estado del proceso.</p>
PRECONDICIÓN	Previamente Autenticado siendo: USUARIO REGISTRADO. Previamente debe consultar y seleccionar el proceso.
ALTERNATIVA	Ninguna.
EXTENSIONES	<p>Si el proceso seleccionado para ser devuelto cumple con las validaciones correspondientes, el mismo se realizara satisfactoriamente.</p> <p>Caso contrario tendremos un mensaje de alerta indicándonos que no ha superado la validación para ser devuelto.</p>

3.4.5. DIGRAMA DE ACTIVIDADES

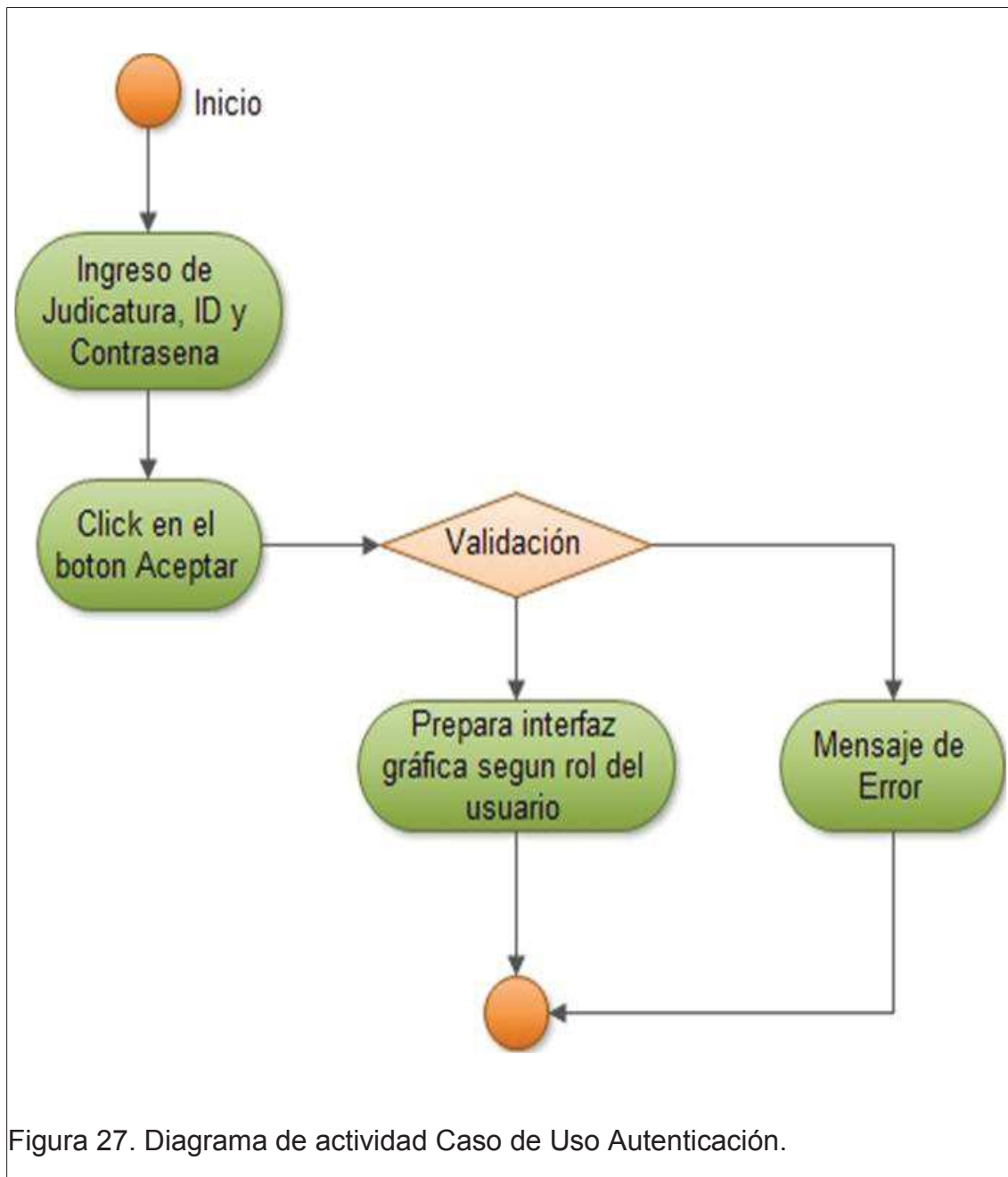
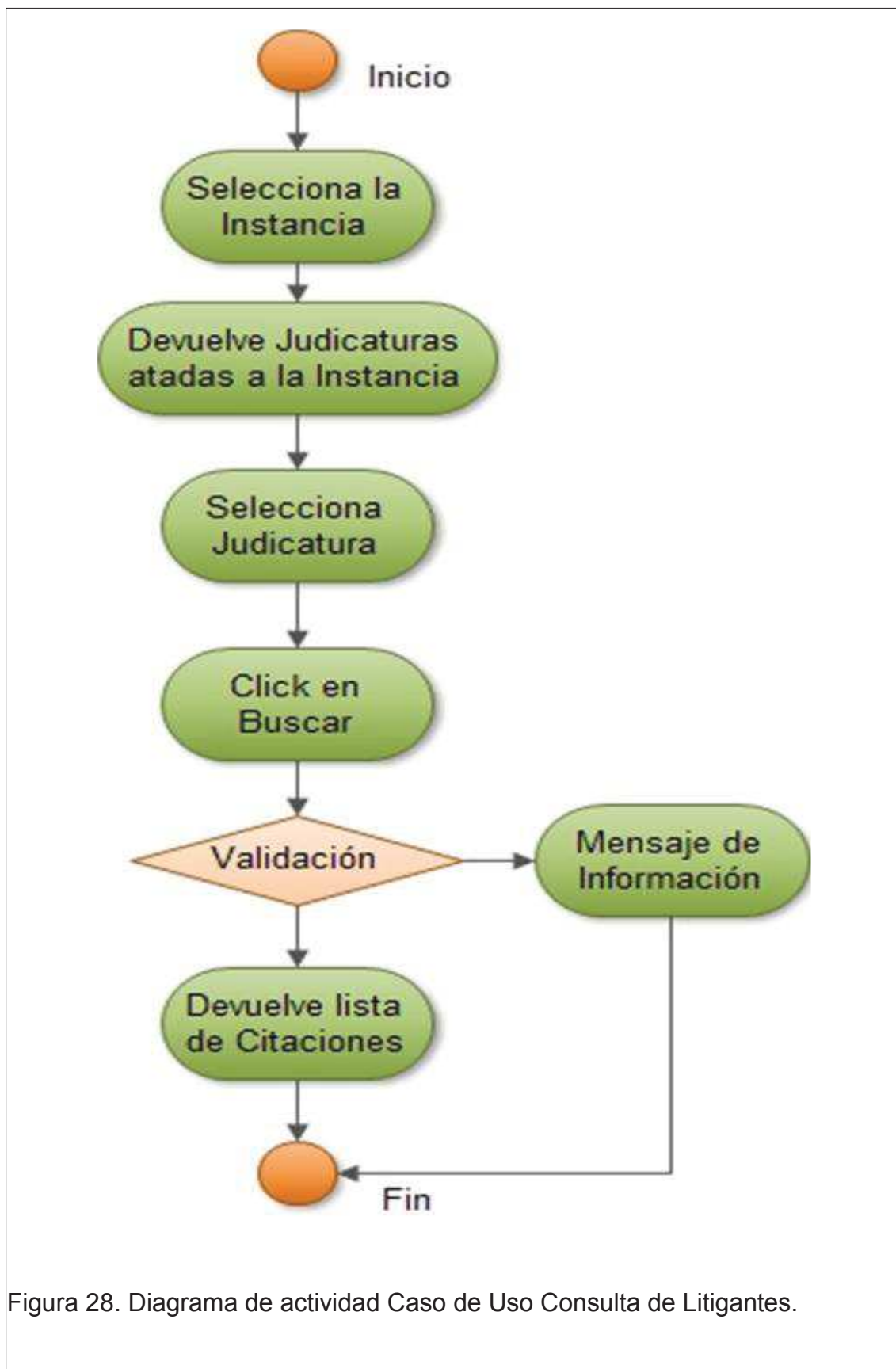
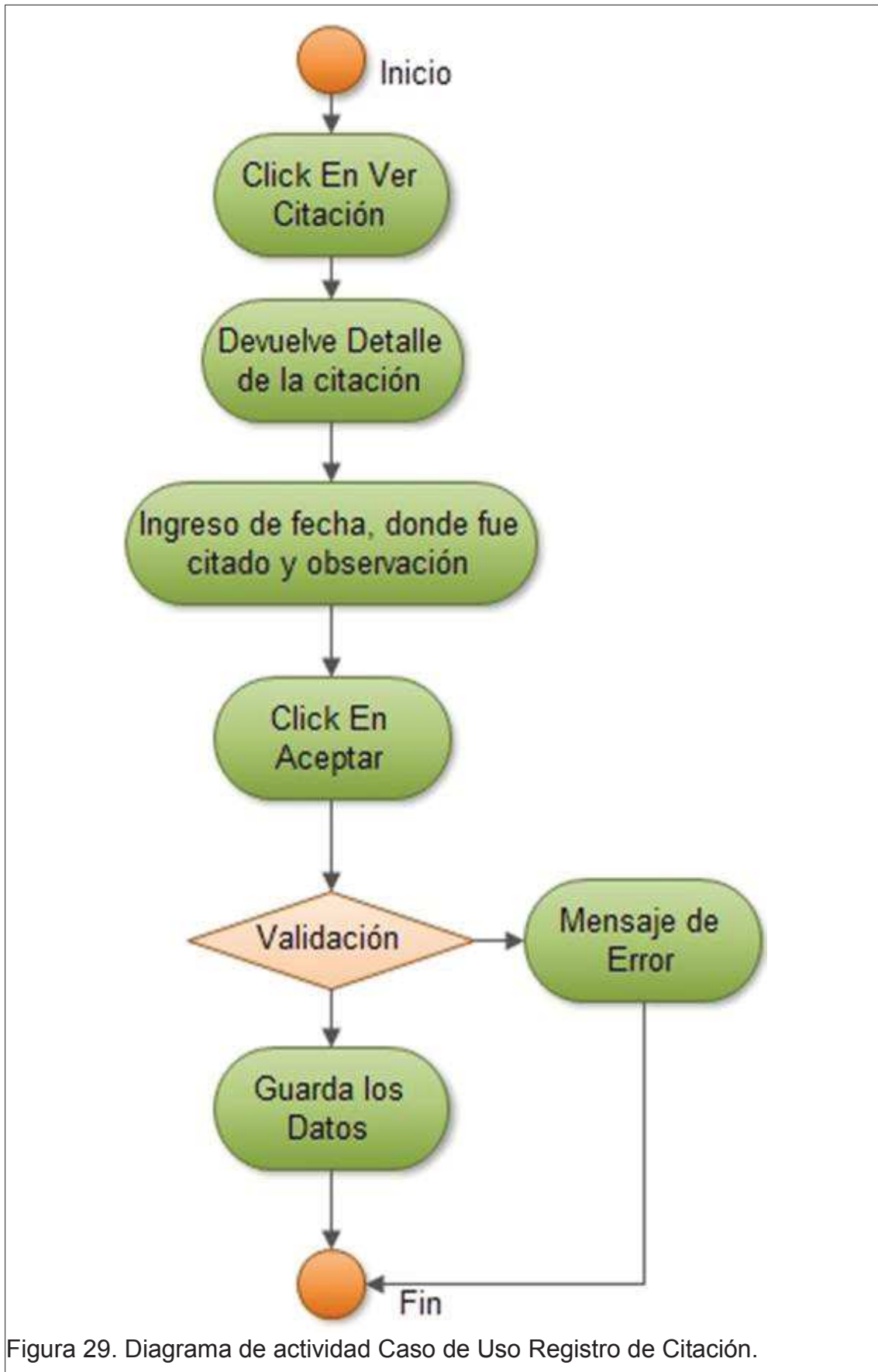


Figura 27. Diagrama de actividad Caso de Uso Autenticación.





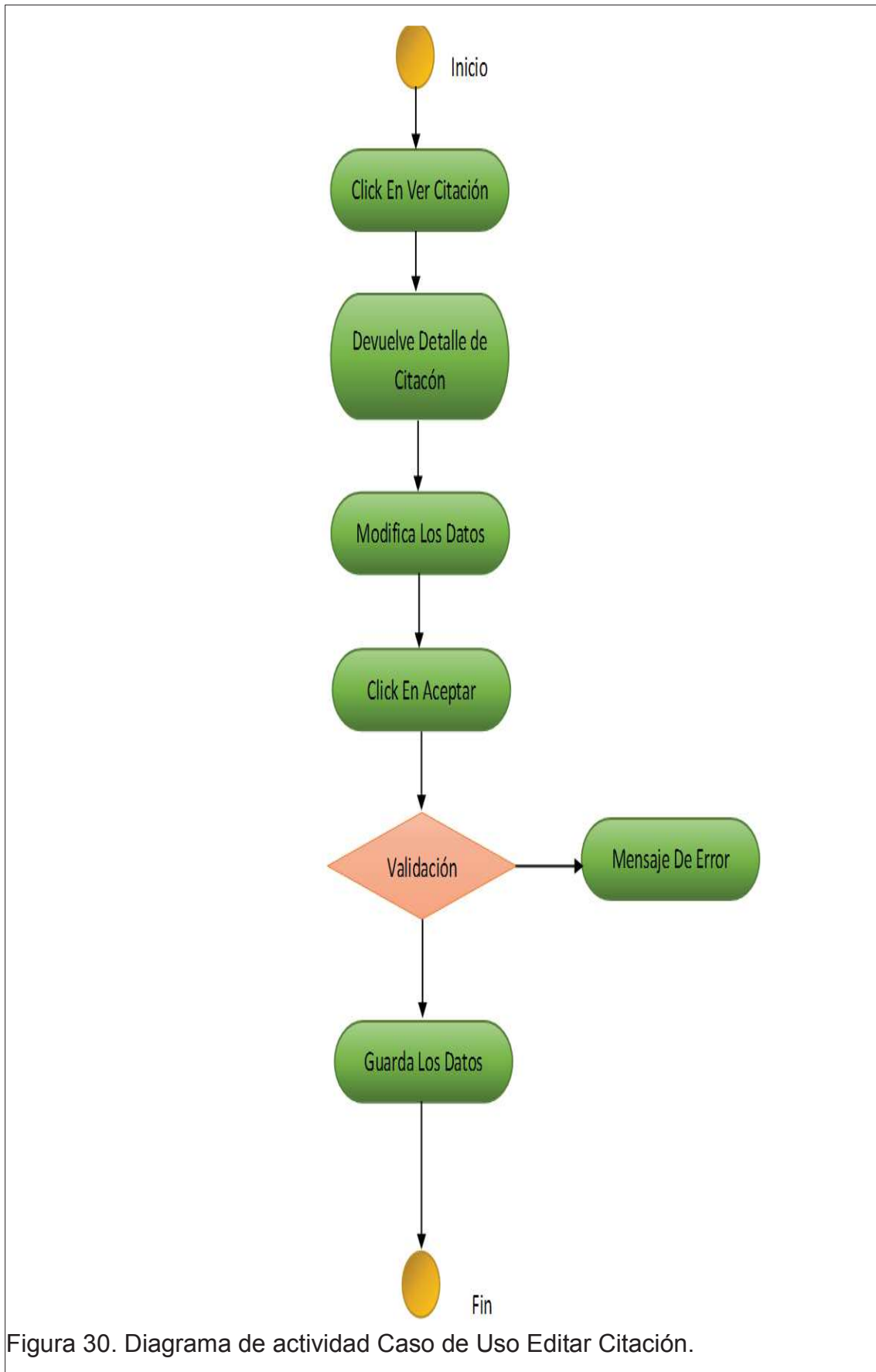


Figura 30. Diagrama de actividad Caso de Uso Editar Citación.

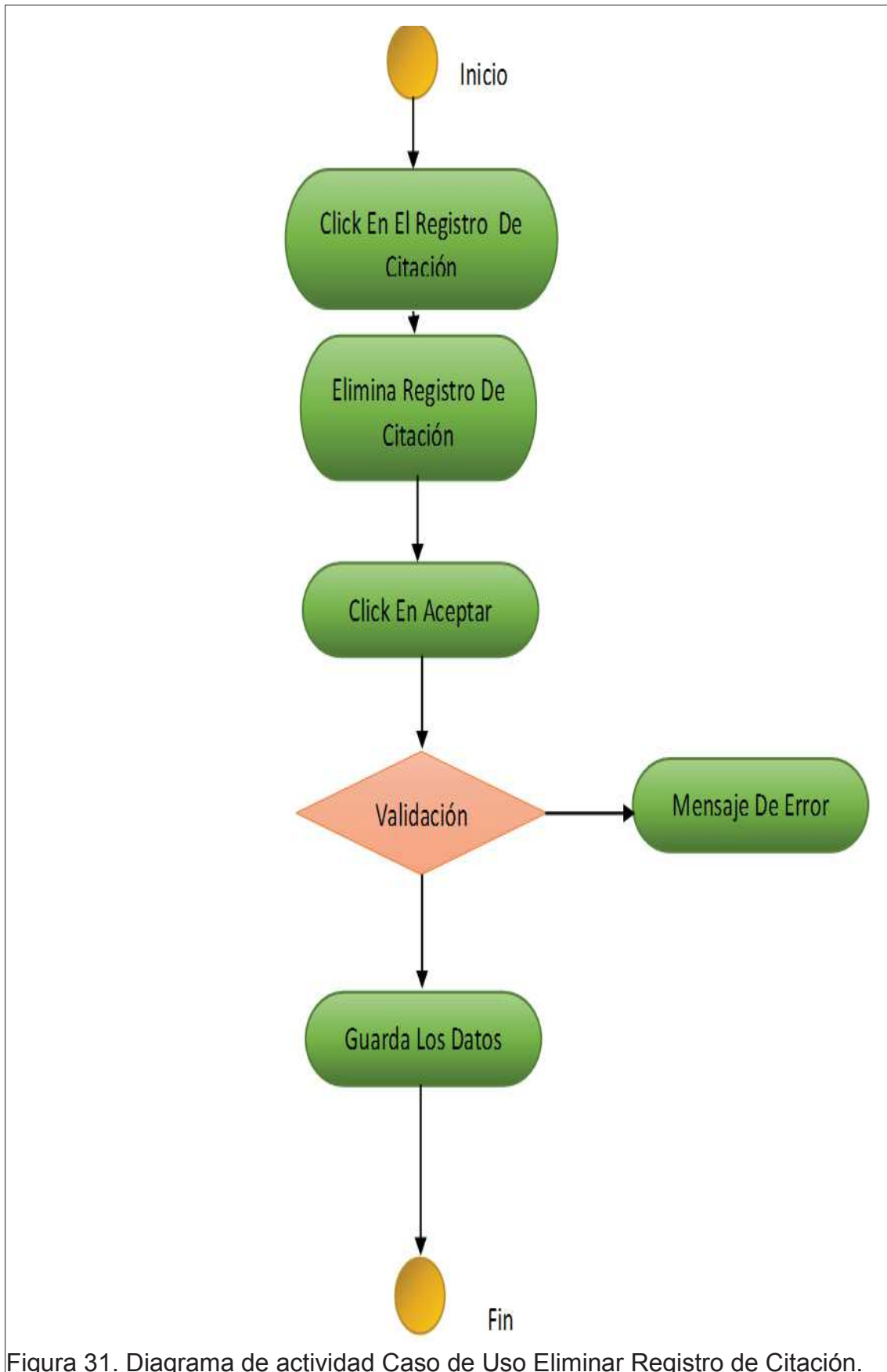


Figura 31. Diagrama de actividad Caso de Uso Eliminar Registro de Citación.

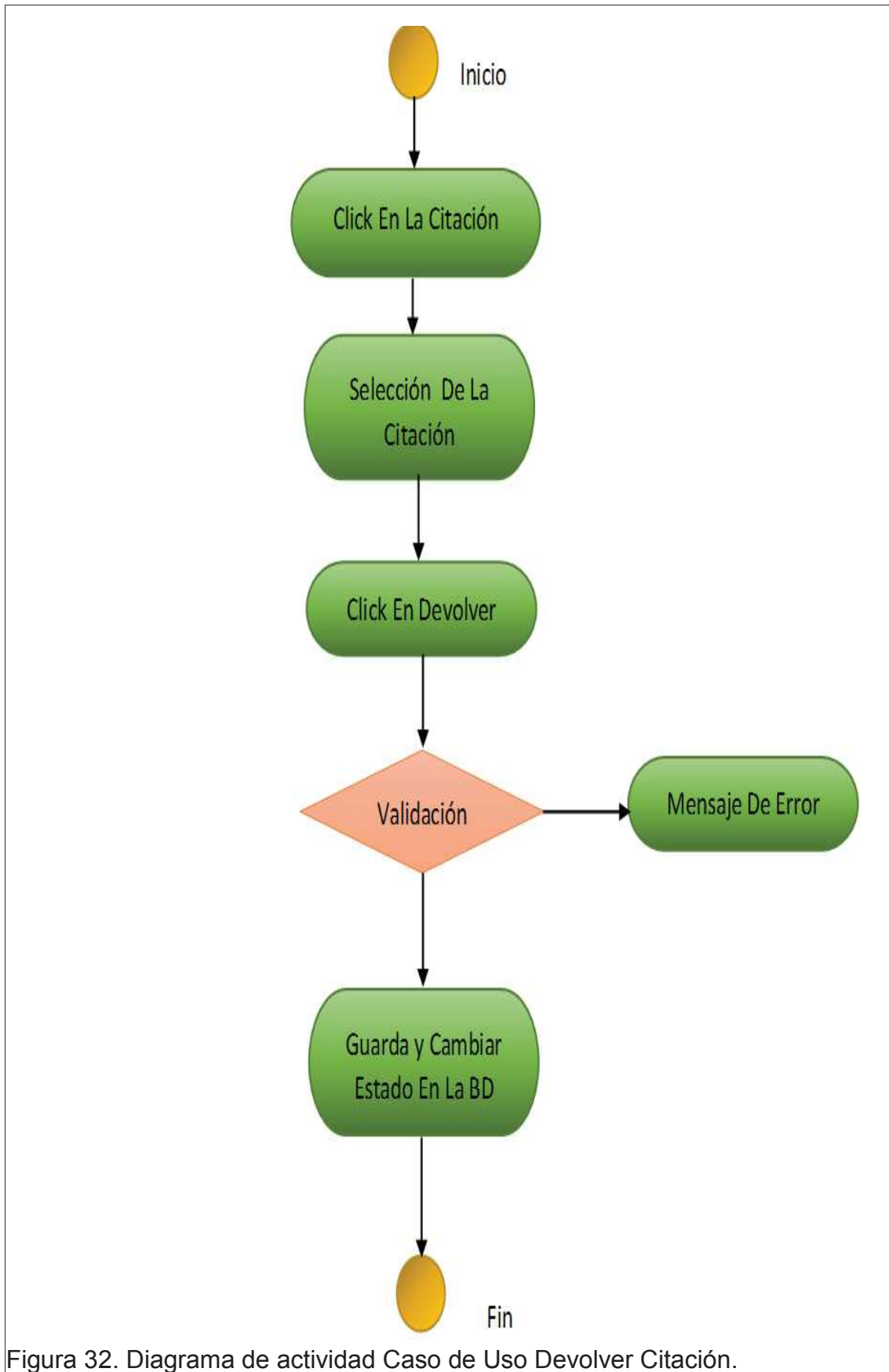


Figura 32. Diagrama de actividad Caso de Uso Devolver Citación.

Secuencia de Eventos

Aquí se establecen los eventos que el actor puede generar los mismos que serán atendidos por cada Caso de Uso.

Todos los eventos siguen una secuencia lógica como ocurriría en la aplicación.

Tabla 14. Secuencia de Eventos Autenticación

CONTROL DE PROYECTOS	FORMATO DE EVENTOS
Nombre del Caso de Uso	Autenticación
Actor Responsable	Usuario Registrado
EVENTO	RESPUESTA DEL SISTEMA
1. Ingresar al Sistema 2. Clic en botón Registrar 3. Ingreso de Judicatura 4. Ingreso de Usuario 5. Ingreso de Contraseña 6. Clic Botón Ingresar 11.a El usuario puede ingresar al sistema con sus respectivos privilegios otorgados por su rol 11. b. Le pide ingresar nuevamente los datos.	3. Capturar la Selección 7. Captura la Selección (datos ingresados) 8. Realiza validaciones. 9. Recoge información de la base de datos. a. INFORMACIÓN CORRECTA 10. a Preparar la interfaz gráfica. b. INFORMACIÓN INCORRECTA 10. b El sistema emite un mensaje de datos incorrectos.

Tabla 15. Secuencia de Eventos Consulta Litigante

CONTROL DE PROYECTOS	FORMATO DE EVENTOS
Nombre del Caso de Uso:	Consulta Litigante
Actor Responsable:	Usuario Registrado
EVENTO	RESPUESTA DEL SISTEMA
<p>1. EVENTO AUTENTICACIÓN</p> <p>2. Selecciona Instancia</p> <p>6. Muestra juzgados.</p> <p>7. Selecciona Juzgado</p> <p>8. Click en botón Buscar</p> <p>15.a. Devuelve todas las citaciones de la Jurisdicción</p> <p>13.b. Pide consultar nuevamente.</p>	<p>3. Captura la Selección</p> <p>4. Valida la información</p> <p>5. Recoge datos de la base de datos</p> <p>9. Captura la Selección</p> <p>10. Valida la información</p> <p>11. Recoge datos de la base de datos</p> <p>a. INFORMACIÓN CORRECTA</p> <p>12.a. Capturar la Selección</p> <p>13.a. Valida la información</p> <p>14.a. Recoge datos de la base de datos</p> <p>b. INFORMACIÓN INCORRECTA</p> <p>12.b. El sistema emite un mensaje de que no existen datos.</p>

Tabla 16. Secuencia de Eventos Registro Citación

CONTROL DE PROYECTOS	FORMATO DE EVENTOS
Nombre del Caso de Uso:	Registro de Citación
Actor Responsable:	Usuario Registrado
EVENTO	RESPUESTA DEL SISTEMA
1. CONSULTA DE CITACIONES 2. Selecciona Citación	
	3. Captura la Selección 4. Valida la información 5. Recoge datos de la base de datos
6. Muestra datos de la citación. 7. Ingresar fecha de visita 8. Ingresar donde fue citado 9. Ingresar observación	10. Captura la Selección (datos ingresados) 11. Valida datos 12. Recoge datos de la base de datos a. INFORMACIÓN CORRECTA
14.a. Despliega mensaje de ingreso satisfactorio 14.b. Pide ingresar los datos nuevamente	13.a. Guarda los datos ingresados b. INFORMACIÓN INCORRECTA 13.b. El sistema emite un mensaje de Error.

Tabla 17. Secuencia de Eventos Editar Citación

CONTROL DE PROYECTOS	FORMATO DE EVENTOS
Nombre del Caso de Uso:	Editar Citación
Actor Responsable:	Usuario Registrado
EVENTO	RESPUESTA DEL SISTEMA
1. CONSULTA DE CITACIONES 2. Selecciona Citación	
	3. Captura la Selección 4. Valida la información 5. Recoge datos de la base de datos
6. Muestra datos de la citación. 7. Modifica la citación	8. Captura la Selección (datos modificados) 9. Valida datos 10. Recoge datos de la base de datos a. INFORMACIÓN CORRECTA
12.a. Despliega mensaje de ingreso satisfactorio 12.b. Pide ingresar los datos nuevamente	11.a. Guarda los datos ingresados b. INFORMACIÓN INCORRECTA 11. b. El sistema emite un mensaje de Error.

Tabla 18. Secuencia de Eventos Eliminar Registro de Citación

CONTROL DE PROYECTOS	FORMATO DE EVENTOS
Nombre del Caso de Uso:	Eliminar Registro de citación
Actor Responsable:	Usuario Registrado
EVENTO	RESPUESTA DEL SISTEMA
1. CONSULTA DE CITACIONES 2. Selecciona Registro	
	3. Captura la Selección 4. Valida la información 5. Recoge datos de la base de datos
6. Elimina la citación	8. Captura la Selección (citación a eliminar) 9. Valida datos 10. Recoge datos de la base de datos a. INFORMACIÓN CORRECTA
12.a. Despliega mensaje de eliminación satisfactorio	11.a. Guarda los cambios b. INFORMACIÓN INCORRECTA 11. b. El sistema emite un mensaje de Error.

Tabla 19. Secuencia de Eventos Devolver Citación

CONTROL DE PROYECTOS	FORMATO DE EVENTOS
Nombre del Caso de Uso:	Devolver Citación
Actor Responsable:	Usuario Registrado
EVENTO	RESPUESTA DEL SISTEMA
<ol style="list-style-type: none"> 1. CONSULTA DE CITACIONES 2. Selecciona Registro 3. Selecciona Litigante 	
	<ol style="list-style-type: none"> 4. Captura la Selección 5. Valida la información 6. Recoge datos de la base de datos
7. Devolver Citación	<ol style="list-style-type: none"> 8. Captura la Selección (citación a eliminar) 9. Valida datos 10. Recoge datos de la base de datos a. INFORMACIÓN CORRECTA
<ol style="list-style-type: none"> 12.a. Despliega mensaje de eliminación satisfactorio 12.b. Despliega mensaje que no es posible devolver. 	<ol style="list-style-type: none"> 11.a. Guarda los cambios en la base de datos b. INFORMACIÓN INCORRECTA 11. b. El sistema emite un mensaje de Error.

3.4.6. DIAGRAMA DE ITERACIONES

Los diagramas de iteración son diagramas que describen como grupos de objetos colaboran para conseguir algún fin. Estos diagramas muestran objetos, así como los mensajes que se pasan entre ellos dentro del caso de uso.

Los diagramas de iteración, representan la forma de como un Actor u Objeto (Clase) se comunica entre sí en petición a un evento. Esto implica recorrer toda

la secuencia de llamadas, de donde se obtienen las responsabilidades claramente. Dicho diagrama se puede obtener, desde el Diagrama Estático de Clases o el de Casos de Uso.

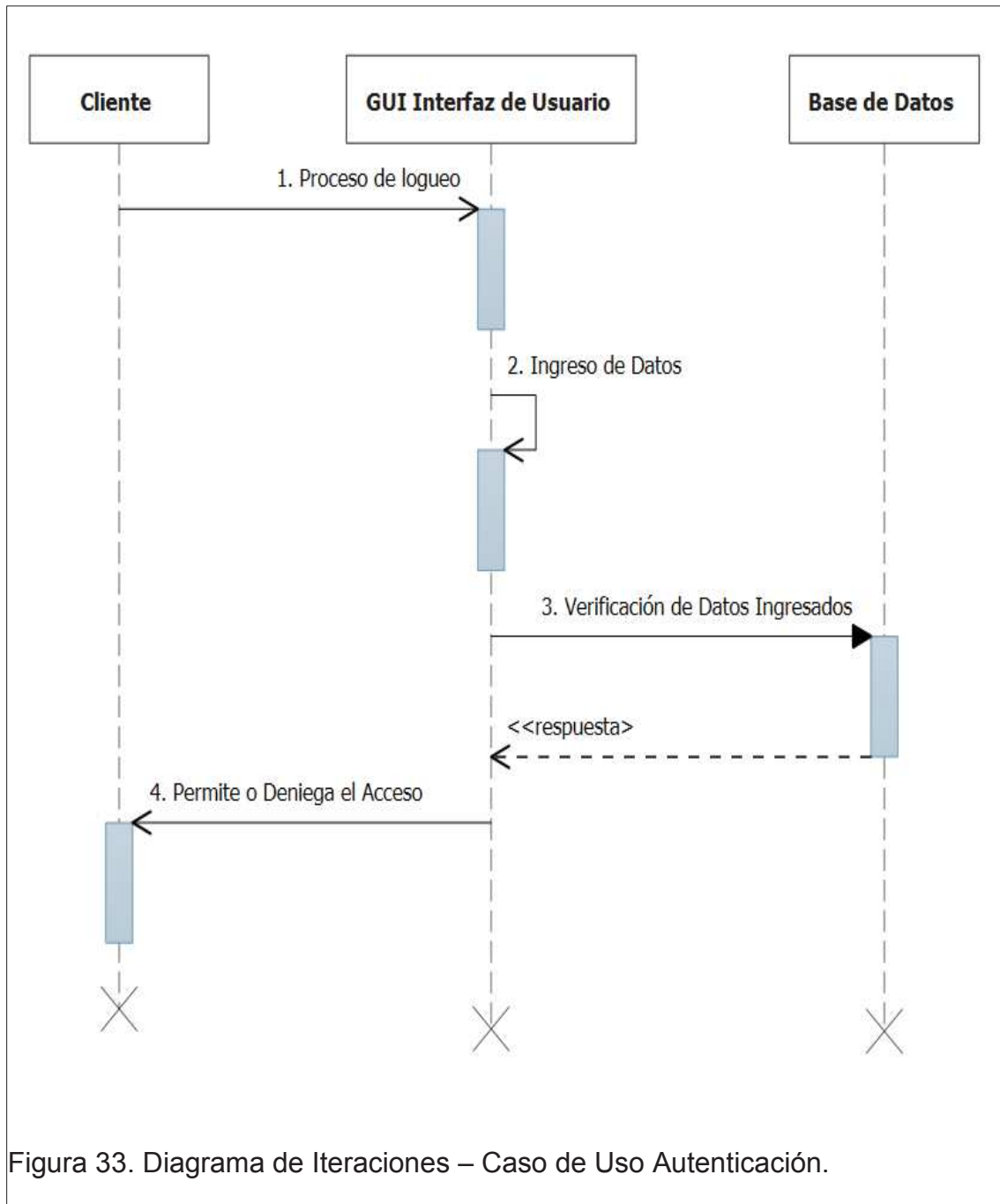


Figura 33. Diagrama de Iteraciones – Caso de Uso Autenticación.

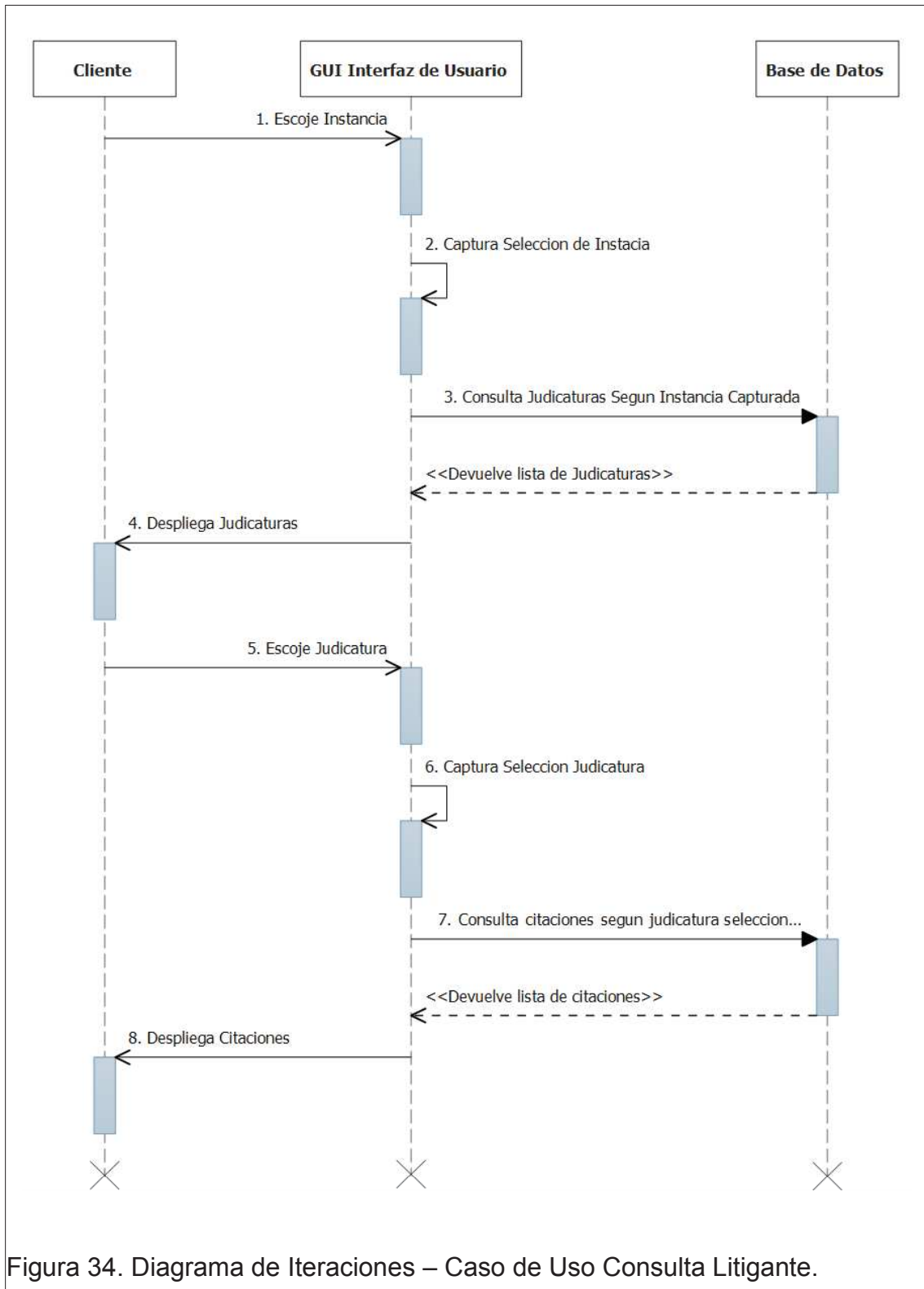


Figura 34. Diagrama de Iteraciones – Caso de Uso Consulta Litigante.

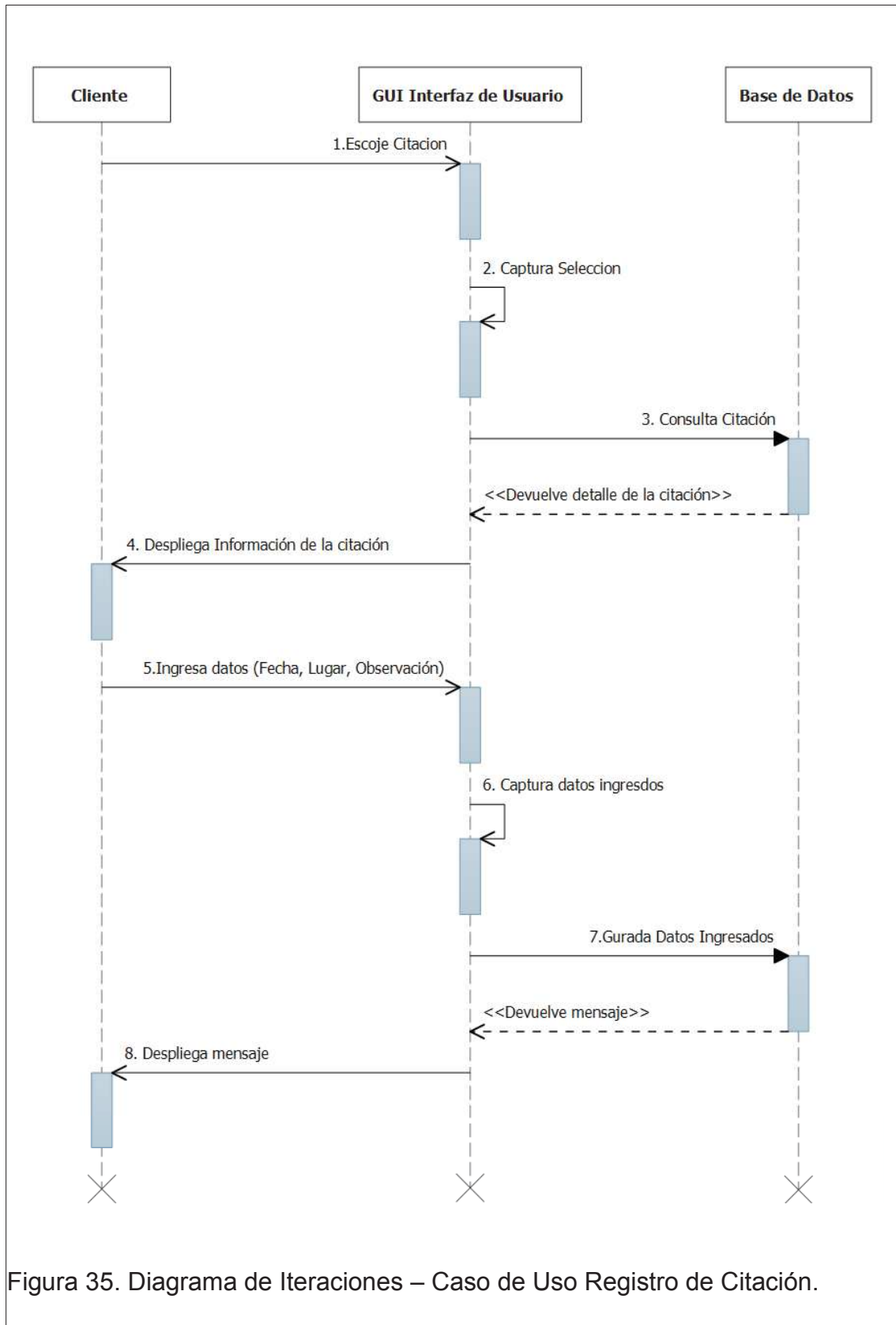


Figura 35. Diagrama de Iteraciones – Caso de Uso Registro de Citación.

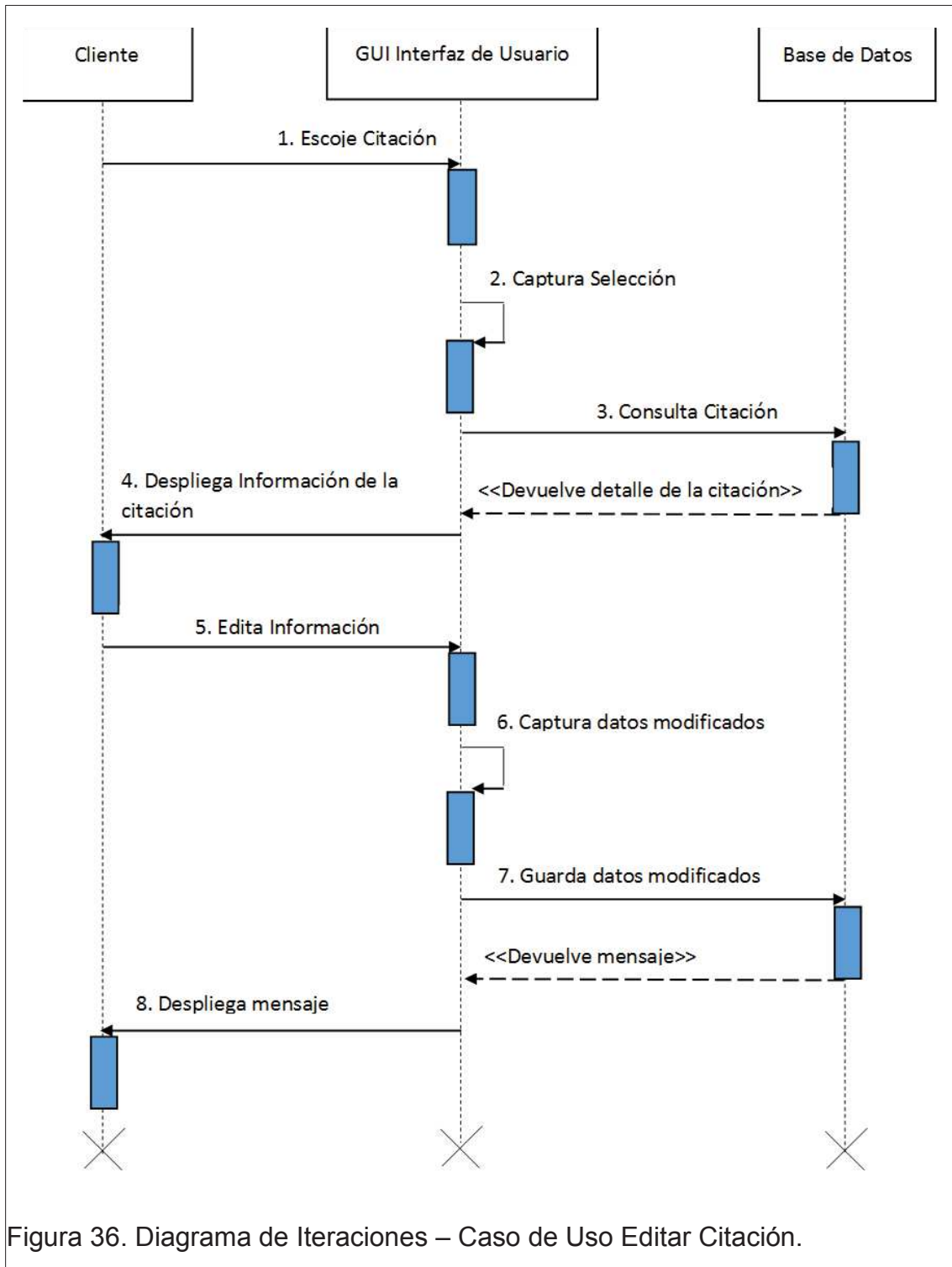
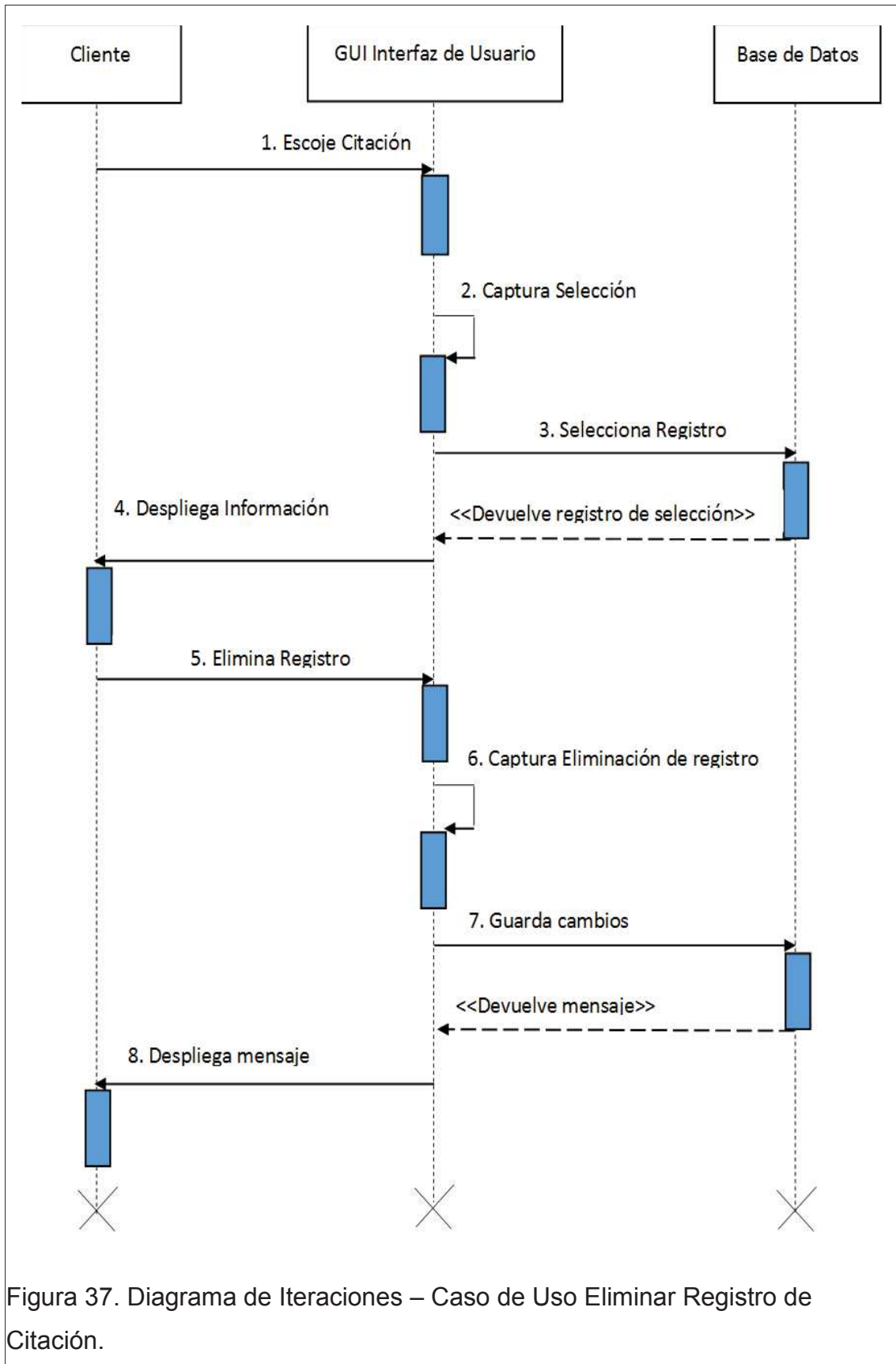


Figura 36. Diagrama de Iteraciones – Caso de Uso Editar Citación.



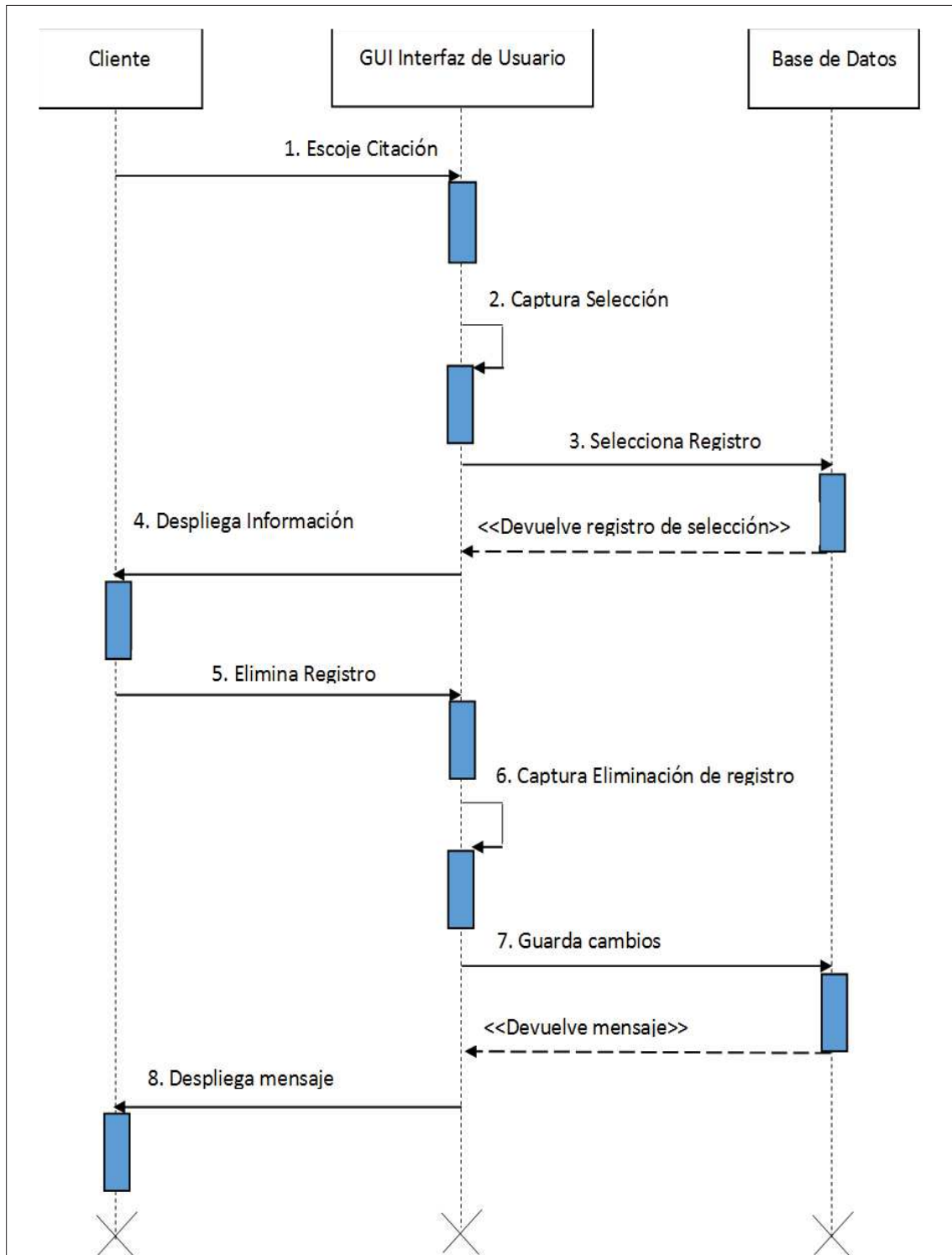
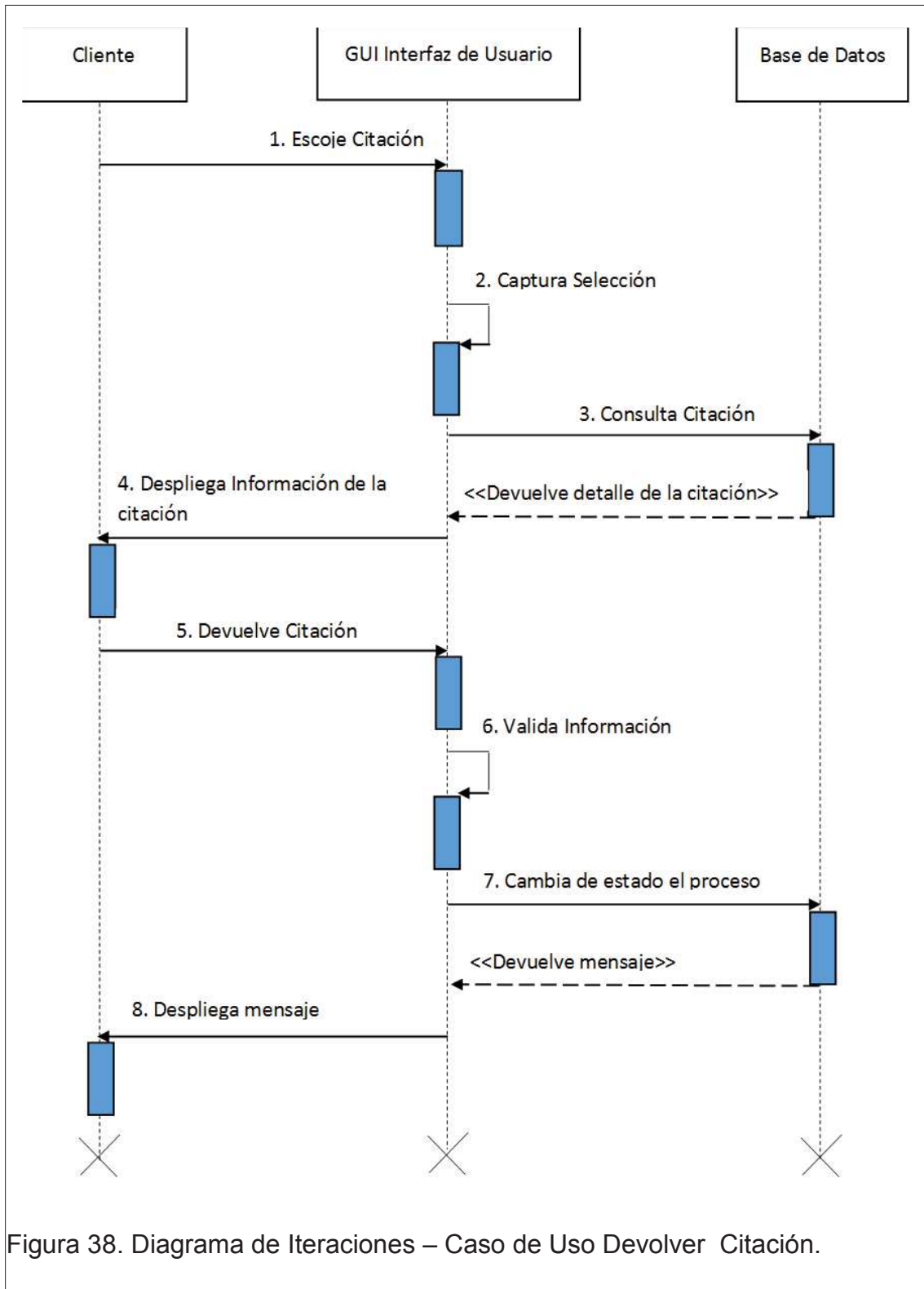


Figura 37. Diagrama de Iteraciones – Caso de Uso Eliminar Registro de Citación.



3.4.7. DICCIONARIO DE DATOS

Tabla Materias					
Columna	Tipo de Dato	Precisión	Longitud	Permite Nulos	Descripción
IdMateria	smallint	NULL	2	NO	Código de la Materia
NombreMateria	char	NULL	60	SI	Nombre de la Materia
Estado	char	NULL	1	SI	Estado de la Materia

Tabla TipoAccion					
Columna	Tipo de Dato	Precisión	Longitud	Permite Nulos	Descripción
IdTipoAccion	int	NULL	4	NO	Código TipoAcción
NombreAccion	char	NULL	70	SI	Nombre de la Acción
IdMateria	smallint	NULL	2	SI	Código de la Materia
Estado	char	NULL	1	SI	Estado que se encuentra
InstruccionFiscal	char	NULL	1	SI	Describe la Instrucción Fiscal

Tabla JudicaturaMaterias					
Columna	Tipo de Dato	Precisión	Longitud	Permite Nulos	Descripción
IdMateria	int	NULL	4	NO	Código de la Materia
IdJudicatura	char	NULL	5	NO	Código de la Judicatura

Tabla DelitosJuicios					
Columna	Tipo de Dato	Precisión	Longitud	Permite Nulos	Descripción
NombreDelito	char	NULL	180	SI	Nombre del Delito
IdDelito	int	NULL	4	SI	Código del Delito
IdRegistro	int	NULL	4	SI	Código del Registro
visible	char	NULL	1	SI	Indica si es visible o no el Delito
IdJuicio	char	NULL	15	NO	Código del Juicio
UltimaModificacion	datetime	NULL	8	SI	Fecha del último cambio que se realizo
DetalleDelito	varchar	NULL	200	SI	Indica el detalle del Delito
IdTitulo	int	NULL	4	SI	Código del Título
IdDetalleDelito	int	NULL	4	SI	Código del Detalle Delito
Flagrante	char	NULL	1	SI	Indica si el delito fue flagrante o no

Tabla GrupoNumeracion					
Columna	Tipo de Dato	Precisión	Longitud	Permite Nulos	Descripción
IdGrupoNumeracion	int	NULL	4	NO	Código Grupo Numeración
Nombre	varchar	NULL	50	NO	Indica el nombre
Caracter	char	NULL	1	SI	Letra que se le puede asignar
DescripcionFe	varchar	NULL	50	SI	Nombre o descripción
Articulo	varchar	NULL	2	SI	Indica el Artículo de pendiendo de la descripción
Estado	char	NULL	1	SI	Indica el Estado en que se encuentra

Tabla Usuarios					
Columna	Tipo de Dato	Precisión	Longitud	Permite Nulos	Descripción
Login	char	NULL	50	NO	Login del Usuario
Nombre	char	NULL	80	SI	Nombre del Usuario
Iniciales	char	NULL	3	SI	Iniciales del Usuario
Categoria	char	NULL	30	SI	Categoría del Usuario
IdJudicatura	char	NULL	5	NO	Código de la Judicatura
PieDeFirma	char	NULL	30	SI	Nombre del Usuario
Password	char	NULL	80	SI	Contraseña del Usuario
Especialidad	char	NULL	30	SI	Indica si el Usuario tiene alguna especialidad
Permisos	char	NULL	7	SI	Indica si el Usuario posee algún tipo de permiso
Estado	char	NULL	1	SI	Indica si el Usuario está habilitado o no
IdCargo	int	NULL	4	SI	Código de Cargo
FechaCtrl	datetime	NULL	8	SI	Fecha de creación del Usuario

Tabla Juicios					
Columna	Tipo de Dato	Precisión	Longitud	Permite Nulos	Descripción
IdJuicio	char	NULL	15	NO	Código del Juicio
IdJudicatura	char	NULL	5	SI	Código de la Judicatura
FechaIngresoJuzgado	datetime	NULL	8	SI	Fecha que se remite al Juzgado
Horaingreso	datetime	NULL	8	SI	Hora que se remite al Juzgado
NoFojas	int	NULL	4	SI	Cuántas hojas ingresan al Juzgado
IdMateria	smallint	NULL	2	SI	Código de la Materia
IdTipoAccion	smallint	NULL	2	SI	Código Tipo Acción
Cuantia	float	53	8	SI	Cuántía que se señala
InstanciaActual	char	NULL	1	SI	Indica la Instancia actual
IdEtapaActual	int	NULL	4	SI	Código Etapa Actual
EstadoActual	char	NULL	1	SI	Señala estado del Juicio
Razon	text	NULL	16	SI	Indica si el Juicio tiene sentada alguna razón
FechaInicio	datetime	NULL	8	SI	Fecha cuando inicia el proceso
IdFormaDeTerminacion	smallint	NULL	2	SI	Código forma de terminación
FechaDeTerminacion	datetime	NULL	8	SI	Fecha de finalización del proceso
TipoIngreso	char	NULL	1	SI	Describe como fue ingresado el proceso
Usuario	char	NULL	30	SI	Indica quien recibió el proceso
Responsable	char	NULL	80	SI	Indica quien ingreso el proceso
Ubicacion	char	NULL	100	SI	Indica la ubicación del proceso
FechaSistema	datetime	NULL	8	SI	Señala la fecha de registro en el sistema
RazonSegunda	char	NULL	100	SI	Indica si el Juicio tiene sentada una segunda razón
LogResponsable	char	NULL	100	SI	Usuario responsable
UltimaModificacion	datetime	NULL	8	SI	Señala fecha de la última modificación
AnexosSegunda	text	NULL	16	SI	Indica si se presentaron anexos
IdInstruccionFiscal	char	NULL	15	SI	Código Instrucción Fiscal
IdProvidenciaActual	int	NULL	4	SI	Código Providencia Actual
FechaIngreso	datetime	NULL	8	SI	Fecha que registra el sistema el ingreso
Maquina	varchar	NULL	80	SI	Nombre de la máquina que se logueo
IdGrupoNumeracion	int	NULL	4	SI	Código grupo numeración
FechaRecibido	datetime	NULL	8	SI	Señala fecha de recibido
NoAdicional	char	NULL	15	SI	Indica si se aumentó algún numero
IdEstadoJuicio	int	NULL	4	SI	Código estado juicio
DetalleEstado	varchar	NULL	100	SI	Describe el detalle del estado
FechaEstado	datetime	NULL	8	SI	Indica la Fecha del estado
logjuez	char	NULL	100	SI	Señala el loguin del juez
Cuerpos	int	NULL	4	SI	Indica el número de cuerpos
OBSERVACION	char	NULL	255	SI	Describe alguna observación
SorteoJuez	char	NULL	5	SI	Indica donde está el proceso

Tabla UsuarioJudicatura					
Columna	Tipo de Dato	Precisión	Longitud	Permite Nulos	Descripción
IdJudicatura	char	NULL	5	NO	Código de la Judicatura
Login	char	NULL	50	NO	Login del Usuario
IdCargo	int	NULL	4	NO	Código de Cargo

Tabla CategoriJudicatura					
Columna	Tipo de Dato	Precisión	Longitud	Permite Nulos	Descripción
IdCategoriaJudicatura	int	NULL	4	NO	Código categoría Judicatura
Nombre	varchar	NULL	50	SI	Nombre de la Instancia
Estado	char	NULL	1	SI	Estado de la Instancia

Tabla Judicaturas					
Columna	Tipo de Dato	Precisión	Longitud	Permite Nulos	Descripción
IdJudicatura	char	NULL	5	NO	Código de la Judicatura
Nombre	char	NULL	160	SI	Nombre de la judicatura
Tipo	char	NULL	6	SI	Tipo de la Judicatura
Ciudad	char	NULL	20	SI	Ciudad de la Judicatura
Direccion	char	NULL	50	SI	Dirección de la Judicatura
Telefono	char	NULL	12	SI	Teléfono de la Judicatura
Descripcion	char	NULL	30	SI	Descripción de la Judicatura
e_mail	char	NULL	50	SI	Correo electrónico de la Judicatura
Orden	int	NULL	4	SI	Número asignado según el nombre de la Judicatura
AliasJudicatura	char	NULL	180	SI	Nombre corto de la Judicatura
IdCategoríaJudicatura	int	NULL	4	SI	Código de Categoría Judicatura
IdCanton	int	NULL	4	SI	Código del Cantón de la Judicatura
NoFormularioTasa	int	NULL	4	SI	Número de Formulario de la Judicatura
NoFormularioDeposito	int	NULL	4	SI	Numero de Formulario para Deposito
NoCuenta	char	NULL	20	SI	Número de Cuenta para el Deposito
Turno	char	NULL	1	SI	Indica si la Judicatura realiza turnos
ActiAnteriores	char	NULL	1	SI	Indica si la Judicatura realizaba otras tareas
IdMateria	int	NULL	4	SI	Código de la Materia
IdTipoJudicatura	char	NULL	3	SI	Código del Tipo Judicatura
Observacion	char	NULL	100	SI	Si se tiene alguna Observación sobre la Judicatura
IdCorteSuperior	varchar	NULL	5	SI	Código de la Corte Superior
Area	char	NULL	20	SI	Área de la Judicatura
Nivel	char	NULL	11	SI	Nivel de la Judicatura

Tabla CitacionesRealizadas					
Columna	Tipo de Dato	Precisión	Longitud	Permite Nulos	Descripción
IdCitacionRealizada	int	NULL	4	NO	Código Citación realizada
IdCitacionJuicio	int	NULL	4	NO	Código Citación Juicio
IdCitacionLitigante	int	NULL	4	NO	Código Citación del litigante
IdJuicio	varchar	NULL	15	SI	Código del Juicio
IdJudicatura	varchar	NULL	5	SI	Código de la Judicatura
TipoCitacion	varchar	NULL	40	SI	Describe el Tipo de Citación
Citador	varchar	NULL	30	SI	Indica el login del Citador
FechaSistema	datetime	NULL	8	SI	Fecha de Ingreso al Sistema
FechaCitacion	datetime	NULL	8	SI	Fecha de realización de la Citación
HoraCitacion	datetime	NULL	8	SI	Hora de realización de la Citación
Realizada	varchar	NULL	2	SI	Indica si se realizó o no la Citación
RecibidoPor	char	NULL	20	SI	Indica quien recibió la Citación
NoIntento	smallint	NULL	2	SI	Indica cuantas citaciones se le realiza
Observaciones	varchar	NULL	250	SI	Describe alguna observación
Firma	varchar	NULL	2	SI	Indica si la persona firmó o no la entrega de la Citación
Direccion	varchar	NULL	200	SI	Indica la Dirección donde se entregó la Citación
Login	varchar	NULL	30	SI	Indica quien ingreso el proceso para dar orden de Citación
RazonCitacion	text	NULL	16	SI	Motivo de la Citación

Tabla CitacionesLitigantes					
Columna	Tipo de Dato	Precisión	Longitud	Permite Nulos	Descripción
IdCitacionLitigante	int	NULL	4	NO	Código Citación del Litigante o procesado
Nombres	char	NULL	200	SI	Nombre del Litigante
IdJuicio	char	NULL	15	SI	Código del Juicio
Direccion	varchar	NULL	200	SI	Dirección del Litigante
Referencia	varchar	NULL	500	SI	Referencia de la Dirección del Litigante
IdZona	int	NULL	4	SI	Código de la zona según la dirección
IdSector	int	NULL	4	SI	Código del sector según la Zona
Citador	varchar	NULL	30	SI	Login del usuario a quien le toca entregar la Citación

Tabla CitacionesJuicios					
Columna	Tipo de Dato	Precisión	Longitud	Permite Nulos	Descripción
IdCitacionJuicio	int	NULL	4	NO	Código Citación Juicio
IdJuicio	varchar	NULL	15	SI	Código del Juicio
IdJudicatura	varchar	NULL	5	SI	Código de la Judicatura
FechaIngreso	datetime	NULL	8	SI	Fecha de ingreso
HoraIngreso	datetime	NULL	8	SI	Hora de Ingreso
FechaSalida	datetime	NULL	8	SI	Fecha de salida
HoraSalida	datetime	NULL	8	SI	Hora de salida
Motivo	varchar	NULL	100	SI	Motivo de la salida del proceso
Estado	varchar	NULL	50	SI	Estado del proceso
UsuarioIngreso	varchar	NULL	50	SI	Login del Usuario quien ingreso la Citación
UsuarioSalida	varchar	NULL	50	SI	Login del Usuario quien termino el proceso de Citación

3.5. IMPLEMENTACIÓN EN LA WEB

Se genera la publicación del proyecto, para lo cual es indispensable tener instalado el IIS. (Ver Anexo 5)

Vale la pena recalcar que los requerimientos son los citados a continuación:

- Instalación de Framework 4.0
- Internet Information Services de Windows correctamente configurado.

Posterior a los requerimientos es necesario subir tal publicación en un dominio virtual.

Los Clientes tendrán acceso a la página mediante un browser del dispositivo móvil y tendrá privilegios dependiendo del tipo de Usuario ingresado.

3.6. PRUEBAS

3.6.1. REALIZACIÓN DE PRUEBAS

Luego de haber concluido la implantación de la aplicación, se espera que los usuarios empiecen con la interacción con el sistema.

Las pruebas se realizan en el periodo de una semana, durante el horario normal de trabajo:

Se pide a los usuarios que ingresen la información de las visitas mediante el dispositivo móvil realizándolo de forma paralela con las hojas de ingreso actuales; A partir de este proceso se espera reparar posibles errores que se puedan producir y a su vez, mejorar cualquier sugerencia brindada.

Tabla 20. Pruebas Autenticación

Tipo Transacción	Usuario	Contraseña	Resultado Esperado	Resultado Obtenido	Estado	Validador
Autenticación	JPEREZ	123456	Usuario Inexistente	Usuario Inexistente	OK	Personal Op.
Autenticación	TMOLINA	molina74	Usuario no pertenece a la Judicatura	Usuario no pertenece a la Judicatura	OK	Personal Op.
Autenticación	IBARRAG	a1a2a3	Contraseña Incorrecta	Contraseña Incorrecta	OK	Personal Op.
Autenticación	IBARRAG	123456	Autenticación Exitosa	Autenticación Exitosa	OK	Personal Op.

Tabla 21. Pruebas de Consulta y Registro de Citaciones

CONSULTA LITIGANTES Y REGISTRO DE CITACIONES												
Tipo Transaccion	Instancia	Materia	Año-Numero	Resultado Esperado	Tipo Causa Esperada	Delito Esperado	Demandado(s) Esperado(s)	Tipo Causa Obtenida	Delito Obtenido	Demandado(s) Obtenidos(s)	Estado	Validador
Consulta											OK	Personal Op.
Registro											OK	Personal Op.
Consulta											OK	Personal Op.
Registro											OK	Personal Op.

3.6.2. RESULTADOS DE PRUEBAS

SwCITAC, fue realizado con según las necesidades solicitadas por los usuarios, sin embargo, no siempre se puede lograr separar estos dos enfoques (desarrollador y usuario); Por tal razón la primera versión que fue implementada, tuvo errores que fueron solucionados.

A continuación se centran las causas y cambios efectuados a favor de las pruebas realizadas.

- Al consultar las citaciones por judicatura desplegaba todas al mismo tiempo sin existir una navegación apropiada para el dispositivo móvil.
- No se puedo visualizar la impresión de las boletas ya que se requiere un hardware específico para dispositivos móviles.

Vale la pena recalcar, como se puede observar, no tuvo errores de relevancia en su ejecución.

Luego de la corrección de los errores presentados se procedió a realizar las pruebas correspondientes, en la Tabla 19 se observa que el sistema valido correctamente en los cuatro intentos que se realizó.

De igual forma en la Tabla 20 se observa que el sistema está consultando y registrando correctamente las citaciones.

3.6.3. REUNIONES DE ENTREGA Y USO

Al finalizar correctamente el desarrollo de la herramienta, se realiza una reunión con los principales usuarios de “Departamento de Citaciones de la Dirección Provincial del Consejo de la Judicatura Distrito-Guayas”.

De esta reunión se obtiene la siguiente evaluación:

3.7 EVALUACIÓN FINAL

Tabla 22. Evaluación Final

Parámetro	Evaluación
Herramienta	La Utilización de UML da una buena comprensión del funcionamiento del sistema, por consiguiente se obtiene un sistema modular y escalable. Como la base de datos está realizada en SQL Server, se puede decir que es una base robusta.
Desempeño	Es la eficiencia desde el punto de vista del usuario (Citadores). Los usuarios se encuentran satisfechos ya que con el sistema no realizan doble trabajo que venían haciendo anteriormente.
Utilización	Los resultados en la fase de pruebas indican que el funcionamiento ha cumplido las expectativas principales del proyecto.
Manejo	El sistema fue realizado pensando en el usuario, de tal manera que sea fácil de manejar.
Contribución	SWCITAC contribuye a los usuarios brindando un servicio: fácil y eficiente.

CAPITULO IV

4. CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

- Se puede concluir que SWCITAC cumple con los objetivos trazados en este proyecto, dado así una satisfacción a los usuarios. La facilidad de uso hacen de este programa un software atractivo y amigable.
- La identificación de los requerimientos del cliente podríamos afirmar que es el escalón más importante y de mucho cuidado en el desarrollo del proyecto, ya que estos brindan las pautas para cumplir las metas y mejorar la productividad del sistema.
- En SWCITAC una ventaja es su modularidad, ya que se redujo el tiempo de trabajo de programación, también el de poder añadir módulos cuando estos sean necesarios a fin de satisfacer las necesidades del cliente.
- SWCITAC da la oportunidad de que varios usuarios naveguen al mismo tiempo en la página, implementando concurrencia en el sistema, por lo que se puede decir que se ha mejorado el trabajo diario de los usuarios (Citadores) realizando el trabajo cotidiano en una forma directa.
- La elección de ASP.NET MVC 4 para este proyecto fue una buena alternativa ya que con las nuevas características que presenta permiten crear aplicaciones que se cargan de forma más rápida, esto es porque se disminuye el tiempo y el tamaño que hacen las paginas a los HTTP a esto se suma los nuevos métodos que permiten un mejor manejo de archivos.
- Al momento de la realización de pruebas se pudo constatar que los usuarios (Citadores), no opusieron ninguna resistencia al cambio o a la utilización de esta aplicación esto se debe a que la misma es muy amigable y fácil de usar.

4.2. RECOMENDACIONES

- Es bueno indicar que ninguna red ni programa será 100% seguro. Hasta ahora no se ha reportado ningún inconveniente sobre manipulación de datos, sin embargo es recomendable tratar de incorporar cada vez mejoras en temas de seguridad.
- Las herramientas escogidas en este proyecto posiblemente no son las mejores de todas, sin embargo para la Dirección Provincial del Consejo de la Judicatura distrito-Guayas y su departamento de Citaciones son las más adecuadas. Para futuras implementaciones se recomienda hacer un análisis mucho más completo de cada herramienta para garantizar que las escogidas funcionen tal como lo requiere la institución.
- Se recomienda adquirir el hardware correspondiente para poder realizar las impresiones de las citaciones, mismas que serán archivadas en la carpeta física correspondiente.
- Se recomienda a la Dirección Provincial del Consejo de la Judicatura Distrito-Guayas ver la prioridad de utilizar esta aplicación ya que se evitaría malos entendidos entre los usuarios externos por el hecho de que se pierden los papeles o las citaciones realizadas así también con los usuarios internos (Citadores) ya que no van a poder manipular su trabajo, esto por la garantía que brinda el presente proyecto.

REFERENCIAS

- Alarcón Aguin, J. M. (2009). Tecnologías ASP.NET 4.0. España: Krasis Press.
- cintiamdominguez.blogspot.com. (18 de 04 de 2012). Base de datos relacional. Obtenido de <http://cintiamdominguez.blogspot.com/2012/04/base-de-datos-relacional.html>
- Collector, G. (s.f.). Garbage Collector. Obtenido de http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/concepto_de_base_de_datos.php
- Consejo de la Judicatura de Transición. (2011). Funcion Judicial de Guayas. Obtenido de www.funcionjudicial-guayas.gob.ec
- Consejo Nacional de Ciencia y Tecnología Perú, S. (1999). Evaluación teórica de la capacidad de procesos de Rational Unified Process. Obtenido de <http://www.scielo.org.pe/pdf/id/v13n2/a11v13n2.pdf>
- Copyright, Scribd Inc. (2013). Lo nuevo en .NET Framework 4. Obtenido de <http://es.scribd.com/doc/63230965/Framework-4-Definiciones>
- Cristalab. (08 de 01 de 2004). Programación orientada a objetos en Visual Basic. NET. Obtenido de <http://www.cristalab.com/tutoriales/programacion-orientada-a-objetos-en-visual-basic-.net-c273/>
- Desarrolloweb.com. (04 de 04 de 2011). Introducción a ASP.NET MVC. Obtenido de <http://www.desarrolloweb.com/articulos/introduccion-asp-net-mvc-dotnet.html?&lang=es&output=json>
- desarrolloweb.com. (s.f.). El motor de vistas Razor. Obtenido de <http://www.desarrolloweb.com/articulos/motor-vistas-razor-dotnet.html>
- es.scribd.com. (30 de 10 de 2010). Especificación de requisitos desoftware. Obtenido de <http://es.scribd.com/doc/59077888/Ejemplo-Formato-IEEE-830>
- espinosa, h. (3 de mayo de 2011). Introducción a .NET Framework 4. Obtenido de <http://frameworkdotnet.wordpress.com/2011/05/03/introduccion-a-net-framework-4/>
- gravitar.biz. (12 de 03 de 2008). SQL Server 2008, nuevas características. Obtenido de <http://www.gravitar.biz/index.php/sql-server/sql-server-2008/>

- GUERRERO GONZALES, PEDRO ENRIQUE. (01 de 07 de 2011). Oracle-vs-sql-server-terminado-8483940. Obtenido de <http://www.slideshare.net/PedroEnrique26/oracle-vs-sql-server-terminado-8483940>
- Indaltronia.com. (17 de 12 de 2010). Guía: Instalación de SQL Server 2008. Obtenido de <http://indaltronia.com/2010/12/guia-instalacion-de-sql-server-2008/>
- iteso.mx. (2000). Proceso de Desarrollo de ERS. Obtenido de http://iteso.mx/~juanjo/materiales/IEEE_Std1233_1998_esp_desarrollo_de_especificacion_de_reque.pdf
- James Martin, James J. Odell. (1997). Object-Oriented Methods. En J. J. James Martin, Object-Oriented Methods. USA: Hardcover.
- Larman, C. (1999). UML y Patrones Introducción al análisis y diseño orientado a objetos. Mexico: Pearson Educación.
- Larman, C. (2003). UML Y Patrones - Una introduccion al análisis y diseño orientado a objetos y al proceso unificado. España: Pearson Education.
- Microsoft Asp.Net. (15 de 08 de 2012). Introducción ASP.NET MVC 4 . Obtenido de <http://www.asp.net/mvc/tutorials/mvc-4/getting-started-with-aspnet-mvc4/intro-to-aspnet-mvc-4>
- Microsoft. (2012). Visual Studio editions. Obtenido de <http://www.microsoft.com/visualstudio/en-us/products/2010-editions/ultimate/overview>
- Microsoft. (s.f.). msdn. Obtenido de <http://msdn.microsoft.com/es-es/library/ms731082.aspx>
- mono-lab. (20 de 02 de 2012). Lo nuevo en ASP.NET MVC 4 Beta. Obtenido de <http://ismapro.wordpress.com/2012/02/20/lo-nuevo-en-asp-net-mvc-4-beta/>
- Oracle.com. (2011). Oracle Database 11g Enterprise Edition. Obtenido de <http://www.oracle.com/us/products/database/enterprise-edition/overview/index.html>
- planetacodigo.com. (29 de 09 de 2011). ASP.NET MVC Request Pipeline. Obtenido de

<http://www.planetacodigo.com/planeta/date/2011/09/?&lang=es&output=json>

Pressman, R. (2002). Ingeniería de Software. Madrid: MacGraw Hill.

Pressman, R. S. (2005). Ingeniería de Software un enfoque práctico. Mexico: Mc Graw Hill.

PROAÑO, ING. DIEGO JAVIER BURBANO. (05 de 05 de 2006). ANALISIS COMPARATIVO DE BASES DE DATOS DE CODIGO ABIERTO. Obtenido de

<http://www.willydev.net/InsiteCreation/v1.0/willycrawler/2008.05.12.articulo.comparacion%20bases%20de%20datos%20open%20y%20propietarias.pdf>

sourceforge, M. (s.f.). Construir Aplicaciones en HTML5: Lo que neces. Obtenido de <http://elcentrohtml5.sourceforge.net/Construir-Aplicaciones-en-HTML5-Lo-que-necesita-saber>

technet.microsoft.com. (11 de 09 de 2012). Features Supported by the Editions of SQL Server 2012. Obtenido de <http://technet.microsoft.com/en-us/library/cc645993.aspx>

Ultimate, Microsoft Visual Studio 2010. (2012). Microsoft Visual Studio 2010 Ultimate Trial - Web Install. Obtenido de <http://www.microsoft.com/en-us/download/details.aspx?id=12752>

Universidad de las Américas Puebla. (13 de 12 de 2012). Colección de Tesis Digitales. Obtenido de http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/rea_c_ji/capitulo3.pdf

UNIVERSIDADPOLITÉCNICADECARTAGENA. (s.f.). Diseño Y desarrollo de una aplicación en .net para el . Obtenido de <http://repositorio.bib.upct.es/dspace/bitstream/10317/2717/1/pfc4259.pdf>

Veloso, Bryan. (01 de 2008). ARQUITECTURA DE ORACLE. Obtenido de <http://infobasedatos.wordpress.com/arquitectura-de-oracle/>

Veracruzana, Universidad. (14 de 12 de 2009). Analisis Comparativo entre SMBD. Obtenido de <http://www.slideshare.net/guest92c0d4/analisis-comparativo-my-sql-vs-oracle-2716900>

web, d. (s.f.). Introducción a ASP.NET MVC. Obtenido de <http://www.desarrolloweb.com/articulos/introduccion-asp-net-mvc-dotnet.html>

Wikipedia, Angeldx7. (25 de 08 de 2008). Proceso Unificado de Rational. Obtenido de http://es.wikipedia.org/wiki/Archivo:Rup_espanol.gif

(s.f.). Obtenido de http://www.google.com.ec/search?q=base+de+datos+relacionales&hl=es&biw=1366&bih=677&prmd=imvnsb&tbm=isch&tbo=u&source=univ&sa=X&ei=rssRUMn_MOTF6wHYtIDQBQ&ved=0CFsQsAQ#q=base+de+datos+relacionales&hl=es&sa=X&tbm=isch&prmd=imvnsb&bav=on.2,or.r_gc.r_pw.r_qf.

ANEXOS

ANEXO 1

MVC

FUENTE: (web, s.f.)

El Patrón Modelo - Vista - Controlador (MVC)

ASP.NET MVC es, básicamente, una implementación del patrón Modelo - Vista - Controlador (MVC) para tecnología ASP.NET. El patrón MVC no es ni nuevo (data de finales de los años 70) ni está pensado para aplicaciones web, pero en realidad en aplicaciones web encaja perfectamente.

Brevemente podemos decir que el patrón MVC separa la lógica (y acceso a datos) de una aplicación de su presentación, usando 3 componentes:

Modelo: Representa las reglas de negocio de la aplicación (y el acceso a datos subyacente).

Vistas: Representan la presentación de la aplicación.

Controlador: Actúan de intermediario entre el usuario y el Modelo y las Vistas. Recogen las peticiones del usuario, interactúan con el modelo y deciden que vista es la que debe mostrar los datos.

En el contexto de ASP.NET MVC:

Toda la lógica de negocio y el acceso a datos es el Modelo (en muchos casos el Modelo puede estar en uno o varios assemblies referenciados).

Las vistas contienen, básicamente, el código que se envía al navegador, es decir el código HTML (y código de servidor asociado, siempre y cuando este código haga cosas de presentación, no de lógica de negocio).

Los controladores reciben las peticiones del navegador y en base a esas, deciden que vista debe enviarse de vuelta al navegador y con qué datos.

Ventajas de ASP.NET MVC

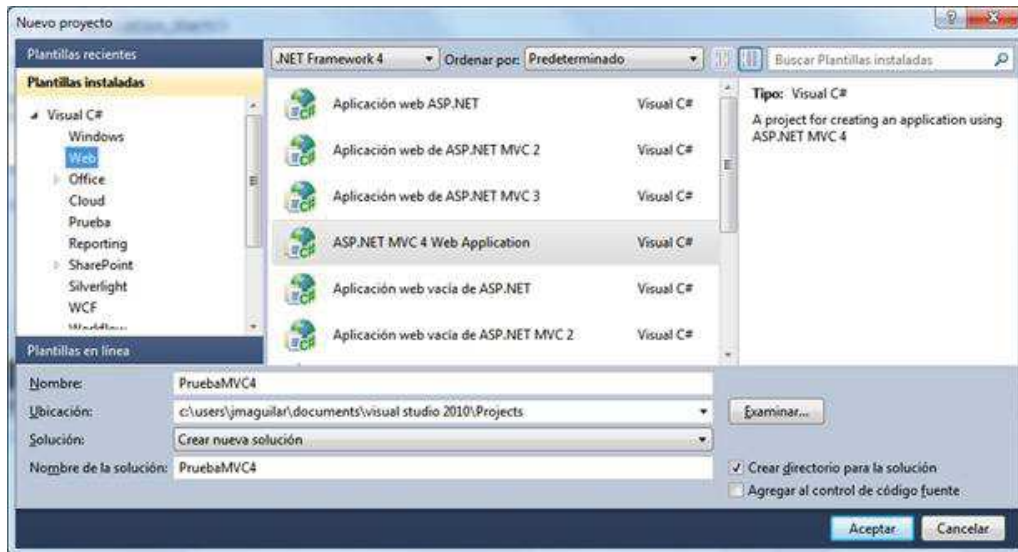
La ventaja que primero salta a la vista de ASP.NET es la facilidad con la que se generan URL semánticas, es decir URL que tengan la forma `http://servidor/ver/productos/cafeteras` en lugar de `http://servidor/productos/ver.aspx?code=cafeteras`. Las URLs semánticas se indexan mejor en los buscadores y son una práctica SEO habitual. No es que en webforms no se puedan hacer, es que en ASP.NET MVC vienen de serie.

Otras ventajas, más a nivel técnico, son que con ASP.NET MVC se facilita mucho el probar nuestra aplicación (especialmente usando pruebas unitarias) y que el uso correcto del patrón MVC facilita la reutilización de código de manera mucho más efectiva que en webforms.

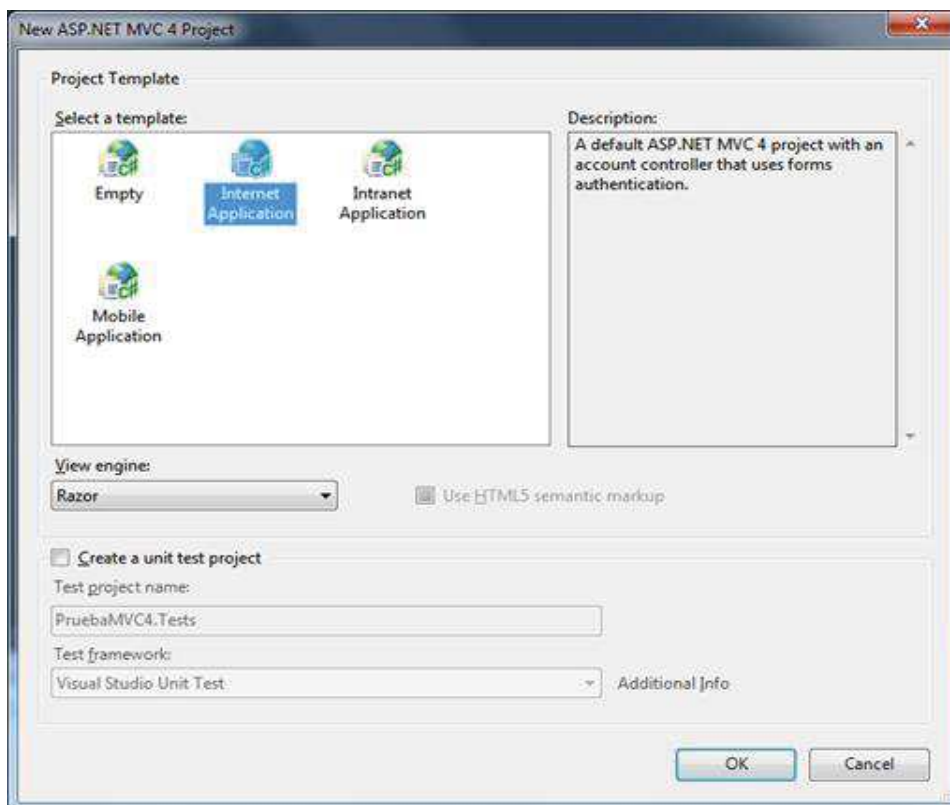
Por supuesto, todo esto tiene un precio: la curva de aprendizaje de ASP.NET MVC puede ser más alta que la de webforms, especialmente si nunca has desarrollado para web. A diferencia de webforms, que te abstrae de HTTP y HTML, ASP.NET MVC está "mucho más cerca de la web", lo que hace necesario conocer HTTP, HTML y Javascript para trabajar con él. De todos modos eso no debería echarte para atrás: si quieres crear aplicaciones web es normal que debas conocer los protocolos y lenguajes en los que se asenta la web.

NOVEDADES MVC 4

Lo nuevo que encontramos al crear un proyecto Web una vez instalado, es la plantilla llamada "ASP.NET MVC 4 Web Application".



Como podemos observar todo está igual hasta aquí: posteriormente se encontrara nuevas novedades.

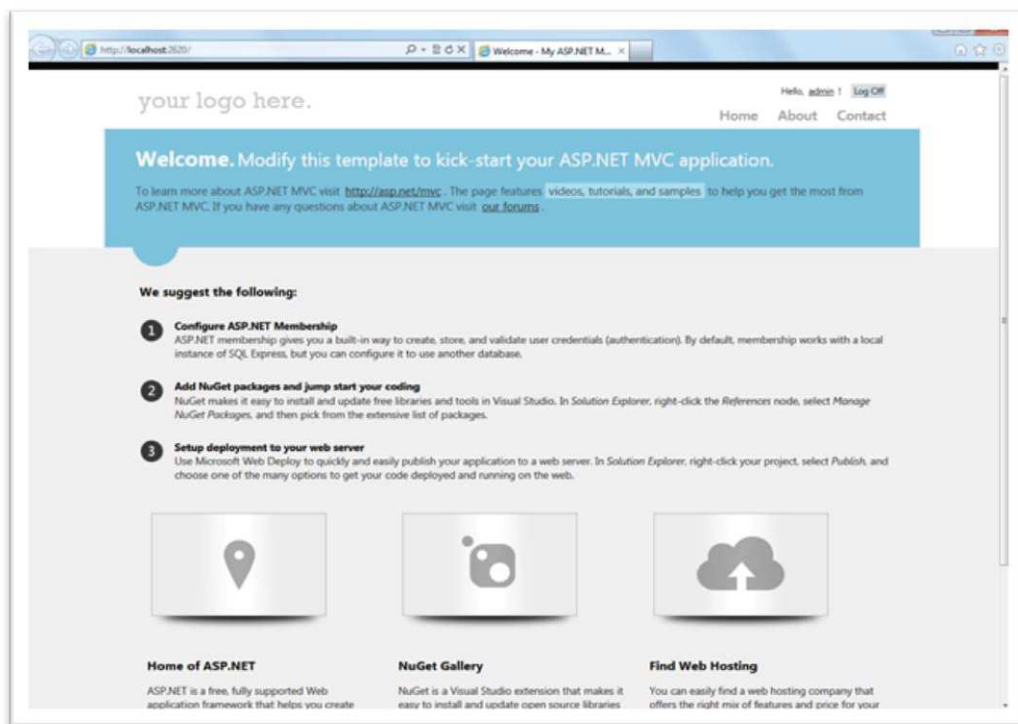


Dentro de las novedades tenemos: el checkbox “Use HTML5 semantic markup” que fue incluido con la última actualización de MVC aparece deshabilitado en todas las plantillas, excepto la vacía. Esto da a pensar que HTML5 ya no es

una opción, es la opción por defecto; se tendrá que esperar para ver si la versión final incluye plantillas no-HTML5, o si se continúa con esta tendencia que han adoptado.

Otra novedad es en las habituales plantillas (Vacía, Aplicación Internet y Aplicación Intranet) se puede observar la nueva plantilla “Mobile Application”. Con esto podemos afirmar que las aplicaciones web para móviles están tomando bastante protagonismo.

Al crear un nuevo proyecto de tipo Internet Application, observamos una plantilla moderna, elegante. Sin duda el cambio estético para mi punto de vista es muy llamativo.

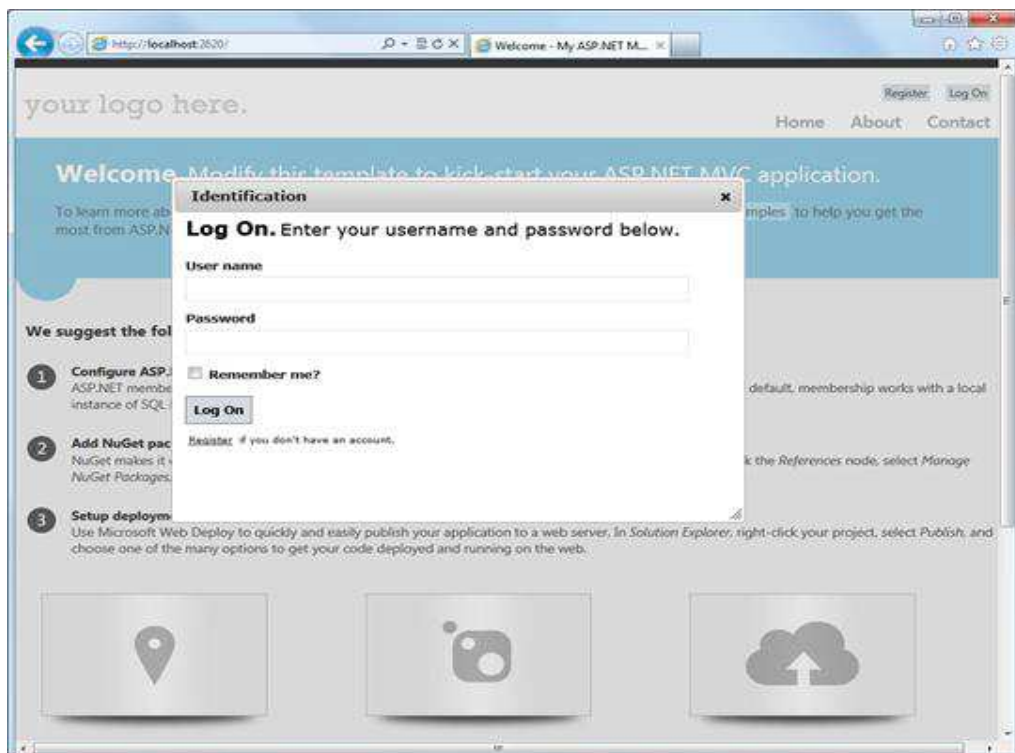


Dentro de la maquetación se utiliza una técnica llamada “renderizado adaptativo”, que no es más el adaptar a distintos anchos la pantalla; por lo que no se tiene ningún inconveniente en visualizar correctamente en dispositivos grandes (captura de pantalla de arriba) como en móviles. Si deseamos comprobar simplemente debemos cambiar el tamaño de la ventana de nuestro navegador.

Lo señalado anteriormente no tiene nada que ver con MVC 4, pero si se deja en manifiesto una clara tendencia del producto hacia los dispositivos móviles.

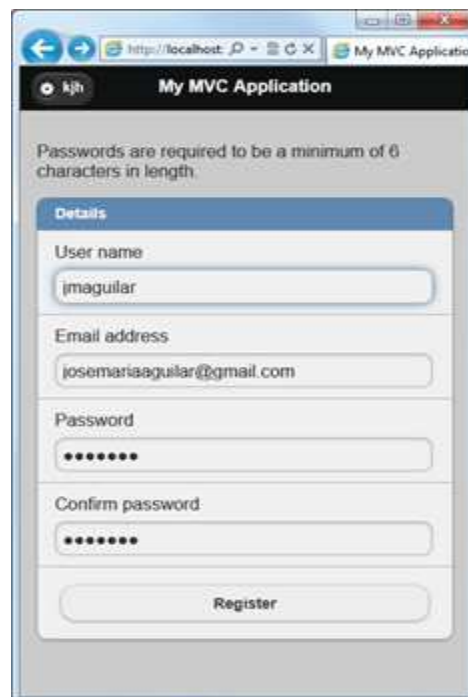


Un cambio radical se observa en el sistema de membresía que habitualmente venía con las plantillas MVC tradicionales, hoy en día si como usuario se tiene activado el scripting (lo que es habitual), las funcionalidades de acceso y registro de usuarios se realizan utilizando cuadros de diálogo basados en jQuery UI, y llamadas Ajax. Si el caso fuere, que no sea posible de utilizar scripts, estas funcionalidades se realizarán a pantalla completa, como se lo ha hecho hasta ahora.



Si hablamos de dispositivos móviles, la pantalla específica, utiliza JQuery Mobile, una popular biblioteca para construir interfaces web específicas para dispositivos táctiles como tablets y smartphones.

Como funcionalidad podemos decir que es idéntica a la plantilla de aplicaciones para Internet, pero mucho más optimizada para la interacción desde estos dispositivos.



“Display Modes

Al no parecer demasiado natural el tener dos plantillas de proyecto independiente, MVC 4 incluye una nueva característica llamada “display modes” (modos de visualización).

Esta característica permite crear una única aplicación y renderizar una u otra vista en función del dispositivo o las condiciones que deseemos.

Por ejemplo, si la acción “Index” del controlador “Home” retorna la vista por defecto, lo habitual es que el framework devuelva al usuario el resultado de procesar la plantilla /Views/Home/Index.cshtml. Pues bien, gracias a los display

modes, si la petición ha sido realizada desde un dispositivo móvil, se retornará la vista `Index.mobile.cshtml` y sólo en caso de no existir, la habitual `Index.cshtml`. Y lo mismo ocurre con los layouts y vistas parciales: se seleccionarán en función del cliente.

También se permite un control más granular sobre los dispositivos. El siguiente código (tomado de las notas de la revisión), registra un nuevo display mode en el sistema. Básicamente, lo que hacemos es asociar el nombre "iPhone" al dispositivo que ha realizado una petición en cuyo user-agent aparece ese mismo texto:

```
DisplayModes.Modes.Insert(0, new DefaultDisplayMode("iPhone"))
{
    ContextCondition = (context => context.Request.UserAgent.IndexOf
        ("iPhone", StringComparison.OrdinalIgnoreCase) >= 0)
});
```

Una vez hecho esto, y siguiendo con el ejemplo anterior, la vista que se intentaría renderizar en este caso sería `Index.iPhone.cshtml`. Qué bueno, eh?

Otra capacidad muy interesante que vamos a tener a nuestra disposición es el browser overriding, o lo que es lo mismo, el poder establecer el navegador del usuario independientemente del utilizado por el mismo. Esto permitirá, por ejemplo, forzar la visualización en modo completo a alguien que está accediendo desde un dispositivo móvil, o justo lo contrario, forzar la visualización del modo móvil a alguien accediendo desde desktop.

El agente de usuario sobrescrito será persistente entre las sucesivas peticiones puesto que se almacenarán por defecto en una cookie, aunque, como es habitual, será posible utilizar otros mecanismos de persistencia personalizados. Para facilitar el desarrollo de este tipo de sistemas, se ha puesto a disposición de los desarrolladores el paquete `jQuery.Mobile.MVC` (descargable a través de Nuget con esta denominación), que incluye un Layout para móviles, un controlador, algunos helpers y una útil vista parcial llamada `ViewSwitcher` que permite conmutar entre el modo móvil y modo desktop.

Bibliotecas de scripting

Hasta MVC 3, en todas las plantillas de proyectos incluían las bibliotecas de scripting de Microsoft relativas a Ajax y los sistemas de validación. Ya en la última versión del framework era más recomendable utilizar las soluciones basadas en jQuery, pero estas bibliotecas seguían pululando por ahí; en ASP.NET MVC 4, simplemente las bibliotecas javascript de Microsoft han dejado de distribuirse en las plantillas de proyectos MVC.

Y como sorpresa, en la carpeta /Scripts, además de los tradicionales Modernizr, jQuery, jQueryUI, jQuery validate, y jQuery unobtrusive, nos encontramos ahora con Knockout, una biblioteca javascript que permite crear interfaces de usuario dinámicos utilizando el patrón MVVM.

Claramente se intenta poner un poco de orden en ese caos en lo que a veces se convierten las vistas cuando son muy dinámicas y utilizan mucho scripting.

Microsoft.Web.Optimizations

Curioseando por las referencias que se añaden a los nuevos proyectos, muchas de ellas en forma de paquetes Nuget preinstalados, me encuentro con Microsoft.Web.Optimizations, una joya que ya deja ver cómo va a resolverse la promesa hecha en el roadmap sobre la compactación y minimización de archivos javascript y estilos.

Se trata de un sistema mediante el que podemos crear paquetes (bundles) de archivos .js o .css y asignarles una URL única. Cuando se realice una petición a esta dirección, se compactarán y minimizarán los archivos originales y se retornará al usuario la copia reducida. Obviamente esto sólo se producirá una vez, puesto que el resultado quedará almacenado en caché.

Por ejemplo, podemos asociar la URL “/myscripts.js” a todos los scripts presentes en la carpeta “~/scripts” del proyecto incluyendo subdirectorios, con lo que podremos descargar todo el javascript que necesita una página

simplemente referenciando desde ella al archivo `myscripts.js`. E igual ocurre con las hojas de estilo.

No es nada que no pudiera hacerse antes incluyendo los procesos de compactación en compilación, pero la verdad es que este nuevo sistema está muy bien, y facilita la creación de

En el caso de los scripts, la compactación no consiste únicamente en reducir los comentarios, espacios y saltos de línea, sino que también son transformados los nombres de las variables, métodos y funciones privadas para reducirlos a su mínima expresión. O sea, que de paso estamos ofuscando el código.

Y lo que es mejor, al tratarse de un paquete independiente que no tiene dependencias con MVC 4 ni con ASP.NET 4.5, podemos usarlo ya en nuestras aplicaciones MVC 3, simplemente descargando el paquete desde Nuget y poniéndolo en marcha. La forma de usarlo, para no extenderme mucho aquí, la dejaré para otro post, aunque ya os adelanto que es realmente sencillo.

Más características incluidas

Durante las pruebas, me he encontrado en las vistas con nuevos helpers como los imprescindibles `Html.IdFor()` y `Html.NameFor()` que ya vimos por aquí hace tiempo, y que son útiles para averiguar respectivamente el id y name que serán asignados al campo de edición de una propiedad en un formulario. Viene también con sus versiones no tipadas, nombrados como `Id()` y `Name()`, y las versiones para el modelo completo `IdForModel()` y `NameForModel()`.

También encontramos un interesante

helper `Html.Value()`, `ValueFor()` y `ValueForModel()` que permite obtener el valor para una propiedad siguiendo la lógica de los habituales helpers de edición, teniendo en cuenta tanto el `ModelState` como los datos de la vista.

Por último, aunque no las he comprobado, en el documento de Release Notes se destacan también los siguientes aspectos: simplificación de controladores asíncronos, utilizando C# 5 y los tipos Task y Task<ActionResult>.

- soporte para la release 1.5 del SDK de Azure.”

Tomado de (planetacodigo.com, 2011)

ANEXO 2

Requisitos

Características de Hardware

Mínimos Requerimientos:

- Procesador de 32 bits (x86) o 64 bits (x64) a 1 gigahercio (GHz) o más.
- Memoria RAM de 1 gigabyte (GB) (32 bits) o memoria RAM de 2 GB (64 bits).
- Espacio disponible en disco rígido de 16 GB (32 bits) o 20 GB (64 bits)

Características de Software

Sistemas Operativos Soportado:

Windows Server 2003 Standard Edition, Windows Server 2003 Enterprise Edition, Windows Server 2003 Small Business Edition, Windows Server 2008 Standard, Windows Server 2008 Enterprise Edition.

Base de Datos:

Microsoft SQL Server 2008 Standart Edition, Microsoft SQL 2008 Enterprise Edition.

Programas:

Microsoft Framework 3.5

Microsoft Framework 3.5 Service Pack 1

Internet Information Services (IIS) 6.0 o superior

Restaurando Base de Datos badalex1

Una vez verificado que todos los programas se encuentran instalados procedemos a restaurar la base de datos.

Abrimos SQL Server, nos conectamos al motor de base de datos.

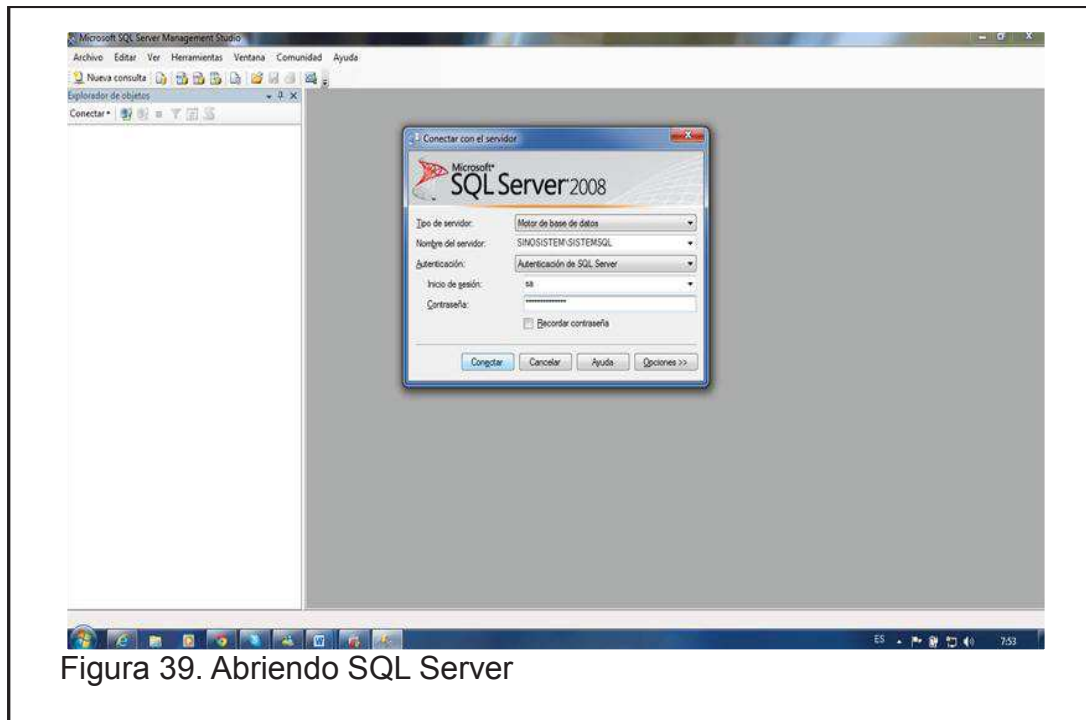


Figura 39. Abriendo SQL Server

Dentro del ambiente del SQL Server procedemos con la restauración del respaldo de la base de datos.

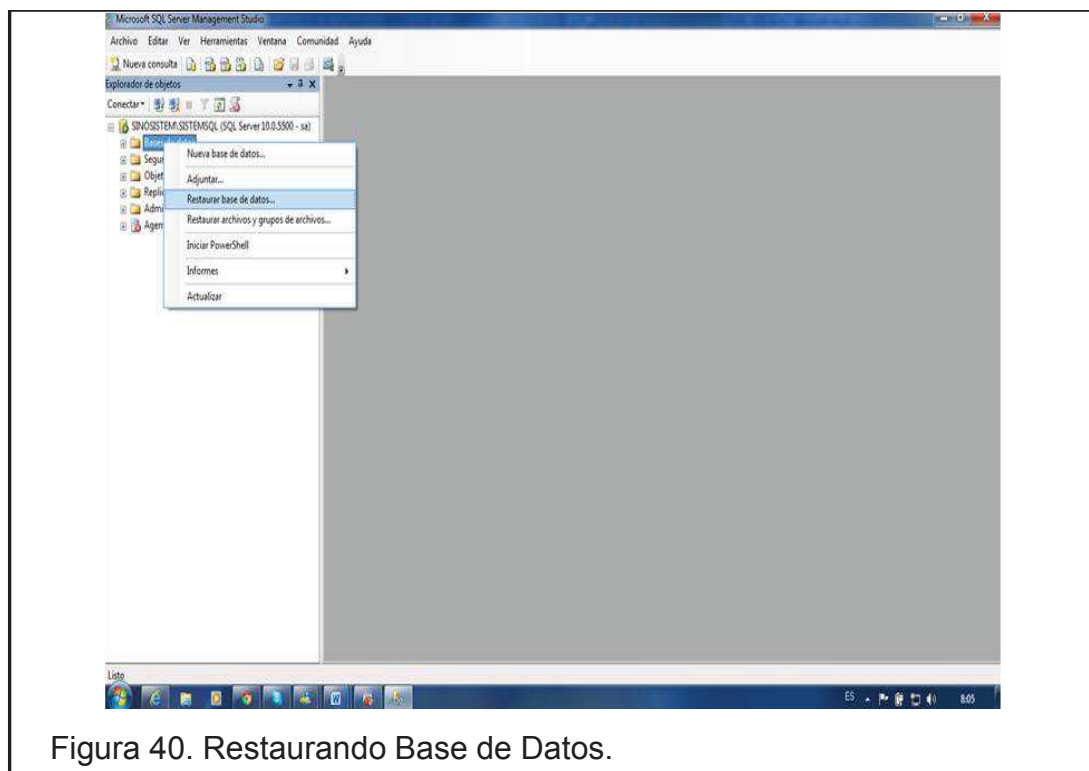


Figura 40. Restaurando Base de Datos.

Siguiendo los pasos de la restauración buscamos la ubicación del respaldo de la base de datos y completamos la misma.

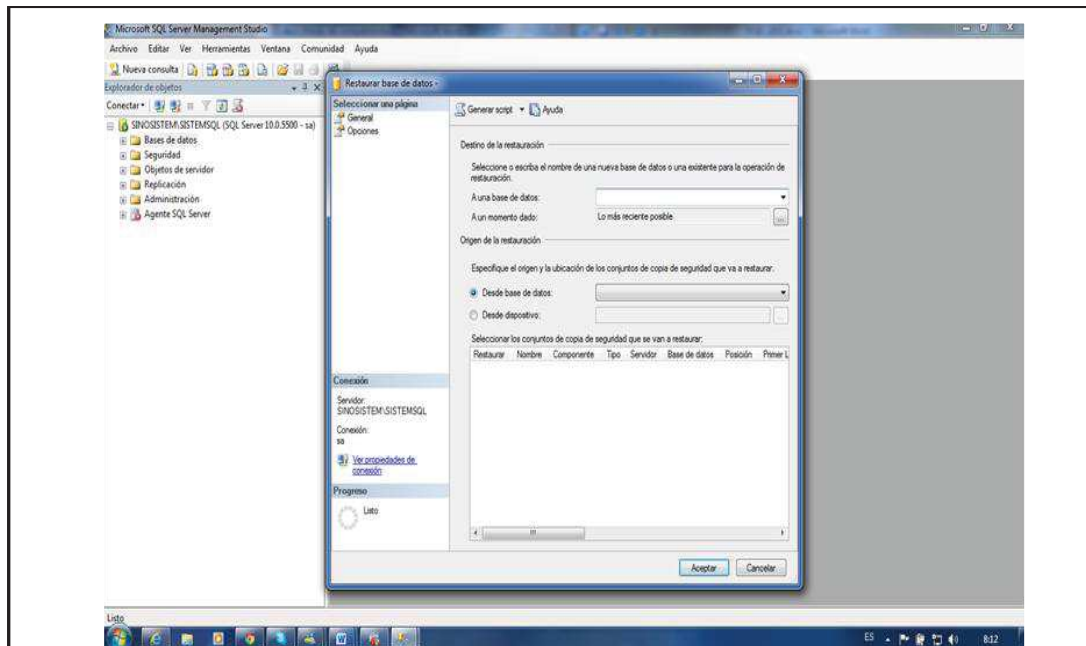


Figura 41. Buscando Base de Datos badalex1.

Una vez finalizada la restauración, la base de datos (badalex1) estaría lista para ser utilizada en la aplicación.

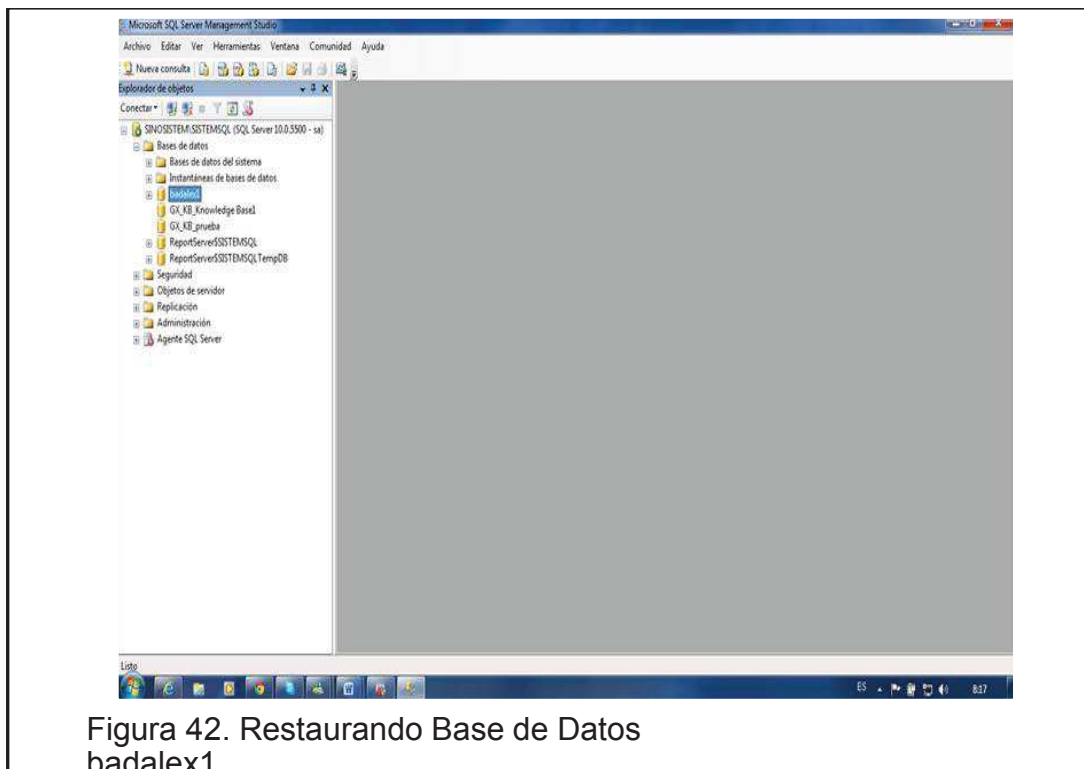


Figura 42. Restaurando Base de Datos badalex1.

Es importante indicar que, por tratarse de una base de datos confidencial no es posible dejar un respaldo de la misma adjunto al presente proyecto.

ANEXO 3

Aplicación

Manual de Usuario

SwCITAC

Ingresamos la dirección Url en el navegador, este nos desplegara una pantalla en la cual ingresaremos nuestro usuario y contraseña.

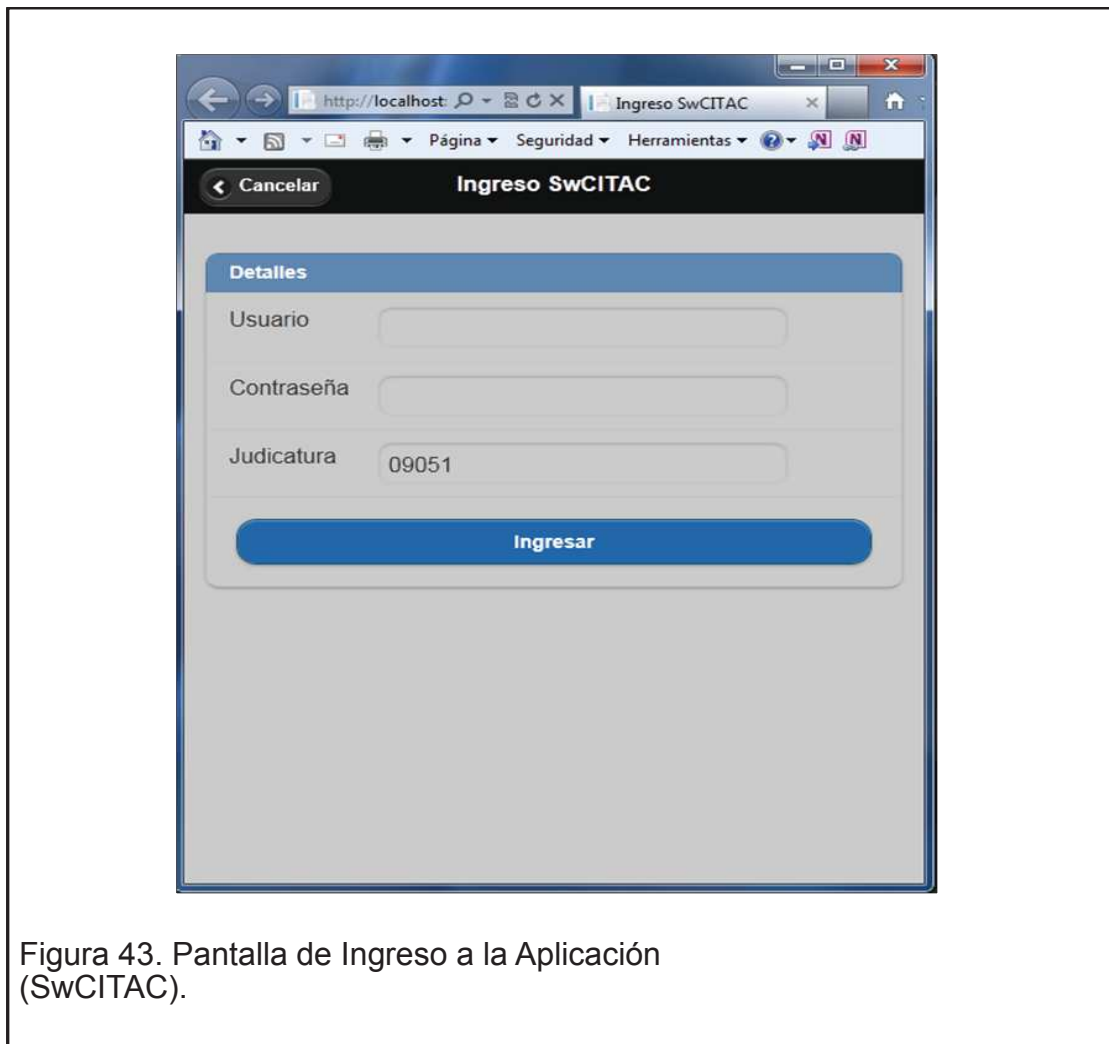
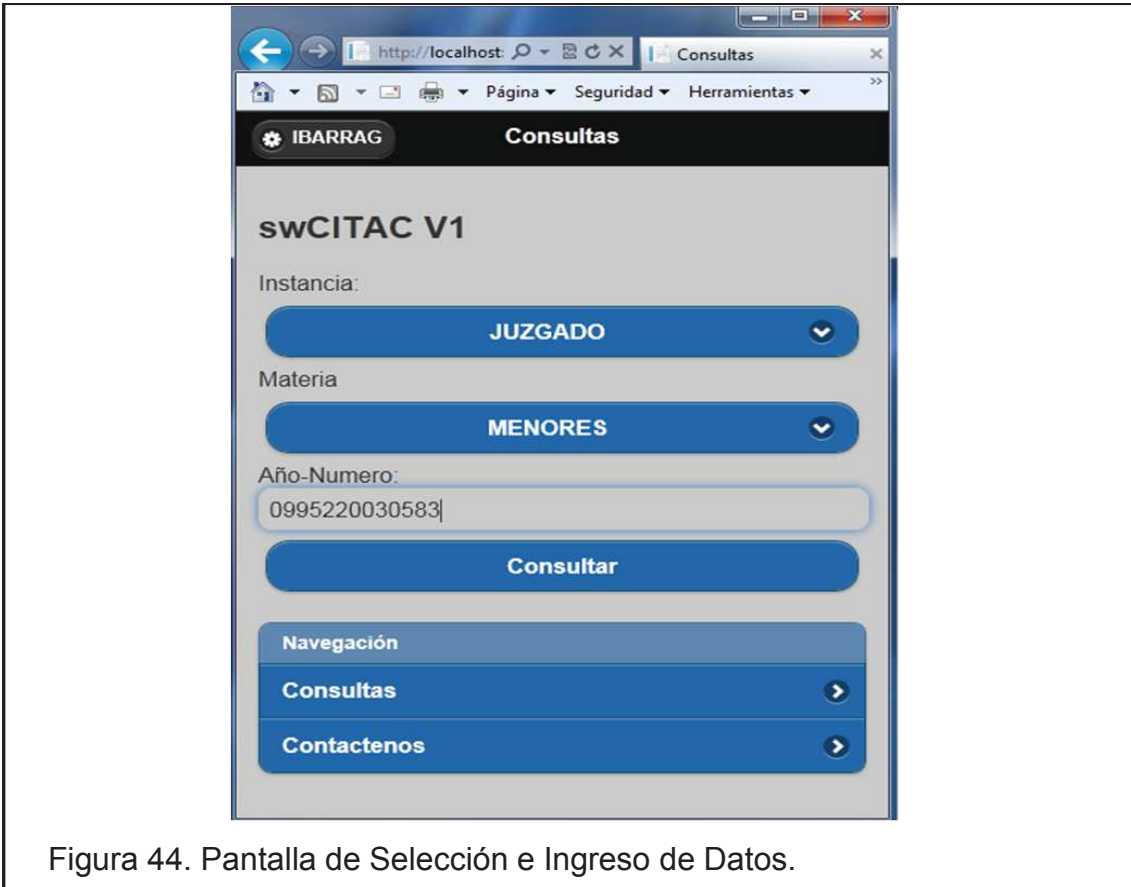


Figura 43. Pantalla de Ingreso a la Aplicación (SwCITAC).

Dentro del sistema en la parte superior izquierda se visualizara el usuario con el que estamos trabajando, escogemos la instancia, la materia e ingresamos el número completo del proceso Ejemplo "0995220030583", seguidamente presionamos el botón consultar.



A continuación se visualizara el proceso con su tipo de causa, el delito para continuar se debe presionar el botón “Citar a: Litigantes”.

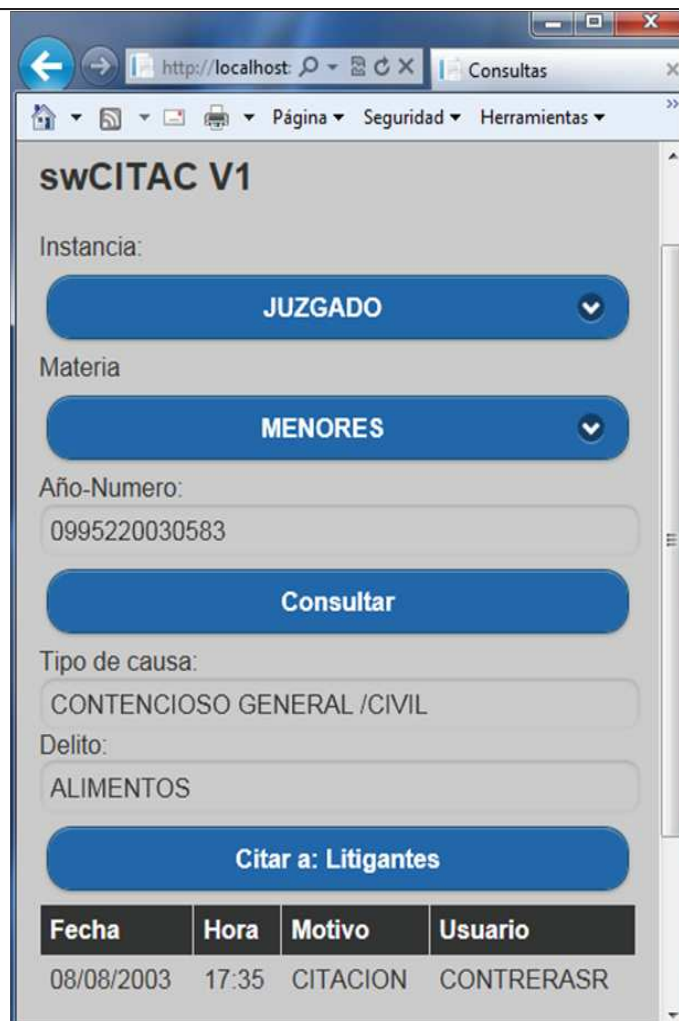
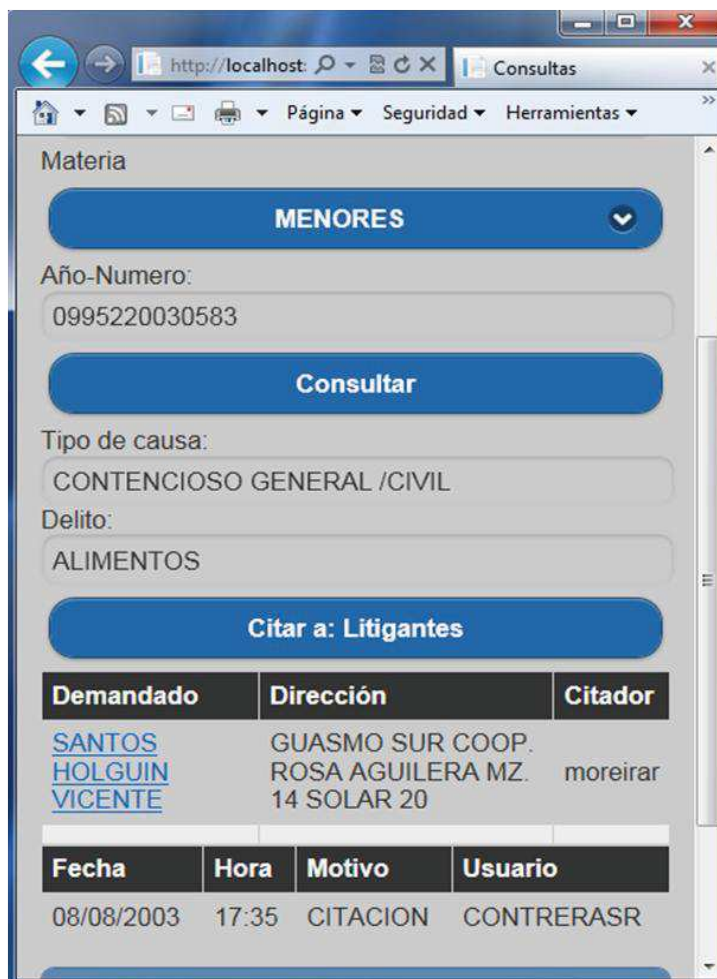


Figura 45. Pantalla de Visualización Datos del Proceso.

Se observa a continuación el o los demandados, nos posicionamos en el nombre del demandado que queremos citar y damos click.



Materia

MENORES

Año-Numero:

0995220030583

Consultar

Tipo de causa:

CONTENCIOSO GENERAL /CIVIL

Delito:

ALIMENTOS

Citar a: Litigantes

Demandado	Dirección	Citador
SANTOS HOLGUIN VICENTE	GUASMO SUR COOP. ROSA AGUILERA MZ. 14 SOLAR 20	moreirar

Fecha	Hora	Motivo	Usuario
08/08/2003	17:35	CITACION	CONTRERASR

Figura 46. Pantalla de visualización de Litigantes.

En este punto ya se puede visualizar los datos del proceso así también se permite escoger el tipo de citación que se va a realizar.

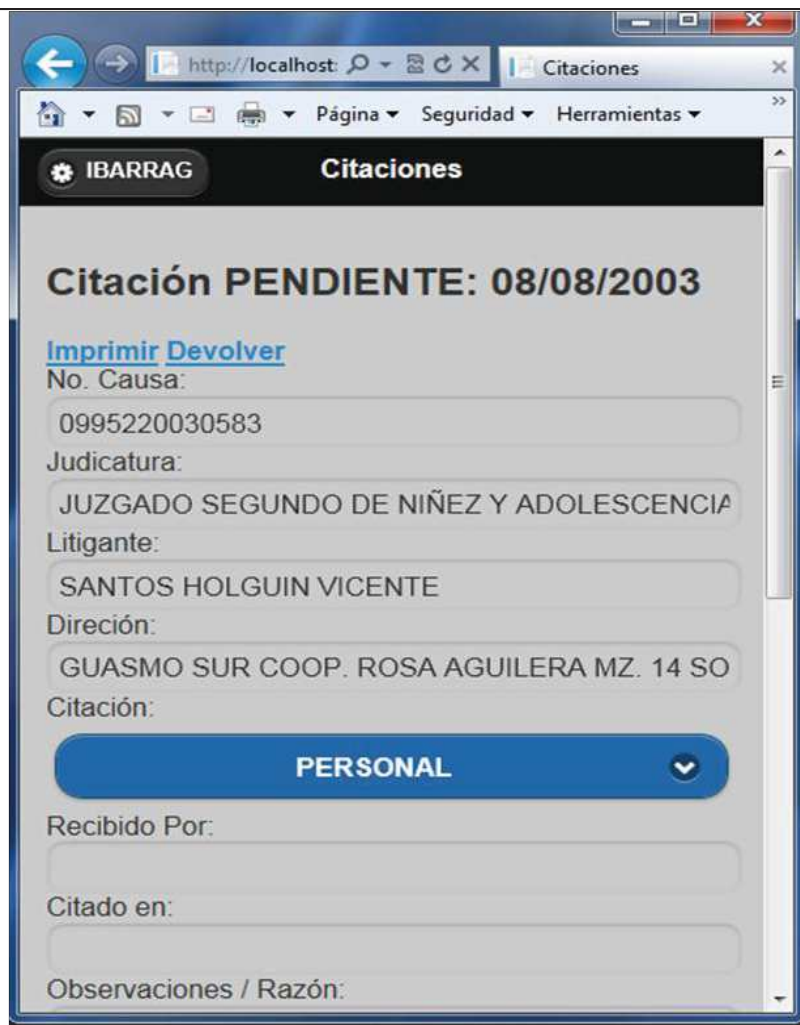


Figura 47. Pantalla de Trabajo-1 .

En la misma pantalla bajando con el scroll de navegación podemos visualizar los campos a llenar tal como se tienen en la hoja de trabajo diaria, una vez concluido con el ingreso de datos se tiene que dar clic en el botón "Guardar". Si en el proceso ya se ha realizado citaciones, se podrá visualizar las mismas, también se tiene la opción de eliminar la citación en el instante si por algún motivo se ingresó algún dato erróneo.

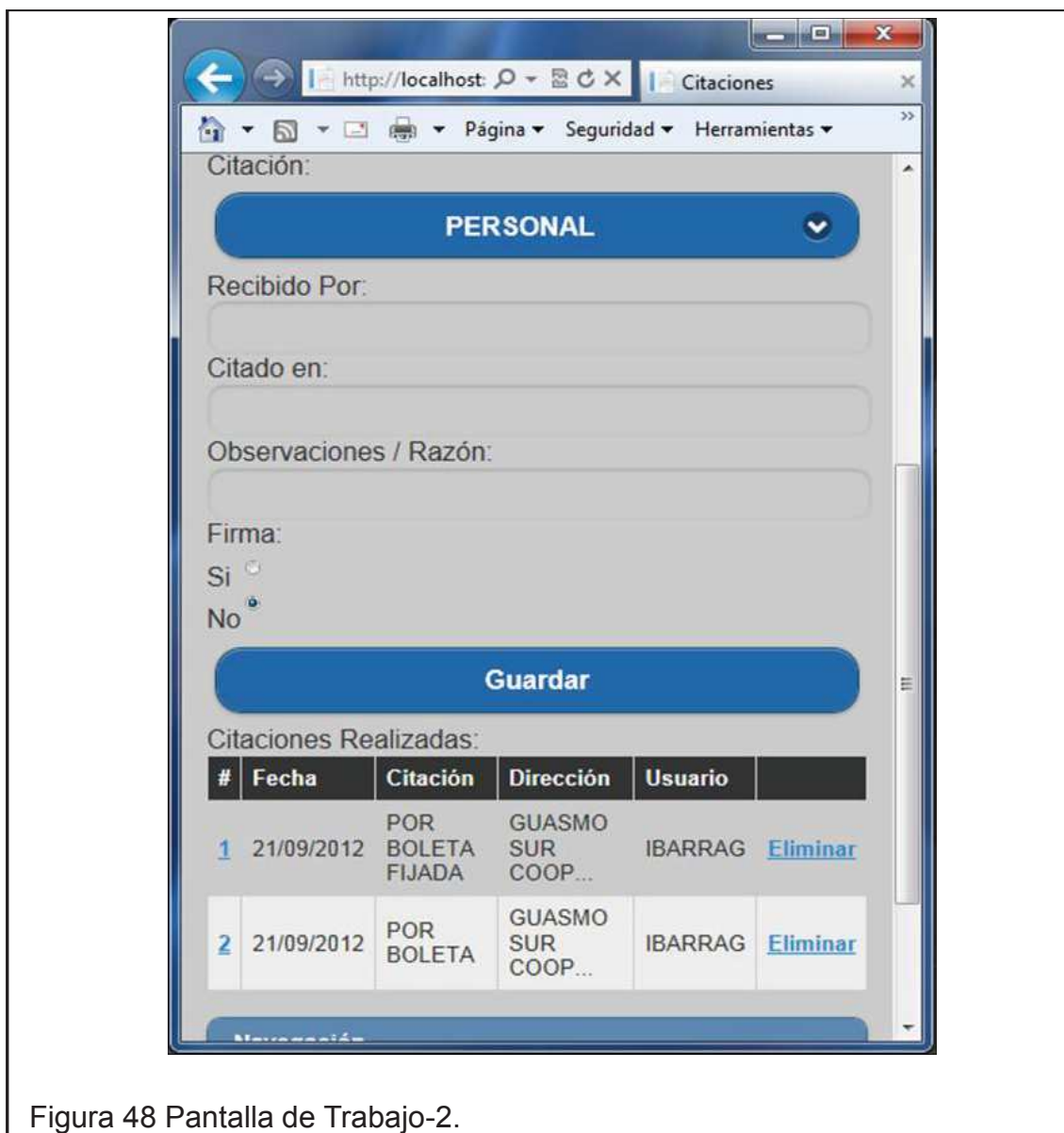


Figura 48 Pantalla de Trabajo-2.

Una vez concluido con el número máximo de citaciones permitidas por la ley, es necesario para que el proceso termine presionar el botón “Devolver”, con esto estaría listo para que el proceso sea devuelto físicamente al Juzgado correspondiente.

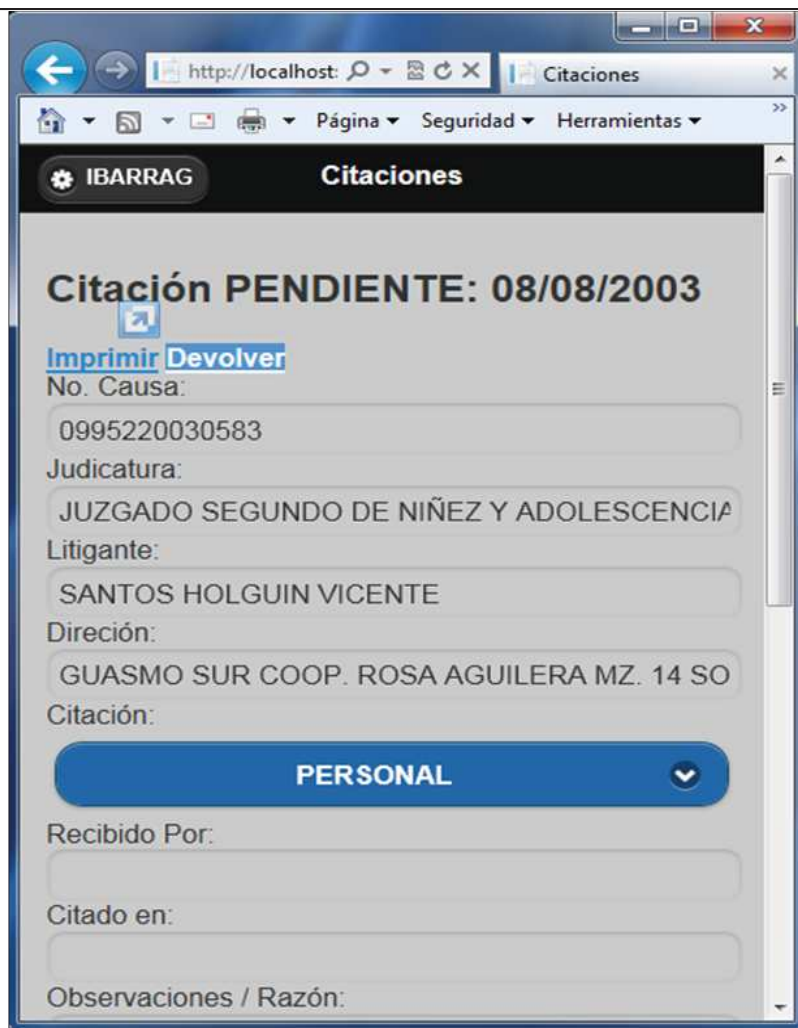


Figura 49. Pantalla de Trabajo-3.

ANEXO 4

IEEE-STD-830-1998: ESPECIFICACIONES DE REQUISITOS DEL SOFTWARE

1.1 Cliente:

Dirección Provincial del Consejo de la Judicatura Distrito Guayas

1.2 Elaborado por

Gonzalo Fernando Ibarra Cordovilla

1.3 Usuarios del Sistema a Diseñar

Personal administrativo del departamento de Citaciones.

2. Consideraciones

2.1 Naturaleza del SRS

a) La Funcionalidad

El sistema tendrá la función de registrar las citaciones que realizan diariamente los Usuarios Judiciales, alimentando en tiempo real la base de datos para que las autoridades o los propios Juzgados mediante la aplicación ya existente SATJE verifiquen la realización de la misma.

c) Velocidad, Disponibilidad, Tiempo de la contestación, tiempo de la recuperación

Tendrá un tiempo de registro inmediato, para que los usuarios puedan acceder a la información o puedan registrar los datos en el sistema.

b) Los Atributos.

Su Base de Datos está fue realizada en primera instancia en SQL 2000, luego fue migrada a SQL 2008 versión en la cual se mantiene hasta la presente fecha, la aplicación utiliza Visual Studio 2010, ASP .NET MVC4 en C#.

La disponibilidad será permanente mientras el servidor este en producción.

c) Las restricciones del diseño impuestas en la aplicación.

No tiene restricciones.

El sistema requiere ser instalado en un servidor que pueda ejecutar sentencias de asp.net mvc, pueda alojar una instancia de servidor de SQL 2008, instalar Framework 4.0, instalar Internet Information Services.

2.2 Ambiente del SRS

El sistema abarca y está desarrollado a los Usuarios del departamento de Citaciones de la Función Judicial Distrito Guayas, para que puedan realizar su trabajo diario a través de un dispositivo móvil y puedan de esta manera registrar la citación en el momento mismo de la entrega.

2.3 Diseño del SRS

2.3.1 Requisitos del plan necesarios

a) El sistema contara con el loguin para que los Usuarios puedan acceder al sistema.

b) Dentro del sistema el Usuario podrá registrar la respectiva citación en el momento mismo de la entrega.

3. Partes del SRS

Tabla de Contenidos

1. Introducción

1.1 Propósito

1.2 Alcance

1.3 Definiciones, siglas, y abreviaciones

1.4 Referencias

1.5 Apreciación global

2. Descripción global

2.1 Perspectiva del producto

2.2 Funciones del producto

2.3 Características del usuario

2.4 Restricciones

3. Los requisitos específicos

3.1 Introducción (Sección 1 del SRS)

La Dirección Provincial del Consejo de la Judicatura distrito Guayas es una institución no financiera pertenece al sector social del gobierno central, la misma que está regulada, por leyes, reglamentos, normas y procedimientos legales; las cuales son emitidas por los organismos rectores (Contraloría General del estado, Consejo de la Judicatura).

En el departamento de Citaciones cuenta con 9 personas que son las encargadas de realizar el trabajo diario.

En la actualidad este proceso se lo realiza en forma manual, en el mejor de los casos al finalizar la jornada de trabajo regresan a la institución a ingresar las citaciones que realizaron; o si no pueden retornar o ingresar lo hacen al siguiente día corriendo el riesgo de que las hojas de trabajo de extravíen perjudicando así al proceso judicial.

3.1.1 Propósito (del SRS)

Esta subdivisión debe:

a) Propósito del SRS

Se va a realizar la especificación de requerimientos con el fin de tener un documento en cual basarse para el análisis del diseño del sistema, con el propósito de automatizar los procesos para los Usuarios Citadores mejorando los tiempos en los procesos informáticos del Departamento.

3.1.2 Alcance del SRS

Dentro de la definición del alcance del sistema están considerado lo siguiente:

a) Identificación del producto de software

SISTEMA PARA EL REGISTRO DE CITACIONES VIA WEB ATRAVES DE TELEFONOS MOVILES.

b) Características principales

- El sistema permitirá al Usuario ingresar con su usuario y clave que actualmente poseen.
- Para consultar un proceso es necesario ingresar todo el número del proceso como esta en el formato de a hoja de trabajo.
- Para facilidad del Usuario el sistema desplegara a continuación la información del proceso, permitiendo al usuario pulsar sobre el litigante a ser citado.

c) Beneficios pertinentes, Objetivos, y Metas

- El sistema permitirá registrar las citaciones realizadas.
- El sistema permitirá modificar la citación si se realizó algún error involuntario.
- El sistema permitirá eliminar la citación si es necesario realizar.
- El sistema valida las citaciones permitidas por usuario de acuerdo a la ley que ampara a las citaciones.
- Una vez finalizado el proceso de citación el sistema permite devolver lógicamente el proceso al Juzgado correspondiente.

3.1.3 Apreciación global

Esta subdivisión debe:

a) Contenido del SRS

En forma general el sistema será el encargado de registrar las citaciones realizadas por los Usuarios Judiciales (Citadores).

3.2 Descripción Global

3.2.1 Perspectiva del Producto

Para que el sistema funcione es necesario que los procesos sean ingresados como se lo viene haciendo con el sistema SATJE, ya que la necesidad por el cual fue creado este sistema es para poder registrar las citaciones en el momento que se entrega la misma.

3.2.2 Funciones del Producto

El sistema deberá cumplir con las siguientes funciones detalladas a continuación.

- El sistema permitirá buscar el proceso.
- El sistema permitirá escoger la judicatura, materia según sea el proceso.
- El sistema permitirá ingresar el número completo del proceso tal como se encuentra en la hoja de trabajo.
- El sistema permitirá escoger al litigante que va a ser citado.
- El sistema permitirá ingresar la información de acuerdo a la hoja de trabajo.
- El sistema validará el número de veces permitido para realizar las citaciones por litigante (Usuario a ser citado).
- El sistema permitirá guardar la información de la citación en el momento mismo que se entrega la citación.
- Una vez terminado el proceso el sistema permitirá devolver el juicio al juzgado de procedencia.

3.2.3 Características del usuario

Para poder manejar el sistema el usuario deberá tener como mínimo las siguientes características.

- Deberá tener conocimientos de manejo del celular para poder ingresar al navegador.
- Deberá conocer y manejar de forma básica algún navegador de Internet como Internet Explorer, Mozilla FireFox, Chrome, etc.

3.2.4 Restricciones

El sistema no funciona si no se tienen disponible navegación o paquete de datos en el celular.

3.2.5 Prorratear los Requisitos (Del SRS)

A continuación una lista de requisitos para la implementación de la aplicación

Requisitos del Servidor

- (IIS) Internet Información Server 6.0 o superior
- .Net Framework 4.0
- Tener restaurado la base de datos en Microsoft Sql Server 2008 Express o Standar

Requisitos del Cliente

- Conexión a navegación a través del celular.
- Internet Explorer 6.0 o superior.

3.3 Requisitos específicos

Esta sección del SRS contiene todos los requisitos del software a un nivel de detalle suficiente para permitirles a los diseñadores diseñar el sistema, a lo largo de esta sección, cada requisito declarado esta externamente perceptible por los usuarios y operadores del sistema. Estos requisitos deben contener una descripción de cada entrada en el sistema, cada salida del sistema, y todas las funciones realizadas por el sistema en la salida a una entrada o en el apoyo de la salida.

3.3.1 Interfaces Externas

El sistema presentara mensajes de existos, o errores al procesar un resgitro.

El sistema permitira verificar cuantas citasiones tiene cada litigante, para lo cual es necesario ingresar al proceso.

3.3.2 Funciones

A continuación tenemos una lista de las funciones que debe realizar el sistema.

- El sistema permitirá buscar el proceso.
- El sistema permitirá escoger la judicatura, materia según sea el proceso.

- El sistema permitirá ingresar el número completo del proceso tal como se encuentra en la hoja de trabajo.
- El sistema permitirá escoger al litigante que va a ser citado.
- El sistema permitirá ingresar la información de acuerdo a la hoja de trabajo.
- El sistema validara el número de veces permitido para realizar las citaciones por litigante (Usuario a ser citado).
- El sistema permitirá guardar la información de la citación en el momento mismo que se entrega la citación.
- Una vez terminado el proceso el sistema permitirá devolver el juicio al juzgado de procedencia.

3.3.3 Atributos del software del sistema.

El sistema tendrá los siguientes atributos y / o características adicionales mencionadas por el cliente al momento de especificación de requerimientos

3.3.3.1 Fiabilidad

La información debe ser totalmente segura al realizar las operaciones de validacion de las posibilidades que se tiene por cada citacion a un litigante.

3.3.3.2 Disponibilidad

La aplicación debe tener disponibilidad inmediata a solicitar la ejecución un tiempo máximo de máximo 10 segundos, para la recuperación de datos un tiempo máximo de 10 segundos, esto dependera de la disponibilidad del proveedor para su navegacion en el dispositivo movil.

3.3.3.3 Seguridad

La seguridad del sistema tendrá las siguientes características:

- Al acceso a los usuarios será únicamente desde la pantalla de loguin del sistema con su respectivo usuario y clave.

- Todos los usuarios del sistema serán validados en la pantalla de inicio del sistema con la clave registrada para poder proteger el sistema de accesos accidentales o malévolos.

3.3.3.4 Mantenimiento

La arquitectura del sistema hace que el mantenimiento y escalabilidad del sistema sean relativamente fáciles de modificar o desarrollar, ya que si se aumenta un nuevo caso de uso se podrán reutilizar cualquiera de los métodos de la capas del sistema.

3.3.6.5 portabilidad

El sistema podrá ser migrado a otro servidor que tenga los requisitos del servidor de la sección (3.2.5) de este documento.

ANEXO 5

PUBLICACION DEL PROYECTO EN EL IIS

Prerrequisito, tener instalado el IIS.

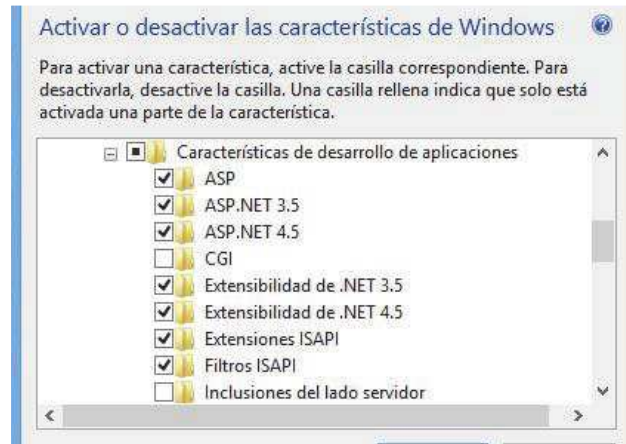


Figura 50. Características de desarrollo de aplicaciones

Dentro de las características de Windows hay que tomar muy en cuenta que las Características de desarrollo de aplicaciones estén habilitadas tal como se muestra en la Figura 50.

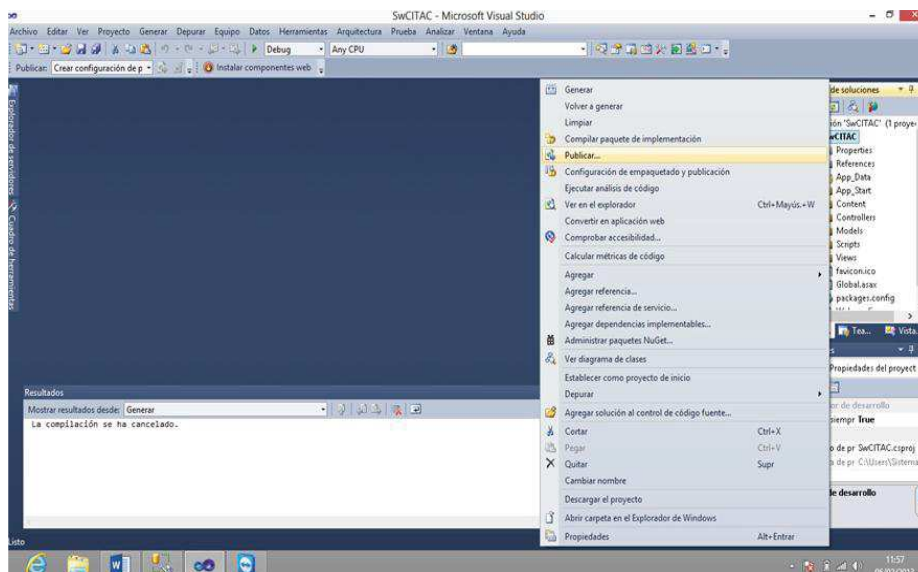
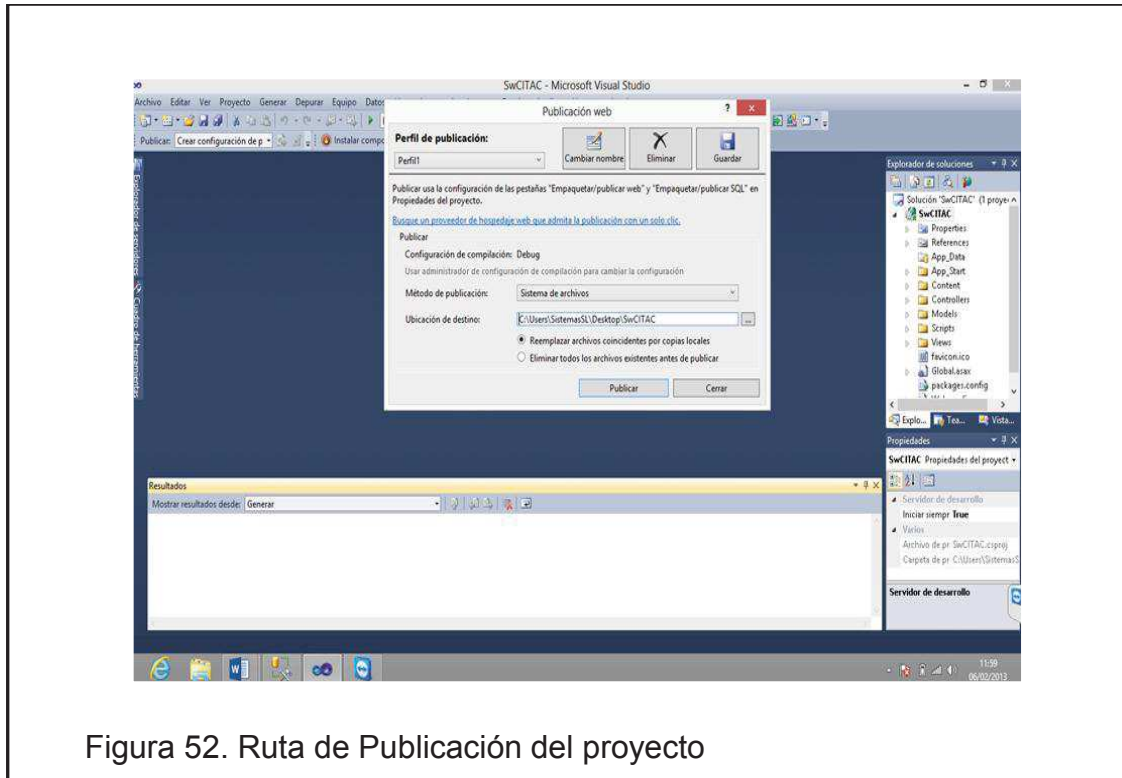


Figura 51. Publicación del proyecto en el IIS

Dentro de la aplicación en nuestro entorno de desarrollo damos clic derecho sobre nuestra aplicación, seleccionando la opción publicar. Tal como se ilustra en la Figura 51.



Seleccionamos el método de publicación que para nuestro caso es “Sistema de archivos”, así también ubicamos el destino de nuestra publicación. En este punto se aconseja crear una carpeta para que los archivos se generen dentro de la misma, para nuestro caso se creó la carpeta SwCITAC. Finalmente se da clic en Publicar tal como se demuestra en la Figura 52.

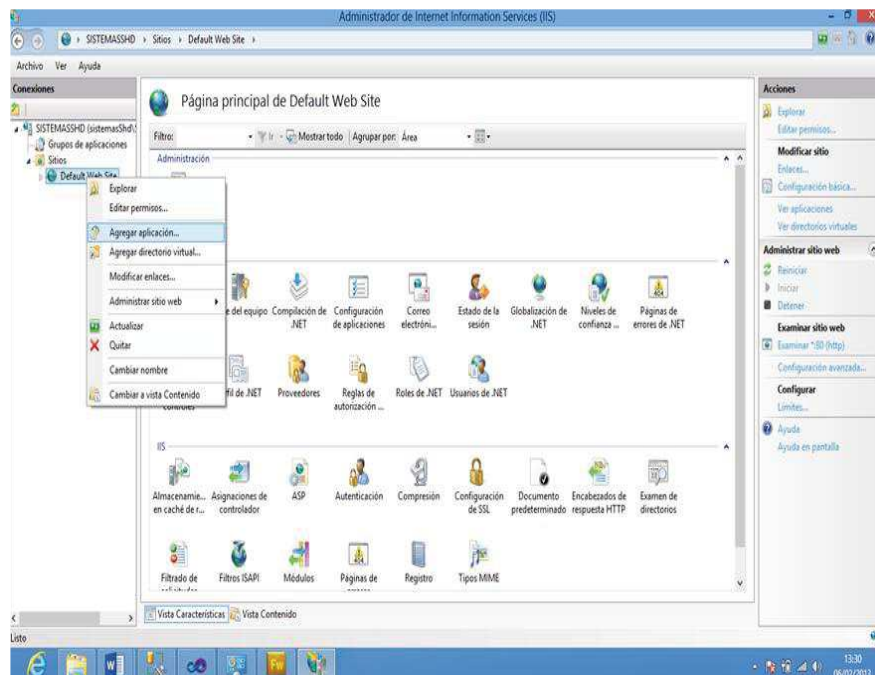


Figura 53. Agregar Aplicación en el IIS

Abrimos el administrador del IIS, damos clic derecho sobre “Default Web Site” seleccionamos la opción “Agregar Aplicación”, tal como se visualiza en la Figura 53.

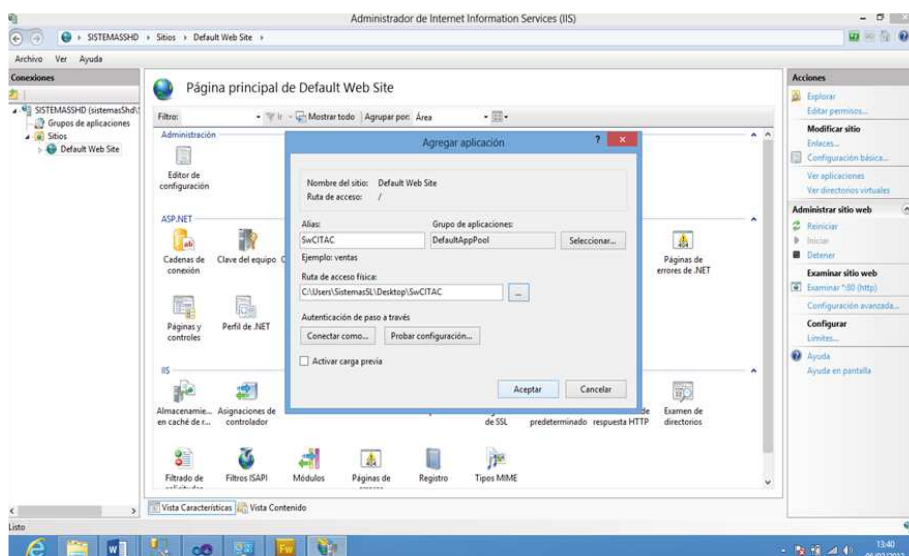
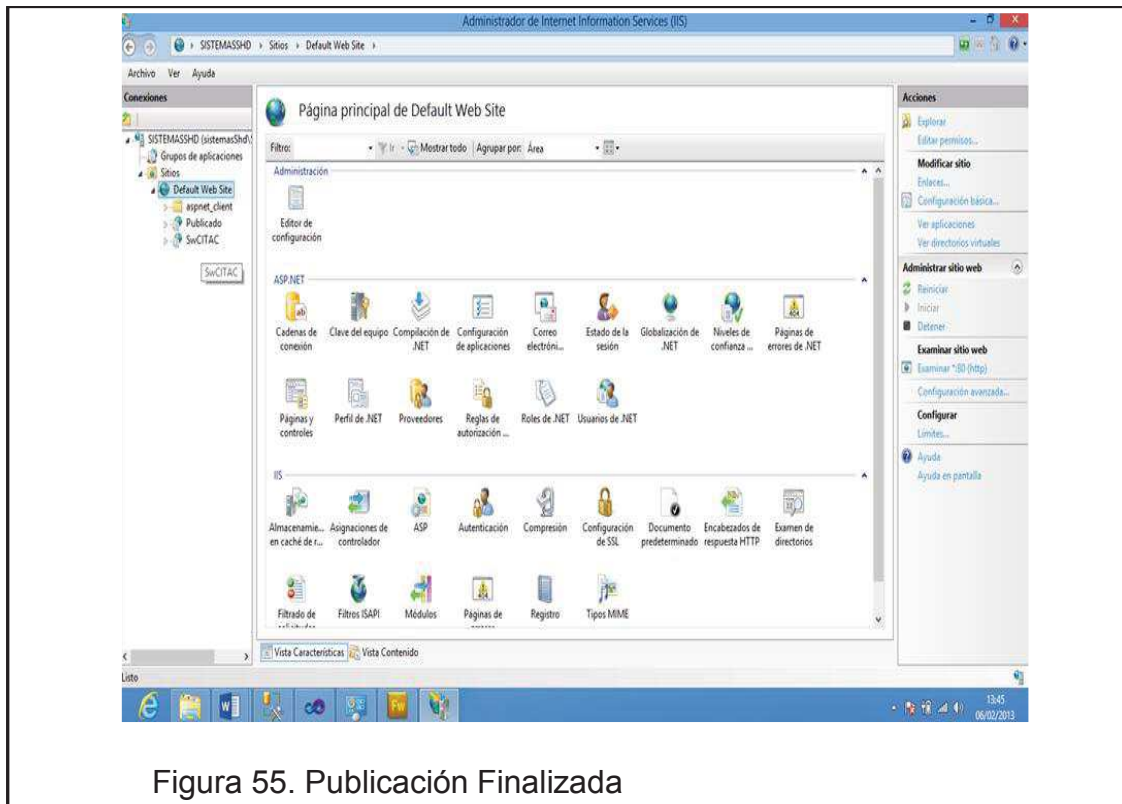


Figura 54. Agregar Aplicación en el IIS

En la pantalla agregar aplicación damos un nombre en la opción “Alias”, en nuestro caso se escogió el nombre “SwCITAC”. Seguidamente se escoge la ruta donde realizamos la publicación, como nuestra publicación se lo hizo en el escritorio de la máquina. Nuestra ruta sería “C:\Users\SistemasSL\Desktop\SwCITAC”, para finalizar damos clic en “Aceptar”, todo esto está ilustrado en la Figura 54.



Ilustrada en la Figura 55, se puede observar en la parte izquierda, que nuestro proyecto se encuentra listo para ser utilizado.

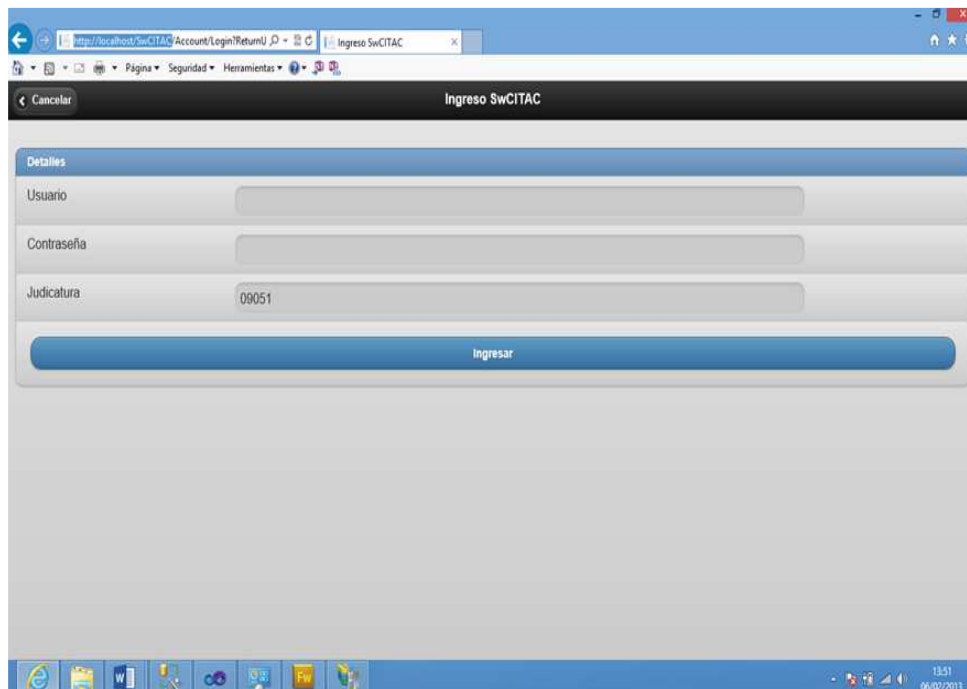


Figura 56. Ejecutando publicación

Para ejecutar la aplicación abrimos nuestro navegador y digitamos, el dominio o nuestra ip pública con la cual salimos al internet seguido del nombre de nuestra aplicación publicada en nuestro IIS. Para nuestro caso tendríamos que digitar http://localhost o ip_publica/SwCITAC, visualizando en la Figura 56 comprobamos que nuestra aplicación esta lista para ser utilizada.

ANEXO 6

SISTEMA PARA EL REGISTRO DE CITACIONES A TRAVES DE TELEFONOS MOVILES CON NAVEGACION WEB (SwCITAC)

ACTA DE LEVANTAMIENTO DE REQUERIMIENTO

PROYECTO:	SwCITAC
RESPONSABLE:	Gonzalo Fernando Ibarra Cordovilla
VERSIÓN:	1.0
LUGAR:	Guayaquil
FECHA:	11 de Agosto de 2012

MÓDULO:	SEGURIDAD
ACTIVIDAD:	AUTENTICACION
<p>1. Se levantan los requerimientos para la autenticación del sistema. Con las siguientes observaciones.</p> <ul style="list-style-type: none">a. El sistema debe poseer una función de logueo el cual permita al usuario ingresar sus datos y estos sean validados en el sistema para permitirle o negarle el ingreso.b. La pantalla inicial debe contener 3 campos de ingreso para el usuario: USUARIO, CONTRASEÑA, y JUDICATURA.c. La información del usuario y su judicatura se encuentra en las tablas Usuarios y UsuarioJudicatura de la base de datos de la empresa.	
COMENTARIOS:	

MÓDULO:	NEGOCIOS	
ACTIVIDAD:	CONSULTA DE LITIGANTES	
<p>1. Se levantan los requerimientos para la consulta de Litigantes. Con las siguientes observaciones.</p> <p>a. El sistema debe poseer una función que permita consultar las citaciones solo y únicamente a los usuarios autenticados mediante varios filtros de consulta.</p> <p>b. La pantalla de consulta contendrá 3 filtros: Instancia, Materia, Año-Numero (Numero proceso judicial completo).</p> <p>c. Al consultar desplegará la información del tipo de causa y el delito, también se desplegará una tabla con las acciones realizadas en la citación.</p>		√
COMENTARIOS:		

MÓDULO:	NEGOCIOS	
ACTIVIDAD:	REGISTRO DE CITACION	
<p>1. Se levantan los requerimientos para la entrega de citaciones. Con las siguientes observaciones.</p> <p>a. El sistema debe poseer una función que permita registrar la entrega de citación en el sistema.</p> <p>b. La pantalla de registro de citaciones debe desplegar el Número de Causa, Judicatura, el Litigante, la dirección.</p> <p>c. Debe existir la opción de escoger el tipo de citación que se va a realizar y registrar quien recibió, donde fue citado, las observaciones, y si firma o no.</p> <p>d. El sistema debe contener: un botón de guardar, que permite registrar la citación en la base de datos.</p>		√
COMENTARIOS:		

MÓDULO:	NEGOCIOS	
ACTIVIDAD:	EDITAR, ELIMINAR REGISTRO DE CITACION	
<p>1. Se levantan los requerimientos para la entrega de citaciones. Con las siguientes observaciones.</p> <ul style="list-style-type: none"> a) El sistema debe poseer una función que permita consultar las citaciones solo y únicamente a los usuarios autenticados mediante varios filtros de consulta. b) La pantalla debe desplegar las citaciones realizadas, permitiendo escoger la citación. c) El sistema debe contener un botón que permita editar la citación. d) El sistema debe contener un botón que permita eliminar la citación. e) El sistema debe contener: un botón de guardar, que permite registrar la citación en la base de datos. 		√
COMENTARIOS:		

MÓDULO:	NEGOCIOS	
ACTIVIDAD:	DEVOLVER CITACION	
<p>1. Se levantan los requerimientos para la entrega de citaciones. Con las siguientes observaciones.</p> <ul style="list-style-type: none"> a) El sistema debe poseer una función que permita consultar las citaciones solo y únicamente a los usuarios autenticados mediante varios filtros de consulta. b) La pantalla debe desplegar las citaciones realizadas, permitiendo escoger la citación. c) El sistema de contener un botón devolver el cual realiza la devolución de citación al juzgado correspondiente una vez concluido el proceso. 		√
COMENTARIOS:		

Para constancia de lo actuado, se firma la presenta acta de levantamiento de requerimientos del módulo de nómina en dos copias de igual valor legal.

Nombre: Gonzalo Fernando Ibarra C. **Nombre:** Personal Operativo