Contenido

[CAPITULO I 1](#_Toc392746395)

[PERFIL 1](#_Toc392746396)

[INTRODUCCION 1](#_Toc392746397)

[ANTECEDENTES 2](#_Toc392746398)

[DESCRIPCION DEL PROBLEMA 2](#_Toc392746399)

[OBJETIVOS 3](#_Toc392746400)

[Objetivo General 3](#_Toc392746401)

[Objetivo Especifico 3](#_Toc392746402)

[JUSTIFICACION 3](#_Toc392746403)

[ALCANCE 4](#_Toc392746404)

[MODULO REGISTRO 4](#_Toc392746405)

[MODULO DOCUMENTOS 4](#_Toc392746406)

[MODULO TRÁMITE 4](#_Toc392746407)

[MODULO SISTEMA Y SEGURIDAD 4](#_Toc392746408)

[METODOLOGIA 5](#_Toc392746409)

[CAPITULO II 6](#_Toc392746410)

[MARCO TEORICO 6](#_Toc392746411)

[ESTUDIO JURIDICO “MIGUEL MORON” 6](#_Toc392746412)

[HERRAMIENTAS CASE 6](#_Toc392746413)

[Rational Rose 6](#_Toc392746414)

[LENGUAJE DE PROGRAMACION 7](#_Toc392746415)

[Java 7](#_Toc392746416)

[GESTOR DE BASE DE DATOS 7](#_Toc392746417)

[MySQL 7](#_Toc392746418)

[CAPITULO III 9](#_Toc392746419)

[CAPTURA DE REQUISITOS 9](#_Toc392746420)

[INTRODUCCION 9](#_Toc392746421)

[REQUISITOS FUNCIONALES 9](#_Toc392746422)

[REQUISITOS NO FUNCIONALES 10](#_Toc392746423)

[IDENTIFICAR CLASES DE DOMINIO 10](#_Toc392746424)

[DIAGRAMA DE DOMINIO INICIAL 11](#_Toc392746425)

[IDENTIFICAR CASOS DE USO 11](#_Toc392746426)

[IDENTIFICAR ACTORES 11](#_Toc392746427)

[DESCRIPCION DE LOS CASOS DE USO 13](#_Toc392746428)

[**1.** **Caso de uso – INICIAR SESION** 13](#_Toc392746429)

[**2.**  **Caso de uso – REGISTRAR SECRETARIA** 14](#_Toc392746430)

[**3. Caso de uso – REGISTRAR ABOGADO** 15](#_Toc392746431)

[**4. Caso de uso – REGISTRAR TITULO** 16](#_Toc392746432)

[**5. Caso de uso – REGISTRAR TELEFONO** 17](#_Toc392746433)

[**6. Caso de uso – REGISTRAR PERSONA** 18](#_Toc392746434)

[**7. Caso de uso – REGISTRAR PLANTILLA** 19](#_Toc392746435)

[**8. Caso de uso – REGISTRAR ARANCEL** 20](#_Toc392746436)

[**9. Caso de uso – REGISTRAR DOCUMENTO EMITIDO** 21](#_Toc392746437)

[**10. Caso de uso – REGISTRAR CASO** 22](#_Toc392746438)

[**11. Caso de uso – REGISTRAR TIPO DOCUMENTO** 23](#_Toc392746439)

[**12. Caso de uso – REGISTRAR DOCUMENTO** 24](#_Toc392746440)

[**13. Caso de uso – REGISTRAR AUTORIDAD Y CARGO** 25](#_Toc392746441)

[**14. Caso de uso – REGISTRAR CITA** 26](#_Toc392746442)

[**15. Caso de uso – REGISTRAR ADMINISTRADOR** 27](#_Toc392746443)

[**16. Caso de uso – GENERAR BACKUP** 28](#_Toc392746444)

[**17. Caso de uso – REALIZAR ROLLBACK** 29](#_Toc392746445)

[DIAGRAMA GENERAL DEL MODELO DE CASO DE USO 29](#_Toc392746446)

[DIAGRAMA GENERAL DE CASO DE USO 30](#_Toc392746447)

[CAPITULO IV 31](#_Toc392746448)

[ANALISIS DE SISTEMA 31](#_Toc392746449)

[ANALISIS DE ARQUITECURA 31](#_Toc392746450)

[DIAGRAMA DE PAQUETES 32](#_Toc392746451)

[DIAGRAMAS DE COLABORACION – ANALISIS 33](#_Toc392746452)

[**1.** **Diagrama de Colaboración – INICIAR SESION** 33](#_Toc392746453)

[**2.** **Diagrama de Colaboración – REGISTRAR SECRETARIA** 33](#_Toc392746454)

[**3.** **Diagrama de Colaboración – REGISTRAR ABOGADO** 33](#_Toc392746455)

[**4.** **Diagrama de Colaboración – REGISTRAR TELEFONO** 33](#_Toc392746456)

[**5.** **Diagrama de Colaboración – REGISTRAR PERSONA** 34](#_Toc392746457)

[**6.** **Diagrama de Colaboración – REGISTRAR PLANTILLA** 34](#_Toc392746458)

[**7.** **Diagrama de Colaboración – REGISTRAR ARANCEL** 34](#_Toc392746459)

[**8.** **Diagrama de Colaboración – REGISTRAR DOCUMENTO EMITIDO** 35](#_Toc392746460)

[**9.** **Diagrama de Colaboración – REGISTRAR CASO** 35](#_Toc392746461)

[**10.** **Diagrama de Colaboración – REGISTRAR TIPO DOCUMENTO** 35](#_Toc392746462)

[**11.** **Diagrama de Colaboración – REGISTRAR DOCUMENTO** 35](#_Toc392746463)

[**12.** **Diagrama de Colaboración – REGISTRAR AUTORIDAD** 36](#_Toc392746464)

[**13.** **Diagrama de Colaboración – REGISTRAR CITA** 36](#_Toc392746465)

[**14.** **Diagrama de Colaboración – REGISTRAR ADMINISTRADOR** 36](#_Toc392746466)

[**15.** **Diagrama de Colaboración – GENERAR BACKUP** 36](#_Toc392746467)

[**16.** **Diagrama de Colaboración – REALIZAR ROLLBACK** 37](#_Toc392746468)

[CAPITULO V 38](#_Toc392746469)

[DIAGRAMAS FINALES 38](#_Toc392746470)

[DIAGRAMA DE DOMINIO FINAL 38](#_Toc392746471)

[DIAGRAMA DE SECUENCIA DE DISEÑO 39](#_Toc392746472)

[**1.** **Diagrama de Secuencia – INICIAR SESIÓN** 39](#_Toc392746473)

[**2.** **Diagrama de Secuencia – REGISTRAR SECRETARIA** 39](#_Toc392746474)

[**3.** **Diagrama de Secuencia – REGISTRAR ABOGADO** 40](#_Toc392746475)

[**4.** **Diagrama de Secuencia – REGISTRAR TELEFONO** 40](#_Toc392746476)

[**5.** **Diagrama de Secuencia – REGISTRAR PERSONA** 41](#_Toc392746477)

[**6.** **Diagrama de Secuencia – REGISTRAR PLANTILLA** 41](#_Toc392746478)

[**7.** **Diagrama de Secuencia – REGISTRAR ARANCEL** 42](#_Toc392746479)

[**8.** **Diagrama de Secuencia – REGISTRAR DOCUMENTO EMITIDO** 42](#_Toc392746480)

[**9.** **Diagrama de Secuencia – REGISTRAR CASO** 43](#_Toc392746481)

[**10.** **Diagrama de Secuencia – REGISTRAR TIPO DOCUMENTO** 43](#_Toc392746482)

[**11.** **Diagrama de Secuencia – REGISTRAR DOCUMENTO** 44](#_Toc392746483)

[**12.** **Diagrama de Secuencia – REGISTRAR AUTORIDAD** 44](#_Toc392746484)

[**13.** **Diagrama de Secuencia – REGISTRAR CITA** 45](#_Toc392746485)

[**14.** **Diagrama de Secuencia – REGISTRAR ADMINISTRADOR** 45](#_Toc392746486)

[**15.** **Diagrama de Secuencia – GENERAR BACKUP** 46](#_Toc392746487)

[**16.** **Diagrama de Secuencia – REALIZAR ROLLBACK** 46](#_Toc392746488)

[DESARROLLO DEL SISTEMA 47](#_Toc392746489)

[INTERFACES 47](#_Toc392746490)

[CREACION DE BD EN MYSQL 49](#_Toc392746491)

# CAPITULO I

# PERFIL

## INTRODUCCION

Desde hace muchos años atrás la información ha sido el pilar fundamental para la toma de decisiones basado en un histórico de datos almacenados con el transcurrir del tiempo se fue mejorando la manipulación de información mediante inventario, el cual se manejaba en hoja de cuadernos, además que la información resguardada proporciona un sin fin de resultado útiles para la persona que maneja la misma

Los sistemas de información administrativa están volviéndose indispensables a gran velocidad para la planificación, la toma de decisiones y el control de la información de datos. La velocidad y exactitud con lo que los directivos pueden recibir información sobre lo que está funcionando bien o lo que está funcionando mal determinaran, en gran medida, la eficacia que tendrá el sistema de control.

Los estudios jurídicos han llegado a formar parte de nuestra de nuestra vida, preservando nuestra seguridad jurídica, tanto personal como empresarial, atendiendo casos que van desde familiares, civiles, agrarios, penales, etc.

Todos alguna vez han tenido la necesidad de un abogado, es por ello que los estudios jurídicos están en constante actividad y a medida de que la población aumenta la cantidad de clientes para ellos.

## 

## ANTECEDENTES

El estudio Jurídico “Miguel Morón y Asoc.” se inicio el 15 de febrero del 2005 teniendo como dueño al Dr. Miguel Morón, el solo inicio este emprendimiento,

Su emprendimiento inició con solo tramites agrarios, actualmente ha incrementado a tramites civiles, familiares, penales.

La oficina está ubicada en la Calle Comercio entre Calle Bush y Tente. Cnel. Sánchez.

## DESCRIPCION DEL PROBLEMA

En la actualidad el estudio Jurídico “Miguel Morón y Asoc.”, presenta los siguientes inconvenientes.

• Mantiene todos los datos de los clientes en un sistema de archivo, tradicional, pudiendo perderse o malograrse el documento.

• El tiempo de búsqueda de los documentos es demasiado largo y tedioso.

• No se cuenta con un registro de todos los trámites realizados por el estudio jurídico.

• Se tiene dificultad en el manejo de la agenda de los abogados ya que los datos de ellos no están bien sincronizados

• No se tiene un registro sobre los clientes, y sus datos personales, de esta manera es un poco dificultoso el obtener contacto con ellos, en caso de ser necesario.

• No se tiene datos actuales sobre los distintos jueces y juzgados del área, y así tener una mejor idea de cómo se desarrollara el caso.

• No se tiene un control de los aranceles y el control de pago de los clientes, quedando muchas veces el cliente con las dudas sobre los desembolsos hechos.

## OBJETIVOS

### Objetivo General

Desarrollar un sistema de información de administración para el Estudio Jurídico “Miguel Morón” de la ciudad de Camiri.

### Objetivo Especifico

• Recopilar la información necesaria para conocer el funcionamiento y la actividad del estudio jurídico, usando entrevistas, encuestas.

* Definir los requerimientos del sistema, utilizando herramientas para la recolección de información, como ser entrevistas, encuestas, observaciones.

• Análisis de los requisitos del sistema, se procederá al análisis de estos requisitos, tratando de enfocar las soluciones informáticas que estén a la altura de estos requisitos.

• Diseñar un sistema de información que cumpla los requisitos, utilizando las herramientas del Proceso Unificado de Desarrollo de Software.

• Desarrollo del sistema de información y escritura de código, se utilizará un lenguaje de programación y un gestor de base de datos que pueda satisfacer los requerimientos del sistema.

• Pruebas y Simulación al sistema

## JUSTIFICACION

Mediante el presente proyecto se pretende mejorar el manejo de información que se utiliza en el estudio jurídico así también como una mejor administración del sistema de archivos, catalogar a los distintos casos y procesos que tiene un cliente, al mismo tiempo llevar un mejor control de los ingresos por los distintos servicios que se brindan, para así brindar un mejor servicio a los clientes.

## ALCANCE

## MODULO REGISTRO

• Registrar Persona

• Registrar Cliente

• Registrar Usuario

• Registrar Autoridad

• Registrar Cargo

• Registrar Título

• Registrar Teléfono

## MODULO DOCUMENTOS

• Registrar Documento

• Registrar Documento Emitido

• Registrar Tipo Documento

• Registrar Plantilla

## MODULO TRÁMITE

• Registrar Caso

• Registrar Tipo Caso

• Registrar Cita

## MODULO SISTEMA Y SEGURIDAD

• Gestionar Respaldo y RollBack

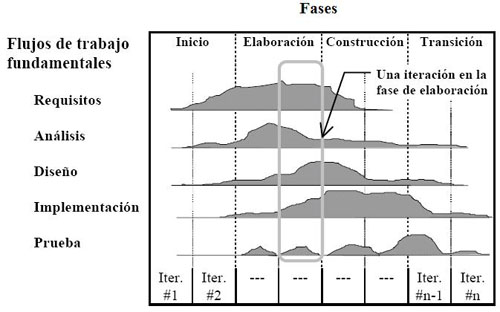
• Gestionar Mantenimiento y Seguridad

## METODOLOGIA

Se utilizara el Proceso Unificado de Desarrollo de Software en la implementación del Sistema de Información para el Estudio Jurídico “Miguel Morón” este proceso tiene un conjunto de actividades necesarias para l transformar los requisitos del cliente en software.

El Proceso Unificado tiene las fases de inicio, elaboración, construcción, transición en las cuales se realizan iteraciones que abarcan los flujos de trabajo, requisitos, análisis, diseño implementación y prueba.

Se utilizara los artefactos que proporciona UML (Lenguaje Unificado de Modelado) para el modelado del sistema.



# CAPITULO II

# MARCO TEORICO

## ESTUDIO JURIDICO “MIGUEL MORON”

Se inicio este emprendimiento el 15 de febrero del 2005, motivado por su padre y amigos del dueño (Miguel Ángel Morón) en que debía empezar un emprendimiento personal a medida de que comenzó en el primer trimestre el ya tenía una cantidad importante de clientes, debido a la gran amistad que tiene con personas del mismo medio, lo cual lo motivo aun mas estableció su oficina.

En la actualidad planea abrir un Buffette de Abogados, por la buena demanda de clientes, para ello se está asesorando de la mejor manera para tener mejores resultados.

## HERRAMIENTAS CASE

Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y dinero. Estas herramientas no pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, calculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.

Sistema de software que intenta proporcionar ayuda automatizada a las actividades del proceso de software.

En tal caso para el desarrollo e implementación del Sistema para el Estudio Jurídico “Miguel Morón” se utilizara la siguiente herramienta case:

### Rational Rose

Rational Software es actualmente conocida como una familia del software de IBM para el despliegue, diseño, construcción, pruebas, administración de proyectos en el proceso de desarrollo de software.

## LENGUAJE DE PROGRAMACION

### Java

Java es un lenguaje de programación con el que podemos realizar cualquier tipo de programa. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Está desarrollado por la compañía Sun Microsystems con gran dedicación y siempre enfocado a cubrir las necesidades tecnológicas más punteras.

Una de las principales características por las que Java se ha hecho muy famoso es que es un lenguaje independiente de la plataforma. Eso quiere decir que si hacemos un programa en Java podrá funcionar en cualquier ordenador del mercado. Es una ventaja significativa para los desarrolladores de software, pues antes tenían que hacer un programa para cada sistema operativo, por ejemplo Windows, Linux, Apple, etc. Esto lo consigue porque se ha creado una Máquina de Java para cada sistema que hace de puente entre el sistema operativo y el programa de Java y posibilita que este último se entienda perfectamente.

La independencia de plataforma es una de las razones por las que Java es interesante para Internet, ya que muchas personas deben tener acceso con ordenadores distintos. Pero no se queda ahí, Java está desarrollándose incluso para distintos tipos de dispositivos además del ordenador como móviles, agendas y en general para cualquier cosa que se le ocurra a la industria.

## GESTOR DE BASE DE DATOS

### MySQL

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 interactuación 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.

# CAPITULO III

# CAPTURA DE REQUISITOS

## INTRODUCCION

La captura de requisitos es el pilar fundamental para el desarrollo de un sistema, por ser la base, el inicio de un trabajo que ir evolucionando según los resultados de este flujo de trabajo, nos ayuda a encontrar, priorizar, detallar y estructurar el modelo de los casos de uso. Obteniendo resultados según requisitos funcionales de los usuarios del software.

## REQUISITOS FUNCIONALES

La captura de los requisitos funcionales se realiza por medio de entrevistas al personal del Estudio Jurídico “Miguel Morón” de acuerdo a la actividad que realiza cada persona (actor), y a las necesidades de cada actor tiene el desempeño de su actividad laboral. A Continuación se presenta la lista de requisitos:

|  |  |
| --- | --- |
| REQUISITOS | ACTOR |
| LOGIN | USERS |
| REGISTRAR SECRETARIA | ADMINISTRADOR-ABOGADO |
| REGISTRAR ABOGADO | ADMINISTRADOR-ABOGADO |
| REGISTRAR TITULO | USERS |
| REGISTRAR TELEFONO | USERS |
| REGISTRAR PERSONA | USERS |
| REGISTRAR PLANTILLA | ABOGADO |
| REGISTRAR ARANCEL | ABOGADO |
| REGISTRAR DOCUMENTO EMITIDO | ABOGADO |
| REGISTRAR CASO | ABOGADO |
| REGISTRAR TIPO DOCUMENTO | ABOGADO-SECRETARIA |
| REGISTRAR DOCUMENTO | ABOGADO-SECRETARIA |
| REGISTRAR AUTORIDAD | ABOGADO-SECRETARIA |
| REGISTRAR CARGO | ABOGADO-SECRETARIA |
| REGISTRAR CITA | SECRETARIA |
| REGISTRAR ADMINISTRADOR | ADMINISTRADOR |
| GENERAR BACKUP | ADMINISTRADOR |
| REALIZAR ROLLBACK | ADMINISTRADOR |

## REQUISITOS NO FUNCIONALES

Los usuarios del sistema requieren de un software que cumpla con los requisitos no funcionales; los cuales se describen a continuación:

* El sistema debe funcionar correctamente para proporcionar a los usuarios eficiencia en el registro de datos y recuperación de datos en reportes (corrección).
* El software debe poseer fiabilidad para asegurar la veracidad de los datos especialmente los datos que comprenden datos personales y documentos de los clientes del estudio jurídico (fiabilidad).
* Debe existir eficiencia en el sistema realizando creación dinámica de formularios para ahorrar recursos de memoria (eficiencia).
* El software debe ser desarrollado con una documentación que pueda ser entendida para realizar cambios o mejoras que se requieran en un futuro; el sistema debe ser desarrollado con un estándar de codificación (mantenimiento).
* Realizar un control de usuario para el ingreso al sistema según su perfil de usuario.

## IDENTIFICAR CLASES DE DOMINIO

En la base de requisitos encontrados se procede a identificar las clases del dominio del sistema las cuales son Arancel, Autoridad, Backup, Cargo, Caso, Cita, Documento, Documento Emitido, Persona, Plantilla, Teléfono, TipoCaso, Tipo Documento, Titulo y Usuario.

## DIAGRAMA DE DOMINIO INICIAL



## IDENTIFICAR CASOS DE USO

Los casos de uso del sistema fueron encontrados según los requisitos de los diferentes usuarios del sistema en la base a los datos que requieren guardar o adquirir, los cuales se detallan en el cuadro.

## IDENTIFICAR ACTORES

Los actores que se encontraron en el desarrollo del software se especifican a continuación:

* **PERSONA.-** Es la agrupación de actores, o la mínima expresión de ellos.
* **CLIENTE.-** Es el actor principal en el registro de las transacciones que se llevan a cabo en el sistema.
* **SECRETARIA.-** Es el encargado de la recepción de archivos del abogado, al mismo tiempo tiene la función de registrar lo que este a su alcance.
* **ABOGADO.-** Tiene el rol fundamental en la toma de decisiones de las actividades dentro del estudio jurídico. Tales como la atención de un cliente, la apertura de un caso, o la emisión de un documento.
* **ADMINISTRADOR.-** El administrador tiene el rol de administrar el sistema siendo la única persona que tiene acceso al sistema.

## DESCRIPCION DE LOS CASOS DE USO

1. **Caso de uso – INICIAR SESION**

****

|  |  |
| --- | --- |
| **Nombre** | **Login** |
| **Actores** | USERS |
| **Iniciador del caso de uso** | Todos |
| **Propósito** | Con este indicador se pretende asignar el área de trabajo para cada usuario distinto. |
| **Resumen** | El usuario ingresa al sistema con sus datos personales y secretos para poder acceder a su panel de usuario |
| **Pre-Condición** | No debe haber hecho Login antes |
| **Flujo de Trabajo** | El usuario ingresa sus datos personales a la interfaz  La interfaz consulta con la capa de negocio  La capa de negocio ejecuta la consulta para poder acceder a la base de datos.  La base de datos devuelve los datos pedidos por la capa de negocio  La capa de negocio evalúa si los parámetros coinciden  Salta la interfaz correspondiente para el usuario si es cierta la condición previa. |

**2. Caso de uso – REGISTRAR SECRETARIA**



|  |  |
| --- | --- |
| **Nombre** | **REGISTRAR SECRETARIA** |
| **Actores** | ABOGADO – ADMINISTRADOR |
| **Iniciador del caso de uso** | ABOGADO –ADMINISTRADOR |
| **Propósito** | Se añade como usuario del sistema a una secretaria que tendrá este rango y responsabilidad. |
| **Resumen** | Los rangos superiores a secretaria pueden añadir a una secretaria, en caso de que RRHH lo admita |
| **Pre-Condición** | No debe estar registrada en la base de datos |
| **Flujo de Trabajo** | Se registra como persona previamente.  Ingresa los datos por la interfaz y esta envía los datos a la capa de negocio.  La capa de negocio evalúa si ya está registrada, de serlo indica un mensaje, sino ingresa los nuevos datos a la BD. |

**3. Caso de uso – REGISTRAR ABOGADO**

****

|  |  |
| --- | --- |
| **Nombre** | **REGISTRAR ABOGADO** |
| **Actores** | ABOGADO – ADMINISTRADOR |
| **Iniciador del caso de uso** | ABOGADO –ADMINISTRADOR |
| **Propósito** | Se añade como usuario del sistema a un abogado que tendrá este rango y responsabilidad. |
| **Resumen** | Los rangos iguales o superiores pueden añadir a un abogado. |
| **Pre-Condición** | No debe estar registrado en la base de datos |
| **Flujo de Trabajo** | Se registra como persona previamente.  Se ingresa por la interfaz los datos a la capa de negocio.  La capa de negocio evalúa si ya está registrada, de serlo indica un mensaje, sino ingresa los nuevos datos a la BD. |

**4. Caso de uso – REGISTRAR TITULO**

****

|  |  |
| --- | --- |
| **Nombre** | **REGISTRAR TITULO** |
| **Actores** | Todos |
| **Iniciador del caso de uso** | Todos |
| **Propósito** | Se guarda un registro de todos los posibles títulos que una persona tenga. Cabe destacar que una persona solo puede tener activo un titulo a la vez. |
| **Resumen** | El User al momento de registrar una persona establece su titulo. |
| **Pre-Condición** | El titulo no debe existir en la lista de la base de datos. |
| **Flujo de Trabajo** | El usuario busca si existe el titulo a añadir  Si el Titulo no existe, a través de la interfaz se añade un nuevo Titulo a la BD. |

**5. Caso de uso – REGISTRAR TELEFONO**

****

|  |  |
| --- | --- |
| **Nombre** | **REGISTRAR TELEFONO** |
| **Actores** | Todos |
| **Iniciador del caso de uso** | Todos |
| **Propósito** | Se pretende obtener una agenda telefónica. |
| **Resumen** | Cada persona tiene su número telefónico. Y se asocia ese nombre con el número de teléfono y se ingresa a la BD. |
| **Pre-Condición** | No debe estar registrado |
| **Flujo de Trabajo** | Los Users piden el dato telefónico a las personas a registrar  Mediante la interfaz se ingresa a la capa de negocios  La capa de negocio revisa que si sea viable registrarlo y lo hace. |

**6. Caso de uso – REGISTRAR PERSONA**

****

|  |  |
| --- | --- |
| **Nombre** | **REGISTRAR PERSONA** |
| **Actores** | Todos |
| **Iniciador del caso de uso** | Todos |
| **Propósito** | Crear un nuevo cliente en la BD |
| **Resumen** | Se crea un cliente con sus datos personales y referenciales. |
| **Pre-Condición** | No debe estar Registrado |
| **Flujo de Trabajo** | Los Users piden el los datos personales a las personas a registrar  Mediante la interfaz se ingresa a la capa de negocios  La capa de negocio revisa que si sea viable registrarlo y lo hace. |

**7. Caso de uso – REGISTRAR PLANTILLA**

****

|  |  |
| --- | --- |
| **Nombre** | **REGRISTAR PLANTILLA** |
| **Actores** | Abogado |
| **Iniciador del caso de uso** | Abogado |
| **Propósito** | Registra una nueva planilla de documento |
| **Resumen** | Se añade un nuevo ítem a la plantilla |
| **Pre-Condición** | No debe estar Registrado |
| **Flujo de Trabajo** | El abogado considera que es necesario añadir un nuevo modelo o plantilla y lo añade  Mediante la interfaz se ingresa a la capa de negocios  La capa de negocio envía la ejecución a la base de dato. |

**8. Caso de uso – REGISTRAR ARANCEL**

****

|  |  |
| --- | --- |
| **Nombre** | **REGRISTAR ARANCEL** |
| **Actores** | Abogado |
| **Iniciador del caso de uso** | Abogado |
| **Propósito** | Registra un nuevo arancel |
| **Resumen** | Se añade un nuevo ítem a la tabla arancel |
| **Pre-Condición** | No debe estar Registrado |
| **Flujo de Trabajo** | El abogado considera que es necesario añadir un nuevo arancel y lo añade  Mediante la interfaz se ingresa a la capa de negocios  La capa de negocio envía la ejecución a la base de datos. |

**9. Caso de uso – REGISTRAR DOCUMENTO EMITIDO**

****

|  |  |
| --- | --- |
| **Nombre** | **REGISTRAR DOCUMENTO EMITIDO** |
| **Actores** | Abogado |
| **Iniciador del caso de uso** | Abogado |
| **Propósito** | Registra en la BD un nuevo documento que acaba de emitir |
| **Resumen** | Se hace esto para poder controlar los documentos que salen del estudio jurídico |
| **Pre-Condición** | Ninguna |
| **Flujo de Trabajo** | El abogado hace la solicitud de la plantilla necesaria para este documento y luego procede a imprimirla  Dicha impresión es registrada en la BD con su respectivo costo. |

**10. Caso de uso – REGISTRAR CASO**

****

|  |  |
| --- | --- |
| **Nombre** | **Login** |
| **Actores** | Abogado |
| **Iniciador del caso de uso** | Persona |
| **Propósito** | Registra en la BD un nuevo caso con su respectivo cliente |
| **Resumen** | Se hace esto para obtener datos previos y establecer un punto de inicio de otros posibles eventos. |
| **Pre-Condición** | Ninguna |
| **Flujo de Trabajo** | El cliente solicita al abogado que se apertura un caso  El abogado toma nota y añade los datos de interés a la BD. |

**11. Caso de uso – REGISTRAR TIPO DOCUMENTO**

****

|  |  |
| --- | --- |
| **Nombre** | **REGISTRAR TIPO DOCUMENTO** |
| **Actores** | Abogado – Secretaria |
| **Iniciador del caso de uso** | Abogado – Secretaria |
| **Propósito** | Para estar más ordenado se tiene clasificación de los distintos tipos de documentos que existen. |
| **Resumen** | Registra un nuevo tipo de documento |
| **Pre-Condición** | No debe estar Registrado |
| **Flujo de Trabajo** | Los actores implicados consideran que es necesario añadir un nuevo tipo de documento y lo añade.  Mediante la interfaz se ingresa a la capa de negocios  La capa de negocio envía la ejecución a la base de datos. |

**12. Caso de uso – REGISTRAR DOCUMENTO**

****

|  |  |
| --- | --- |
| **Nombre** | **REGISTRAR DOCUMENTO** |
| **Actores** | Abogado – Secretaria |
| **Iniciador del caso de uso** | Abogado – Secretaria |
| **Propósito** | Todos los documentos recepcionados serán añadidos a la base de datos |
| **Resumen** | Registra un nuevo documento en la BD |
| **Pre-Condición** | Ninguna |
| **Flujo de Trabajo** | Los actores implicados consideran que es necesario añadir un nuevo documento y lo añade.  Mediante la interfaz se ingresa a la capa de negocios  La capa de negocio envía la ejecución a la base de datos. |

**13. Caso de uso – REGISTRAR AUTORIDAD Y CARGO**

****

|  |  |
| --- | --- |
| **Nombre** | **REGISTRAR AUTORIDAD** |
| **Actores** | Abogado – Secretaria |
| **Iniciador del caso de uso** | Abogado – Secretaria |
| **Propósito** | Se pretender obtener todos los datos de las autoridades competentes a la justicia. |
| **Resumen** | Registra una nueva autoridad y su cargo. |
| **Pre-Condición** | No debe estar Registrado |
| **Flujo de Trabajo** | Los actores implicados consideran que es necesario añadir una nueva autoridad y lo añade.  Mediante la interfaz se ingresa a la capa de negocios  La capa de negocio envía la ejecución a la base de datos. |

**14. Caso de uso – REGISTRAR CITA**

****

|  |  |
| --- | --- |
| **Nombre** | **REGISTRAR CITA** |
| **Actores** | Secretaria |
| **Iniciador del caso de uso** | Secretaria |
| **Propósito** | El propósito es crear un registro de todas las citas que tendrán el abogado(s). |
| **Resumen** | Se registra en la BD la el día y la hora de la cita. |
| **Pre-Condición** | Ninguna |
| **Flujo de Trabajo** | Mediante la Interfaz la Secretaria va añadiendo de manera paulatina las citas que se solicitan  La base de datos, guarda para luego mostrar de manera ordenada las citas y planificarlas en un calendario. |

**15. Caso de uso – REGISTRAR ADMINISTRADOR**

****

|  |  |
| --- | --- |
| **Nombre** | **Login** |
| **Actores** | Administrador |
| **Iniciador del caso de uso** | Administrador |
| **Propósito** | Se añade como usuario del sistema a un administrador que tendrá este rango y responsabilidad. |
| **Resumen** | Solo un Administrador puede añadir. |
| **Pre-Condición** | No debe estar registrado en la base de datos |
| **Flujo de Trabajo** | Se registra como persona previamente.  Se ingresa por la interfaz los datos a la capa de negocio.  La capa de negocio evalúa si ya está registrada, de serlo indica un mensaje, sino ingresa los nuevos datos a la BD. |

**16. Caso de uso – GENERAR BACKUP**

****

|  |  |
| --- | --- |
| **Nombre** | **Generar BackUp** |
| **Actores** | **A**dministrador |
| **Iniciador del caso de uso** | **A**dministrador |
| **Propósito** | **Obtener una copia segura de la BD** |
| **Resumen** | Mediante una interfaz externa el Administrador crea una copia de seguridad. |
| **Pre-Condición** | El servidor debe estar OUT LINE |
| **Flujo de Trabajo** | El administrador detiene los servicios del servidor  Ingresa a una interfaz externa al sistema  Y con ella accede a la base de datos para realizar una copia de seguridad |

**17. Caso de uso – REALIZAR ROLLBACK**

****

|  |  |
| --- | --- |
| **Nombre** | **REALIZAR ROLLBACK** |
| **Actores** | **A**dministrador |
| **Iniciador del caso de uso** | **A**dministrador |
| **Propósito** | Poder volver a un determinado momento de estado en la BD |
| **Resumen** | Utilizando una herramienta propia del sistema se obtiene todos los Insert, Delete y Update que se realizaron |
| **Pre-Condición** | El servidor debe estar OUT LINE |
| **Flujo de Trabajo** | El administrador detiene los servicios del servidor  Ingresa a una interfaz externa al sistema  Usando los datos proporcionados por la tabla Backup del sistema se realiza una restauración completa de la base de datos. |

## 

## DIAGRAMA GENERAL DEL MODELO DE CASO DE USO

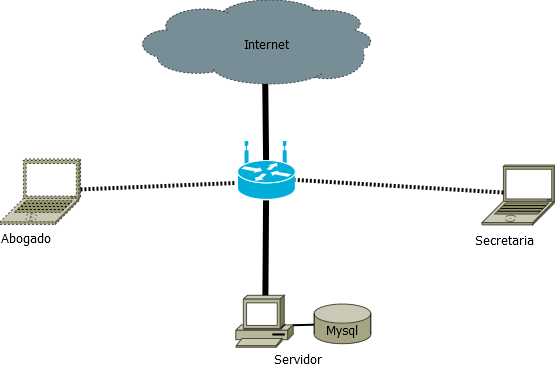
## DIAGRAMA GENERAL DE CASO DE USO



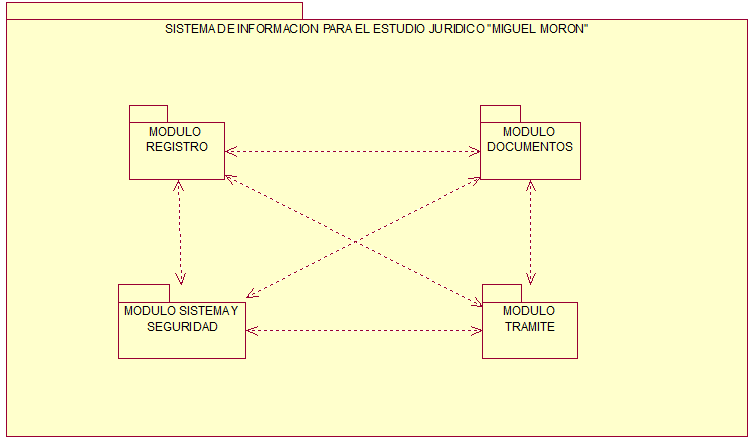
# CAPITULO IV

# ANALISIS DE SISTEMA

## ANALISIS DE ARQUITECURA



## DIAGRAMA DE PAQUETES



## DIAGRAMAS DE COLABORACION – ANALISIS

1. **Diagrama de Colaboración –** **INICIAR SESION**

****

1. **Diagrama de Colaboración – REGISTRAR SECRETARIA**

****

1. **Diagrama de Colaboración – REGISTRAR ABOGADO**

****

1. **Diagrama de Colaboración – REGISTRAR TELEFONO**

****

1. **Diagrama de Colaboración – REGISTRAR PERSONA**

****

1. **Diagrama de Colaboración – REGISTRAR PLANTILLA**

****

1. **Diagrama de Colaboración – REGISTRAR ARANCEL**

****

1. **Diagrama de Colaboración – REGISTRAR DOCUMENTO EMITIDO**

****

1. **Diagrama de Colaboración – REGISTRAR CASO**

****

1. **Diagrama de Colaboración – REGISTRAR TIPO DOCUMENTO**

****

1. **Diagrama de Colaboración – REGISTRAR DOCUMENTO**

****

1. **Diagrama de Colaboración – REGISTRAR AUTORIDAD**

****

1. **Diagrama de Colaboración – REGISTRAR CITA**

****

1. **Diagrama de Colaboración – REGISTRAR ADMINISTRADOR**

****

1. **Diagrama de Colaboración – GENERAR BACKUP**

****

1. **Diagrama de Colaboración – REALIZAR ROLLBACK**

****

# CAPITULO V

# DIAGRAMAS FINALES

## DIAGRAMA DE DOMINIO FINAL



## DIAGRAMA DE SECUENCIA DE DISEÑO

1. **Diagrama de Secuencia – INICIAR SESIÓN**

****

1. **Diagrama de Secuencia – REGISTRAR SECRETARIA**

****

1. **Diagrama de Secuencia – REGISTRAR ABOGADO**

****

1. **Diagrama de Secuencia – REGISTRAR TELEFONO**

****

1. **Diagrama de Secuencia – REGISTRAR PERSONA**

****

1. **Diagrama de Secuencia – REGISTRAR PLANTILLA**

****

1. **Diagrama de Secuencia – REGISTRAR ARANCEL**

****

1. **Diagrama de Secuencia – REGISTRAR DOCUMENTO EMITIDO**

****

1. **Diagrama de Secuencia – REGISTRAR CASO**

****

1. **Diagrama de Secuencia – REGISTRAR TIPO DOCUMENTO**

****

1. **Diagrama de Secuencia – REGISTRAR DOCUMENTO**

****

1. **Diagrama de Secuencia – REGISTRAR AUTORIDAD**

****

1. **Diagrama de Secuencia – REGISTRAR CITA**

****

1. **Diagrama de Secuencia – REGISTRAR ADMINISTRADOR**

****

1. **Diagrama de Secuencia – GENERAR BACKUP**

****

1. **Diagrama de Secuencia – REALIZAR ROLLBACK**



## DESARROLLO DEL SISTEMA

## INTERFACES











## CREACION DE BD EN MYSQL

/\*Table structure for table `arcane` \*/

CREATE TABLE `arancel` (

`id\_ara` int(6) NOT NULL AUTO\_INCREMENT,

`nombre` varchar(30) NOT NULL,

`costo` float NOT NULL DEFAULT '0',

`flag` int(1) NOT NULL,

PRIMARY KEY (`id\_ara`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*Table structure for table `backup` \*/

CREATE TABLE `backup` (

`id\_back` int(11) NOT NULL AUTO\_INCREMENT,

`fecha\_hora` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

`sql` text,

PRIMARY KEY (`id\_back`)

) ENGINE=InnoDB AUTO\_INCREMENT=5 DEFAULT CHARSET=latin1;

/\*Table structure for table `cargo` \*/

CREATE TABLE `cargo` (

`id\_car` int(6) NOT NULL AUTO\_INCREMENT,

`nombre` varchar(20) NOT NULL,

PRIMARY KEY (`id\_car`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*Table structure for table `caso` \*/

CREATE TABLE `caso` (

`id\_cas` int(6) NOT NULL AUTO\_INCREMENT,

`id\_tip` int(6) NOT NULL,

`ci` varchar(10) NOT NULL,

`descripcion` text,

`inicio` date DEFAULT NULL,

`fin` date DEFAULT NULL,

`pagado` float DEFAULT '0',

PRIMARY KEY (`id\_cas`,`id\_tip`,`ci`),

UNIQUE KEY `id\_cas` (`id\_cas`),

KEY `FK\_caso\_persona` (`ci`),

KEY `FK\_caso\_tipo\_Caso` (`id\_tip`),

CONSTRAINT `FK\_caso\_tipo\_Caso` FOREIGN KEY (`id\_tip`) REFERENCES `tipo\_caso` (`id\_tip`),

CONSTRAINT `FK\_caso\_persona` FOREIGN KEY (`ci`) REFERENCES `persona` (`ci`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*Table structure for table `cita` \*/

CREATE TABLE `cita` (

`id\_cit` int(11) NOT NULL AUTO\_INCREMENT,

`ci` varchar(10) NOT NULL,

`id\_cas` int(6) DEFAULT NULL,

`fecha\_hora` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

`costo` float NOT NULL DEFAULT '0',

`pagado` float NOT NULL DEFAULT '0',

`activa` int(1) NOT NULL DEFAULT '1',

PRIMARY KEY (`id\_cit`,`ci`),

UNIQUE KEY `id\_cit` (`id\_cit`),

KEY `FK\_cita\_persona` (`ci`),

KEY `FK\_cita\_Caso` (`id\_cas`),

CONSTRAINT `FK\_cita\_Caso` FOREIGN KEY (`id\_cas`) REFERENCES `caso` (`id\_cas`) ON DELETE SET NULL ON UPDATE SET NULL,

CONSTRAINT `FK\_cita\_persona` FOREIGN KEY (`ci`) REFERENCES `persona` (`ci`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*Table structure for table `documento` \*/

CREATE TABLE `documento` (

`ci` varchar(10) NOT NULL,

`id\_tip` int(6) NOT NULL,

`copia` int(2) NOT NULL AUTO\_INCREMENT,

`ubicacion` varchar(50) NOT NULL,

`ubicacion\_fisica` varchar(50) NOT NULL,

PRIMARY KEY (`ci`,`id\_tip`,`copia`),

UNIQUE KEY `copia` (`copia`),

KEY `FK\_documento\_tipo\_documento` (`id\_tip`),

CONSTRAINT `FK\_documento\_persona` FOREIGN KEY (`ci`) REFERENCES `persona` (`ci`),

CONSTRAINT `FK\_documento\_tipo\_documento` FOREIGN KEY (`id\_tip`) REFERENCES `tipo\_documento` (`id\_tip`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*Table structure for table `documento\_emitido` \*/

CREATE TABLE `documento\_emitido` (

`ci` varchar(10) NOT NULL,

`id\_pla` int(6) NOT NULL,

`vez` int(6) NOT NULL,

`costo` float NOT NULL,

`pagado` float NOT NULL,

PRIMARY KEY (`ci`,`id\_pla`),

KEY `FK\_documento\_emitido\_plantilla` (`id\_pla`),

CONSTRAINT `FK\_documento\_emitido\_persona` FOREIGN KEY (`ci`) REFERENCES `persona` (`ci`),

CONSTRAINT `FK\_documento\_emitido\_plantilla` FOREIGN KEY (`id\_pla`) REFERENCES `plantilla` (`id\_pla`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*Table structure for table `ocupa` \*/

CREATE TABLE `ocupa` (

`id\_car` int(6) NOT NULL,

`ci` varchar(10) NOT NULL,

`rep` int(3) NOT NULL AUTO\_INCREMENT,

`inic\_gest` date NOT NULL,

`fin\_gest` date NOT NULL,

`observacion` text,

PRIMARY KEY (`id\_car`,`ci`,`rep`),

UNIQUE KEY `rep` (`rep`),

KEY `FK\_ocupa\_persona` (`ci`),

CONSTRAINT `FK\_ocupa\_cargo` FOREIGN KEY (`id\_car`) REFERENCES `cargo` (`id\_car`),

CONSTRAINT `FK\_ocupa\_persona` FOREIGN KEY (`ci`) REFERENCES `persona` (`ci`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*Table structure for table `persona` \*/

CREATE TABLE `persona` (

`ci` varchar(10) NOT NULL,

`nombre` varchar(30) DEFAULT NULL,

`apellido\_paterno` varchar(30) DEFAULT NULL,

`apellido\_materno` varchar(30) DEFAULT NULL,

`direccion` varchar(100) DEFAULT NULL,

`id\_tit` int(6) NOT NULL DEFAULT '0',

PRIMARY KEY (`ci`),

KEY `FK\_persona\_titulo` (`id\_tit`),

CONSTRAINT `FK\_persona\_titulo` FOREIGN KEY (`id\_tit`) REFERENCES `titulo` (`id\_tit`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*Table structure for table `plantilla` \*/

CREATE TABLE `plantilla` (

`id\_pla` int(6) NOT NULL AUTO\_INCREMENT,

`nombre` varchar(30) NOT NULL,

`directorio` text NOT NULL,

PRIMARY KEY (`id\_pla`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*Table structure for table `telefono` \*/

CREATE TABLE `telefono` (

`ci` varchar(10) NOT NULL,

`numero` varchar(20) NOT NULL,

PRIMARY KEY (`ci`,`numero`),

CONSTRAINT `FK\_telefono\_persona` FOREIGN KEY (`ci`) REFERENCES `persona` (`ci`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*Table structure for table `tipo\_caso` \*/

CREATE TABLE `tipo\_caso` (

`id\_tip` int(11) NOT NULL AUTO\_INCREMENT,

`nombre` varchar(100) NOT NULL,

PRIMARY KEY (`id\_tip`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*Table structure for table `tipo\_documento` \*/

CREATE TABLE `tipo\_documento` (

`id\_tip` int(11) NOT NULL AUTO\_INCREMENT,

`tipo` varchar(25) NOT NULL,

PRIMARY KEY (`id\_tip`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*Table structure for table `titulo` \*/

CREATE TABLE `titulo` (

`id\_tit` int(6) NOT NULL AUTO\_INCREMENT,

`titulo` varchar(30) NOT NULL,

`abreviatura` varchar(15) NOT NULL,

PRIMARY KEY (`id\_tit`)

) ENGINE=InnoDB AUTO\_INCREMENT=2 DEFAULT CHARSET=latin1;

/\*Table structure for table `usuario` \*/

CREATE TABLE `usuario` (

`id\_usu` int(6) NOT NULL AUTO\_INCREMENT,

`user` varchar(10) DEFAULT NULL,

`pass` varchar(40) DEFAULT NULL,

`activo` int(1) DEFAULT NULL,

`ci` varchar(10) NOT NULL,

`tipo` int(1) DEFAULT NULL,

PRIMARY KEY (`id\_usu`),

KEY `FK\_usuario\_persona` (`ci`),

CONSTRAINT `FK\_usuario\_persona` FOREIGN KEY (`ci`) REFERENCES `persona` (`ci`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB AUTO\_INCREMENT=2 DEFAULT CHARSET=latin1;