

UNIVERSIDAD CATÓLICA LOS ÁNGELES
CHIMBOTE

**FACULTAD DE INGENIERÍA
ESCUELA PROFESIONAL DE INGENIERÍA DE
SISTEMAS**

**ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN
SISTEMA DE APOYO AL SEGUIMIENTO DE
PROCESOS JUDICIALES EN EL ESTUDIO JURÍDICO
SALAS CALDERON & ASOCIADOS DE LA CIUDAD DE
ICA. 2015**

**TESIS PARA OPTAR EL TÍTULO PROFESIONAL
DE INGENIERO DE SISTEMAS**

AUTOR

MELLENDEZ PEREZ, JORGE DANIEL

ASESOR

PLASENCIA LATOUR, JOSE ELIAS

CHIMBOTE – PERÚ

2016

JURADO EVALUADOR DE TESIS

Dr. José Guillermo Saldaña Tirado
Presidente

Mgter. Oscar Arquímedes Ascón Valdivia
Secretario

Mgter. Andrés David Epifanía Huerta Miembro

DEDICATORIA

Dedico este trabajo principalmente a Dios, por haberme dado la vida y permitirme el haber llegado hasta este momento tan importante de mi formación profesional. A

mi incondicional amiga Ing. Evelyn Sánchez Mejía, por ser el pilar más importante y por demostrarme siempre su cariño y apoyo incondicional sin importar nuestras diferencias de opiniones.

Jorge Daniel

AGRADECIMIENTO

A todas aquellas personas que, de una u otra forma, colaboraron o participaron en la realización de esta investigación, hago extensivo mi más sincero agradecimiento.

Jorge Daniel

RESUMEN

El presente Informe de Tesis titulado “ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE APOYO AL SEGUIMIENTO DE PROCESOS JUDICIALES EN EL ESTUDIO JURÍDICO SALAS CALDERON & ASOCIADOS DE LA CIUDAD DE ICA. 2015”, de tipo descriptivo, de corte transversal, con diseño no experimental, tuvo como objetivo principal el , diseño e implementación de un sistema de apoyo al seguimiento de procesos judiciales en el estudio Jurídico Salas Calderón & Asociados de la ciudad de Ica. El universo está conformado por el estudio Jurídico Salas Calderón & Asociados de la ciudad de Ica, la cual cuenta con 61 trabajadores. Para la recolección de requerimientos funcionales y no funcionales se realizaron entrevistas y encuestas físicas. Asimismo, para el proceso de diseño e implementación del sistema, se tendrá en cuenta la metodología XP, la cual nos proporciona las herramientas necesarias para el corrector desarrollo de la aplicación. Finalmente luego del , diseño e implementación del Sistema de información se concluye que su implementación permitirá mejorar de manera sustancial el seguimiento de documentos, se debe de tener en cuenta que, para realizar dichas mejoras, tanto los trabajadores como la gerencia deberán garantizar el uso al 100% del Software, alimentando la base de datos con todos los expedientes judiciales existentes en el estudio Jurídico, así mismo se concluye que el Sistema de Información ha colaborado satisfactoriamente en la gestión de los procesos judiciales del Estudio Jurídico SALAS CALDERON & ASOCIADOS, habiendo disminuido el tiempo total en brindar información al cliente acerca de su proceso

PDOEUDVFODYH: Programación Extrema, UML, Sistema de Información, Seguimiento de expedientes, PHP, Mysql.

ABSTRACT

This report thesis titled, descriptive, cross-sectional, with no experimental design, its main objective the analysis, design and implementation of a system to assist the monitoring of court cases in the law firm Salas Calderon & Associates Ica. The universe is made by the law firm Salas Calderon & Asociados city of Ica, which has 8 employees. For the collection of functional requirements and nonfunctional interviews and physical surveys were conducted. Also, for the process of design and implementation of web-based system it will take into account the XP methodology, which provides the necessary tools for correcting application development. Finally after the analysis, design and implementation of information system it is concluded that its implementation will improve substantially tracking documents, must be taken into account that, for such improvements, both workers and management must ensure the use 100% Software, feeding the database with all existing case files at the law firm, also concludes that the information system has worked successfully in the management of judicial processes Law Firm SALAS CALDERON & ASSOCIATES, having decreased the total time to provide information to customers about process

KH\ZRUGV: Extreme Programming, UML, Information System, File tracking, PHP, MySQL.

Contenido

DEDICATORIA	iii
AGRADECIMIENTO	
iv	
RESUMEN.....	v
ABSTRACT	
vi	
I. INTRODUCCIÓN	1
II. REVISIÓN DE LA LITERATURA	7
2.1. Conceptos jurídicos	7
2.1.1. Conceptos básicos	7
2.1.2. Descripción de un proceso judicial	8
2.1.3. Procesos principales de un estudio de abogados	10
2.2. Sistemas de información	11
2.3. BASE DE DATOS	16
2.4. METODOLOGÍA DE DESARROLLO DE SOFTWARE	22
2.5. LENGUAJE DE MODELAMIENTO UNIFICADO (UML)	36
2.6. LENGUAJE DE PROGRAMACIÓN ⁽²⁶⁾	48
2.7. SISTEMA DE GESTIÓN DE BASE DE DATOS:	55
III. METODOLOGÍA	60
3.1. DISEÑO DE LA INVESTIGACIÓN	60
3.2. Población y muestra	60

3.3.	DEFINICION Y OPERACIONALIZACIÓN DE LAS VARIABLES:	
	62	
3.4.	Técnicas e instrumentos:	
	63	
3.5.	Plan de :	
	63	
3.6.	Matriz de Consistencia:	
	64	
3.7.-	Principio Éticos	65
IV.	RESULTADOS	
	66	
4.1.	FACTIBILIDAD TÉCNICA	
	67	
4.2.	FACTIBILIDAD OPERATIVA	67
4.3.	FACTIBILIDAD ECONÓMICA	68
4.4.	MODELADO DEL NEGOCIO	68
4.4.2.	DISEÑO	71
	4.4.2.1. DIAGRAMAS DE CASOS DE USO	71
	4.4.2.2. DIAGRAMA DE ACTIVIDADES	72
	4.4.2.3. Diagramas de Secuencia	84
	4.4.2.4. Diagrama de Clases	86
4.4.3.	Construcción del Sistema: Fase de Construcción	89
V.	CONCLUSIONES	
	116	
	RECOMENDACIONES	
	117	
	BIBLIOGRAFIA	118
	ANEXOS	124

INDICE DE GRÁFICOS

Gráfico N° 1	71
Gráfico N° 2	71
Gráfico N° 3	72
Gráfico N° 4	74
Gráfico N° 5	76
Gráfico N° 6	78
Gráfico N° 7	80
Gráfico N° 8	82
Gráfico N° 9	84
Gráfico N° 10	85
Gráfico N° 11	86
Gráfico N° 12	87
Gráfico N° 13	88
Gráfico N° 14	89
Gráfico N° 15	90

Gráfico N° 16
90

Gráfico N° 17
91

Gráfico N° 18
92

Gráfico N° 19
92

I. INTRODUCCIÓN

En la actualidad el avance tecnológico viene abriendo nuevas puertas de oportunidades para las diferentes profesiones, el uso de tecnologías de la información y comunicación por parte de los diferentes profesionales van generando un valor adicional a su labor, no solo las grandes empresas se benefician con la implementación y uso de las tecnologías de la información y comunicación, en la actualidad pequeñas empresas o negocios personales usan las tecnologías para optimizar sus procesos, aumentando de esta forma su ventaja ante sus posibles competidores, esto se logra gracias a la reducción de tiempo al automatizar sus diferentes procesos. Los estudios jurídicos brindan servicios que dan solución a los diferentes conflictos sociales originados por múltiples razones. Esto hace que los litigantes acudan con la necesidad de que su problema sea solucionado, originando así un proceso judicial, el cual debe ser llevado de una forma adecuada para de esta manera buscar la satisfacción de las partes involucradas. Si definimos el problema del estudio Jurídico SALASCALDERON & ASOCIADOS de la ciudad de Ica, se tiene una deficiente gestión de los procesos judiciales; ya que se presenta una serie de problemas que dan lugar a una molestia por parte de los clientes, debido a que no se les brinda una correcta y rápida información acerca del estado de los procesos judiciales que se están llevando, así como de las actuaciones y documentos judiciales que se están manejando durante el proceso. Ya como sean las notificaciones escritas, apelaciones y propiamente dicho los expedientes. Además no existe un control acerca de un registro de clientes con los que se está trabajando actualmente, ni se cuenta con una cartera de los clientes actualizada ya que en algunas ocasiones se desea conocer con cuántos clientes están trabajando y no se tiene disponible

dicha información. De lo analizado, nos encontramos con algunos incumplimientos en la atención al cliente, que se refleja en la demora de la información. Existen deficiencias en la administración de los documentos judiciales, originando el extravío de ciertos documentos de suma importancia para el Estudio Jurídico, al mismo tiempo causan molestias y trabas en el desarrollo del proceso judicial viéndose en algunas ocasiones afectado. A la vez existen deficiencias en la administración de las actividades judiciales,

En la actualidad para la categorización del abogado según su grado de experiencia, se suelen utilizar nombres como socio (caso de los propietarios de la firma); asociados (caso de abogados especialistas con participación en las ganancias, pero sin derechos políticos o bien abogados externos que prestan sus servicios a la firma en casos especiales); "junior" (caso de los abogados con menor experiencia y que se encuentran subordinados a los socios o "senior").

La utilidad de éste tipo de colectivos, es que, respecto al cliente o usuario, éste puede acceder a un grupo de abogados que manejan distintas áreas del Derecho con algún mayor o menor grado de especialización. Sin embargo, a su vez tiene como consecuencia la despersonalización de la labor desempeñada por el abogado, ya que éste no conoce de forma integral los casos del cliente, sino sólo aquellos que por área de conocimiento o dedicación aborda.

Otro inconveniente desde el punto de vista práctico, es que por la gran infraestructura que se requiere para el mantenimiento del equipo de abogados, los costes aumentan sustancialmente, y la atención y destinación de los asuntos tratados, en la mayoría de los casos se ven atendidos no por los abogados con mayor experiencia, terminando en manos

de los abogados junior, en el Estudio Jurídico Salas Calderón & Asociados de la ciudad de Ica, se vienen llevando actualmente diferentes casos judiciales, los cuales son registrados de manera manual en cuadernos que fácilmente se pierden, esto perjudica a los clientes y al estudio jurídica, ya que causa demora en la solución de problemas, de la misma forma es difícil hacerle el seguimiento a un determinado expediente para conocer su estado ya que normalmente cada expediente es observado y corregido por el profesional que lleva el caso, y es este profesional el que conoce los elementos que intervienen en dicho caso, como el delito, demandante y demandado, distrito judicial, etc.

De acuerdo a la problemática descrita en los párrafos anteriores, planteamos la siguiente pregunta de investigación: ¿Cómo beneficiaría él , diseño e implementación de un sistema de apoyo al seguimiento de procesos judiciales en el Estudio Jurídico Salas Calderón & Asociados de la ciudad de Ica?

Para responder a la pregunta de investigación expresada anteriormente se ha establecido el siguiente objetivo general: Determinar la medida en que el Sistema de Información, influye en la Gestión de los Procesos Judiciales del Estudio Jurídico SALAS CALDERON & ASOCIADOS de la ciudad de Ica.

Y los objetivos específicos son:

- Analizar la problemática actual del consultorio jurídico
- Seleccionar la metodología adecuada para el modelado del sistema de información
- Obtener los requerimientos funcionales y no funcionales del sistema de información
- Analizar y especificar los requerimientos de hardware.

- Seleccionar el lenguaje de programación y el SGBD a usar
- El sistema de información permite llevar una adecuada administración de los clientes y sus datos respectivos con los que se está trabajando el Estudio Jurídico, además posibilita a los abogados conocer la cartera de clientes específica dependiendo de los distritos de Ica, así como los abogados y empleados que laboran en el Estudio Jurídico.

El sistema permitirá una correcta gestión de los procesos judiciales, registrando los datos de un proceso determinado, así como el o los clientes que se encuentran involucrados, trayendo consigo una rápida búsqueda del expediente deseado, disminuyendo así los tiempos requeridos normalmente para esta actividad. La secretaria podrá llevar una adecuada administración de los documentos judiciales, ya que mediante la digitalización de estos se enlazarán con el proceso judicial respectivo, teniendo así un acceso rápido a la información ya que se obviará la búsqueda innecesaria en algunos casos del expediente físico, evitando consigo una demora y buscando así la satisfacción por parte de los clientes, así mismo permitirá una organización de las diferentes actividades judiciales que se ejecutan durante el proceso judicial, proporcionando así el acceso a la consulta diaria o por un determinado proceso judicial, donde el abogado o secretaria podrá realizar la programación de estas, verificando cuales fueron cumplidas o incumplidas.

De esta forma el presente estudio se justifica, por ser de suma importancia para el desarrollo de las actividades dentro del Consultorio jurídico

En el año 2005 Contreras F. en su tesis “Diseño de un modelo para la implementación de un sistema de gestión documental en áreas u organizaciones jurídicas “ el cual en este proyecto de grado se enfocó desde un principio en las entidades jurídicas aprovechando las necesidades de dichas tenían en el manejo documental sin embargo esto no significa que los resultados alcanzados a otras áreas o entidades como por ejemplo en el manejo de historiales médicos en establecimientos de salud donde también se encuentran deficiencias en el manejo de un alto volumen de documentos El objetivo general se enfocó a diseñar un modelo para la implementación de un sistema documental en un entorno jurídico Levantar la información conceptual, procedimental, técnica y tecnológica para la implementación de un modelo que permita la aplicación de un sistema de gestión documental y para identificar la naturaleza del entorno jurídico Identificar y definir en conjuntos de características organizacionales en los aspectos conceptuales, procedimentales técnicos y tecnológicos que será usada para la construcción del modelo de acuerdo a las necesidades y realidad organizacional ⁽¹⁾

En el año 2012 Salazar C. presenta la tesis “Implementación de un sistema de gestión y control de un despacho jurídico” la cual trata sobre la implementación de un sistema de información para el ámbito de la gestión y control de despachos jurídicos que implica el seguimiento adecuado de los procesos y trámites, administración eficiente de las actividades relacionadas con los procesos, recuperación y tratamiento eficiente de la información generada en los despachos, etc. El sistema se ha construido siguiendo una metodología ágil de desarrollo scrum, que ha permitido controlar eficientemente cada una de las actividades y ser flexible a los cambios sugeridos por el cliente. La arquitectura que se ha utilizado para desarrollar el sistema fue la arquitectura modelo-vista-controlador que

ha permitido desacoplar la interfaz del usuario de la lógica del negocio y el modelo de datos. La arquitectura física del sistema es cliente-servidor, dicha arquitectura se ejecutará en la intranet que posee el despacho. Al implementar un sistema de información en el ámbito de la gestión y control de despachos jurídicos, se ha logrado organizar, recuperar, controlar, administrar, manipular, y dar seguimiento a la información generada al trabajar con procesos judiciales y sus respectivas actividades.⁽²⁾

En el año 2013 Escobar M. realizó su tesis titulada “, diseño e implementación de un sistema de apoyo a los procesos judiciales “ el cual tiene como objetivo analizar los procesos que se llevan en un estudio jurídico y que los objetivos que pueda ser tomada como base para implementarse en otros estudios jurídicos y también en , haciendo las mejoras necesarias, en una instancia mayor como podría ser un juzgado o fiscalía y que beneficie de esta manera al sistema judicial del país el cual, en la actualidad, es ineficiente en el desarrollo de los procesos judiciales debido a retardos, que a su vez son originados por la congestión de expedientes o por la pérdida de éstos. la población tiene una percepción negativa del sistema judicial del país debido a ciertas ineficiencias que se presentan en el desarrollo de un proceso judicial⁽³⁾

II. REVISION DE LA LITERATURA

2.1. Conceptos jurídicos

2.1.1. Conceptos básicos

El Derecho es el sistema de regulación que establece las normas básicas de conductas sociales las cuales permiten una convivencia libre de conflictos. Sin embargo, no siempre es así. Cuando se produce una controversia entre partes y éstas no logran llegar a acuerdo alguno, será necesaria la intervención de un mediador y se constituirá un proceso, el cual tendrá como contenido dicha controversia. Los causales de dicha controversia tienen distintos orígenes, los principales son la discordancia de puntos de vista entre dos personas, el incumplimiento de alguna norma vigente, el reclamo por alguna violación de derechos de las personas, etc. Entonces, un proceso judicial está definido por una discordancia entre dos partes: persona o institución que inicia el proceso y la persona o institución sobre la que se inicia el proceso (generalmente denominados demandante y demandado, respectivamente). Dicha causa será resuelta por un tercer integrante, llamado juez, que se encargará de discernir cuál de las dos partes se fundamenta con mayor base en el estado de derecho y partir de ello dará un fallo. Los procesos judiciales están clasificados por materias, entre las principales se encuentran la civil, penal, laboral, de familia y comercial. Si bien es cierto cada materia tiene sus propias características que las distinguen del resto (por ejemplo, las partes se denominan demandante - demandado en un proceso civil y agraviado - inculpaado en un proceso penal), todas comparten ciertos principios comunes y se basan en la materia civil. Aunque un proceso puede presentar diversas variantes a lo largo de su desarrollo, en general su esencia se mantiene. No obstante, se debe tener en cuenta las etapas del proceso diferenciadas por cada uno de los

órganos jurisdiccionales (aquellos que tienen la misión de administrar justicia, es decir, de juzgar y hacer ejecutar lo juzgado). Dichas etapas se denominan instancias y los órganos jurisdiccionales son los siguientes:

- Juzgados de Paz (No Letrados y Letrados): el menor nivel jerárquico en que se encuentra organizado el Poder Judicial.
- Juzgados de Primera Instancia (Especializados o Mixtos): el segundo nivel jerárquico en que se encuentra organizado el Poder Judicial del Perú.
- Cortes Superiores: el tercer nivel jerárquico en que se organiza el Poder Judicial. Sólo se encuentran bajo la autoridad de la Corte Suprema de la República y es, en la mayoría de procesos, el último organismo que conoce un proceso.
- Corte Suprema: máximo órgano jurisdiccional del Perú. Su competencia se extiende a todo el territorio del país.

Mediante las instancias se puede realizar una impugnación a un fallo (esto se describirá a detalle en la siguiente sección) y es la Corte Suprema la última instancia ante la cual se pueden apelar todos los procesos judiciales. Todo proceso tiene su equivalente expediente, el cual contiene las piezas escritas del proceso, agregadas sucesivamente y en orden de presentación, con las que se forma un solo cuerpo foliado con número y letras ⁽⁴⁾

2.1.2. Descripción de un proceso judicial

En esta sección se describirá los pasos por los que atraviesa un proceso (se tomará de base un proceso civil que es el más general).

Cuando se presenta un conflicto, las partes se acercan a un Juzgado de Paz, donde el Juez de Paz buscará la conciliación entre las partes en desacuerdo. Ahora bien, si no se logra

un acuerdo entre ambas partes, el Juez de Paz debe remitir el caso al siguiente nivel del sistema judicial, es decir, a los Juzgados de Primera Instancia. Aquí se inicia el proceso judicial propiamente dicho. ⁽⁵⁾

El procedimiento se inicia presentando una **demanda** por escrito la cual se interpone ante un juez especializado en la materia correspondiente. Esta demanda debe ser **contestada** por el demandado, dentro de un plazo cuya extensión varía según los diferentes procedimientos que se indiquen en el Código de la materia respectiva. Tramitado el proceso de la manera antedicha, y suponiendo que no hubiera problemas que requieran complicaciones procesales, el juez expedirá una resolución validando el proceso (a lo que se denomina **saneamiento del proceso**). Luego del saneamiento del proceso, se realiza una **audiencia conciliatoria** que tiene por finalidad la conciliación entre las partes. Si hay conciliación se levanta un acta que equivale a una sentencia con autoridad de cosa juzgada. Si no hay conciliación, se continuará con el proceso probatorio de los hechos, para lo cual se procederá a enumerar los puntos controvertidos entre las partes y, en especial, los que van a ser materia de prueba. Posteriormente se realizará la actuación de las pruebas dirigidas por el Juez bajo sanción de nulidad en la **audiencia de pruebas**. Terminada la audiencia, la causa queda expedita para ser sentenciada.

Uno de los medios impugnatorios existentes es el recurso de **apelación**, mediante el cual las partes solicitan se anule, revoque o reforme un fallo judicial. La apelación habilita la competencia de la segunda instancia en el proceso (por ejemplo, si se había iniciado un proceso ante un Juez Especializado Civil, la apelación hará que el proceso se continúe en la sala civil de la Corte Superior que corresponde al distrito judicial del juez en mención) La Sala competente de la Corte Superior **sentenciará en segunda instancia**. Se

interpone la **casación** (recurso extraordinario que tiene por objeto anular la sentencia judicial en segunda instancia), para que la Sala competente de la Corte Suprema de la

República revise cualquiera de las siguientes causales:

- Aplicación indebida o la interpretación errónea de una norma de derecho material o doctrina jurisprudencial.
- Inaplicación de una norma de derecho material o doctrina jurisprudencial.
- Contravención de las normas que garantizan el derecho a un debido proceso o a la infracción de las formas esenciales para la eficacia y validez de los actos procesales. El proceso termina en definitiva con la **sentencia resolviendo la casación**. Ningún otro medio impugnatorio puede interponerse contra ella.⁽⁶⁾

2.1.3. Procesos principales de un estudio de abogados

En base a la información que presentan diversos Estudios de Abogados

y a partir de la clasificación estándar de procesos de negocio y métricas desarrollados

por "OpenLegal Standards Initiative" se pueden agrupar los procesos básicos de un

estudio de

abogados en tres grupos:

- Gestión de personal. Todo estudio de abogados cuenta con un staff de abogados quienes serán los encargados de realizar el seguimiento y gestión de sus procesos.
- Atención al cliente. Todo estudio de abogados ofrece un servicio de asesoría legal a sus clientes y gestiona la información de éstos para brindar una mejor calidad de servicio.

Gestión de procesos. Todo estudio de abogados lleva el control de los

procesos de sus clientes asignando para ello a un abogado de su staff, quien le hará el

respectivo seguimiento.⁽⁷⁾

2.2. Sistemas de información

Un sistema de información según Fernández A., (2006), Por definición es un conjunto de elementos que interactúan entre sí con un fin común; que permite que la información esté disponible para satisfacer las necesidades en una organización, un sistema de información no siempre requiere contar con recurso computacional aunque la disposición del mismo facilita el manejo e interpretación de la información por los usuarios. Los elementos que interactúan entre sí son: el equipo computacional (cuando esté disponible), el recurso humano, los datos o información fuente, programas ejecutados por las computadoras, las telecomunicaciones y los procedimientos de políticas y reglas de operación. Un Sistema de Información realiza cuatro actividades básicas:

Entrada de información: proceso en el cual el sistema toma los datos que requiere.

Almacenamiento de información: puede hacerse por computadora o archivos físicos para conservar la información.⁽⁸⁾

Procesamiento de la información: permite la transformación de los datos fuente en información que puede ser utilizada para la toma de decisiones

Salida de información: es la capacidad del sistema para producir la información procesada o sacar los datos de entrada al exterior.

Los usuarios de los sistemas de información tienen diferente grado de participación dentro de un sistema y son el elemento principal que lo integra, así se puede definir usuarios primarios quienes alimentan el sistema, usuarios indirectos que se benefician de los resultados pero que no interactúan con el sistema, usuarios gerenciales y directivos

quienes tienen responsabilidad administrativa y de toma de decisiones con base a la información que produce el sistema. ⁽⁹⁾

a. CARACTERÍSTICAS DE LOS SISTEMAS DE INFORMACIÓN

Los Sistemas de Información están constituidos por características comunes que se definen a continuación:

Entrada de Información: Es el proceso mediante el cual el Sistema de Información toma los datos que requiere para procesar la información. Las entradas pueden ser manuales o automáticas. Las manuales son aquellas que se proporcionan en forma directa por el usuario, mientras que las automáticas son datos o información que provienen o son tomados de otros sistemas o módulos.

Procesamiento de Información: Es la capacidad del Sistema de Información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida. Estos cálculos pueden efectuarse con datos introducidos recientemente en el sistema o bien con datos que están almacenados. Esta característica de los sistemas permite la transformación de datos fuente de información que puede ser utilizada para la toma de decisiones. **Salida**

de Información: La salida es la capacidad de un Sistema de Información para exponer la información procesada al exterior, bien sea a través de un monitor, en papel o por cualquier otro medio.

Almacenamiento de Información: El Almacenamiento es una de las actividades o capacidades más importante que tiene una computadora, ya que a través de esta propiedad el sistema puede recordar la información guardada en la sesión o proceso anterior. Esta información suele ser almacenada en estructuras de información denominadas archivos.

(10)

b. CICLO DE VIDA DE UN SISTEMA DE INFORMACIÓN

El ciclo de vida de los sistemas consta de cinco fases. Las primeras cuatro –planificación, , diseño e implementación – se dedican a la creación. La quinta fase se dedica al uso. En todas las fases deben intervenir los usuarios y pueden participar especialistas en información si no se está practicando la computación de usuario final en su forma más pura. Las actividades dentro del ciclo de vida de un sistema, tanto de los usuarios como de los especialistas en información, se admitan desde varios puntos de supervisión dentro de la compañía. Los ejecutivos establecen políticas y hacen planes que establecen el margo general para el uso de las computadoras. En un nivel un poco más bajo, un comité especial llamado director del MIS puede controlar todos los ciclos de vida de la compañía. A medida que cada uno de los ciclos de vida pasa por las fases de desarrollo, los jefes de proyecto supervisan a los miembros del equipo. El ciclo de vida de los sistemas es una aplicación del enfoque de sistemas a la tarea de crear y usar un sistema basado en computadora. Como tal, el ciclo de vida de los sistemas es una metodología, pero en su patrón está influyendo la necesidad de desarrollar sistemas con mayor rapidez. Se logra una respuesta rápida en la creación de sistemas refinando el ciclo de vida usando herramientas para desarrollo basadas en computadora. Dos refinamientos son el uso de prototipos y la creación rápida de aplicaciones (RAD), y estas herramientas pertenecen a una categoría llamada ingeniería de software asistida por computador.⁽¹⁰⁾

c. CICLO DE VIDA CLÁSICO DEL DESARROLLO DE SISTEMAS ⁽¹¹⁾

El método de ciclo de vida para el desarrollo de sistemas es el conjunto de actividades que los analistas, diseñadores y usuarios realizan para desarrollar e implantar un sistema de información. El método del ciclo de vida para el desarrollo de sistemas consta de 6 fases:

1). Investigación Preliminar: La solicitud para recibir ayuda de un sistema de información puede originarse por varias razones: sin importar cuales sean estas, el proceso se inicia siempre con la petición de una persona.

2). Determinación de los requerimientos del sistema: El aspecto fundamental del de sistemas es comprender todas las facetas importantes de la parte de la empresa que se encuentra bajo estudio. Los analistas, al trabajar con los empleados y administradores, deben estudiar los procesos de una empresa para dar respuesta a las siguientes preguntas clave:

¿Qué es lo que hace?

¿Cómo se hace?

¿Con que frecuencia se presenta?

¿Qué tan grande es el volumen de transacciones o decisiones?

¿Cuál es el grado de eficiencia con el que se efectúan las tareas?

¿Existe algún problema? ¿Qué tan serio es? ¿Cuál es la causa que lo origina?

3). Diseño del sistema: El diseño de un sistema de información produce los detalles que establecen la forma en la que el sistema cumplirá con los requerimientos identificados durante la fase de los especialistas en sistemas se refieren, con frecuencia, a esta etapa como diseño lógico en contraste con la del desarrollo del software, a la que denominan diseño físico.

4). Desarrollo del software: Los encargados de desarrollar software pueden instalar software comprobando a terceros o escribir programas diseñados a la medida del solicitante. La elección depende del costo de cada alternativa, del tiempo disponible para escribir el software y de la disponibilidad de los programadores.

Por lo general, los programadores que trabajan en las grandes organizaciones pertenecen a un grupo permanente de profesionales.

5). Prueba de sistemas: Durante la prueba de sistemas, el sistema se emplea de manera experimental para asegurarse de que el software no tenga fallas, es decir, que funciona de acuerdo con las especificaciones y en la forma en que los usuarios esperan que lo haga. Se alimentan como entradas conjunto de datos de prueba para su procesamiento y después se examinan los resultados.

6). Implantación y evaluación: La implantación es el proceso de verificar e instalar nuevo equipo, entrenar a los usuarios, instalar la aplicación y construir todos los archivos de datos necesarios para utilizarla. Una vez instaladas, las aplicaciones se emplean durante muchos años. Sin embargo, las organizaciones y los usuarios cambian con el paso del tiempo, incluso el ambiente es diferente con el paso de las semanas y los meses.

Por consiguiente, es indudable que debe darse mantenimiento a las aplicaciones. La evaluación de un sistema se lleva a cabo para identificar puntos débiles y fuertes. La evaluación ocurre a lo largo de cualquiera de las siguientes dimensiones:

- Evaluación operacional: Valoración de la forma en que funciona el sistema, incluyendo su facilidad de uso, tiempo de respuesta, lo adecuado de los formatos de información, confiabilidad global y nivel de utilización.

- Impacto organizacional: Identificación y medición de los beneficios para la organización en áreas tales como finanzas, eficiencia operacional e impacto competitivo.

También se incluye el impacto sobre el flujo de información externo e interno.

- Opinión de los administradores: evaluación de las actividades de directivos y administradores dentro de la organización, así como de los usuarios finales.

- Desempeño del desarrollo: La evaluación de proceso de desarrollo de acuerdo con criterios tales como tiempo y esfuerzo de desarrollo, concuerdan con presupuestos y estándares, y otros criterios de administración de proyectos. También se incluye la valoración de los métodos y herramientas utilizados en el desarrollo. ⁽¹¹⁾

2.3. BASE DE DATOS

a. DEFINICIÓN

Una base de datos es el conjunto de datos informativos organizados en un mismo contexto para su uso y vinculación.

Se le llama base de datos a los bancos de información que contienen datos relativos a diversas temáticas y categorizados de distinta manera, pero que comparten entre sí algún tipo de vínculo o relación que busca ordenarlos y clasificarlos en conjunto.

Una base de datos puede ser de diverso tipo, desde un pequeño fichero casero para ordenar libros y revistas por clasificación alfabética hasta una compleja base que contenga datos de índole gubernamental en un Estado u organismo internacional. Recientemente, el término base de datos comenzó a utilizarse casi exclusivamente en referencia a bases construidas a partir de software informático, que permiten una más fácil y rápida organización de los datos. Las bases de datos informáticas pueden crearse a partir de

software o incluso de forma online usando Internet. En cualquier caso, las funcionalidades disponibles son prácticamente ilimitadas.

Las bases de datos se clasifican como estáticas – en casos en que sólo sirven para su lectura y almacenamiento – o dinámicas – la información se modifica y puede ser actualizada -. También, según su contenido pueden ser bibliográficas, de texto completo, directorios, o de tipo biblioteca.

Los usos de las bases de datos son tan múltiples que, por ejemplo, pueden utilizarse en una biblioteca o archivo que guarda libros para su consulta frecuente, pero también pueden emplearse para guardar material biológico, como un archivo genético o un banco de esperma que almacena esperma para ser utilizado en fertilización.

Este tipo de clasificadores contribuyen, por supuesto, a labores diarias de casi todo tipo de empleos, como registrar las comunicaciones que se sostienen con clientes, almacenar datos de proveedores, archivar datos, números y todo tipo de información que el usuario quiera poner en relación o encontrar fácilmente con sólo ingresar una palabra clave. ⁽¹²⁾

b. Tipos de bases de datos⁽¹³⁾

Existen muchas empresas con diferentes giros y dependiendo del giro será el tipo de procesamiento que se le dará a la información, esto determinará el tipo de base de datos a utilizar. Existen diferentes tipos de bases de datos, pero las más comunes son las OLTP y OLAP.

Las bases de datos de tipo OLTP (On Line Transaction Processing) también son llamadas bases de datos dinámicas lo que significa que la información se modifica en tiempo real, es decir, se insertan, se eliminan, se modifican y se consultan datos en línea durante la

operación del sistema. Un ejemplo es el sistema de un supermercado donde se van registrando cada uno de los artículos que el cliente está comprando y a su vez el sistema va actualizando el Inventario.

Las bases de datos de tipo OLAP (On Line Analytical Processing) también son llamadas bases de datos estáticas lo que significa que la información en tiempo real no es afectada, es decir, no se insertan, no se eliminan y tampoco se modifican datos; solo se realizan consultas sobre los datos ya existentes para el y toma de decisiones. Este tipo de bases de datos son implementadas en Business Intelligence para mejorar el desempeño de las consultas con grandes volúmenes de información.

La necesidad de implementar un tipo u otro dependerá del giro y necesidades de cada empresa. Es muy importante identificar el tipo de base de datos que se requiere antes de implementar un manejador de base de datos. Por ejemplo, podemos citar al manejador de base de datos “DB2 10.5 with BLU Acceleration”, la cual proporciona una solución para la parte transaccional y para la parte de analíticos. ⁽¹³⁾

c. **NORMALIZACIÓN**⁽¹⁴⁾

Fundamentos de la normalización

La normalización es el proceso de organizar los datos de una base de datos. Se incluye la creación de tablas y el establecimiento de relaciones entre ellas según reglas diseñadas tanto para proteger los datos como para hacer que la base de datos sea más flexible al eliminar la redundancia y las dependencias incoherentes. Los datos redundantes desperdician el espacio de disco y crean problemas de mantenimiento. Si hay que cambiar datos que existen en más de un lugar, se deben cambiar

de la misma forma exactamente en todas sus ubicaciones. Un cambio en la dirección de un cliente es mucho más fácil de implementar si los datos sólo se almacenan en la tabla Clientes y no en algún otro lugar de la base de datos.

¿Qué es una "dependencia incoherente"? Aunque es intuitivo para un usuario mirar en la tabla Clientes para buscar la dirección de un cliente en particular, puede no tener sentido mirar allí el salario del empleado que llama a ese cliente. El salario del empleado está relacionado con el empleado, o depende de él, y por lo tanto se debería pasar a la tabla Empleados. Las dependencias incoherentes pueden dificultar el acceso porque la ruta para encontrar los datos puede no estar o estar interrumpida.

Hay algunas reglas en la normalización de una base de datos. Cada regla se denomina una "forma normal". Si se cumple la primera regla, se dice que la base de datos está en la "primera forma normal". Si se cumplen las tres primeras reglas, la base de datos se considera que está en la "tercera forma normal". Aunque son posibles otros niveles de normalización, la tercera forma normal se considera el máximo nivel necesario para la mayor parte de las aplicaciones.

Al igual que con otras muchas reglas y especificaciones formales, en los escenarios reales no siempre se cumplen los estándares de forma perfecta. En general, la normalización requiere tablas adicionales y algunos clientes consideran éste un trabajo considerable. Si decide infringir una de las tres primeras reglas de la normalización, asegúrese de que su aplicación se anticipa a los problemas que puedan aparecer, como la existencia de datos redundantes y de dependencias incoherentes.

Primera forma normal

Elimine los grupos repetidos de las tablas individuales.

Cree una tabla independiente para cada conjunto de datos relacionados.

Identifique cada conjunto de datos relacionados con una clave principal.

No use varios campos en una sola tabla para almacenar datos similares. Por ejemplo, para realizar el seguimiento de un elemento del inventario que proviene de dos orígenes posibles, un registro del inventario puede contener campos para el Código de proveedor 1 y para el Código de proveedor 2.

¿Qué ocurre cuando se agrega un tercer proveedor? Agregar un campo no es la respuesta, requiere modificaciones en las tablas y el programa, y no admite fácilmente un número variable de proveedores. En su lugar, coloque toda la información de los proveedores en una tabla independiente denominada Proveedores y después vincule el inventario a los proveedores con el número de elemento como clave, o los proveedores al inventario con el código de proveedor como clave.

Segunda forma normal

Cree tablas independientes para conjuntos de valores que se apliquen a varios registros.

Relacione estas tablas con una clave externa.

Los registros no deben depender de nada que no sea una clave principal de una tabla, una clave compuesta si es necesario. Por ejemplo, considere la dirección de un cliente en un sistema de contabilidad. La dirección se necesita en la tabla Clientes, pero también en las tablas Pedidos, Envíos, Facturas, Cuentas por cobrar y Colecciones. En lugar de almacenar la dirección de un cliente como una entrada independiente en cada una de estas tablas, almacénela en un lugar, ya sea en la tabla Clientes o en una tabla Direcciones independiente.

Tercera forma normal

Elimine los campos que no dependan de la clave.

Los valores de un registro que no sean parte de la clave de ese registro no pertenecen a la tabla. En general, siempre que el contenido de un grupo de campos pueda aplicarse a más de un único registro de la tabla, considere colocar estos campos en una tabla independiente. Por ejemplo, en una tabla Contratación de empleados, puede incluirse el nombre de la universidad y la dirección de un candidato. Pero necesita una lista completa de universidades para enviar mensajes de correo electrónico en grupo. Si la información de las universidades se almacena en la tabla Candidatos, no hay forma de enumerar las universidades que no tengan candidatos en ese momento. Cree una tabla Universidades independiente y vincúlela a la tabla Candidatos con el código de universidad como clave.

EXCEPCIÓN: cumplir la tercera forma normal, aunque en teoría es deseable, no siempre es práctico. Si tiene una tabla Clientes y desea eliminar todas las dependencias posibles entre los campos, debe crear tablas independientes para las ciudades, códigos postales, representantes de venta, clases de clientes y cualquier otro factor que pueda estar duplicado en varios registros. En teoría, la normalización merece el trabajo que supone. Sin embargo, muchas tablas pequeñas pueden degradar el rendimiento o superar la capacidad de memoria o de archivos abiertos.

Puede ser más factible aplicar la tercera forma normal sólo a los datos que cambian con frecuencia. Si quedan algunos campos dependientes, diseñe la aplicación para que pida al usuario que compruebe todos los campos relacionados cuando cambie alguno.

Otras formas de normalización

LA CUARTA FORMA NORMAL:

También llamada Forma normal de Boyce Codd (BCNF, Boyce Codd Normal Form), y la quinta forma normal existen, pero rara vez se consideran en un diseño real. Si no se aplican estas reglas, el diseño de la base de datos puede ser menos perfecto, pero no debería afectar a la funcionalidad. ⁽¹⁴⁾

2.4. METODOLOGÍA DE DESARROLLO DE SOFTWARE

A decir de Menéndez y Barsanallana (15), las metodologías de software son el conjunto de políticas, reglas, procedimientos que definen los pasos a seguir para llegar a la culminación de un proyecto de software garantizando la eficacia y eficiencia del desarrollo durante su ciclo.

Por tanto, el escoger adecuadamente la metodología a seguir durante el desarrollo de software puede determinar el éxito o no del proyecto a realizar, por esta razón detallamos algunas de las principales metodologías más cercanas a nuestra línea de investigación:

a. RATIONAL UNIFIED PROCESS - RUP

Rational Unified Process, es un marco de desarrollo de software dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental, pretende implementar las mejores prácticas en ingeniería de software, con el objetivo de asegurar la producción de software de calidad, dentro de los plazos y presupuestos predecibles,

Provee un enfoque disciplinado en la asignación de tareas y responsabilidades dentro de una organización de desarrollo. Su meta es asegurar la producción de software de muy alta calidad que satisfaga las necesidades de los usuarios finales, dentro de un calendario y presupuesto predecible. ⁽¹⁶⁾

RUP tienen principios fundamentales de desarrollo que son:

Adaptar el proceso, que se refiere a que los procesos deben adaptarse al tamaño de los proyectos o de la organización. Enfocarse a la calidad, el control de calidad deberá ser llevado a cabo a lo largo de toda la producción. Balancear prioridades, se debe encontrar un balance que satisfaga los deseos de todos. Colaboración entre equipos, los proyectos de desarrollo de software no son llevados a cabo por una sola persona, sino varias o varios equipos de desarrollo, los cuales deben contar con una buena comunicación para que esto les permita coordinar esfuerzos. Demostrar valor iterativamente, los proyectos, aunque sea de manera interna deben entregarse de manera iterada en cada iteración se analiza el avance, estabilidad, calidad del producto. Elevar el nivel de abstracción, esto previene a los ingenieros de Software ir directamente de los requerimientos del cliente a la codificación, un nivel alto de abstracción permite discusiones sobre diversos niveles de arquitectura, los cuales se pueden acompañar por representaciones visuales de la arquitectura como por ejemplo utilizando UML.⁽¹⁷⁾

RUP está formado por dos dimensiones:

Una horizontal que representa el ciclo de vida, fases del proyecto de acuerdo al transcurso del tiempo.

Una vertical que agrupa actividades definidas lógicamente por la naturaleza del proyecto, iteraciones.

La primera dimensión representa el aspecto dinámico del proceso conforme se va desarrollando, se expresa en términos de fases, iteraciones e hitos.

La segunda dimensión representa el aspecto estático del proceso: cómo es descrito en términos de componentes del proceso, disciplinas, actividades, flujos de trabajo, artefactos y roles. ⁽¹⁷⁾

Sánchez (2012), define las siguientes fases para el modelo RUP:

Inicio: se hace un plan de fases, se identifican los principales casos de uso y se elimina los medios.

Elaboración: se hace un plan de proyectos, se contempla los casos de uso y se eliminan los medios.

Construcción: se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario.

Transición: se instala el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto, suelen surgir nuevos requisitos a ser analizados. ⁽¹⁸⁾

Según Rueda (2006), Actividades, son los procesos que llegan a determinar en cada iteración. Representan una unidad de trabajo desempeñada por un determinado rol.

Roles, definen el comportamiento de las personas o entes involucrados en cada proceso.

Artefactos, es un elemento que el proyecto produce y utiliza para componer el producto final, puede ser un documento, un modelo o un elemento de modelo.

Flujos de Trabajo, constituyen la secuencia de actividades que producen resultados visibles por medio de la integración de los roles y las actividades, artefactos y disciplinas.

Un Flujo de Trabajo es una relación de actividades que nos producen unos resultados observables. Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software. En RUP,

los flujos de trabajo son secuencias realizadas por los diferentes roles así como la relación entre los mismos, estas actividades nos dan resultados observables.⁽¹⁹⁾

A continuación, se detallan los principales Flujos de Trabajos de Procesos:

Modelado del Negocio: en este flujo de trabajo se pretende entender la organización donde se va a implementar el producto. RUP proporciona un lenguaje y proceso común para ambos ámbitos. Para el moldeamiento del negocio se utilizan los Casos de Uso del Negocio, que aseguran un común entendimiento entre los interesados en el negocio y la organización.

Entre los objetivos de este Flujo, tenemos:

Entender la estructura y la dinámica de la organización para la cual el sistema va ser desarrollado (organización objetivo).

Entender el problema actual en la organización objetivo e identificar potenciales mejoras.

Asegurar que clientes, usuarios finales y desarrolladores tengan un entendimiento común de la organización objetivo.

Derivar los requisitos del sistema necesarios para apoyar a la organización objetivo.

Requerimientos: Determina que tiene que hacer el sistema para lo que se debe establecer requerimientos, documentar funcionalidad y restricciones, identificar actores y casos de uso.

Entre los objetivos de este flujo tenemos:

Establecer y mantener un acuerdo entre clientes y otros stakeholders sobre lo que el sistema podría hacer.

Proveer a los desarrolladores un mejor entendimiento de los requisitos del sistema.

Definir el ámbito del sistema.

Proveer una base para la planeación de los contenidos técnicos de las iteraciones.

Proveer una base para estimar costos y tiempo de desarrollo del sistema.

Definir una interfaz de usuarios para el sistema, enfocada a las necesidades y metas del usuario.

y Diseño: flujo de trabajo que describe como se implementará el sistema para lo cual se deben ejecutar las tareas y funciones descritas en los casos de uso. Como producto final se obtendrá el modelo de diseño, el modelo de (opcional) y la documentación de la arquitectura del Software.

Sus objetivos pueden ser enumerados de la siguiente manera:

Transformar los requisitos del diseño del futuro sistema.

Desarrollar una arquitectura para el sistema.

Adaptar el diseño para que sea consistente con el entorno de implementación, diseñando para el rendimiento.

Implementación: en este flujo se implementan las clases y objetos en ficheros fuente, binarios, ejecutables y demás. Se hacen las pruebas de unidad. El resultado final de este flujo de trabajo es un sistema ejecutable.

Sus objetivos son:

Definir la organización del código.

Implementar clases y objetos en forma de componentes.

Probar los componentes desarrollados

Integrar los componentes en un sistema integrado.

Pruebas: este flujo de trabajo es el encargado de evaluar la calidad del producto que se está desarrollando, pero no para aceptar o rechazar el producto al final del proceso de desarrollo, sino que debe integrarlo en todo el ciclo de vida.

Entre sus objetivos podemos identificar:

Encontrar y documentar defectos en la calidad del software.

Generalmente asesora sobre la calidad del software percibida.

Provee la validación de los supuestos realizados en el diseño y especificación de requisitos por medio de demostraciones concretas.

Verificar las funciones del producto de software según lo diseñado.

Verificar que los requisitos tengan su apropiada implementación.

Desarrollo: el objetivo de este flujo de trabajo es producir con éxito distribuciones del producto y distribuirlo a los usuarios.

Probar el producto en su entorno de ejecución final.

Empaquetar el software para su distribución.

Distribuir el software.

Instalar el software.

Proveer asistencia y ayuda a los usuarios.

Formar a los usuarios y al cuerpo de ventas.

Migrar el software existente o convertir bases de datos.

RUP es el conjunto de procesos que acompañado de la notación UML conforman una metodología de desarrollo que sigue procesos disciplinados para asignar tareas y responsabilidades, detallando y documentando todo el proceso de desarrollo.⁽²⁰⁾

b. EXTREME PROGRAMMING – XP⁽²¹⁾

La metodología XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre. Kent Beck (19), el padre de la metodología, describe la filosofía de la metodología XP es sin cubrir los detalles técnicos y de implantación de las prácticas.

Es una de las llamadas Metodologías ágiles de desarrollo de software más exitosas de los tiempos recientes, nace como nueva disciplina de desarrollo de software centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, se preocupa por el aprendizaje de los desarrolladores, y propicia un buen clima de trabajo. Está basada en los valores de simpleza, comunicación, feedback y coraje.

Auer y Miller (2001), destacan entre sus características fundamentales, las siguientes: Desarrollo iterativo e incremental, se comienza con un sistema con la principal funcionalidad y se va añadiendo una a una las funcionalidades restantes.

Pruebas Unitarias, se basa en las pruebas realizadas a los principales procesos, de tal manera que adelantándonos en algo hacia el futuro, podamos hacer pruebas de las fallas que pudieran ocurrir. Es como si nos adelantáramos a obtener los posibles errores. Re-fabricación, se basa en la reutilización de código, para lo cual se crean patrones o Modelos estándares, siendo más flexible al cambio.

Programación en Parejas, propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento.

Integración del Equipo de Programación con el Cliente, Es recomendable que el grupo de programación tenga contacto directo y continuo con el cliente.

Corrección de todos los errores, se debe corregir todos los errores antes de agregar funcionalidades.

Refactorización del código, si es necesario reescribir código para hacerlo más legible o mejorar su funcionalidad.

Propiedad del código compartida, crea funcionalidad que puede ser utilizada por otras personas dentro del proyecto.

Simplicidad, fácil de desarrollar, probar y mantener, se irá incrementando su complejidad si el cliente lo ve necesario. La metodología XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios.

La Programación Extrema “Extreme Programming” (XP) no es un conjunto de reglas a seguir, sino una forma de trabajar en armonía con los valores personales y organizacionales, que tiene su punto de partida en cuatro valores fundamentales ⁽²¹⁾:

1. COMUNICACIÓN

El extreme Programming se nutre del ancho de banda más grande que se puede obtener cuando existe algún tipo de comunicación: la comunicación directa entre personas. Es muy importante entender cuáles son las ventajas de este medio.

Cuando dos (o más) personas se comunican directamente pueden no solo consumir las palabras formuladas por la otra persona, sino que también se aprecian los gestos, miradas, etc. que hace su compañero.

Sin embargo, en una conversación mediante el correo electrónico, hay muchos factores que hacen de esta una comunicación, por así decirlo, mucho menos efectiva.

2. CORAJE

El coraje es un valor muy importante dentro de la programación extrema. Un miembro de un equipo de desarrollo extremo debe de tener el coraje de exponer sus dudas, miedos, experiencias sin "embellecer" éstas de ninguna de las maneras. Esto es muy importante ya que un equipo de desarrollo extremo se basa en la confianza para con sus miembros. Faltar a esta confianza es una falta más que grave.

3. SIMPLICIDAD

Dado que no se puede predecir cómo va a ser en el futuro, del software que está en desarrollo; un equipo de programación extrema intenta mantener el software lo más sencillo posible. Esto quiere decir que no se va a invertir ningún esfuerzo en hacer un desarrollo que en un futuro pueda llegar a tener valor. En la metodología XP frases como "...en un futuro se va a necesitar..." o "Hacer un sistema genérico de..." no tienen ningún sentido ya que no aportan ningún valor en el momento.

4. RETROALIMENTACIÓN

La agilidad se define (entre otras cosas) por la capacidad de respuesta ante los cambios que se hacen necesarios a lo largo del camino. Por este motivo uno de los valores que lo hace más ágil es el continuo seguimiento o retroalimentación que se recibe a la hora de desarrollar en un entorno ágil de desarrollo. Esta retroalimentación se toma del cliente, de

los miembros del equipo, en cuestión de todo el entorno en el que se mueve un equipo de desarrollo ágil.

En el caso de la programación extrema se identifican diferentes roles: un equipo de gestión o de diseño, uno de desarrollo y los clientes finales. La relación entre los equipos de diseño y desarrollo son distintas a lo definido en las metodologías tradicionales. Es así que entre sus principales fases tenemos ⁽²¹⁾:

Fase 1: Planificación del proyecto

El primer paso para cualquier proyecto que siga XP, son las Historias de Usuario, éstas tienen la misma finalidad que los casos de uso pero con algunas diferencias:

Son escritas por el cliente en 3 o 4 líneas utilizando un lenguaje no técnico, sin detalles y sin hacer referencias a diseños o algoritmos para la codificación.

Las Historias de Usuario son usadas para estimar tiempos y también en la fase de pruebas para verificar que el sistema cumple con lo deseado. Luego de definir las historias, es preciso elaborar un plan de publicaciones o Release Plan, que indicará las historias de usuario que se crearán para cada versión del programa y las fechas en las que se publicarán estas versiones.

Todo proyecto que utilice XP se ha de dividir en Iteraciones de 3 semanas de duración, al inicio de cada iteración se definirá el Release Planing y al final se determinarán las Historias de Usuario que no pasaron las pruebas de aceptación.

La Velocidad del Proyecto es la medida de la rapidez con que se desarrolla el proyecto; la estimación es sencilla, basta con contar el número de historias de usuario que se pueden implementar por una iteración XP como metodología recomienda la Programación en

Pareja, pues incrementa la productividad y la calidad del software desarrollado, este trabajo involucra a dos programadores en el mismo equipo.

Es necesario que los programadores mantengan Reuniones Diarias para que se expongan sus problemas, soluciones e ideas de forma conjunta, las reuniones tienen que ser fluidas y todo el mundo tiene que tener voz y voto.

Fase 2: Diseño

La metodología XP sugiere que se manejen Diseños Simples que procuren hacer todo lo menos complicado posible para conseguir un diseño entendible y fácil de implementar. Usar Glosarios de Términos y una correcta especificación de nombres de clases y métodos ayudará a comprender el diseño y facilitará una futura reutilización de código. No se debe añadir Funcionalidad Extra al programa aunque se piense en un futuro que ésta será utilizada, esto implica un desperdicio de tiempo y recursos.

Refactorizar es mejorar y modificar la estructura y codificación de códigos ya creados sin alterar su funcionalidad. Es muy común usar códigos ya creados que contienen funcionalidades que no serán usadas y diseños obsoletos.

Fase 3: Codificación

El cliente es una parte más del equipo de desarrollo; su presencia es indispensable en las distintas fases de XP. A la hora de codificar una historia de usuario su presencia es aún más necesaria. No olvidemos que los clientes son los que crean las historias de usuario y negocian los tiempos en los que serán implementadas.

Antes del desarrollo de cada historia de usuario el cliente debe especificar detalladamente lo que ésta hará y también tendrá que estar presente cuando se realicen las pruebas que verifiquen que la historia implementada cumple la funcionalidad especificada.

La optimización del código siempre se debe dejar para el final. Hay que hacer que funcione y que sea correcto, más tarde se puede optimizar. XP afirma que la mayoría de los proyectos que necesiten más tiempo extra que el planificado para ser finalizados no podrán ser terminados a tiempo se plantea la solución de realizar un nuevo "Release plan" para concretar los nuevos tiempos de publicación y de velocidad del proyecto.

Fase 4: Pruebas

Uno de los pilares de la metodología XP es el uso de pruebas para comprobar el funcionamiento de los códigos que vayamos implementando. Las pruebas permiten verificar que un cambio en la estructura de un código no tiene por qué cambiar su funcionamiento.

Además las pruebas sirven para evaluar las distintas tareas en las que ha sido dividida una historia de usuario. Para asegurar el funcionamiento final de una determinada historia de usuario se deben crear "Test de aceptación"; estos test son creados y usados por los clientes para comprobar que las distintas historias de usuario cumplen su cometido.⁽²¹⁾

c. MICROSOFT SOLUTION FRAMEWORK (MSF)

Microsoft Developer Network ⁽²²⁾, define a la metodología MSF como del tipo de metodologías ágiles, que está enfocada a dirigir proyectos o soluciones de innovación, en ella no se detalla ni se hace énfasis de la organización ni el tamaño del equipo de desarrollo, está más bien centrada en la gestión y administración del proyecto para lograr el impacto deseado.

Involucra indudablemente la calidad ya que prevé liberar una solución si ésta aún tiene fallos o desperfectos para ello propone seleccionar un grupo de prueba piloto el cual es

una VERSION BETA y cumplido un tiempo de prueba ya es liberada la versión formal o VERSION ALFA en la cual está garantizada la calidad. Microsoft Solution Framework (MSF), es un modelo de procesos que combina dos modelos muy comunes en proyectos de desarrollo, el modelo en cascada y el modelo en espiral.

Consta de cinco etapas, en las cuales se generan entregables concretos que ayudan a resolver los requerimientos del cliente de una manera objetiva. Estas cinco etapas de MSF son (22):

VISIÓN: fase en que se debe realizar un estudio de lo que pretendemos en el futuro haga nuestra aplicación o nuestro proyecto, por lo que se realizara un documento de estrategia y alcance en donde debe quedar pactada la necesidad de funcionalidad y servicio que se debe contar en la solución. Debemos crear los equipos de trabajo junto con el plan de trabajo, para asegurar el éxito del proyecto es importante tener en cuenta el de riesgos y plan de contingencia.

PLANIFICACIÓN: en esta fase básicamente se concreta claramente cómo va a estar estructurada nuestra solución para ello se crea un documento de planificación y diseño de la arquitectura, diseñar las pruebas de concepto donde se plantean los diferentes escenarios para probar la validez de los criterios utilizados para el diseño, debemos establecer métricas.

DESARROLLO: en la etapa de desarrollo debemos codificar las aplicaciones y realizar las configuraciones necesarias para que la solución funcione, es importante hacer pruebas continuamente así se verifica la calidad del producto continuamente a lo largo del desarrollo y no únicamente al final del proceso.

ESTABILIZACIÓN: fase en que se debe seleccionar el entorno de prueba piloto e identificar las deficiencias con un grupo reducido de usuarios para corregirlas y así en el futuro no tener problemas cuando se use la solución por todos, ocasionalmente a esta etapa se le llama BETA, debemos crear un plan de gestión de incidencias, realizar una revisión de documentación final de la arquitectura y elaboración de plan de despliegue o implementación.

DESPLIEGUE O IMPLEMENTACIÓN: en esta etapa final ya se ha comprobado la calidad de la solución por lo cual está lista para ser publicada, en este sentido debemos liberar la solución y crear un registro de mejoras y sugerencias, revisar las guías y manuales y entrega de proyecto final.

Para entender el modelo MSF, es importante entender a claridad el modelo en cascada, y el modelo en espiral, así como las ventajas y desventajas de estos, luego se podrá visualizar de qué manera MSF juna las mejores prácticas de ambos para dar a como resultado un modelo altamente práctico.⁽²²⁾

d. MODELO EN CASCADA

Se basa en que se avanza a la siguiente fase del proyecto, siempre y cuando la etapa anterior del proyecto está finalizada por completo. Así mismo una vez que se ha pasado a la siguiente etapa del proyecto, no se admite retroceso alguno a la etapa anterior.

Esto funciona bien para proyectos donde claramente se pueden delinear los requerimientos (no modificables) del proyecto en una fase inicial. Estas etapas cerradas, facilitan el planeamiento y asignación de recursos para cada etapa.

Sin embargo se complica si los requerimientos tomados en un inicio se modifican en una etapa posterior del proyecto.⁽²³⁾

e. MODELO EN ESPIRAL

No define etapas claras dentro del desarrollo del proyecto, sin embargo, está abierto a cambios por el cliente en cualquier momento. Es decir, se trata de un desarrollo a la par con el levantamiento de requerimientos, ya que el cliente provee retroalimentación en cualquier etapa del proyecto. Esto puede ser muy efectivo cuando se necesita un desarrollo rápido en aplicaciones sumamente pequeñas, pero deja abierta la posibilidad de catástrofes en el proyecto por dos razones:

La primera, la tarea de por sí durísima, de medir los recursos necesarios para el desarrollo, se complica aún más.

La segunda, al no haber puntos de control, la tarea de desarrollo, aunque se esté trabajando con un equipo pequeño, se puede volver algo caótico. ⁽²³⁾

2.5. LENGUAJE DE MODELAMIENTO UNIFICADO (UML)

El lenguaje de Modelamiento Unificado (UML – Unified Modeling Language), es definido por Krall, como un estándar que se ha adoptado a nivel internacional por numerosos organismos y empresas para crear esquemas, diagramas y documentación relativa a los desarrollo de software y programas informáticos).

UML entrega una forma de modelar cosas conceptuales como lo son los procesos de negocio y funciones de sistema, además de cosas concretas como lo son escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables.

Los principales beneficios de UML son:

- Mejores tiempos totales de desarrollo (de 50 % o más).
- Modelar sistemas utilizando conceptos orientados a objetos.
- Establecer conceptos y artefactos ejecutables.
- Encaminar el desarrollo del escalamiento en sistemas complejos de misión crítica.
- Crear un lenguaje de modelado utilizado tanto por humanos como por máquinas.
- Mejor soporte a la planeación y al control de proyectos.
- Alta reutilización y minimización de costos.

UML proporciona un conjunto estandarizado de herramientas para documentar el y diseño de un sistema de software. El conjunto de herramientas de UML incluye diagramas que permiten a las personas visualizar la construcción de un sistema de información, similar a la forma en que un conjunto de planos permite a las personas visualizar la construcción de un edificio. Ya sea que se está trabajando independientemente o como parte de un equipo grande de desarrollo de sistemas, la documentación que crea con UML proporciona un medio eficaz de comunicación entre el equipo de desarrollo y el equipo de negocios en un proyecto.

Los diagramas de UML se han convertido en una herramienta de gran aceptación y uso durante el diseño de software. Se aplica a multitud de diferentes tipos de sistemas, dominios y métodos o procesos:

Como lenguaje de propósito general, se enfoca en el corazón de un conjunto de conceptos para la adquisición, compartición y utilización de conocimientos emparejados con mecanismos de extensión.

Como un lenguaje para modelamiento ampliamente aplicable, puede ser aplicado a diferentes tipos de sistemas (software y no software), dominios (negocios versus software) y métodos o procesos.

Como un lenguaje de modelamiento soportable por herramientas, las herramientas ya están disponibles para soportar la aplicación del lenguaje para especificar, visualizar, construir y documentar sistemas.

Como un lenguaje para modelamiento industrialmente estandarizado, no es un lenguaje cerrado, propiedad de alguien, sino más bien, un lenguaje abierto y totalmente extensible reconocido por la industria

Los tres elementos que forman el modelo conceptual de UML, son: los bloques básicos de construcción del lenguaje, las reglas que se aplican sobre esos bloques y los mecanismos comunes de UML.

Existen tres tipos de bloques de construcción:

- Elementos: son los modelos UML (clases, casos de uso, estados, anotaciones).
- Relaciones: ligan elementos entre sí, establecen la forma en que interactúan.
- Diagramas: representaciones gráficas de un grupo de elementos y sus relaciones. UML maneja tres tipos de relaciones. Una relación es una conexión semántica entre elementos del modelo. En el UML se definen relaciones de asociación, generalización

y dependencia. La agregación y composición son casos especiales de relaciones de asociación. Gómez⁽²⁴⁾.

TIPOS DE RELACIONES

Asociación: Una asociación es una relación estructural entre varios elementos. Una relación de asociación implica que los objetos de los distintos elementos de la relación están conectados entre sí y se pueden comunicar. Una relación de asociación se representa gráficamente con una línea continua entre los elementos relacionados.

Generalización: Una generalización es una relación de especialización. Los elementos especializados (hijos) son elementos que derivan de un elemento general (padre). Los elementos hijos mantienen la estructura y el funcionamiento del elemento padre pero de una forma más especializada. Su representación gráfica es la de una línea dirigida con punta triangular.

Composición: Es un tipo de agregación donde la relación de posesión es tan fuerte como para marcar otro tipo de relación. Las clases en UML tienen un tiempo de vida determinado, en las relaciones de composición, el tiempo de vida de la clase que es parte del todo (o agregado) vienen determinado por el tiempo de vida de la clase que representa el todo, por lo tanto es equivalente a un atributo, aunque no lo es porque es una clase y puede funcionar como tal en otros casos.

Dependencia: Una dependencia es una relación entre dos elementos (un elemento utiliza a otro). Una relación de dependencia entre dos elementos implica que los cambios que se produzcan en un elemento pueden afectar al otro pero no necesariamente a la inversa. Las dependencias se representan con una línea discontinua.

Entre los diagramas UML se pueden agrupar en ⁽²⁴⁾:

a. DIAGRAMA ESTRUCTURAL ⁽²⁵⁾

1. Diagrama de Clases

Muestra el conjunto de clases que participan o forman parte de un sistema, junto con las relaciones que existen entre dichas clases. Muestra de una manera estática la estructura de la información que maneja el sistema y la visibilidad que tiene cada una de las clases, dada por sus relaciones con los demás en el modelo.

Componentes:

Clases: una clase se representa por un rectángulo en el cual se escriben tres secciones: en la sección superior se coloca el nombre de la clase; en la intermedia se presentan los atributos que caracterizan a la clase y en la sección inferior se listan sus métodos u operaciones.

Interfaces, son clases abstractas, lo que significa que no es posible crear instancias directamente a partir de ellas. Pueden contener operaciones, pero no atributos. Las clases pueden heredar de las interfaces (a través de una asociación de realización) y de estos diagramas sí es posible crear instancias.

Enumeraciones, son simples listas de valores. Un ejemplo típico de esto sería una enumeración de los días de la semana. Las opciones de una enumeración se llaman «literales de enumeración». Al igual que los tipos de datos, no pueden relacionarse con las clases, pero las clases sí pueden hacerlo con ellos.

Tipo de datos, son primitivas construidas normalmente en algunos lenguajes de programación. Algunos ejemplos comunes son los enteros y los booleanos. No pueden tener relación con clases, pero las clases sí pueden relacionarse con ellos.

Paquetes, en lenguajes de programación, representan un espacio de nombres en un diagrama se emplean para representar partes del sistema que contienen más de una clase, incluso cientos de ellas.

2. Diagrama de Componentes

Los Diagramas de Componentes muestran los componentes del software (ya sea las tecnologías que lo forman como Kparts, componentes CORBA, Java Beans o simplemente secciones del sistema claramente distintas) y los artilugios de que está compuesto como los archivos de código fuente, las librerías o las tablas de una base de datos.

Los Diagramas de Componentes prevalecen en el campo de la arquitectura de software pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema. Debido a que los Diagramas de Componentes son más parecidos a los Diagramas de Casos de Usos, éstos son utilizados para modelar la vista estática y dinámica de un sistema. Muestra la organización y las dependencias entre un conjunto de componentes. Los componentes pueden tener interfaces (es decir clases abstractas con operaciones) que permiten asociaciones entre componentes. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes. Cada diagrama describe un apartado del sistema.

DIAGRAMA DE ESTRUCTURA COMPUESTA

Un Diagrama de Estructura Compuesta es un tipo de diagrama de estructura estática en el Lenguaje de Modelado Unificado (UML), que muestra la estructura interna de una clase y las colaboraciones que esta estructura hace posibles. Esto puede incluir partes internas,

puertas mediante las cuales, las partes interactúan con cada una de las otras o mediante las cuales, instancias de la clase interactúan con las partes y con el mundo exterior, y conectores entre partes o puertas.

Una estructura compuesta es un conjunto de elementos interconectados que colaboran en tiempo de ejecución para lograr algún propósito. Cada elemento tiene algún rol definido en la colaboración.

3. Diagrama de Paquetes

En el Lenguaje Unificado de Modelado, un Diagrama de Paquetes muestra cómo un sistema está dividido en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones. Dado que normalmente un paquete está pensado como un directorio, los Diagramas de Paquetes suministran una descomposición de la jerarquía lógica de un sistema.

Los paquetes están normalmente organizados para maximizar la coherencia interna dentro de cada paquete y minimizar el acoplamiento externo entre los paquetes. Con estas líneas maestras sobre la mesa, los paquetes son buenos elementos de gestión.

Cada paquete puede asignarse a un individuo o a un equipo, y las dependencias entre ellos pueden indicar el orden de desarrollo requerido.

4. Diagrama de Despliegue

Muestran la disposición física de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. Estos diagramas muestran la configuración en funcionamiento del sistema, incluyendo su hardware y su software.

Un Diagrama de Despliegue modela la arquitectura en tiempo de ejecución de un sistema mostrando la configuración de los elementos de hardware y mostrando cómo los elementos y artefactos del software se trazan en esos nodos.

5. Diagrama de Objetos

Los Diagramas de Objetos son utilizados durante el proceso de y Diseño de los sistemas informáticos en la metodología UML. Se puede considerar un caso especial de un Diagrama de Clases en el que se muestran instancias específicas de clases (objetos) en un momento particular del sistema.

Los Diagramas de Objetos utilizan un subconjunto de los elementos de un Diagrama de Clase. Los Diagramas de Objetos no muestran la multiplicidad ni los roles, aunque su notación es similar a los diagramas de clase.

Una diferencia con los Diagramas de Clase es que el compartimiento de arriba va en la forma Nombre de objeto: Nombre de clase. Por ejemplo, Miguel: Persona.

b. DIAGRAMA COMPORTAMIENTOS ⁽²⁵⁾

1. Diagrama de Actividad

Es un diagrama de flujo del proceso multi-proposito que se usa para modelar el comportamiento del sistema. Es importante recalcar que aunque un Diagrama de Actividad es muy similar en definición a un Diagrama de Flujo, estos no son lo mismo.

Se pudiera considerar que un Diagrama de Actividad describe el problema, mientras un Diagrama de Flujo describe la solución.

En el Lenguaje de Modelado Unificado, un Diagrama de Actividades representa los flujos de trabajo paso a paso de negocio y operacionales de los componentes en un sistema. Un Diagrama de Actividades muestra el flujo de control general.

2. Diagrama de Casos de Uso

Muestran la relación entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa. En el Diagrama de Casos de Uso se representa también el sistema como una caja rectangular con el nombre en su interior. Los casos de uso están en el interior de la caja del sistema, y los actores fuera, y cada actor está unido a los casos de uso en los que participa mediante una línea.

Los elementos de un Caso de Uso, son:

Elementos Diagrama de Casos de Uso en UML

Casos de uso: es una descripción de la secuencia de interacciones que se producen entre un actor y el sistema, cuando el actor usa el sistema para llevar a cabo una tarea específica. Expresa una unidad coherente de funcionalidad, y se representa en el Diagrama de Caso de Uso mediante una elipse con el nombre del caso de uso en su interior. El nombre del caso de uso debe reflejar la tarea específica que el actor desea llevar a cabo usando el sistema.

Actores: un actor es algo como un comportamiento, como una persona (identificada por un rol), un sistema informatizado u organización, y que realiza algún tipo de interacción con el sistema. Se representa mediante una figura humana dibujada con palotes.

Relaciones entre Casos de Uso: un Caso de Uso, en principio, debería describir una tarea que tiene un sentido completo para el usuario. Sin embargo, hay ocasiones en la que es

útil describir una interacción con un alcance menor como caso de uso. La razón para utilizar estos casos de uso no completos en algunos casos, es mejorar la comunicación con el equipo de desarrollo, el manejo de la documentación de casos de uso.

3. Diagrama de Estado

En UML, un Diagrama de Estados es un diagrama utilizado para identificar cada una de las rutas o caminos que puede tomar un flujo de información luego de ejecutarse cada proceso. Permite identificar bajo qué argumentos se ejecuta cada uno de los procesos y en qué momento podrían tener una variación. El Diagrama de Estados permite visualizar de una forma secuencial la ejecución de cada uno de los procesos.

Los Diagramas de Estado muestran los diferentes estados de un objeto durante su vida, y los estímulos que provocan los cambios de estado en un objeto.

Los Diagramas de Estado ven a los objetos como máquinas de estado o autómatas finitos que pueden estar en un conjunto de estados finitos y que pueden cambiar su estado a través de un estímulo perteneciente a un conjunto finito. Por ejemplo, un objeto de tipo NetServer puede tener durante su vida uno de los siguientes estados:

- Listo.
- Escuchando.
- Trabajando.
- Detenido.

Y los eventos que pueden producir que el objeto cambie de estado son:

Se crea el objeto.

- El objeto recibe un mensaje de escucha.
- Un cliente solicita una conexión a través de la red.

- Un cliente finaliza una solicitud.
- La solicitud se ejecuta y ser termina.
- El objeto recibe un mensaje de detención.

4. Diagrama de Secuencia

Muestra una interacción ordenada según la secuencia temporal del evento. En particular, muestra los objetos participantes en la interacción y los mensajes que intercambian ordenados según su secuencia de tiempo.

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso. Mientras que el diagrama de casos de uso permite el modelado de una vista business del escenario, el diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario y mensajes intercambiados entre los objetos.

c. DIAGRAMA DE COMUNICACIÓN ⁽²⁵⁾

En el Lenguaje Unificado de Modelado (UML) 2.0, un Diagrama de Comunicación es una versión simplificada del diagrama de colaboración de la versión de UML 1.x. Un Diagrama de Comunicación modela las interacciones entre objetos o partes en términos de mensajes en secuencia. Los Diagramas de Comunicación representan una combinación de información tomada desde el diagrama de clases, secuencia, y diagrama de casos de uso describiendo tanto la estructura estática como el comportamiento dinámico de un sistema.

Los Diagramas de Comunicación y de secuencia describen información similar, y con ciertas transformaciones, pueden ser transformados unos en otros sin dificultad.

Para mantener el orden de los mensajes en un diagrama de comunicación, los mensajes son etiquetados con un número cronológico y colocado cerca del enlace por el cual se desplaza el mensaje.

Leer un Diagrama de Comunicación conlleva comenzar en el mensaje 1.0, y seguir los mensajes desde un objeto hasta el siguiente, sucesivamente.

Diagrama de Interacción

El Diagrama de Interacción, representa la forma en como un Cliente (Actor) u Objetos (Clases) se comunican entre sí en petición a un evento. Esto implica recorrer toda la secuencia de llamadas, de donde se obtienen las responsabilidades claramente. Dicho diagrama puede ser obtenido de dos partes, desde el Diagrama Estático de Clases o el de Casos de Uso (son diferentes).

Los componentes de un diagrama de interacción son:

Un Objeto o Actor: el rectángulo representa una instancia de un Objeto en particular, y la línea punteada representa las llamadas a métodos del objeto.

Mensaje de un objeto a otro objeto: se representa por una flecha entre un objeto y otro, representa la llamada de un método (operación) de un objeto en particular.

Mensaje de un objeto a sí mismo: no solo llamadas a métodos de objetos externos pueden realizarse, también es posible visualizar llamadas a métodos desde el mismo objeto en estudio.

d. DIAGRAMA DE TIEMPOS⁽²⁵⁾

Un Diagrama de Tiempos o cronograma es una gráfica de formas de onda digitales que muestra la relación temporal entre varias señales, y cómo varía cada señal en relación a las demás.

Un cronograma puede contener cualquier número de señales relacionadas entre sí. Examinando un diagrama de tiempos, se puede determinar los estados, nivel alto o nivel bajo, de cada una de las señales en cualquier instante de tiempo especificado, y el instante exacto en que cualquiera de las señales cambia de estado con respecto a las restantes.

2.6. LENGUAJE DE PROGRAMACION ⁽²⁶⁾

Un lenguaje de programación es un idioma artificial diseñado para expresar computaciones que pueden ser llevadas a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana. Está formado de un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Al proceso por el cual se escribe, se prueba, se depura, se compila y se mantiene el código fuente de un programa informático se le llama programación.

También la palabra programación se define como el proceso de creación de un programa de computadora, mediante la aplicación de procedimientos lógicos, a través de los siguientes pasos:

- El desarrollo lógico del programa para resolver un problema en particular.
- Escritura de la lógica del programa empleando un lenguaje de programación específico (codificación del programa)

- Ensamblaje o compilación del programa hasta convertirlo en lenguaje de máquina.
- Prueba y depuración del programa.
- Desarrollo de la documentación

a. PHP ⁽²⁵⁾

PHP Hypertext Pre-processor. Lenguaje de programación usado generalmente en la creación de contenidos para sitios web. Es un lenguaje interpretado especialmente usado para crear contenido dinámico web y aplicaciones para servidores, aunque también es posible crear aplicaciones gráficas utilizando la biblioteca GTK+. Generalmente los scripts en PHP se embeben en otros códigos como HTML, ampliando las posibilidades del diseñador de páginas web enormemente. La interpretación y ejecución de los scripts PHP se hacen en el servidor, el cliente (un navegador que pide una página web) sólo recibe el resultado de la ejecución y jamás ve el código PHP. Permite la conexión a todo tipo de servidores de base de datos como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite. PHP es una alternativa a otros sistemas como el ASP.NET/C#/VB.NET de Microsoft o a ColdFusion de Macromedia, a JSP/Java de Sun Microsystems, y a CGI/Perl. La ventaja con los de Microsoft o Macromedia es que es totalmente gratuito, no hay que pagar licencias.

b. ASP.NET ⁽²⁷⁾

ASP.NET es un modelo de desarrollo Web unificado que incluye los servicios necesarios para crear aplicaciones Web empresariales con el código mínimo. ASP.NET forma parte de .NET Framework y al codificar las aplicaciones ASP.NET tiene acceso a las clases en .NET Framework. El código de las aplicaciones puede escribirse en cualquier lenguaje

compatible con el Common Language Runtime (CLR), entre ellos Microsoft Visual Basic, C#, JScript .NET y J#. Estos lenguajes permiten desarrollar aplicaciones ASP.NET que se benefician del Common Language Runtime, seguridad de tipos, herencia, etc.

ASP.NET incluye:

- Marco de trabajo de página y controles
- Compilador de ASP.NET
- Infraestructura de seguridad
- Funciones de administración de estado
- Configuración de la aplicación
- Supervisión de estado y características de rendimiento
- Capacidad de depuración
- Marco de trabajo de servicios Web XML
- Entorno de host extensible y administración del ciclo de vida de las aplicaciones
- Entorno de diseñador extensible

MARCO DE TRABAJO DE PÁGINAS Y CONTROLES

El marco de trabajo de páginas y controles ASP.NET es un marco de trabajo de programación que se ejecuta en un servidor Web para generar y representar de forma dinámica páginas Web ASP.NET. Las páginas Web ASP.NET se pueden solicitar a cualquier explorador o dispositivo del cliente y ASP.NET representa el marcado (como HTML) al explorador que realizó la solicitud. Como norma, puede utilizar la misma

página para varios exploradores, porque ASP.NET representa el mercado adecuado para el explorador que realiza la solicitud. Sin embargo, puede diseñar una página Web ASP.NET para ejecutarse en un explorador determinado, como Microsoft Internet Explorer 6, y aprovechar así todas las características de ese explorador. ASP.NET es compatible con los controles móviles de los dispositivos preparados para trabajar en Web como teléfonos celulares, PC portátiles y asistentes digitales personales (PDA).

Las páginas Web ASP.NET están completamente orientadas a objetos. En las páginas Web ASP.NET se puede trabajar con elementos HTML que usen propiedades, métodos y eventos. El marco de trabajo de páginas ASP.NET quita los detalles de implementación relacionados con la separación de cliente y servidor inherente a las aplicaciones Web presentando un modelo unificado que responde a los eventos de los clientes en el código que se ejecuta en el servidor. El marco de trabajo también mantiene automáticamente el estado de la página y de los controles que contenga durante el ciclo vital de procesamiento de la página. Para obtener más información, vea Información general sobre páginas Web ASP.NET.

El marco de trabajo de páginas y controles ASP.NET también permite encapsular la funcionalidad común de la interfaz de usuario en controles fáciles de usar y reutilizables. Los controles se escriben una vez, se pueden utilizar en varias páginas y se integran en la página Web ASP.NET en la que se colocan durante la representación.

El marco de trabajo de páginas y controles ASP.NET también proporciona funciones para controlar la apariencia y el funcionamiento general de los sitios Web a través de temas y

máscaras. Se pueden definir temas y máscaras y, a continuación, aplicarlos en las páginas o controles. Para obtener más información, vea Información general sobre temas y máscaras de ASP.NET.

Además de los temas, es posible definir páginas principales que se crean para conseguir un diseño coherente en las páginas de la aplicación. Una página principal única define el diseño y el comportamiento estándar deseados para todas las páginas (o un grupo de páginas) de la aplicación. A continuación, se pueden crear páginas de contenido individuales con el contenido específico de la página que se desee mostrar. Cuando los usuarios solicitan las páginas de contenido, las combinan con la página principal con el fin de generar un resultado que combine el diseño de la página principal con el de la página de contenido. Para obtener más información, vea Información general sobre las páginas principales ASP.NET.

COMPILADOR DE ASP.NET

Compila todo el código de ASP.NET, lo que permite el establecimiento inflexible de tipos, las optimizaciones de rendimiento y el enlace en tiempo de compilación, entre otras ventajas. Una vez que se ha compilado el código, el Common Language Runtime compila una vez más código de ASP.NET en código nativo, lo que permite un mayor rendimiento.

ASP.NET incluye un compilador que compilará todos los componentes de la aplicación, incluidos las páginas y los controles, en un ensamblado que el entorno de host de

ASP.NET puede utilizar a continuación para atender las solicitudes del usuario. Para obtener más información, vea Información general sobre la compilación de ASP.NET.

INFRAESTRUCTURA DE SEGURIDAD

Además de las características de seguridad de .NET, ASP.NET proporciona una infraestructura de seguridad avanzada para autenticar y autorizar el acceso de los usuarios y realizar otras tareas relacionadas con la seguridad. Puede autenticar usuarios con la autenticación de Windows suministrada por IIS o puede administrar la autenticación con su propia base de datos de usuario utilizando la autenticación mediante formularios ASP.NET y la suscripción ASP.NET. Además, puede administrar la autorización a las capacidades e información de su aplicación Web mediante los grupos de Windows o su propia base de datos de funciones personalizada utilizando las funciones de ASP.NET.

c. JSP⁽²⁸⁾

Java Server Pages (JSP) es una tecnología que nos permite mezclar HTML estático con HTML generado dinámicamente. Muchas páginas Web que están construidas con programas CGI son casi estáticas, con la parte dinámica limitada a muy pocas localizaciones. Pero muchas variaciones CGI, incluyendo los servlets, hacen que generemos la página completa mediante nuestro programa, incluso aunque la mayoría de ella sea siempre lo mismo. JSP nos permite crear dos partes de forma separada.

¿CUÁLES SON LAS VENTAJAS DE JSP?

- **Contra Active Server Pages (ASP).** ASP es una tecnología similar de Microsoft. Las ventajas de JSP están duplicadas. Primero, la parte dinámica está escrita en Java, no en Visual Basic, otro lenguaje específico de MS, por eso es mucho más poderosa y fácil de usar. Segundo, es portable a otros sistemas operativos y servidores Web.
- **Contra los Servlets.** JSP no nos da nada que no pudiéramos en principio hacer con un servlet. Pero es mucho más conveniente escribir (y modificar!) HTML normal que tener que hacer un billón de sentencias `println` que generen HTML. Además, separando el formato del contenido podemos poner diferentes personas en diferentes tareas: nuestros expertos en diseño de páginas Web pueden construir el HTML, dejando espacio para que nuestros programadores de servlets inserten el contenido dinámico.
- **Contra Server-Side Includes (SSI).** SSI es una tecnología ampliamente soportada que incluye piezas definidas externamente dentro de una página Web estática. JSP es mejor porque nos permite usar servlets en vez de un programa separado para generar las partes dinámicas. Además, SSI, realmente está diseñado para inclusiones sencillas, no para programas "reales" que usen formularios de datos, hagan conexiones a bases de datos, etc.
- **Contra JavaScript.** JavaScript puede generar HTML dinámicamente en el cliente. Esta es una capacidad útil, pero sólo maneja situaciones donde la información dinámica está basada en el entorno del cliente. Con la excepción de las cookies, el HTTP y el envío de formularios no están disponibles con JavaScript. Y, como se ejecuta en el cliente, JavaScript no puede acceder a los recursos en el lado del servidor, como bases de datos, catálogos, información de precios, etc.

d. LENGUAJE PYTHON ⁽²⁹⁾

Es un lenguaje de programación creado en el año 1990 por Guido van Rossum, es el sucesor del lenguaje de programación ABC. Python es comparado habitualmente con Perl. Los usuarios lo consideran como un lenguaje más limpio para programar. Permite la creación de todo tipo de programas incluyendo los sitios web.

Su código no necesita ser compilado, por lo que se llama que el código es interpretado. Es un lenguaje de programación multiparadigma, lo cual fuerza a que los programadores adopten por un estilo de programación particular:

- ✓ Programación orientada a objetos.
- ✓ Programación estructurada.
- ✓ Programación funcional.
- ✓ Programación orientada a aspectos

Ventajas:

- ✓ Libre y fuente abierta.
- ✓ Lenguaje de propósito general.
- ✓ Gran cantidad de funciones y librerías.
- ✓ Sencillo y rápido de programar.
- ✓ Multiplataforma.
- ✓ Licencia de código abierto (Opensource).
- ✓ Orientado a Objetos.
- ✓ Portable.

Desventajas:

- ✓ Lentitud por ser un lenguaje interpretado.

2.7. SISTEMA DE GESTIÓN DE BASE DE DATOS:

Álvarez G, menciona que un sistema de gestión de bases de datos (SGBD) es el conjunto de programas que permiten definir, manipular y utilizar la información que contienen las bases de datos, realizar todas las tareas de administración necesarias para mantenerlas operativas, mantener su integridad, confidencialidad y seguridad. Una BD nunca se accede o manipula directamente sino a través del SGBD. Se puede considerar al SGBD como el interfaz entre el usuario y la BD.

El funcionamiento del SGBD está muy interrelacionado con el del sistema operativo, especialmente con el sistema de comunicaciones. El SGBD utilizará las facilidades del sistema de comunicaciones para recibir las peticiones del usuario (que puede estar utilizando un terminal físicamente remoto) y para devolverle los resultados. ⁽³⁰⁾

2.7.1. Funciones de un SGBD:

Mendoza J. Un SGBD debe proporcionar un amplio surtido de funcionalidades para poder cumplir adecuadamente su misión. Normalmente se clasifican en definición, manipulación y utilización.

- **Función de definición:** Permite describir los elementos de datos, sus estructuras, sus interrelaciones y sus validaciones a nivel externo, lógico e interno. Esta función es realizada por una parte del SGBD denominada lenguaje de definición de datos (LDD o DDL, Data Definición Lenguaje).

- Función de manipulación: Permite buscar, añadir, suprimir y modificar los datos de la BD. Esta función es realizada por una parte del SGBD denominada lenguaje de manipulación de datos (LMD o DML, Data Manipulación Lenguaje).
- Función de utilización: Incluye otras funcionalidades tales como: modificar la capacidad de los registros, cargar archivos, realizar copias de seguridad, arranque, protección frente a accesos no autorizados, gestión de la concurrencia, estadísticas de utilización, etc. ⁽³¹⁾.

a. MYSQL⁽³²⁾

MySQL es un sistema de administración de bases de datos (*Database Management System, DBMS*) para bases de datos relacionales. Así, MySQL no es más que una aplicación que permite gestionar archivos llamados de bases de datos.

Existen muchos tipos de bases de datos, desde un simple archivo hasta sistemas relacionales orientados a objetos. MySQL, como base de datos relacional, utiliza múltiples tablas para almacenar y organizar la información. MySQL fue escrito en C y C++ y destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más utilizados como PHP, Perl y Java y su integración en distintos sistemas operativos. También es muy destacable, la condición de open source de MySQL, que hace que su utilización sea gratuita e incluso se pueda modificar con total libertad, pudiendo descargar su código fuente. Esto ha favorecido muy positivamente en

su desarrollo y continuas actualizaciones, para hacer de MySQL una de las herramientas más utilizadas por los programadores orientados a Internet.

b. MICROSOFT SQL SERVER⁽³³⁾

Microsoft SQL Server es un sistema de manejo de bases de datos del modelo relacional, desarrollado por la empresa Microsoft. El lenguaje de desarrollo utilizado (por línea de comandos o mediante la interfaz gráfica de Management Studio) es Transact-SQL (TSQL), una implementación del estándar ANSI del lenguaje SQL, utilizado para manipular y recuperar datos (DML), crear tablas y definir relaciones entre ellas (DDL). Dentro de los competidores más destacados de SQL Server están: Oracle, MariaDB, MySQL, PostgreSQL. SQL Server solo está disponible para sistemas operativos Windows de Microsoft. Puede ser configurado para utilizar varias instancias en el mismo servidor físico, la primera instalación lleva generalmente el nombre del servidor, y las siguientes - nombres específicos (con un guion invertido entre el nombre del servidor y el nombre de la instalación).

c. ORACLE⁽³⁴⁾

Oracle Database es un sistema de gestión de base de datos de tipo objeto-relacional (ORDBMS, por el acrónimo en inglés de *Object-Relational Data Base Management System*), desarrollado por Oracle Corporation. Se considera a *Oracle Database* como uno de los sistemas de bases de datos más completos, destacando: soporte de transacciones, estabilidad, escalabilidad, y soporte multiplataforma.

Su dominio en el mercado de servidores empresariales había sido casi total hasta que recientemente tiene la competencia del Microsoft SQL Server y de la oferta de otros RDBMS con licencia libre como PostgreSQL, MySQL o Firebird.

Las últimas versiones de Oracle han sido certificadas para poder trabajar bajo GNU/Linux.

d. POSTGRE SQL⁽³⁵⁾

PostgreSQL es un Sistema de gestión de bases de datos relaciona lo orientado a objetos y libre, publicado bajo la licencia PosgreSQL, similar a la BSD o la MIT. Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre y/o apoyados por organizaciones comerciales. Dicha comunidad es denominada el PGDG (*PostgreSQL Global Development Group*).

III. METODOLOGIA

3.1. DISEÑO DE LA INVESTIGACION

La investigación es de tipo descriptivo, ya que se describirá y documentará la situación problemática de la empresa para lograr conocer de manera más completa el entorno donde se desarrollará la investigación. Así mismo la investigación realizada es de tipo aplicativo, ya que gracias a la información obtenida se determinó con mayor exactitud deficiencia en los procesos de seguimiento de expedientes. Finalmente es una investigación no experimental, porque se realiza sin manipular deliberadamente las variables, es decir se trata de una investigación donde no hacemos variar intencionadamente las variables independientes, lo que se hace es observar los fenómenos, tal y como se dan en su contexto natural, para después analizarlos.

3.2. Población y muestra

Diseño no experimental, de tipo descriptivo, de corte transversal.

Donde:

M = Muestra O = Observación

M => O

Población y muestra:

Población Muestral: Se tendrá en cuenta como población muestra a todos los trabajadores del Consultorio Jurídico.

3.3. DEFINICION Y OPERACIONALIZACIÓN DE LAS VARIABLES:

VARIABLE	CONCEPTUALIZACIÓN	DIMENSIONES	INDICADOR (ES)	ESCALA MEDICIÓN	DEFINICIÓN OPERACIONAL
Diseñar	<p>Sistema de Gestión, permite acortar distancias físicas entre las diferentes dependencias</p>	<p>Sistema de Gestión</p>	<p>Base de datos</p> <p>Interfaces</p> <p>Tratamiento de la Información</p>	Ordinal	Alto
Implementar	<p>Control automatizado de la documentación del centro: permite garantizar la eficiente administración de los documentos internos o externos</p>	<p>Control de Operaciones</p>	<p>Recepción, envío y seguimiento de documentación</p>		Medio
	<p>Suministro de Información, automatizar las operaciones y suministrar información confiable de manera oportuna y segura.</p>	<p>Acceso a la información</p>	<p>Reportes y Consultas</p>		Bajo

3.4. Técnicas e instrumentos:

Se usará la entrevista para el del sistema de información, y un cuestionario para validar la funcionalidad del sistema. Como técnicas de recolección de datos se usará lo siguiente. ✓

Observación directa: Con esta técnica se podrá tener una percepción más clara del problema planteado, pudiendo observar la situación desde el enfoque de los usuarios, es decir, administrativos, para obtener un mejor entendimiento acerca de los problemas actuales y de la acción que se debe tomar para solucionar estos.

✓ **Entrevista no Estructurada:** La aplicación de entrevistas permitirá indagar de manera más profunda los detalles de la situación planteada.

✓ **Encuestas:** Esta técnica será aplicada de manera escrita, y con ella se recolectará información valiosa de parte de los usuarios para optimizar el proceso de trámite documentario.

3.5. Plan de :

Para identificar y determinar cómo es el proceso de trámite documentario de la institución, se aplicarán entrevistas y encuestas físicas los cuales serán tabulados en archivos Excel y con los puntajes totales obtenidos nos permitirán definir nuestros ampliamente los resultados. Asimismo para el proceso de diseño e implementación del sistema , se tendrá en cuenta la metodología RUP, que nos proporciona guías para poder conocer todo el camino a recorrer antes de empezar con la implementación asegurando así la calidad del producto final.

3.6. Matriz de Consistencia:

PROBLEMA	OBJETIVOS	HIPÓTESIS	INDICADORES	MÉTODOS	INSTRUMENTOS
Problema Principal	Objetivo General	Hipótesis General			
¿En qué medida el Sistema de Información influye en la Gestión de los Procesos Judiciales del Estudio Jurídico SALAS CALDERON & ASOCIADOS de la ciudad de Ica?	<p>Objetivo General</p> <p>Determinar la medida en que el Sistema de Información influye en la Gestión de los Procesos Judiciales del Estudio Jurídico SALAS CALDERON & ASOCIADOS de la ciudad de Ica.</p>	<p>Si se usa el Sistema de Información entonces se influye significativamente en la Gestión de los Procesos Judiciales del Estudio Jurídico SALAS CALDERON & ASOCIADOS de la ciudad de Ica.</p>	<p>Tiempo total en brindar información al cliente acerca de su proceso</p> <ul style="list-style-type: none"> - Tiempo de registro de un expediente - Tiempo de búsqueda de expediente - Tiempo de consulta de actividades judiciales por expediente - Tiempo de emisión de reporte de actividades judiciales por día - Porcentaje de expedientes omitidos con actividades judiciales programadas por día 	<p>Tipo de Investigación:</p> <p>Aplicada</p> <p>Tipo de Investigación:</p> <p>Pre-experimental</p> <p>Universo:</p> <p>Todos los trabajadores del estudio Jurídico Salas calderón de la Provincia de Ica.</p> <p>Población:</p> <p>Todos los procesos judiciales del Estudio Jurídico SALAS CALDERON & ASOCIADOS de la Provincia de Ica comprendidos en el período de Diciembre del 2015 a Enero del año 2016 que consta de 200 procesos.</p>	<p>Guía de Entrevistas</p> <p>Cuestionario</p> <p>Guía de apuntes de Observación</p> <p>Formato Digital</p>

3.7.- Principio Éticos

El Derecho es el sistema de regulación que establece las normas básicas de conductas sociales las cuales permiten una convivencia libre de conflictos. Sin embargo, no siempre es así. Cuando se produce una controversia entre partes y éstas no logran llegar a acuerdo alguno, será necesaria la intervención de un mediador y se constituirá un proceso, el cual tendrá como contenido dicha controversia. Los causales de dicha controversia tienen distintos orígenes, los principales son la discordancia de puntos de vista entre dos personas, el incumplimiento de alguna norma vigente, el reclamo por alguna violación de derechos de las personas, etc.

Entonces, un proceso judicial está definido por una discordancia entre dos partes: persona o institución que inicia el proceso y la persona o institución sobre la que se inicia el proceso (generalmente denominados demandante y demandado, respectivamente). Dicha causa será resuelta por un tercer integrante, llamado juez, que se encargará de discernir cuál de las dos partes se fundamenta con mayor base en el estado de derecho y partir de ello dará un fallo.

Los procesos judiciales están clasificados por materias, entre las principales se encuentran la civil, penal, laboral, de familia y comercial. Si bien es cierto cada materia tiene sus propias características que las distinguen del resto (por ejemplo, las partes se denominan demandante - demandado en un proceso civil y agraviado

- inculpado en un proceso penal), todas comparten ciertos principios comunes y se basan en la materia civil. Aunque un proceso puede presentar diversas variantes a lo largo de su desarrollo, en general su esencia se mantiene. No obstante, se debe tener en cuenta las etapas del proceso diferenciadas por cada uno de los órganos jurisdiccionales (aquellos que tienen la misión de administrar justicia, es decir, de juzgar y hacer ejecutar lo juzgado). Dichas etapas se denominan instancias y los órganos jurisdiccionales son los siguientes:

- Juzgados de Paz (No Letrados y Letrados): el menor nivel jerárquico en que se encuentra organizado el Poder Judicial.
- Juzgados de Primera Instancia (Especializados o Mixtos): el segundo nivel jerárquico en que se encuentra organizado el Poder Judicial del Perú.
- Cortes Superiores: el tercer nivel jerárquico en que se organiza el Poder Judicial. Sólo se encuentran bajo la autoridad de la Corte Suprema de la República y es, en la mayoría de procesos, el último organismo que conoce un proceso.
- Corte Suprema: máximo órgano jurisdiccional del Perú. Su competencia se extiende a todo el territorio del país.

Mediante las instancias se puede realizar una impugnación a un fallo (esto se describirá a detalle en la siguiente sección) y es la Corte Suprema la última instancia ante la cual se pueden apelar todos los procesos judiciales. Todo proceso tiene su equivalente expediente, el cual contiene las piezas escritas del proceso, agregadas sucesivamente y en orden de presentación, con las que se forma un solo cuerpo foliado con número y letras

IV. RESULTADOS

4.1. FACTIBILIDAD TÉCNICA

Como resultado de este estudio técnico se determinó que, en los actuales momentos, la empresa posee la infraestructura tecnológica en cuanto a Hardware necesario para el desarrollo del sistema de información.

De igual forma en cuanto a Software no contó con los principales softwares de desarrollo y se adquirió 3 de estos a versión express para realizar el piloto de prueba siendo el costo igual a cero, pero para la validez del uso se debe adquirir más adelante.

4.2. FACTIBILIDAD OPERATIVA

En cuanto a factibilidad operacional, el sistema propuesto si es factible ya que se cumplen los siguientes puntos:

- Los usuarios están muy interesados en la automatización de los procesos ya que les generará poder tener un mayor control sobre los procesos judiciales con los que están trabajando.
- Tanto los abogados como las secretarias han participado en el planeamiento del proyecto proporcionando información necesaria acerca de la gestión de los procesos judiciales.
- El acceso a la información será de una forma más rápida, trayendo consigo un beneficio tanto a los clientes como al Estudio Jurídico.
- El nivel de cultura de los empleados en informática está en un nivel intermedio por lo que facilitará la aplicación del presente proyecto.

- La necesidad y deseo de un cambio en el sistema actual, expresada por los usuarios y el personal involucrado con el mismo, llevó a la aceptación de un nuevo sistema, que, de una manera más sencilla y amigable, cubra todos sus requerimientos y expectativas.

4.3. FACTIBILIDAD ECONÓMICA

A continuación, se presenta un estudio que dio como resultado la factibilidad económica del desarrollo del Sistema de Información. Se determinaron los recursos para desarrollar y mantener en operación la aplicación, haciendo una evaluación donde se puso de manifiesto el equilibrio existente entre los costos del sistema y los beneficios que se derivaron de éste, lo cual permitió observar de una manera más precisa las bondades del sistema de información propuesto.

4.4. MODELADO DEL NEGOCIO

Durante esta fase, se establece el caso de negocio para el sistema y delimita el alcance del proyecto. Para lograr esto debe identificar todas las entidades externas con las cuales el sistema interactúe (los actores) y definir la naturaleza de esta interacción a un nivel alto, para esto se realizaron entrevistas con las diferentes entidades involucradas. A continuación, se muestran los resultados de esta fase.

4.4.1. REQUERIMIENTOS

REQUERIMIENTOS FUNCIONALES

- El abogado registrará un nuevo cliente en el sistema.
- El abogado podrá consultar y modificar los datos del cliente en el sistema
- El sistema permite la administración de los juzgados.

- El sistema permite la administración de los abogados y trabajadores respectivos del Estudio jurídico.
- El abogado registrará un nuevo proceso judicial con los datos y clientes respectivos en el sistema.
- El abogado o secretaria podrán registrar los documentos judiciales escaneados pertinentes en el sistema.
- El sistema permite la administración de los documentos judiciales de entrada y salida enlazándolos con una carpeta por proceso judicial.
- La secretaria podrá administrar las actividades judiciales de los procesos judiciales respectivos en el sistema.
- El abogado podrá consultar las actividades judiciales por día ya sean pendientes o ejecutadas en el sistema.
- El sistema mostrará al inicio de sesión las actividades judiciales pendientes

REQUERIMIENTOS NO FUNCIONALES

Usabilidad

Toda la interfaz deberá ser en formato Windows (Xp, Windows 7, Windows 8) acorde a las necesidades del Estudio Jurídico.

Todos los textos y mensajes de la interfaz gráfica se mostrarán en español.

Rendimiento

La aplicación debe ofrecer un buen desempeño del sistema ante una alta demanda acorde a los requerimientos funcionales y no funcionales de la solución; el tiempo promedio no mayor a 5 segundos sobre operaciones transaccionales.

El sistema soportará la administración de los tipos de usuarios pertinentes, así como la restricción de acceso a la información.

Licencia e instalación

No se requerirán licencias de usuario para el funcionamiento del software piloto en las computadoras de los usuarios, pero para la validez del uso se debe adquirir más adelante.

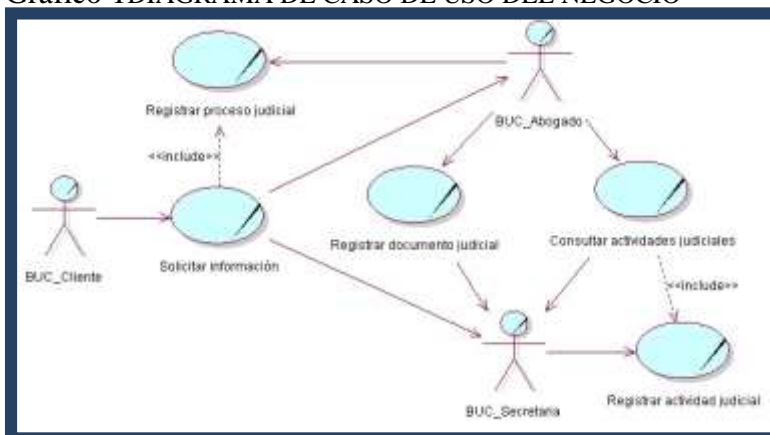
Restricciones de Diseño e implementación

- La aplicación se desarrollará bajo tecnología J2SE.
- La base de datos a utilizar será MySql.

4.4.2. DISEÑO

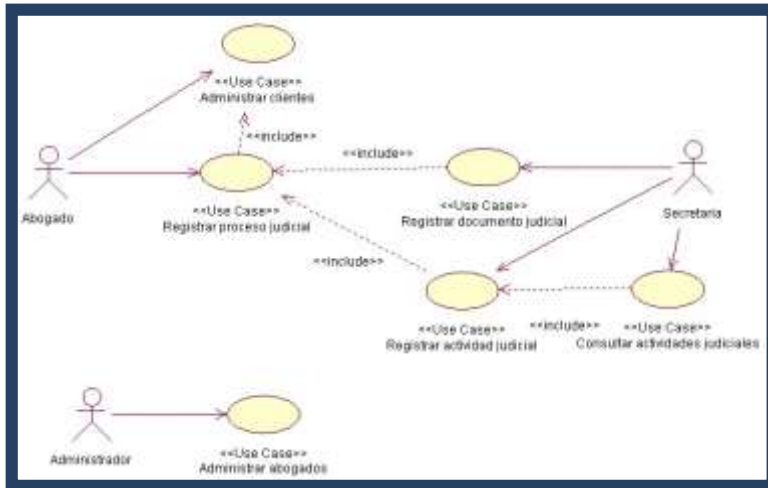
4.4.2.1. DIAGRAMAS DE CASOS DE USO

Gráfico 1 DIAGRAMA DE CASO DE USO DEL NEGOCIO



Fuente: Elaboración propia

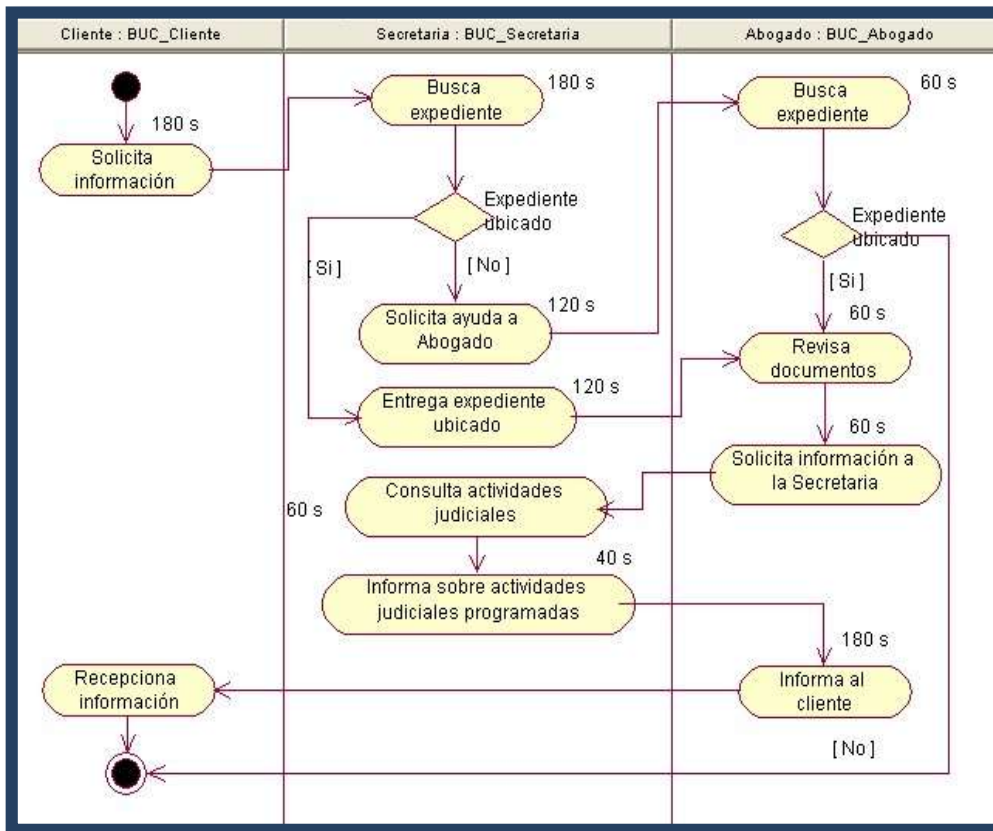
Gráfico N° 2 DIAGRAMA DE CASO DE USO DEL SISTEMA



Fuente: Elaboración propia

4.4.2.2. DIAGRAMA DE ACTIVIDADES

Gráfico N° 3 BUSQUEDA DE INFORMACIÓN



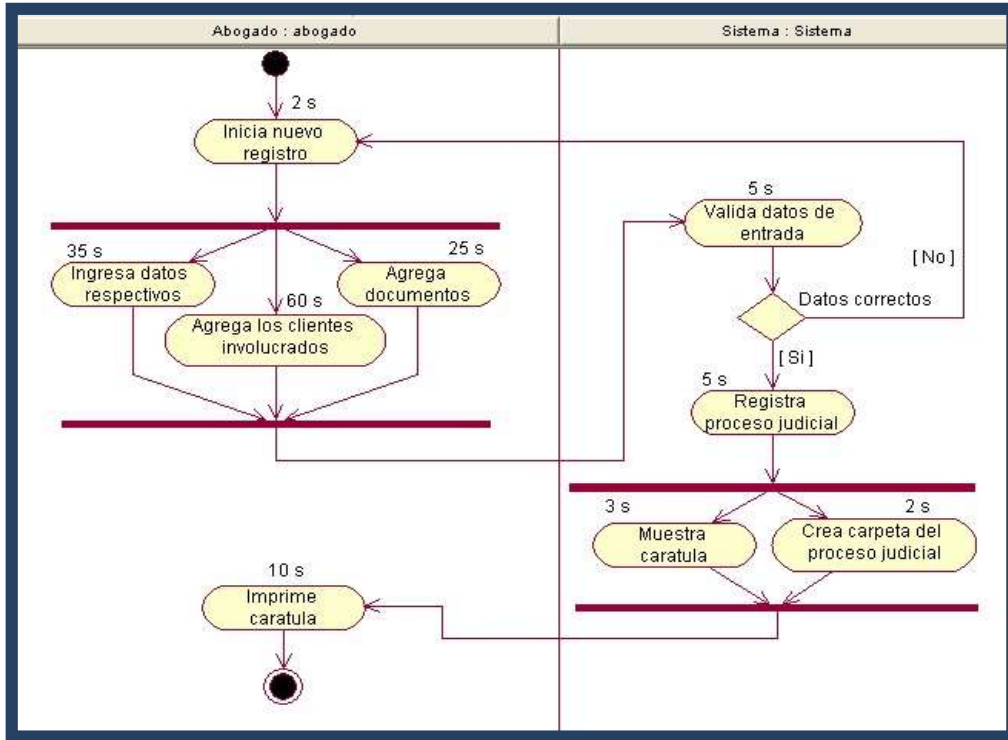
Fuente: Elaboración Propia

CASO DE USO:	Solicitar información
ACTORES:	Cliente, Secretaria, Abogado

FLUJO		DETALLE
1.-	Cliente solicita información	1.1.- Desea conocer el avance de su proceso judicial
2.-	Secretaria busca el expediente	2.1.- Busca en el archivador de los expedientes dependiendo del tipo de proceso 2.2.- En caso de no encontrarlo solicita ayuda al abogado para su búsqueda
3.-	Secretaria entrega expediente al abogado	
4.-	Abogado revisa expediente	4.1.- Revisa los documentos judiciales para obtener conclusiones
5.-	Secretaria consulta actuaciones judiciales	5.1.- Consulta cuaderno de actuaciones judiciales para observar las actuaciones judiciales programadas
		teniendo en cuenta el número de expediente
6.-	Abogado informa al cliente	6.1.- Informa al cliente acerca del estado de su proceso judicial y ejecución de posibles actuaciones judiciales

PRECONDICION:	Proceso judicial registrado
POSTCONDICION:	Informar el estado del proceso judicial

Gráfico N° 4 Registrar proceso judicial

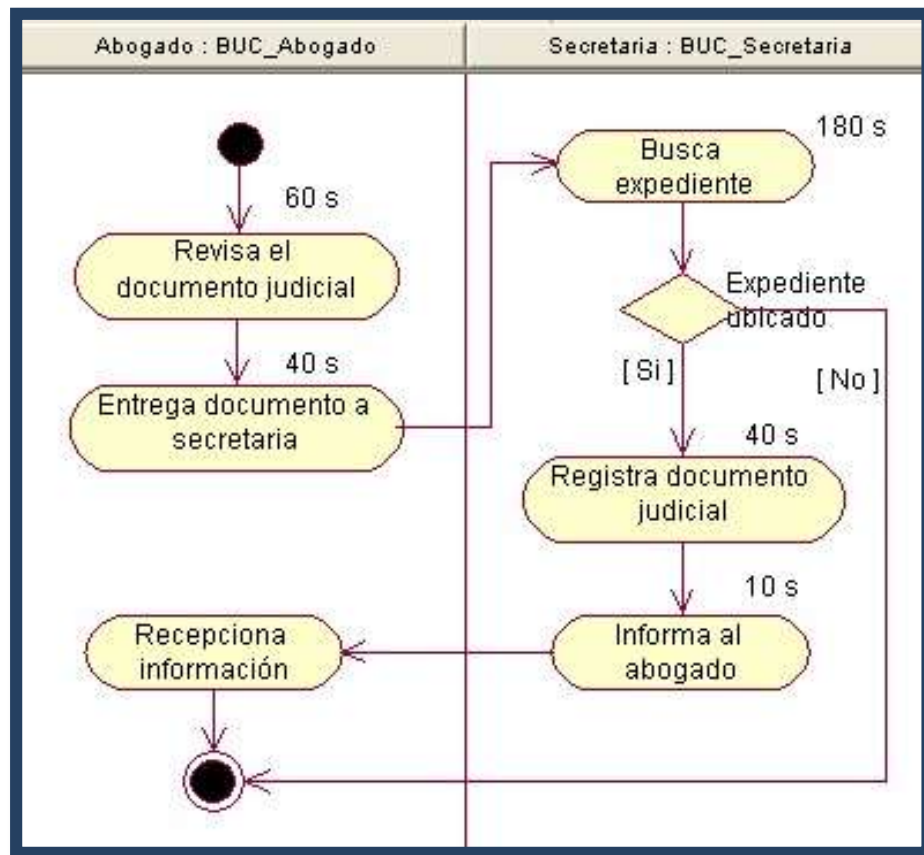


Fuente: Elaboración Propia Fuente: Elaboración Propia

CASO DE USO:	Registrar proceso judicial
ACTORES:	Abogado, Cliente
FLUJO	DETALLE

1.- Abogado busca un fólder nuevo	
2.- Abogado escribe los datos respectivos del proceso judicial	2.1.- En algunas ocasiones se registran solo algunos datos
3.- Abogado solicita datos del cliente	3.1.- Abogado escribe los datos del cliente
4.- Abogado archiva el expediente según la materia	
PRECONDICION:	Documento judicial presentado
POSTCONDICION:	Registrar documentos judiciales

5 Registrar documento judicial



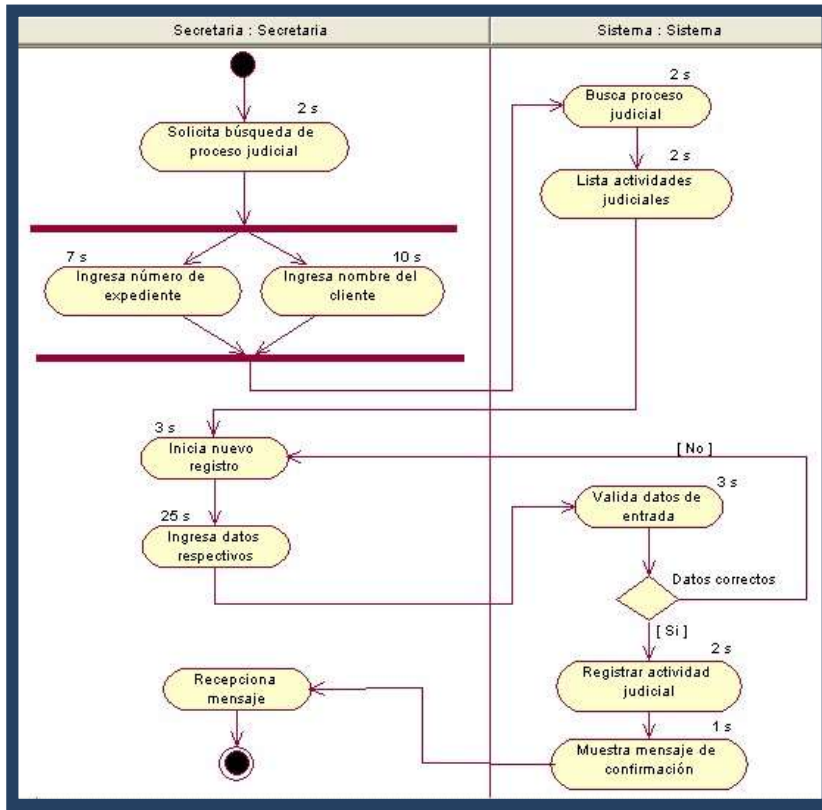
Fuente: Elaboración Propia

Gráfico N°

CASO DE USO:	Registrar documento judicial
ACTORES:	Abogado, Secretaria
FLUJO	DETALLE
1.- Abogado revisa el documento judicial	1.1.- Abogado lee el documento para conocer de que se trata
2.- Abogado entrega el documento a la secretaria para su registro	
3.- Secretaria busca expediente judicial	3.1.- Busca en el archivador de los expedientes dependiendo del tipo de proceso 3.2.- En caso de no encontrarlo solicita ayuda al abogado para su búsqueda
4.- Secretaria registra el documento judicial	

PRECONDICION:	Proceso judicial registrado
POSTCONDICION:	Consultar documento

6 Registrar actividad judicial



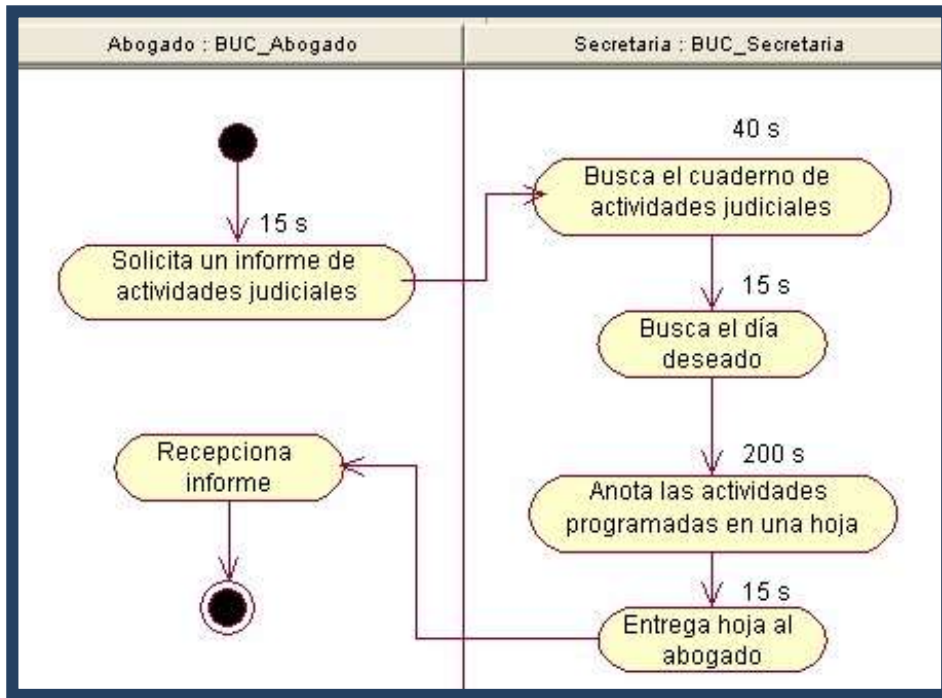
Fuente: Elaboración Propia

CASO DE USO:	Registrar actividad judicial
ACTORES:	Secretaria
FLUJO	DETALLE
1.- Abogado programa actividades judiciales	1.1.- Abogado informa a la secretaria

Gráfico N°

<p>2.- Secretaria busca el cuaderno de actividades judiciales</p>	
<p>3.- Secretaria busca el día que se desea programar</p>	
<p>4.- Secretaria registra los datos respectivos</p>	<p>4.1.- En los datos respectivos se incluye el número de expediente y una descripción de la actividad judicial</p>
<p>PRECONDICION:</p>	<p>Proceso judicial registrado</p>
<p>POSTCONDICION:</p>	<p>Consultar actividad judicial</p>

7Consultar actividades judiciales



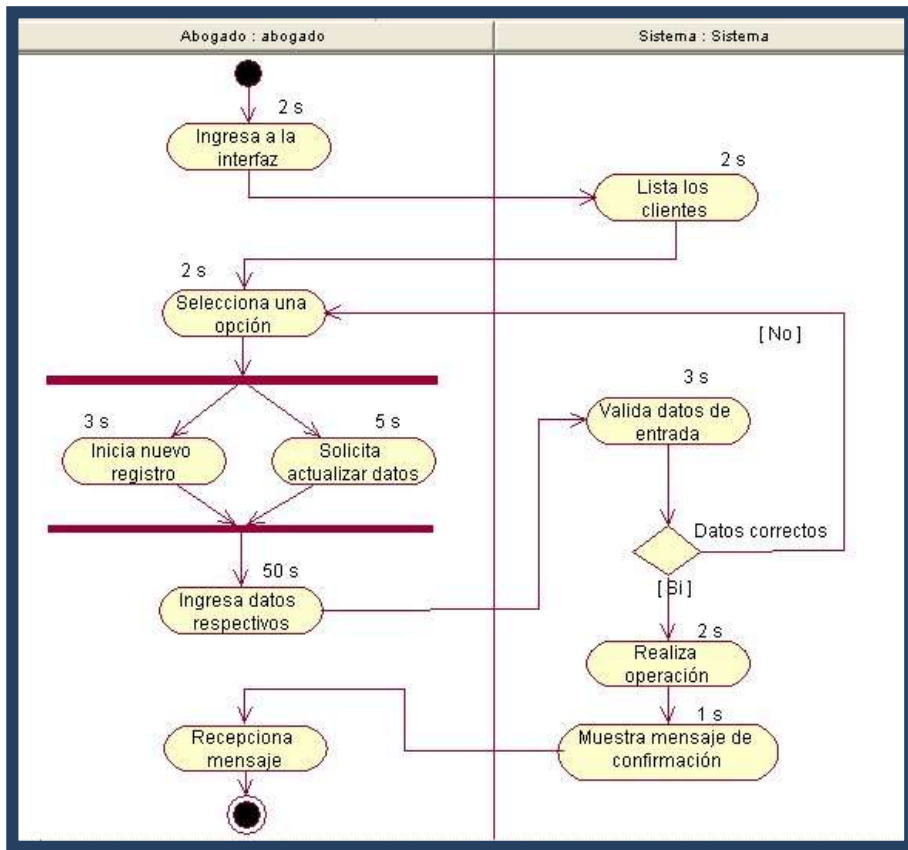
Fuente: Elaboración Propia

CASO DE USO:	Consultar actividades judiciales
ACTORES:	Abogado, Secretaria
FLUJO	DETALLE
1.- Abogado solicita un informe actividades judiciales	
2.- Secretaria busca el cuaderno de actividades judiciales	

Gráfico N°

3.- Secretaria busca el día deseado	
4.- Secretaria anota las actividades programadas en una hoja	4.1.- Anota los datos respectivos como el número de expediente y descripción
PRECONDICION:	Actividad judicial registrada
POSTCONDICION:	Emitir reportes de actividades

8REGISTRO DE USUARIOS



Fuente: Elaboración Propia

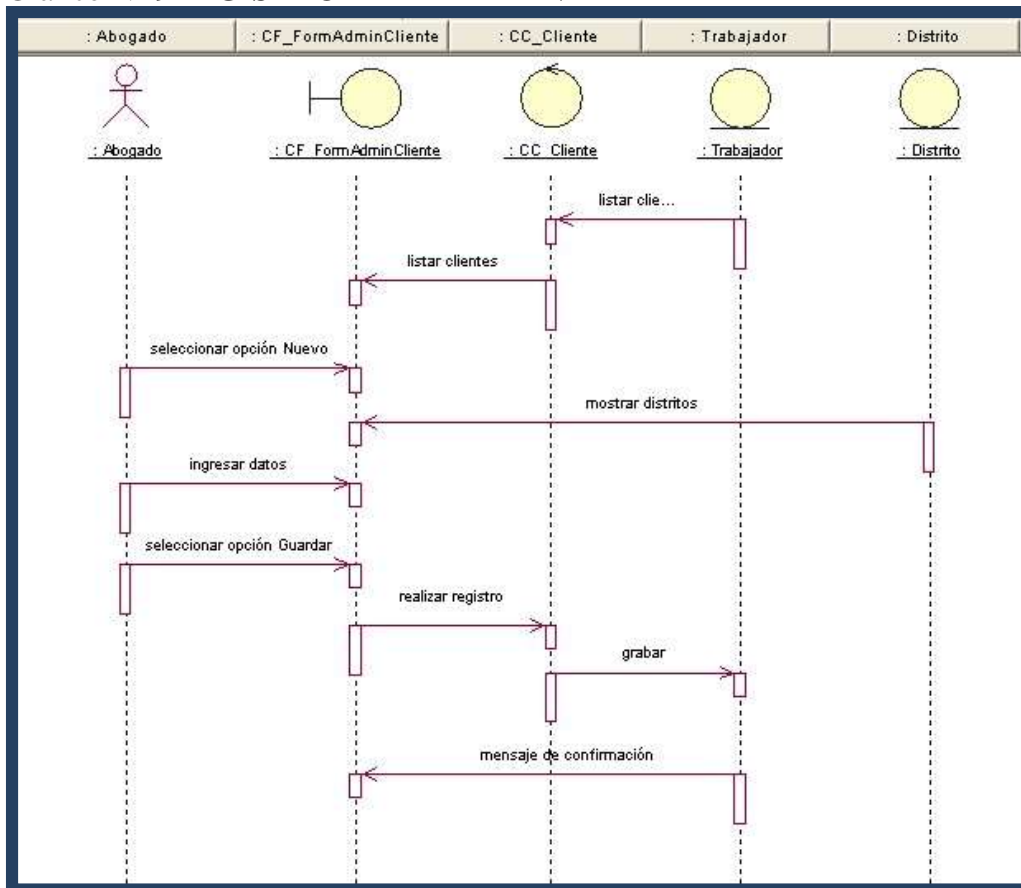
CASO DE USO:	Administrar clientes
ACTORES:	Abogado, Sistema
FLUJO	DETALLE
1.- Sistema lista todos los clientes del Estudio Jurídico	

Gráfico N°

2.- Abogado inicia un nuevo registro	2.1.- Abogado también puede actualizar los datos de un cliente
3.- Abogado ingresa los datos respectivos del cliente	3.1.- Estos datos incluye el Nombre / Razón Social, DNI / RUC, Distrito, Dirección, Teléfono / Celular, Correo Electrónico
4.- Sistema valida los datos	
5.- Sistema realiza la operación respectiva	
PRECONDICION:	Validar datos del abogado
POSTCONDICION:	Asignar los clientes a los procesos judiciales respectivos

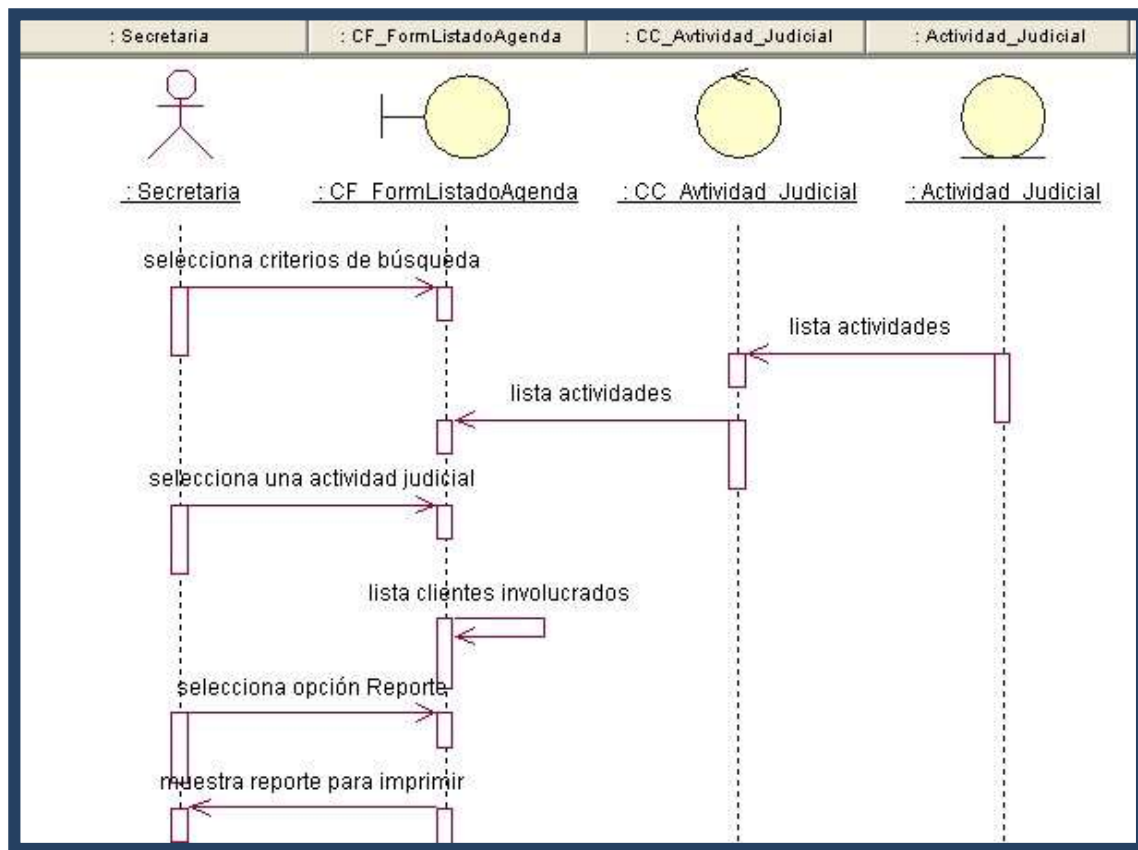
4.4.2.3. Diagramas de Secuencia

Gráfico N° 9 REGISTRO DE EXPEDIENTE



Fuente: Elaboración Propia

Gráfico N° 10 REGISTRAR ACTIVIDADES JUDICIALES

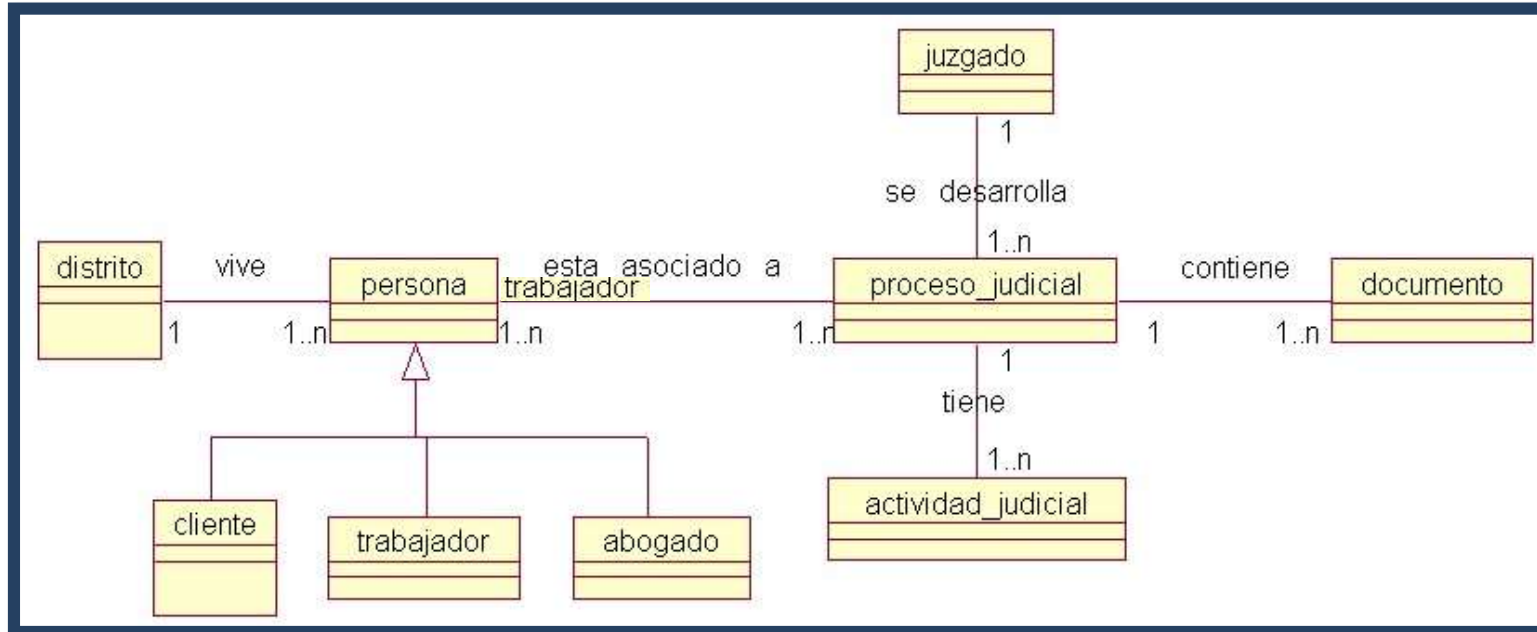


Fuente: Elaboración Propia

4.4.2.4. Diagrama de Clases

1. Modelo Conceptual

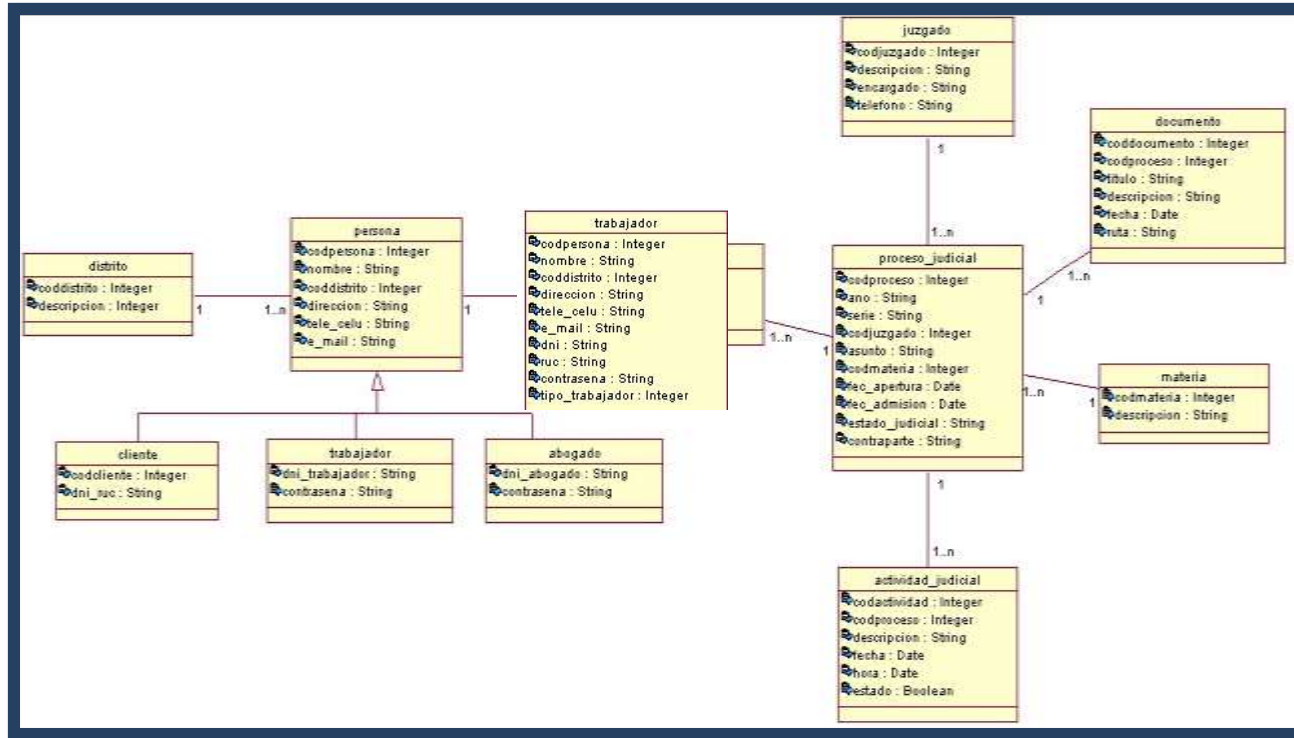
Gráfico N° 11 Modelo Conceptual



Fuente: Elaboración Propia

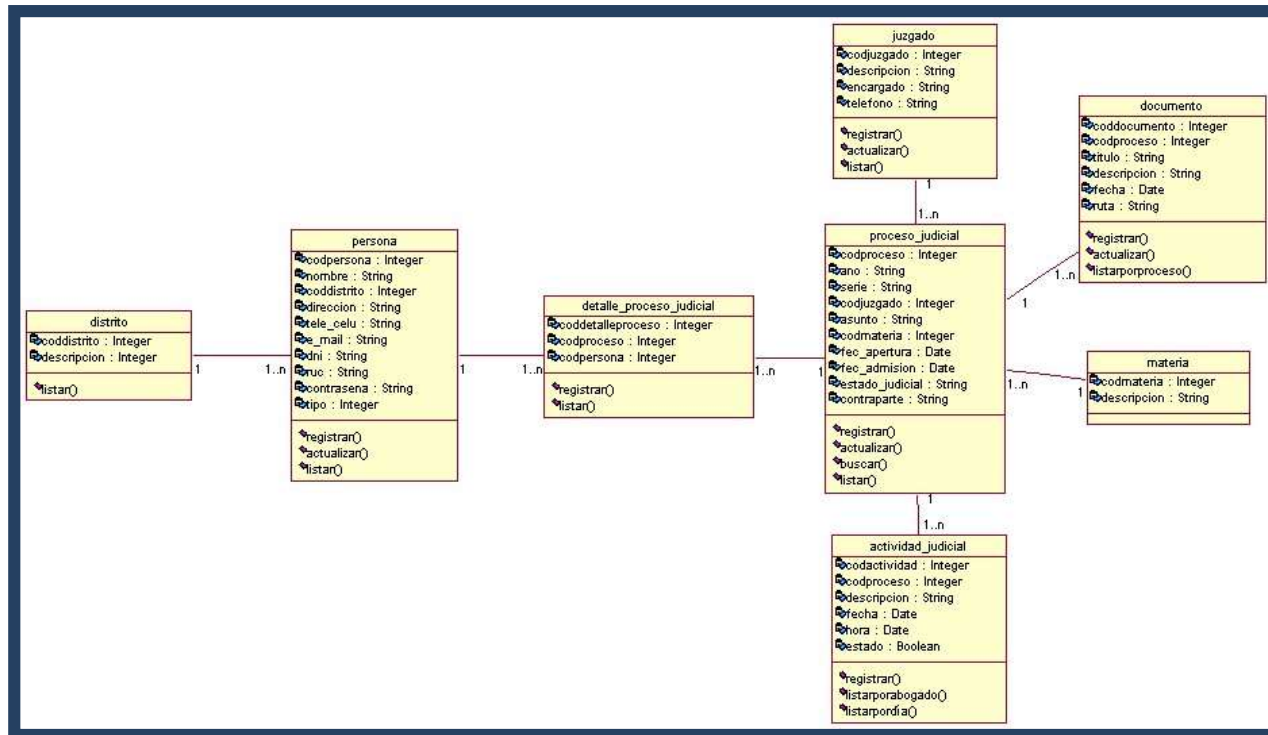
2. Modelo Lógico

Gráfico N° 12 Modelo Lógico



Fuente: Elaboración Propia

3. Modelo Físico Gráfico N° 13 Modelo Físico



Fuente: Elaboración Propia

4.4.3. Construcción del Sistema: Fase de Construcción

1. Arquitectura del Sistema

La arquitectura a utilizar será Cliente-Servidor. Se distinguen dos secciones, el cliente, donde se encuentran los usuarios del sistema y que accederá al sistema por medio de su computadora, y la segunda la conforma el servidor, en donde residen los datos, las reglas y lógica de la misma.

2. Elaboración de producto final

2.1. INTERFACES

A continuación, se presentan las interfaces de las principales ventanas.

Inicio de Sesión

En esta ventana el usuario inicia sesión con su respectivo nombre de usuario y contraseña para el ingreso al sistema.

Gráfico N° 14 Interfaz de Inicio de sesión

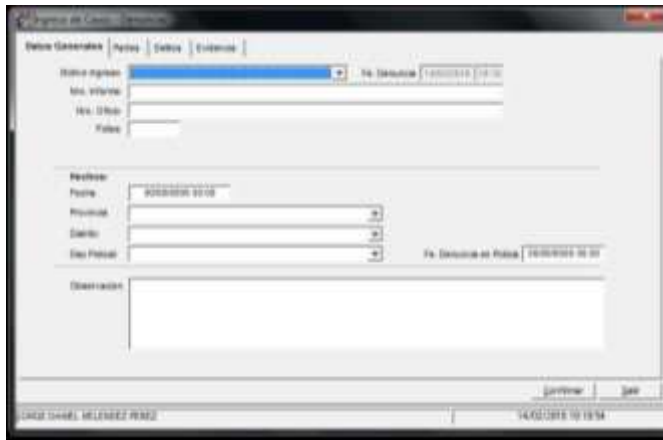


Fuente: Elaboración Propia

Menú Principal

En esta interfaz en donde se muestra los menús y accesos respectivos a las diferentes ventanas, además se registren el acceso de las mismas dependiendo del tipo de usuario.

Gráfico N° 15 Interfaz menú principal

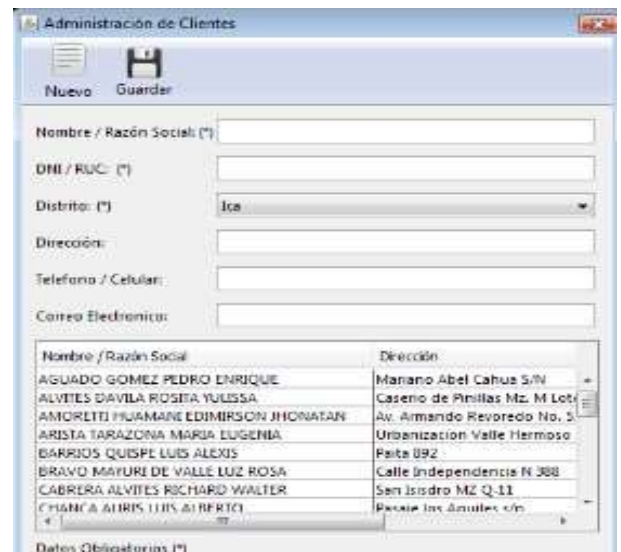


Fuente: Elaboración Propia

Administración de Clientes

En esta interfaz el abogado puede administrar los distintos clientes del Estudio Jurídico, registrándolos o modificando sus datos respectivos.

Gráfico N° 16 Interfaz Administración de clientes



Fuente: Elaboración Propia

Administración de Abogados

En esta interfaz el administrador puede gestionar los abogados que laboran en el Estudio Jurídico, registrándolos o modificando sus datos respectivos. Gráfico N° 17 Interfaz Administración de Abogados

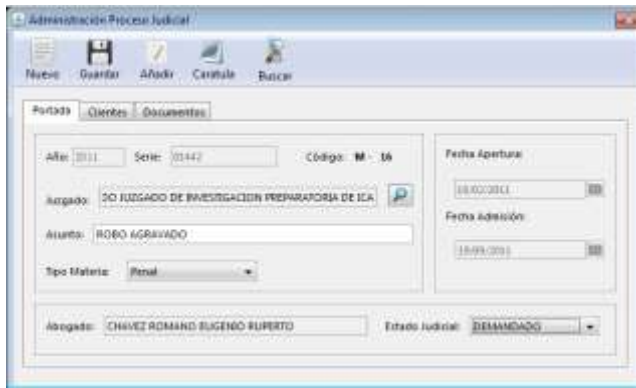
Apellidos y Nombres	Dirección
CHAVEZ ROMANO EUGENIO RUPERTO	San Joaquin IV Etapa
LUJA QUIJANDRIA MARIELA	Santa Maria

Fuente: Elaboración Propia

Administración Proceso judicial

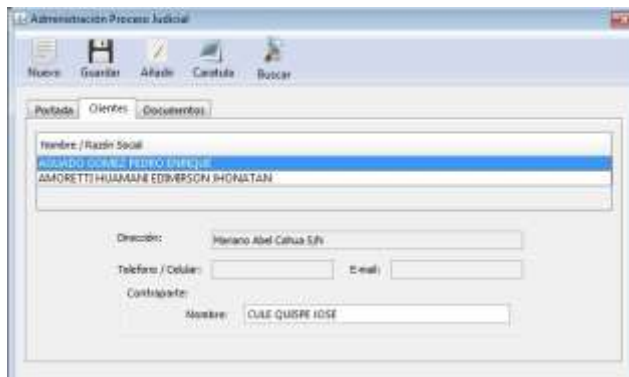
En esta interfaz el abogado registra los datos del proceso judicial y seleccionar los clientes involucrados. Además, muestra los documentos judiciales respectivos del proceso.

Gráfico N° 18 Interfaz proceso judicial



Fuente: Elaboración Propia

Gráfico N° 19 Interfaz Proceso Judicial



Fuente: Elaboración Propia

2.2. TRANSCT SQL

```
DROP PROCEDURE IF EXISTS `sp_validarabogado` $$
```

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `sp_validarabogado` (IN
```

```
pdni CHAR(8), IN pcontrasena CHAR(6))
```

```
BEGIN
```

```
SELECT codpersona FROM persona WHERE dni=pdni AND contrasena
```

```
=pcontrasena;
```

```
END $$
```

DELIMITER ;

DROP PROCEDURE IF EXISTS `sp_validartrabajador` \$\$

CREATE DEFINER=`root`@`localhost` PROCEDURE

`sp_validartrabajador`(IN pdni CHAR(8), IN pcontrasena CHAR(6))

BEGIN

SELECT codpersona FROM trabajador WHERE dni_trabajador=pdni

AND contrasena =pcontrasena;

END\$\$

DELIMITER ;

//click en el botón Aceptar

//usuarioabogado

if(cbbtipo.getSelectedIndex()==0){

codpersona=abo.validar(txtusuario.getText(),

String.valueOf(pwdcontraseña.getPassword()));

if(codpersona==0){JOptionPane.showMessageDialog(null,"DatosIncorrectos","I
nformación",JOptionPane.INFORMATION_MESSAGE);

 } else{

JOptionPane.showMessageDialog(null,"Bienvenido","Información",JOptionPane
.INFORMATION_MESSAGE); usuario="ABOGADO";

hacer();

 }

 }

//usuariotrabajador


```

else
                                if(cbbtipo.getSelectedIndex()==1){
codpersona=traba.validar(txtusuario.getText(),
String.valueOf(pwdcontraseña.getPassword()));
if(codpersona==0){
JOptionPane.showMessageDialog(null,"DatosIncorrectos","Información",JOption
nPane.INFORMATION_MESSAGE);
        } else{
JOptionPane.showMessageDialog(null,"Bienvenido","Información",JOptionPane
.INFORMATION_MESSAGE); usuario="TRABAJADOR";
hacer();      }      }

```

DROP PROCEDURE IF EXISTS `sp_registrarcliente`\$\$

```

CREATE DEFINER=`root`@`localhost` PROCEDURE `sp_registrarcliente`(IN
pnombre VARCHAR(100),IN pcoddistrito INT(11),IN pdireccion
VARCHAR(100),IN ptele VARCHAR(50),IN pemail VARCHAR(50),IN
pdniruc VARCHAR(11))

```

BEGIN

```

        INSERT INTO persona (nombre,coddistrito,direccion,tele_celu,e_mail)
VALUES (pnombre,pcoddistrito,pdireccion,ptele,pemail);

```

END\$\$

DELIMITER ;

DROP PROCEDURE IF EXISTS `sp_modificarcliente`\$\$

```

CREATE DEFINER=`root`@`localhost` PROCEDURE `sp_modificarcliente`(IN
pcodpersona INT(11),IN pnombre VARCHAR(100),IN pcoddistrito INT(11),IN
pdireccion VARCHAR(100),IN ptele VARCHAR(50),IN pemail
VARCHAR(50),IN pdniruc VARCHAR(11))
BEGIN
    UPDATE persona SET nombre=pnombre, coddistrito=pcoddistrito,
direccion=pdireccion, tele_celu=ptele, e_mail=pemail
    WHERE
codpersona=pcodpersona;
    END$$
DELIMITER ;

```

//click en el botón Guardar

```

//guardar cliente
if(Integer.parseInt(lblcodpersona.getText())==0){
if(validar()==0){
    }else{
if((cli.registrar(txtnombre.getText(),Integer.parseInt(lblcoddistrito.getText()),
txtdireccion.getText(),txttelefono.getText(), txtemail.getText(),
txtdni.getText()))==1){
JOptionPane.showMessageDialog(null,"Registro
Correcto","Informacion",JOptionPane.INFORMATION_MESSAGE);
listarcliente();

```

```

        }
else{JOptionPane.showMessageDialog(null,"RegistroIncorrecto","Informacion",
JOptionPane.INFORMATION_MESSAGE);
        }
    }
}
else{
//actualizarcliente
if((cli.actualizar(Integer.parseInt(lblcodpersona.getText()),txtnombre.getText(),I
nteger.parseInt(lblcoddistrito.getText()),
txtdireccion.getText(),txttelefono.getText(),          txtemail.getText(),
txtdni.getText()))==1){
JOptionPane.showMessageDialog(null,"DatosActualizados","Informacion",JOpti
onPane.INFORMATION_MESSAGE);
listarcliente(); btnuevo.setEnabled(true);
        }
else{JOptionPane.showMessageDialog(null,"Datos          no
Actualizados","Informacion",JOptionPane.INFORMATION_MESSAGE);
}
}
}

```

//creamos procedimientos almacenados

DROP PROCEDURE IF EXISTS `sp_administrarabotraba`\$\$

```

CREATE DEFINER=`root`@`localhost` PROCEDURE `sp_administrarabotraba`(IN
    ptipo INT(11),IN pnombre VARCHAR(100),IN pcoddistrito INT(11),IN
    pdireccion VARCHAR(100),IN ptele VARCHAR(50),IN pemail
    VARCHAR(50),IN pdni VARCHAR(11),IN ppass CHAR(6),IN pcodpersona
    INT(11))
BEGIN
    IF ptipo=1 THEN
        INSERT INTO persona (nombre,coddistrito,direccion,tele_celu,e_mail)
        VALUES (
    ELSEIF ptipo=2 THEN
        INSERT INTO persona (nombre,coddistrito,direccion,tele_celu,e_mail)
        VALUES (pnombre,pcoddistrito,pdireccion,ptele,pemail);
    ELSEIF ptipo=3 THEN
UPDATE    persona    SET    nombre=pnombre,    coddistrito=pcoddistrito,
    direccion=pdireccion,tele_celu=ptele,e_mail=pemail WHERE
    codpersona=pcodpersona;
    ELSEIF ptipo=4 THEN
UPDATE    persona    SET    nombre=pnombre,    coddistrito=pcoddistrito,
    direccion=pdireccion, tele_celu=ptele,e_mail=pemail WHERE
    codpersona=pcodpersona;
        END IF;
    END$$
DELIMITER ;

```

//click en el botón Guardar

```
int tipo = Integer.parseInt(lbltipo.getText());          if(validar()==1){
if(Integer.parseInt(lblcodpersona.getText())==0){
    //registrar abogado
if(tipo==1){
    if((abo.administrar(1,txtnombre.getText(),Integer.parseInt(lblcoddistrito.getText(
)), txtdireccion.getText(),txttelefono.getText(), txtemail.getText(),
txtdni.getText(),String.valueOf(pwdcontraseña.getPassword()),0))==1){
        JOptionPane.showMessageDialog(null,"RegistroCorrecto","Informacion",JOptio
nPane.INFORMATION_MESSAGE);
listarabogado();
    }
else{JOptionPane.showMessageDialog(null,"RegistroIncorrecto","Informacion",JOptio
nPane.INFORMATION_MESSAGE);
}
}
//registrar trabajador
else if(tipo == 2){
if((traba.administrar(2,txtnombre.getText(),Integer.parseInt(lblcoddistrito.getTex
t()), txtdireccion.getText(),txttelefono.getText(), txtemail.getText(),
txtdni.getText(),String.valueOf(pwdcontraseña.getPassword()),0))==1){
        JOptionPane.showMessageDialog(null,"RegistroCorrecto","Informacion",JOptio
```

```

        nPane.INFORMATION_MESSAGE);

listarabogado();

        }

else{JOptionPane.showMessageDialog(null,"RegistroIncorrecto","Informacion",JOption
        Pane.INFORMATION_MESSAGE);

}    }

        }

else{

        //actualizarabogado

if(tipo==1){

        if((abo.administrar(3,txtnombre.getText(),Integer.parseInt(lblcoddistrito.getText(
        )),    txtdireccion.getText(),txttelefono.getText(),    txtemail.getText(),
        txtdni.getText(),String.valueOf(pwdcontraseña.getPassword()),Integer.parseInt(1
        blcodpersona.getText()))==1){

        JOptionPane.showMessageDialog(null,"DatosActualizados","Informacion",JOpti
        onPane.INFORMATION_MESSAGE);

listarabogado();

        }

else{    JOptionPane.showMessageDialog(null,"Datos no
        Actualizados","Informacion",JOptionPane.INFORMATION_MESSAGE);

        }

        }

        //actualizar trabajador

elseif(tipo    ==    2){

```

```

if((traba.administrar(4,txtnombre.getText(),Integer.parseInt(lblcoddistrito.getTex
t()),   txtdireccion.getText(),txttelefono.getText(),      txtemail.getText(),
txtdni.getText(),String.valueOf(pwdcontraseña.getPassword()),Integer.parseInt(l
blcodpersona.getText()))==1){

JOptionPane.showMessageDialog(null,"DatosActualizados","Informacion",JOpti
onPane

.INFORMATION_MESSAGE);

listartrabajador();

    }

else{JOptionPane.showMessageDialog(null,"Datos                               no
Actualizados","Informacion",JOptionPane.INFORMATION_MESSAGE);

    }    }    }

} else{

}

```

DROP PROCEDURE IF EXISTS `sp_registrarproceso`\$\$

```

CREATE DEFINER=`root`@`localhost` PROCEDURE `sp_registrarproceso`(IN pano
CHAR(4),IN pserie VARCHAR(10),IN pcodjuzgado INT(11),IN pasunto
VARCHAR(100),IN pcodmateria INT(11),IN pfec_apertura DATE,IN
pestadojudicial VARCHAR(10),IN pcodabo INT(11),IN pcontra
VARCHAR(150))

```

BEGIN

INSERT

INTO

```

proceso_judicial(ano,serie,codjuzgado,asunto,codmateria,fec_apertura,fec_admis
ion,estado_judicial,contraparte) VALUES
(pano,pserie,pcodjuzgado,pasunto,pcodmateria,pfec_apertura,CURDATE(),pesta
dojudicial,pcontra);

```

```

INSERT INTO detalle_proceso_judicial (codproceso,codpersona) VALUES
((SELECT codproceso FROM proceso_judicial ORDER BY codproceso DESC
LIMIT 1),pcodabo);

```

```

END$$

```

```

DELIMITER ;

```

//click en el botón guardar

```

String dato = new SimpleDateFormat("yyyy-MM-dd").format(txtfecha1.getDate());
intconta=0,conta2=0,conta3=1; if (txtaño.isEnabled()==true){
    //registrar
    if (validar()==0){
        }
    else{
        if((proceso.registrar(txtaño.getText(),
txtserie.getText(),Integer.parseInt(lblcodjuzgado.getText()),txtasunto.getText(),m
ate.obtenercodigo(String.valueOf(cbbmateria.getSelectedItem())) , dato,
String.valueOf(cbbestado.getSelectedItem()),
Integer.parseInt(lblcodabogado.getText()),txtcontra.getText())==1){
for(int =0;i<tbclientes.getRowCount();i++){
if((detaproceso.registrar(0,Integer.parseInt(String.valueOf(modelocli.getValueAt
(i, 4))))==1){ conta++;

```



```

        }
    }

for(int =0;i<tbdocumentos.getRowCount();i++){

    if((docu.registrar(0,0,

String.valueOf(modelodocu.getValueAt(i,0)),String.valueOf(modelodocu.getVal
ueAt(i,      1)),      String.valueOf(modelodocu.getValueAt(i,
2)),rutas(String.valueOf(modelodocu.getValueAt(i,
3)),String.valueOf(modelodocu.getValueAt(i,      5))),"Documento
"+String.valueOf(conta3),String.valueOf(modelodocu.getValueAt(i, 6))))==1){
conta2++; conta3++;

    }

    }

    if(conta==tbclientes.getRowCount()      &&
conta2==tbdocumentos.getRowCount()){

JOptionPane.showMessageDialog(null,"RegistroCorrecto","Información",JOptio
nPane.INFORMATION_MESSAGE);

lblcodproceso.setText(String.valueOf(proceso.buscar2(txtaño.getText(),
txtserie.getText())));

this.cargartablacliente(Integer.parseInt(lblcodproceso.getText()));

this.cargartabladocumento(Integer.parseInt(lblcodproceso.getText())); botones();

repor.caratula(Integer.parseInt(lblcodproceso.getText()));  }

else{JOptionPane.showMessageDialog(null,"RegistroIncorrecto","Información",JOption
Pane.INFORMATION_MESSAGE);

btnuevo.setEnabled(true);

```

```

        }
    }
else{JOptionPane.showMessageDialog(null,"RegistroIncorrecto","Información",JOption
    Pane.INFORMATION_MESSAGE);        }        }        } else{
    //actualizar        conta3=tbdocumentos.getRowCount();
if((proceso.modificar(Integer.parseInt(lblcodproceso.getText()),Integer.parseInt(l
    blcodjuzgado.getText()),txtasunto.getText(),mate.obtenercodigo(String.valueOf(
    cbbmateria.getSelectedItem())) , String.valueOf(cbbestado.getSelectedItem()),
    txtcontra.getText())==1){
        for(inti=0;i<tbclientes.getRowCount();i++){
            if((detaproceso.registrar(Integer.parseInt(lblcodproceso.getText()),Integer.parseI
                nt(String.valueOf(modelocli.getValueAt(i, 4))))==1){
    conta++;        }
        }
        for(inti=0;i<tbdocumentos.getRowCount();i++){
            if(Integer.parseInt(String.valueOf(modelodocu.getValueAt(i,4)))==0){
                if((docu.registrar(Integer.parseInt(lblcodproceso.getText()),0,
                    String.valueOf(modelodocu.getValueAt(i,0)),String.valueOf(modelodocu.getVal
                        ueAt(i,        1)),        String.valueOf(modelodocu.getValueAt(i,
                            2)),rutas(String.valueOf(modelodocu.getValueAt(i,
                                3)),String.valueOf(modelodocu.getValueAt(i,        5)),"Documento
                                    "+String.valueOf(conta3),String.valueOf(modelodocu.getValueAt(i, 6))))==1){
    conta2++; conta3++;

```

```

        }
    }
else{
    if((docu.registrar(Integer.parseInt(lblcodproceso.getText()),Integer.parseInt(String.valueOf(modelodocu.getValueAt(i,4))),
    String.valueOf(modelodocu.getValueAt(i,0)),String.valueOf(modelodocu.getValueAt(i, 1)), String.valueOf(modelodocu.getValueAt(i, 2))," ")==1){ conta2++;
        }
    }
    }
    if(conta==tblclientes.getRowCount() &&
    conta2==tbdocumentos.getRowCount()){
        JOptionPane.showMessageDialog(null,"DatosActualizados","Información",JOptionPane.INFORMATION_MESSAGE);
        this.cargartablacliente(Integer.parseInt(lblcodproceso.getText()));
        this.cargartabladocumento(Integer.parseInt(lblcodproceso.getText()));
        proceso.obtenerdatos(Integer.parseInt(lblcodproceso.getText()), txtaño,
        txtserie,txtasunto,lblcodjuzgado,txtjuzgado,cbbestado, lblfec1,lblfec2,
        cbbmateria,lblcodabogado,txtabogado,txtcontra);
        txtfecha1.setDate(convertirfecha(lblfec1.getText()));
        txtfecha2.setDate(convertirfecha(lblfec2.getText()));
    }
else{JOptionPane.showMessageDialog(null,"Datos no
    Actualizados","Información",JOptionPane.INFORMATION_MESSAGE);
    }
else{JOptionPane.showMessageDialog(null,"RegistroIncorrecto","Información",
    JOptionPane.INFORMATION_MESSAGE);
}

```

```
}
```

```
}
```

```
}
```

DROP PROCEDURE IF EXISTS `sp_obtenerproceso`\$\$

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `sp_obtenerproceso`(IN
```

```
pcodproceso INT(11))
```

```
BEGIN
```

```
SELECT proceso_judicial.ano, proceso_judicial.serie ,proceso_judicial.asunto,  
proceso_judicial.codjuzgado, juzgado.descripcion,
```

```
proceso_judicial.estado_judicial, proceso_judicial.fec_apertura,
```

```
proceso_judicial.fec_admision, materia.descripcion,
```

```
detalle_proceso_judicial.codpersona, persona.nombre,  
proceso_judicial.contraparte FROM proceso_judicial
```

```
INNER JOIN juzgado ON juzgado.codjuzgado=proceso_judicial.codjuzgado
```

```
INNER JOIN materia ON materia.codmateria=proceso_judicial.codmateria
```

```
INNER JOIN detalle_proceso_judicial ON
```

```
proceso_judicial.codproceso=detalle_proceso_judicial.codproceso
```

```
INNER JOIN persona ON
```

```
persona.codpersona=detalle_proceso_judicial.codpersona
```

```
INNER JOIN abogado ON abogado.codpersona=persona.codpersona
```

```
WHERE proceso_judicial.codproceso=pcodproceso;
```

```
END$$
```

```
DELIMITER ;
```

DROP PROCEDURE IF EXISTS `sp_buscarprocesoxcliente`\$\$

```

CREATE DEFINER=`root`@`localhost` PROCEDURE `sp_buscarprocesoxcliente`(IN
    pcod INT(11))
BEGIN
    SELECT      CONCAT(proceso_judicial.ano,'-',      proceso_judicial.serie),
    proceso_judicial.asunto, materia.descripcion, proceso_judicial.estado_judicial,
    detalle_proceso_judicial.codproceso FROM detalle_proceso_judicial
INNER          JOIN          proceso_judicial          ON
    detalle_proceso_judicial.codproceso=proceso_judicial.codproceso
INNER JOIN materia ON materia.codmateria=proceso_judicial.codmateria
    WHERE detalle_proceso_judicial.codpersona=pcod;
END$$

```

```
DELIMITER ;
```

```
//creamos método buscarexpediente
```

```

modeloex=proce.buscar(codpersona); tbprocesos.setModel(modeloex);
tbprocesos.getColumnModel("Año          -          Serie").setPreferredWidth(25);
    tbprocesos.getColumnModel("Asunto").setPreferredWidth(25);
    tbprocesos.getColumnModel("Materia").setPreferredWidth(25);
    tbprocesos.getColumnModel("Estado Judicial").setPreferredWidth(25);
TableColumncolumn = null;      column=tbprocesos.getColumnModel().getColumn(4);
    tbprocesos.getColumnModel().removeColumn(column);

```

```
DROP PROCEDURE IF EXISTS `sp_registrardocumento`$$
```

```

CREATE DEFINER=`root`@`localhost` PROCEDURE `sp_registrardocumento`(IN
    pcodproceso INT(11),IN pcoddocumento INT(11),IN ptitulo VARCHAR(50),IN
    pdescripcion VARCHAR(200),IN pfecha DATE,IN pruta VARCHAR(100))

```

```

BEGIN

    DECLARE v_codproceso INT DEFAULT 0;

    IF pcodproceso=0 THEN

        SELECT codproceso INTO v_codproceso FROM proceso_judicial

ORDER BY codproceso DESC LIMIT 1;

        INSERT INTO documento (codproceso,titulo,descripcion,fecha,ruta)

VALUES (v_codproceso,ptitulo,pdescripcion,pfecha,pruta);

    ELSE

        IF pcoddocumento=0 THEN

            INSERT INTO documento (codproceso, titulo, descripcion, fecha,

ruta) VALUES (pcodproceso,ptitulo,pdescripcion,pfecha,pruta);

        ELSE

            UPDATE documento SET titulo=ptitulo,

descripcion=pdescripcion, fecha=pfecha WHERE

coddocumento=pcoddocumento;

        END IF;

    END IF;

END$$

DELIMITER ;

```

//click en el botón aceptar

```

if(this.getTitle().equals("Nuevo Documento"))

agregar(); else

```

```

if(this.getTitle().equals("EditarDocumento"))
modificar();

//creamos método agregar

//validan datos if(txttitulo.getText().equals(""))
){
JOptionPane.showMessageDialog(null,
    "Ingresetitulo", "Error", JOptionPane.ERROR_MESSAGE);
txttitulo.requestFocus();
}
else if(txtruta.getText().equals("")){
JOptionPane.showMessageDialog(null, "Ingrese ruta del
    documento", "Error", JOptionPane.ERROR_MESSAGE);
}
else{
//agregan datos al modelo documento
Object nuevo[]=
    {txttitulo.getText(),txtdescripcion.getText(),lblfecha.getText(),txtruta.getText(),"
    0",lbldirectorio.getText(),txtextension.getText()};
mainApp.modelodocu.addRow(nuevo); intanswer =
JOptionPane.showConfirmDialog(null, "¿Desea agregar otro documento",
    "Información",
    JOptionPane.YES_NO_OPTION,JOptionPane.INFORMATION_MESSAGE); if
(answer == JOptionPane.YES_OPTION)

```

```

{      limpiar();
txttitulo.requestFocus();      }
else dispose();      }
validarboton();

```

DROP PROCEDURE IF EXISTS `sp_registraractividad` \$\$

```

CREATE DEFINER=`root`@`localhost` PROCEDURE
`sp_registraractividad`(IN pcodproceso INT(11),IN pdescpcion
VARCHAR(200),IN pfecha DATE,IN phora TIME)
BEGIN
INSERT INTO actividad_judicial (codproceso,descripcion,fecha,hora,estado)
VALUES (pcodproceso,pdescripcion,pfecha,phora,FALSE);
END$$
DELIMITER ;

```

DROP PROCEDURE IF EXISTS `sp_modificaractividad` \$\$

```

CREATE DEFINER=`root`@`localhost` PROCEDURE `sp_modificaractividad`(IN
pcodactuacion INT(11),IN pdescpcion VARCHAR(200),IN pfecha DATE,IN
phora TIME,IN pestado BOOLEAN)
BEGIN
UPDATE actividad_judicial SET descripcion=pdescpcion, fecha=pfecha, hora=phora,
estado=pestado WHERE codactividad=pcodactuacion;
END$$
DELIMITER ;

```



```

//creamos método listar modeloactu=actu.listar(codproceso);

this.tbactuaciones.setModel(modeloactu);

tbactuaciones.getColumnModel("Descripcion").setPreferredWidth(300);

tbactuaciones.getColumnModel("Fecha").setPreferredWidth(100);

tbactuaciones.getColumnModel("Hora").setPreferredWidth(100);

TableColumn column = null;

column=tbactuaciones.getColumnModel().getColumn(4);

tbactuaciones.getColumnModel().removeColumn(column);

//click en el botón guardar if(Integer.parseInt(lblcodproceso.getText())==0){

JOptionPane.showMessageDialog(null,"Debe buscar un proceso judicial", "Error",

JOptionPane.ERROR_MESSAGE);

} else{ try{

if(Integer.parseInt(lblcodactuacion.getText())==0){

//registrar if(validar()==0){

}else{

Date fecha2=formats.parse(lblfecha.getText());

Date hors=hora.parse(txthora.getText()); java.sql.Time timest = new

java.sql.Time(hors.getTime()); java.sql.Date fec = new

java.sql.Date(fecha2.getTime());

if((actu.registrar(Integer.parseInt(lblcodproceso.getText()),txtdescripcion.getText

```

```

        ), fec,timest))==1){
    JOptionPane.showMessageDialog(null,"RegistroCorrecto","Informacion",JOption
    nPane.I
        INFORMATION_MESSAGE);

this.listar(Integer.parseInt(lblcodproceso.getText()));

        }

else{JOptionPane.showMessageDialog(null,"RegistroIncorrecto","Informacion",JOption
    Pane.INFORMATION_MESSAGE);

}

        }

        }

else{

        //actualizar

Date fecha2=formats.parse(lblfecha.getText());

        Date hors=hora.parse(txthora.getText()); java.sql.Timetimest = new
java.sql.Time(hors.getTime()); java.sql.Datefec = new java.sql.Date(fecha2.getTime());

if((actu.modificar(Integer.parseInt(lblcodactuacion.getText()),txtdescripcion.getText(),
fec,timest,Boolean.parseBoolean(lblestado.getText()))==1){

JOptionPane.showMessageDialog(null,"DatosActualizados","Informacion",JOpti
    onPane.INFORMATION_MESSAGE);

this.listar(Integer.parseInt(lblcodproceso.getText()));

        }

else{

JOptionPane.showMessageDialog(null,"Datos
        no
        Actualizados","Informacion",JOptionPane.INFORMATION_MESSAGE);

```

```

        }
    }
} catch (Exception ex) {
System.out.print(ex);
}
}
}

```

DROP PROCEDURE IF EXISTS `sp_listaractividadxdia`\$\$

```

CREATE DEFINER=`root`@`localhost` PROCEDURE `sp_listaractividadxdia`(IN
    pfecha DATE,IN pcodabo INT(11))
BEGIN
    SELECT CONCAT(p.ano,'-',p.serie), a.descripcion, a.hora, a.estado,
    a.codproceso, a.codactividad FROM actividad_judicial a
    INNER JOIN proceso_judicial p ON p.codproceso=a.codproceso
    INNER JOIN detalle_proceso_judicial d ON d.codproceso=p.codproceso
    WHERE a.fecha=pfecha AND d.codpersona=pcodabo ORDER BY hora;
END$$
DELIMITER ;

```

//creamos método listaractividad

```

public void listaractividad(java.sql.Date fecha, int codabogado){
    modeloactu=actu.listar2(fecha,codabogado); tbactividades.setModel(modeloactu);
    tbactividades.getColumnModel("Expediente").setPreferredWidth(50);
    tbactividades.getColumnModel("Descripción").setPreferredWidth(300);
}

```

```
tbactividades.getColumn("Hora").setPreferredWidth(13);  
tbactividades.getColumn("Estado").setPreferredWidth(13); removercoluactividad();  
}
```

V. CONCLUSIONES

Como conclusiones podemos destacar lo siguiente:

1. La implementación del Sistema de Información en el consultorio Jurídico permitirá mejorar de manera sustancial el seguimiento de documentos, se debe de tener en cuenta que, para realizar dichas mejoras, tanto los trabajadores como la gerencia deberán garantizar el uso al 100% del Software, alimentando la base de datos con todos los expedientes judiciales existentes en el estudio Jurídico.
2. La implementación del Sistema de Información permitirá establecer de forma clara los procesos judiciales a cargo del estudio Jurídico, su estado, su avance y los actores que intervienen en el.
3. El sistema de información podrá ser implementado en los diferentes puntos del estudio Jurídico,
4. Se concluye que el Sistema de Información ha colaborado satisfactoriamente en la gestión de los procesos judiciales del Estudio Jurídico SALASCALDERON & ASOCIADOS, habiendo disminuido el tiempo total en brindar información al cliente acerca de su proceso, lo cual fue logrado gracias al uso del Sistema de

Información, permitiendo integrar toda la información de los procesos judiciales, logrando una rápida búsqueda del expediente deseado, y evitando al cliente la larga espera por ser atendido.

RECOMENDACIONES

1. Implementar el sistema de información propuesto por presentar mejoras sustanciales en el proceso seleccionado para la investigación.
2. El personal debe estar en constante capacitación con las nuevas tecnologías a fin de garantizar la eficiencia en su trabajo.
3. En el tema de la seguridad de los datos, se debe tomar en cuenta normas para definir contraseñas de los usuarios, para restringir así el acceso a cierta información.
4. Se sugiere que el Sistema de Información sea implementado en otros Estudios Jurídicos de la ciudad de Ica, debido a los beneficios que se pueden obtener con dicha implementación.
5. Realizar un rediseño de procesos, de tal forma que se pueda prestar un mejor servicio orientado principalmente a la satisfacción del cliente.

BIBLIOGRAFIA

1. Contreras FA. Diseño de un modelo para la implementación de un sistema de gestión documental en áreas u organizaciones jurídicas. [Tesis de Grado]. Bogotá: Pontificia Universidad Javeriana; 2015. 87 p.
2. Salazar CM. Implementación de un sistema de gestión y control de un despacho jurídico. [Tesis de Grado]. Loja: Universidad Particular Técnica de Loja; 2012. 248 p.
3. Escobar MC. , diseño e implementación de un sistema de apoyo al seguimiento de procesos judiciales para un estudio de abogados. [Tesis de Grado]. Lima: Pontificia Universidad Católica del Perú; 2010. 85 p.
4. Poder Judicial. Descripción de las Leyes [Internet]. Poder Judicial Peruano. Marzo 2011. [Revisado 02 de enero 2015]. Disponible desde [:http://www.pj.gob.pe/docinteres/leyes/tercera_seccion.htm](http://www.pj.gob.pe/docinteres/leyes/tercera_seccion.htm)
5. RUBIO Correa, Marcial, El sistema jurídico: introducción al derecho, 8va edición corregida y aumentada, PUCP Fondo Editorial, Lima, 2006, ISBN: 848930985X.
6. Código Procesal Civil, Lima: Gaceta Jurídica, 2007, ISBN: 9789972208720.
7. Rodríguez EM. Conceptos sobre procesos Judiciales[Internet]. Estudio Rodrigo, Elías & Medrano. Agosto 2013 [Revisado el 03 de enero del 2015]. Disponible desde: <http://www.estudiorodrigo.com/home.htm>
8. Fernández Alarcón, V. (2006). Desarrollo de Sistemas de Información: Una Metodología Basada en el Modelado (Primera Edición ed.). Barcelona:

Ediciones UPC.

9. Sistema de Información en las Empresas. [En línea].; 2014 [Fecha de Acceso 12 Febrero 2015]. URL disponible en: http://sistemas-de-de-informacion.blogspot.com/p/actividades-basicas-de-un-sistema-de_06.html
10. Sistemas de información gerencial. Séptima edición, Raymond McLeod Jr. México, 2000, Prentice Hall Hispanoamericana S.A.
11. Analisis y Diseño de Sistemas. Sexta Edición, Kenneth E. Kendall. New Jersey, 2005, University School Of New Jersey
12. Rodriguez GM. Base de Datos. [Internet] Definiciones ABC. Abril 2014. [Recuperado 04 de enero del 20015]. Disponible desde:
<http://www.definicionabc.com/tecnologia/base-de-datos.php>
13. Aguiniano Moirales JD. Características y Tipos de Base de datos [Internet]. Junio del 2014. [Recuperado 14 de diciembre del 2015]. Disponible desde:
https://www.ibm.com/developerworks/ssa/data/library/tipos_bases_de_datos/ 14.
- Microsoft. Fundamentos y normalización de base de datos. [Internet]. Enero 2013. [Recuperado 10 de diciembre del 2015]. Disponible desde:
<https://support.microsoft.com/es-es/kb/283878>
15. Rafael Barzanalla. Ingeniería del Software, Metodología de desarrollo. [Internet]. Enero 2014. [Recuperado 10 de diciembre del 2015]. Disponibe desde:
<http://www.um.es/docencia/barzana/IAGP/IAGP2-Metodologias-dedesarrollo.html>
16. Meza Miloslavvic. Metodología de desarrollo de Software RUP. [Trabajo de Graduación Para Optar el Grado de Ing. de Ciencias de la Computación]. El

- Salavador: Universidad Don Bosco. 2009. Disponible desde:
http://www.academia.edu/13247296/MONOGRAFIA_SOBRE_LA_METODOLOGIA_DE_DESARROLLO_DE_SOFTWARE_RUP
17. Moya Moreno J.J. Nuevos Enfoques de la Ingeniería del Software [Internet]. DocPlayer. Diciembre del 2015. [Recuperado 04 de enero del 2016]. Disponible desde: <http://docplayer.es/9901844-Capitulo-i-nuevos-enfoques-de-la-ingenieriade-software.html>
 18. Sánchez E. Fases RUP. [En línea].; 2012 [Fecha de Acceso 10 Marzo 2015]. URL disponible en: <http://es.slideshare.net/syboldin/rup-11345894>.
 19. Rueda J. Aplicación de la Metodología RUP para el desarrollo rápido de aplicaciones basado en el estandar J2EE. [En línea].; 2006 [Fecha de Acceso 21 Febrero 2015]. URL disponible en: http://biblioteca.usac.edu.gt/tesis/08/08_0308_CS.pdf
 20. Ecured. Ecured Conocimiento con todos y para todos. [En línea].; 2010 [Fecha de Acceso 2 Marzo 2015]. URL disponible en: http://www.ecured.cu/index.php/Kent_Beck
 21. Auer K y Miller R. Extreme Programming Applied. Primera Edición ed. Boston: Addison Wesley; 2001
 22. Microsoft C. Microsoft Develoér Network. [En línea].; 2012 [Fecha de Acceso 12 Marzo 2015]. URL disponible en: <https://msdn.microsoft.com/es-es/library/jj161047.aspx>.
 23. INTRODUCCION AL PROYECTO DE INGENIERIA. Enrique Villamil García,

Ingeniero Universidad Nacional de Buenos Aires Instituto Tecnológico de Buenos Aires Miguel J. García Hernández, Ph. D. Universidad Politécnica de Cataluña.
Buenos Aires, diciembre de 2003

24. Gomez E. El Proceso Unificado de Desarrollo de Software (RUP). [En línea].; 2010 [Fecha de Acceso 22 Febrero 2015]. URL disponible en:
<http://moleculax.blogspot.com/2008/06/el-proceso-unificado-de-desarrollode.html>
25. N Z. Acerca de UML: El lenguaje de modelación unificada. [En línea].; 1977 [Fecha de Acceso 28 Febrero 2015]. URL disponible en:
http://www2.scielo.org.ve/scielo.php?script=sci_arttext&pid=S1012-70892008000200004&lng=pt&nrm=i
26. Ecured. Lenguajes de Programación [Internet]. Ecured. Enero 2014. [Recuperado Diciembre 2015]. Disponible desde:
http://www.ecured.cu/Lenguaje_de_Programaci%C3%B3n
27. Microsft. Asp y Asp.Net [Internet]. Microsft. Marzo del 2014. [Recuperado: 16 de enero del 2016.]. Disponible desde:
[https://msdn.microsoft.com/eses/library/4w3ex9c2\(v=vs.100\).aspx](https://msdn.microsoft.com/eses/library/4w3ex9c2(v=vs.100).aspx)
28. Alberto Cardona. Servelts y JSP. [Internet]. Programación.Net. Mayo 2011. [Recuperado: 01 de enero del 2016]. Disponible desde:
http://programacion.net/articulo/servlets_y_jsp_82
29. Pérez, V., D. Los diferentes lenguajes de programación para la web. [Monografía en internet]. Bolivia. 2007, Noviembre. Disponible en:

<http://www.maestrosdelweb.com/editorial/los-diferentes-lenguajes-de-programacion-para-la-web/>

30. Álvarez, G. Sistema de gestión de base de datos. [Monografía en Internet]. Venezuela, 2009, Diciembre. Disponible en: <http://es.slideshare.net/jeissonlarry/sistema-gestin-de-bases-de-datos>
31. Mendoza, J., H., Fundamentos de la gestión de base de datos. [Monografía en Internet]. Mexico, 2011, Junio. Disponible en: <http://www.monografias.com/trabajos56/sistemas-bases-de-datos/sistemas-basesde-datos2.shtml>
32. Desarrollo Web. ¿Qué es MYSQL?. [Internet]. EsepEstudio. Agosto 2005- [Recuperado el 12 de diciembre del 2015]. Disponible desde: <http://www.espestudio.com/noticias/que-es-mysql>
33. Wikipedia. Microsoft SQL Server. [Internet]. Wikipedia. Enero del 2014. [Recuperado 12 de diciembre del 2015]. Disponible desde: https://es.wikipedia.org/wiki/Microsoft_SQL_Server
34. Oracle. Oracle Soluciones en respaldo de Datos. [Internet]. Noviembre 2014. [Recuperado 10 de noviembre del 2015]. Disponible desde: <https://www.google.com/search?q=sql+server&ie=utf-8&oe=utf-8#q=oracle>
35. Wikipedia. POstgreSQL. [Internet]. Wikipedia. Mayo del 2015. [Recuperado 12 de diciembre del 2015]. Disponible desde:

ANEXOS

Presupuesto

De igual forma en cuanto a Software no contó con los principales softwares de desarrollo y se adquirió 3 de estos a versión prueba para realizar el piloto de prueba siendo el costo igual a cero, pero para la validez del uso se debe adquirir más adelante.

NetBeans IDE	versión de prueba	Costo 0-00 soles	mysql
server	versión de prueba	Costo 0-00 soles	
Rational Rose family	versión de prueba	Costo 0-00 soles	

ANEXO N° 01

Guía de Entrevista a los Abogados del Estudio

Jurídico SALASCALDERON&ASOCIADOS de la Ciudad de Ica

1. ¿Actualmente cómo se está desarrollando la gestión de los procesos judiciales en el Estudio Jurídico?
2. ¿Está usted conforme con la actual gestión de los proceso judiciales que se desarrolla en el Estudio Jurídico?
3. ¿Cuál cree usted que son los principales problemas en la gestión de los procesos judiciales?

4. ¿Qué tan frecuente se suscitan estos problemas?
5. ¿Cree usted que estos problemas afectan el servicio a los clientes?
6. ¿Le gustaría contar con una herramienta que le ayudará a solucionar estos problemas que tiene?

ANEXO N° 02

Imágenes del Estudio Jurídico SALASCALDERON & ASOCIADOS de la Ciudad de Ica









Estudio de abogados
Salas Calderón & Asociados
RUC.- 10215601868

“Año De La Consolidación Del Mar De Grau”

LOS ABOGADOS QUE SUSCRIBEN, HABILITADOS EN EL EJERCICIO DE LA DEFENSA, CON CARNET N°2541 DEL COLEGIO DE ABOGADOS DE ICA

Que el ciudadano MELENDEZ PEREZ, JORGE DANIEL, identificado con DNI N° 42715431, con código un Universitario N° 7009092001 Boltero Católico se le ha autorizado realizar sus prácticas de Elaboración de tests sobre diseño e implementación de un sistema de apoyo a estudio jurídico Salas Calderón y Asociados de la ciudad de Ica del estudio se representó en esta oficina legal desde el día 07 de diciembre del 2015 en horario de las 8:00 horas hasta las 13:00 horas encargándose de realizar las indagaciones que considere pertinentes en los expedientes de las diversas materias así como tipos de los escritos de las demandas denuncias notificaciones y cuanto documento a realizarse dentro de las obligaciones del estudio jurídico siendo merecedor a las facilidades pertinentes habiendo demostrado puntualidad dedicación responsabilidad disciplina y sobre todo honestidad

Se expide la presente certificación en honor a la verdad y los hechos suscitados.

En la ciudad de Ica a los 29 días de Enero del 2016

Atte.


Eugenio J. Torres Romazo
N° 1960
RUC. 10215601868


Marcelo M. Lora Quijada
ABOGADO
C.A.T. 2097 - RUC. 1499

ANEXO N° 06

Caratulas de los expedientes físicos

J.P. 1114-2010
SECRET. CARLOS
BRANDIA
2 DO CIVIL ICA
DEMANDANTE: OFELIA
HUAMANI
CABRERA
DEMANDADO: JOSE
MENDOZA
HERNANDEZ
MATERIA: COFOBRI
NULIDAD DE

Fuente: El investigador, Perú, 2015

Fuente: El investigador, Perú, 2015

-Parcona.
Jackeline Angelica
Apcho Soto
Alimentos

ANEXO N° 07

Modelo de caratula de los Expedientes

ESTUDIO JURÍDICO	
Código:	M - 16
Datos Generales	
N° de Expediente:	2011-01442
Fecha Apertura:	
Fecha Admisión:	
Estado Judicial:	DEMANDADO
Contraparte:	CULE QUISPE JOSE
Datos Judiciales	
Tipo Materia:	Penal
Juzgado:	SEGUNDO JUZGADO DE INVESTIGACION
Encargado:	ENEYDA CONTRERAS ZAMORA
Clientes	
AMORETTI HUAMANI EDIMIRSON JHONATAN	
AGUADO GOMEZ PEDRO ENRIQUE	
Código:	M - 16

131

GLOSARIO DE TÉRMINOS

Diagrama	La presentación gráfica de un conjunto de elementos, usualmente representado como un grafo conectado de vértices y arcos.
Flujo de trabajo	Realización de un caso de uso o parte de él puede describirse en términos de diagramas de actividad que incluye a los trabajadores participantes, las actividades que realiza y los artefactos que produce.
Gestión	Por los trámites o diligencias necesarios para resolver un asunto. También refiere a dirigir, gobernar, disponer y organizar en orden a lograr un objetivo propuesto. Entonces una gestión puede estar orientada a resolver un problema específico.
Herramientas CASE	Las herramientas CASE (ComputerAidedSoftwareEngineering, Ingeniería de Software Asistida por Computadora) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costos de las mismas en términos de tiempo y de dinero.

13
2

Heterocomposición	Es la solución del conflicto por un tercero o la voluntad de este, en muchos casos aun en contra de la voluntad de las partes.
Hito	Es una tarea de duración cero que simboliza el haber conseguido un logro importante en el proyecto. Los hitos son una forma de conocer el avance del proyecto sin estar familiarizado con el proyecto y constituyen un trabajo de duración cero porque simbolizan un logro, un punto, un momento en el proyecto.
Metodología	Hace referencia al conjunto de procedimientos basados en principios lógicos, utilizados para alcanzar una gama de objetivos que rigen en una investigación científica.
Modelado	Procesos de abstracción en el modelamiento. Modelo de dato. Modelos conceptuales, lógicos y físicos. Modelo entidad relación extendido. Formalismo individual. Modelo relacional. Modelos orientados al objeto.
UML	Es UnifiedModelingLanguage, es un lenguaje (una forma de escribir y de modelar), es un lenguaje de modelado de datos, nos va a servir para modelar el sistema.
Rational Rose v 7	Es una herramienta de diseño unificada orientada al objeto del software de la lengua que modela (UML) prevista para

	modelar de la representación visual y la construcción componente de los usos del software del empresarial.
Requerimiento	Es una necesidad documentada sobre el contenido, forma o funcionalidad de un producto o servicio.