



# **UNIVERSIDAD AUTÓNOMA DE ICA**

**FACULTAD DE CIENCIAS, INGENIERÍA Y ADMINISTRACIÓN**

**ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**

## **TITULO**

**“SOFTWARE DE CONTROL DOCUMENTARIO  
PARA EL MEJORAMIENTO DE LA GESTIÓN  
ADMINISTRATIVA EN LA UNIVERSIDAD  
AUTÓNOMA DE ICA DE CHINCHA – 2015.”**

## **TESIS**

Presentada para optar el Título Profesional de  
**INGENIERO DE SISTEMAS**

## **PRESENTADA POR:**

Hernández Sifuentes, Omar Ernesto

CHINCHA - PERÚ – Mayo - 2015

**SOFTWARE DE CONTROL DOCUMENTARIO PARA  
EL MEJORAMIENTO DE LA GESTIÓN  
ADMINISTRATIVA EN LA UNIVERSIDAD  
AUTÓNOMA DE ICA DE CHINCHA – 2015**

---

Hernández Sifuentes, Omar

**AUTOR**

---

Ing. Moreno Heredia, Armando

**ASESOR**

Presentada a la Facultad de Ciencias, Ingeniería y administración de la  
Universidad Autónoma de Ica. Para optar el Título de:

**APROBADO POR:**

---

**PRESIDENTE DEL JURADO**

---

**SECRETARIO DEL JURADO**

---

**VOCAL DEL JURADO**

Mayo del 2015

## **DEDICATORIA**

Este proyecto está dedicado a mis padres quienes son fuente de inspiración, apoyo en nuestra superación a lo largo de nuestra vida profesional, por brindarnos todo su apoyo incondicional, por el cariño y esmero y consejos constantes, me permiten realizar uno de los sueños a lo largo de la vida, gracias a ellos salí adelante.

## **AGRADECIMIENTOS**

Primeramente agradecer a Dios por todos los días de vida, por habernos colmado de bendiciones y siempre guiándonos por el buen camino para lograr nuestros objetivos a lo largo de nuestra formación profesional.

Agradecer a los destacados Ingenieros que a lo largo de nuestra carrera dan lo mejor de ellos para brindarnos un aprendizaje con óptimo desempeño, gracias a ellos nos hemos ido forjando.

Agradecer a la familia completa y a todas las personas que me apoyaron de manera directa e indirecta para la culminación de este proyecto.

## **INDICE**

RESUMEN

ABSTRACT

INTRODUCCION

### **CAPITULO I – ANALISIS DEL OBJETO DE ESTUDIO**

#### **1.1. UBICACIÓN**

1.1.1. GEOGRAFÍA

1.1.2. LIMITE

1.1.3. EXTENSIÓN

1.1.4. PLANO DE UBICACIÓN

1.1.5. DATOS GENERALES DEL DISTRITO

1.1.6. DATOS GENERALES DE LA EMPRESA

1.1.7. ORGANIGRAMA Y JERARQUIAS

1.1.8. INFRAESTRUCTURA TECNOLÓGICA,  
HARDWARE Y SOFTWARE

#### **1.2. COMO SURGE EL PROBLEMA**

1.2.1. SITUACIÓN PROBLEMÁTICA

1.2.2. FORMULACIÓN DEL PROBLEMA

1.2.3. JUSTIFICACIÓN E IMPORTANCIA

1.2.4. FORMULACION DE OBJETIVOS

#### **1.3. COMO SE MANIFIESTA Y QUE CARACTERISTICA TIENE**

#### 1.4. METODOLOGÍA

1.4.1. FORMULACION DE HIPOTESIS

1.4.2. VARIABLES DE OPERACIONALIZACIÓN

#### 1.5. DISEÑO DE METODOLOGÍA

1.5.1. TIPO DE ESTUDIO Y DISEÑO DE  
CONTRASTACIÓN DE HIPÓTESIS

1.5.2. POBLACION, MUESTRA DE ESTUDIO Y  
MUESTREO

### **CAPITULO II – MARCO TEORICO**

#### 2.1. BASES TEORICA

2.1.1. METODOLOGÍA

2.1.2. HERRAMIENTAS DE SOFTWARE

2.1.3. SISTEMA OPERATIVO

2.1.4. REDES

#### 2.2. DEFINICION DE TERMINOS BASICOS

### **CAPITULO III – RESULTADOS Y DISCUSION**

#### 3.1. ANALISIS E INTERPRETACION DE LOS DATOS

3.1.1 DESCRIPCIÓN Y SUSTENTACIÓN DE LA  
SOLUCIÓN

3.1.2 ANÁLISIS

3.1.3 CRONOGRAMA DE ACTIVIDADES DEL  
PROYECTO

3.1.4 DIAGRAMA GENERAL DE LA EMPRESA

3.1.5 DIAGRAMA DEL ÁREA DESIGNADA

3.1.6 DIAGRAMA DEL PROCESO ACTUAL

3.1.7 STAKEHOLDER INTERNOS Y EXTERNOS

3.1.8 DIAGRAMA DE PROCESO DE EXPEDIENTE  
INTERNO

3.2. DIAGRAMA DE CASO DE USO DEL SISTEMA

3.2.1. DIAGRAMA DE INICIO DE SESION

3.2.2. DIAGRAMA DEL SISTEMA

3.2.3. DIAGRAMA DE MANTENIMIENTO DE USUARIO

3.2.4. DIAGRAMA DE REPORTES

3.3. DIAGRAMA DE SECUENCIA

3.3.1. DIAGRAMA DE PROCESAMIENTO DE  
EXPEDIENTE INTERNO

3.3.2. DIAGRAMA DE INICIO DE SESION

3.3.3. DIAGRAMA DE MANTENIMIENTO DE DATOS

3.3.4. DIAGRAMA DE MANTENIMIENTO DE USUARIO

3.3.5. DIAGRAMA DE CONSULTA DE DATOS

3.3.6. DIAGRAMA DE REPORTES

3.4. DIAGRAMA DE COLABORACION DEL SISTEMA

3.4.1. DIAGRAMA DE PROCESAMIENTO DE  
EXPEDIENTE INTERNO

3.4.2. DIAGRAMA DE INICIO DE SESION

3.4.3. DIAGRAMA DE MANTENIMIENTO DE DATOS

3.4.4. DIAGRAMA DE MANTENIMIENTO DE USUARIO

3.4.5. DIAGRAMA DE CONSULTA DE DATOS

3.4.6. DIAGRAMA DE REPORTES

3.5. DIAGRAMA DE ACTIVIDADES DEL SISTEMA

3.5.1. DIAGRAMA DE PROCESAMIENTO DE  
EXPEDIENTE INTERNO

3.5.2. DIAGRAMA DE INICIO DE SESION

3.5.3. DIAGRAMA DE MANTENIMIENTO DE DATOS

3.5.4. DIAGRAMA DE MANTENIMIENTO DE USUARIO

3.5.5. DIAGRAMA DE CONSULTA DE DATOS

3.5.6. DIAGRAMA DE REPORTES

3.6. MODELADO DE LA BASE DE DATOS

3.7. INTERFACE DEL SISTEMA

3.8. FASE DE PROGRAMACION DEL SISTEMA

CONCLUSIONES

RECOMENDACIONES

BIBLIOGRAFIA

ANEXOS

## RESUMEN

En la actualidad, se vive una revolución que ha dado lugar a la sociedad de la información, también denominada sociedad digital, comandada por las nuevas tecnologías, donde la informática juega un papel fundamental en todos los ámbitos. Una de las utilidades más importantes de la informática es facilitar información en forma oportuna y veraz, lo cual, por ejemplo, puede tanto facilitar el desarrollo de un cálculo en corto tiempo, como permitir el control de procesos críticos en las diferentes ramas profesionales.

Hoy en día el eficiente seguimiento y control de documentos representa un reto en las empresas o instituciones en donde es gestionado un gran volumen de información, y que en la mayoría de los casos el registro se hace de manera manual y una vez archivada la información representa cierto grado de dificultad extraer información en torno a un documento en específico en relación a la gestión que se le ha dado.

Por lo anterior, las tecnologías de la información han permitido hacer grandes avances en la gestión documental, mediante los Sistemas de Gestión Documental. Estos sistemas permiten informatizar la gestión, haciendo los procesos más ágiles y eficientes.

Al respecto una de las soluciones con la que cuenta el presente proyecto **“Software de control documentario para el mejoramiento de la gestión administrativa en la Universidad Autónoma De Ica de chincha – 2015”**, en apoyo a la gestión documental, es el módulo de seguimiento y de control de documentos, el cual consiste en una serie de formas y reportes que le ayudarán al personal de una dependencia a controlar de una manera sencilla los documentos que allí se gestionan.

## ABSTRACT

Today, we are experiencing a revolution that has resulted in the information society, also called digital society, led by the new technologies, where the computer plays a vital role in all areas. One of the most important utilities of computing is to provide information in a timely and accurate, which, for example, can both facilitate the development of a calculation in a short time, such as allowing control of critical processes in the different professional branches.

Today the efficient monitoring and control of documents is a challenge in enterprises or institutions where it managed a large volume of information, and in most cases the record is done manually and once the information is archived some difficulty extracting information about a specific document relating to the management that has been given.

Therefore, the information technology has enabled great advances in making document management through Document Management Systems. These systems allow management computerization, making the process more streamlined and efficient.

In this regard one of the solutions with which this project has "**Software for Control Documents for improving administrative management in the Private University of Ica, Chincha - 2015**", in support of the document management module is the monitoring and control document, which consists of a number of forms and reports that will help staff a unit to control a simple way documents are managed there.

## INTRODUCCIÓN

En el siglo XXI el hombre y sus investigaciones científicas hacen que la tecnología camine a pasos agigantados en la vida de las organizaciones y personas, permitiendo con sus avances tecnológicos: fiabilidad, calidad, agilidad y reducción de espacio físico en el trabajo.

Efectivamente la informática se constituye en la tecnología intelectual dominante, dando lugar a formas de conocimientos y por lo tanto a otras formas de memoria.

Es compromiso de la sociedad moderna prepararse para enfrentar los retos que las innovaciones científico-tecnológicas presentan día a día y así formar parte de sus avances, adquiriendo ventajas que permitan ser competitivos.

Es indispensable en la actualidad un Sistema que nos brinde apoyo para la Administración de Datos y Documentos, así como de un adecuado seguimiento, software que permita tener eficiencia tanto en control así como en el manejo de los documentos incorporados a un Sistema de Gestión.

El presente Proyecto de Gestión de Documentos materializa nuestra intención de implantar mecanismos eficientes que optimicen y modernicen la gestión interna, partiendo desde las actividades básicas como la gestión documental en las diversas Áreas de Trabajo de la UNIVERSIDAD AUTONOMA DE ICA - Chincha, a fin de automatizar los procesos, minimizar tiempos y esfuerzos a favor de una administración más eficiente de cara a los usuarios.

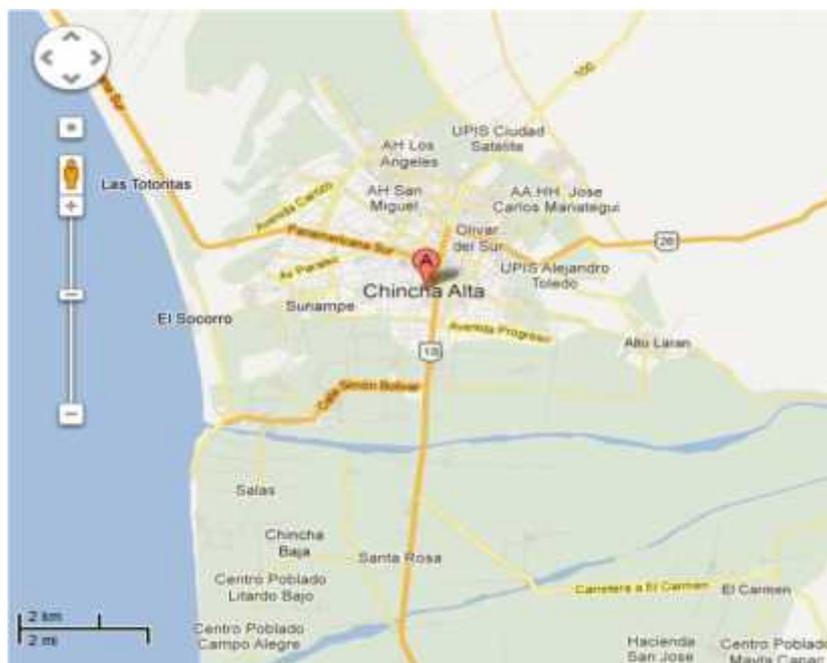
Conscientes de la magnitud de ésta intención, éste proyecto concentra sus esfuerzos en sentar bases sólidas consistentes y a la vez flexibles al cambio, a fin de, a partir de ellas, desarrollar y fortalecer una moderna y eficiente gestión documental, lo cual se evidencia en las primeras fases del mismo.

# **CAPITULO I: ANÁLISIS DEL OBJETO DE ESTUDIO**

## 1.1 Ubicación

### 1.1.1 Geográfica

El distrito de Chinchá Alta es uno de los once distritos peruanos que forman la Provincia de Chinchá en el Departamento de Ica, bajo la administración del Gobierno regional de Ica.



### 1.1.2 Limite

El Distrito de Chinchá Alta limita:

**Por el Norte:**

Con el distrito de Sunampe y Grocio Prado.

**Por el Este:**

Con el distrito de Chinchá Baja y El Carmen.

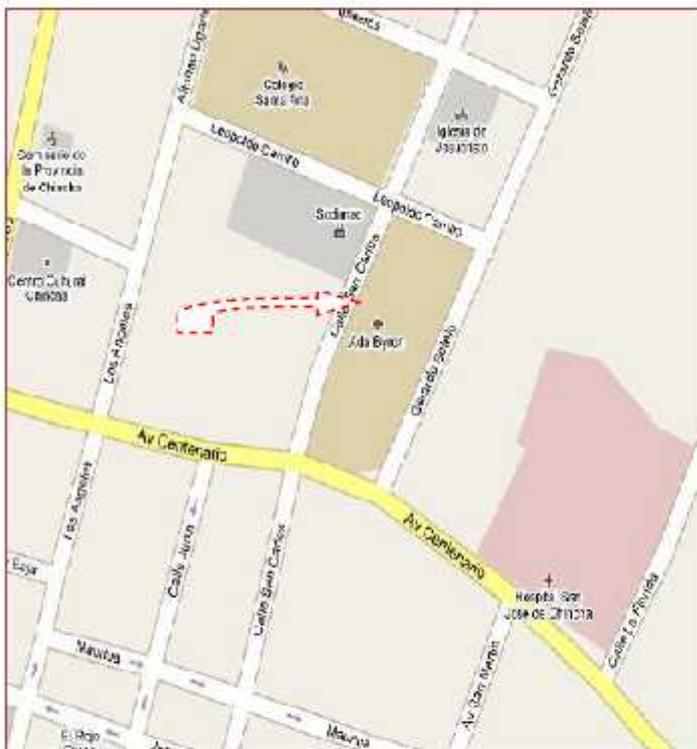
**Por el Oeste:**

Con el Océano Pacífico.

### 1.1.3 Extensión

Tiene una superficie aproximada de 238.34. Km<sup>2</sup>.

### 1.1.4 Plano de Ubicación



Institución	Ada A. Byron S.A.C
Ciudad	Chíncha Alta
Provincia	Chíncha
Departamento	Ica
Longitud	-75.015152
Lattud	9.189967



## 1.1.5 DATOS GENERALES DEL DISTRITO

### ❖ Referencias históricas

El distrito de Chincha Alta su población es urbana y está conformada fundamentalmente por comerciantes, profesionales, industriales, ambulantes, empleados artesanos.

El resto son campesinos, principalmente medianos y pequeños agricultores, así como una gran cantidad de obreros hombres, mujeres y niños de ambos sexos.

El chinchano común es de color trigueño (ascendencia africana) y talla mediana, por lo general de complexión fuerte, robusta, laboriosa, amante del trabajo agrícola. Gusta mucho del deporte, como el atletismo, box y fútbol. De conformidad con el X Censo Nacional de Población y V de Vivienda realizado en 2005 por el Instituto Nacional de Estadística e Informática (INEI), la población de Chincha Alta (distrito) asciende a 56.085 habitantes, de los cuales 27.364 habitantes (48,79%) son hombres y 28.721 habitantes (51,21%) son mujeres. La tasa de crecimiento anual es de 1,1% y la densidad poblacional es de 217,09 habitantes/km<sup>2</sup>.

La edad quinquenal más alta corresponde al rango de 10-14 años, con un 10,25% y la más baja al rango comprendido entre 95-99 años con un 0,08%. La población alfabeta, asciende a 49.348 habitantes, lo que representa el 93,07% contra los analfabetos que ascienden a 3.672 habitantes o el 6,93%. El nivel educativo más alto alcanzado corresponde al de secundaria completa con un 22,93% y el más bajo a educación inicial con un 2,72%. chinchano

## 1.1.6 DATOS GENERALES DE LA EMPRESA

- **Nombre de Empresa** : UNIVERSIDAD AUTONOMA DE ICA
- **RUC** : 20452777399
- **Fecha de Fundación** : 10/12/2005
- **Tipo de Sociedad** : Privada

- **Estado de la Empresa** : Activo
- **Actividad Económica** : Prestación de servicios a la comunidad
- **Dirección Principal** : Av. Abelardo Alva Maúrtua N° 499
- **Población** : Ica / Chincha / Chincha Alta
- **Teléfono** : 262439
- **Nro. Trabajadores** : 30

#### ❖ **Visión de la empresa**

Ser una universidad innovadora de liderazgo académico y de investigación a nivel regional, nacional e internacional, generando beneficios y bienestar para sus miembros y la sociedad.

#### ❖ **Misión de la empresa**

Desarrollamos una propuesta innovadora de liderazgo académico potenciando los talentos y habilidades personales y profesionales, practicando valores corporativos a través de un modelo educativo propio basado en competencias e investigación.

## ❖ Actividad principal de la empresa

La UNIVERSIDAD AUTONOMA DE ICA brinda el servicio de educación superior a los jóvenes de la provincia de Chincha.

### Actividades:

- Educación Superior – Universitario.
- Actividades de recreación.
- Actividades de Proyección Social en la Provincia de Chincha.
- Cursos – Tallares – Seminarios – Congresos para los estudiantes de la Universidad y público en general.
- Servicio de Orientación Vocacional
- Cursos Extra Curriculares.
- Cursos externos como Maestrías en convenio con otras universidades.

Todas estas actividades agrupadas en forma general con lo siguiente:

**LA DOCENCIA UNIVERSITARIA.-** es la función por la que se trasmite en el proceso académico los conocimientos, se dirige la aplicación de éstos a la realidad espacio/temporal y se desarrolla la actitud heurística. Se cumple a través de la Facultad y sus distintas escuelas profesionales:

- Escuela Profesional de Ingeniería de Sistemas.

-Escuela Profesional de Ingeniería de Industrias Alimentarias.

- Escuela Profesional de Ingeniería de Industrial.

- Escuela Profesional de Ingeniería Civil.

- Escuela Profesional de Administración.

- Escuela Profesional de Enfermería.

- Escuela Profesional de Psicología.

**LA INVESTIGACIÓN UNIVERSITARIA.-** se orienta hacia la búsqueda de conocimientos nuevos y a la comprobación de los ya existentes. Su meta es la superación del acervo existente, el entrenamiento de los estudiantes/profesionales en los métodos del conocimiento y actuar sobre los problemas sociales prioritarios. Las unidades estructurales encargadas de la política de esta función y las unidades de base a través de las cuales se cumple son las Escuelas Profesionales.

**LA EXTENSIÓN.-** es la función que vincula entre sí el conocimiento y el ámbito de la realidad y de la sociedad. Su misión es extender hacia la sociedad la actividad orgánica de la Universidad, integrando en aquella el modelo de práctica profesional multifacética que postula la filosofía institucional, devolviéndole parte de sus aportes y recogiendo del seno vivo de su universo las orientaciones del saber popular. Se cumple a través de los distintos eventos realizados por la Universidad, denominados eventos de proyección social, que hasta la fecha se han conseguido grades logros.

**LA PLANIFICACIÓN.-** es la función que mediante el diagnóstico de la realidad orienta la toma de decisiones conducentes al logro de las otras funciones y de la misión. Esta función es realizada por la junta directiva en coordinación con los jefes de escuelas profesionales.

**LA ADMINISTRACIÓN.-** es una función que se sustenta, entre otras, en forma de actividades específicas que sirven de apoyo a todo el proceso institucional. Su órgano central es la Dirección de Administración, pero se manifiesta en todas las unidades docentes, docentes/administrativas y administrativas.

## 1.1.8. Infraestructura Tecnológica, Hardware y Software

### - HARDWARE

Hardware	Características	Cantidad
<b>SERVIDORES ADAPTADOS</b>		
Servidor de App - Datos	Intel Core 2 Duo – 2Gb RAM – 2.9 GHZ	01
Servidor Administrador de Red	Pentium IV – Intel 2Gb RAM	01
<b>COMPUTADORAS</b>		
Centro de Desarrollo		03
Áreas de Producción		00
Laboratorios A –D		50

#### Característica PC laboratorios A

- o 15 pc PHENOM II 3.2 GHZ. DDR3 2GB
- o 10 pc PENTIUM 4 3.06 GHZ. DDR2 1GB.

#### Características PC laboratorio D

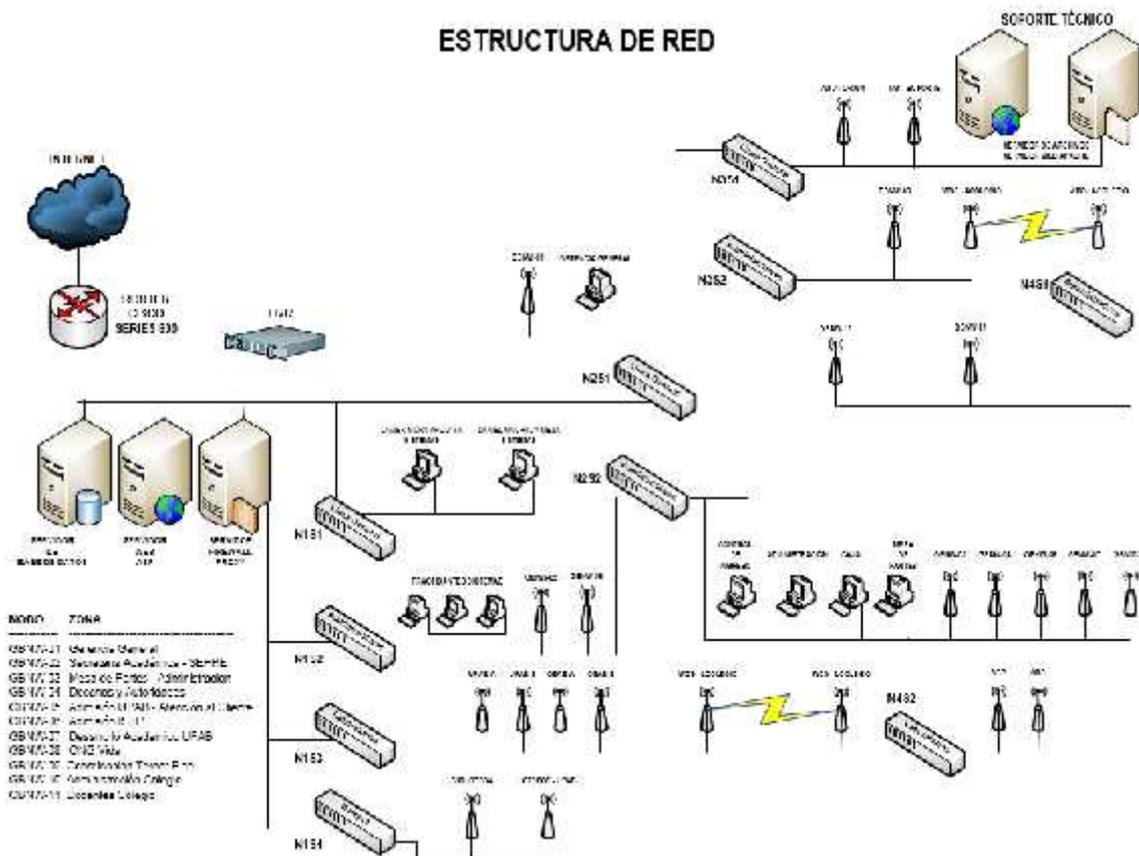
- o 20 pc INTEL CORE 2 DUO 2.93 GHZ/ 100K/ 2GB
- o 5 pc PHENOM II 3.2 GHZ. DDR3 2GB.

### - SOFTWARE DE DESARROLLO.

Software	Versión	Cantidad
<b>CENTRO DE DESARROLLO</b>		
Microsoft SQL Server 2003 R2	Enterprise	01
Microsoft Visual Studio 2008	Professional	01
Microsoft Windows Server	Enterprise	01
Visual Fox Pro	v. 9	01
Eset NOD32 Antivirus	Business	3 (x5)
<b>CENTRO DE DESARROLLO - PRODUCCIÓN - LABORATORIOS</b>		
Windows XP Profesional	SP 3	18
Microsoft Office 2007	Profesional	18

- SOFTWARE DE PRODUCCIÓN.

NOMBRE	AREA
SIGECO	CAJA – DIDART – ADMISIÓN – ATENCIÓN AL CLIENTE
CHEKE	DIRECCION DE ADMINISTRACIÓN – SECRETARIA DE ADMINISTRACIÓN
UPAD	ADMISION
SIGECO V 2.0	ATENCIÓN AL CLIENTE
SGE – UPICA	ATENCIÓN AL CLIENTE
CONTROL PERSONAL	DE JEFATURA DE PERSONAL
CONTABILIDAD	PDT



## 1.2 Como surge el problema

### 1.2.1 Situación Problemática

La Universidad Autónoma De Ica SAC. Se encuentra ubicada en Av. Abelardo Alva Maúrtua N° 499, Chincha Alta, con RUC 20452777399, con fecha de funcionamiento 29 de Mayo del 2006 con RESOLUCION 136-2006-CONAFU, funciones que brinda la Universidad es el servicio de Estudio de Grado Superior Universitario a toda la provincia de Chincha Alta y sus alrededores, teniendo como visión de la empresa: **Ser una universidad innovadora de liderazgo académico y de investigación a nivel regional, nacional e internacional, generando beneficios y bienestar para sus miembros y la sociedad.** Así mismo tiene como misión: **Desarrollamos una propuesta innovadora de liderazgo académico potenciando los talentos y habilidades personales y profesionales, practicando valores corporativos a través de un modelo educativo propio basado en competencias e investigación.**

El área de Tramite Documentario – OCTA actualmente cuenta con problemas en la demora/retraso en las respuestas de cada documento emitido a las diversas áreas correspondientes para su debido análisis y respuesta en un tiempo determinado para su devolución a OCTA, siendo una molestia para las personas que requieren una respuesta inmediata para poder solucionar su necesidad. Teniendo un control de trámite documentario hecho por el área en Microsoft Office Excel, en el cual se registra la persona que emite el documento y la fecha de recepción al área de OCTA, asunto del trámite, quien recepciona y evalúa el documento, duración del trámite documentario (días), fecha de retorno del documento a OCTA, fecha de entrega del documento al solicitante y si lo recibió/rechazo, registra una observación del documento en trámite.

**NOTA:** En algunos casos el tramite documentario puede tener una duración mínima de 2 días a 15 días o 1 mes, así mismo la

documentación no solo pasa por una área sino por 4 o 5 áreas dependiendo cual sea el asunto. Ejemplo: Tramite de Titulación (interfiere 4 Áreas).

Por lo que con el presente proyecto se busca que crear un seguimiento en tiempo real de todos los documentos y trámites realizados en la UNIVERSIDAD AUTONOMA DE ICA, así como lograr un mejor manejo de toda la información existente dentro de la empresa, asegurando con esto un control más exacto y un archivo de datos de manera segura de todos los documentos, logrando búsquedas más rápidas, evitar pérdidas de información y generar reportes cada ciertos periodos de todos los documentos que se expiden en el área.

Por estas razones que permitirá hacer un trabajo más organizativo y participativo de todas las áreas es que se presenta el proyecto como una solución al proceso que en la actualidad se realiza de forma manual.

## 1.2.2 Formulación del Problema

### ❖ Problema General

¿Software de control documentario beneficiara para el mejoramiento de la gestión administrativa en la Universidad Autónoma De Ica de Chincha – 2015?

### ❖ Problemas Específicos

- ¿El Software de control documentario a implantar ayudara a automatizar y optimizar la gestión documental en la Universidad Autónoma De Ica?
- ¿Ayudara el Software de control documentario en obtener un informe en tiempo real del estado de sus trámites realizados en el Universidad a los estudiantes?

- ¿El Software de control documentario beneficiara para disponer reportes estadísticos de todos los documentos?
- ¿El Software de control documentario ayudara a minimizar el tiempo de los trámites documentarios?
- ¿El Software de control documentario ayudara a mejorar el control de la gestión administrativa?

### **1.2.3 Justificación e Importancia**

#### **❖ Justificación Teórica**

Actualmente en la Universidad Autónoma De Ica cuenta con el área de Sistemas; sin embargo en cada área no se cuenta con sistemas de información para la automatización de los procesos, al manejarse 6 diferentes Base de Datos que no se encuentran integradas, ocasionando un inadecuado flujo de información, de trabajo así como pérdidas de información y de tiempo, respecto al área estudiada OCTA al no tener un sistema de información ha ocasionado el malestar de los alumnos al momentos de realizar o ir a consultar el estado de su trámite.

El control y archivo de documentos se realiza en su mayoría en todas las áreas de forma manual, es decir a través de cuadernos donde realizan los apuntes de todos los documentos expedidos y aceptados en cada área (Libro de Cargo) y por medio de un formato de Microsoft Office Excel que lleva el control y seguimiento de los tramites documentarios que llegan al área de OCTA.

### ❖ **Justificación Práctica**

Por lo estudio realizado en el Área de Tramite Documentario (OCTA) requiere implantar un software de control documentario para el mejoramiento de la gestión administrativa en la Universidad Autónoma De Ica de manera eficiente y eficaz, dejando poco a poco el trabajo manual que se realiza en el área actualmente para ir automatizando sus procesos por medio del software de gestión y a su vez aminorar el tiempo de respuesta como el seguimiento de cada uno de los documentos y sus respectivas respuestas para el área y/o persona interesa. Viendo a su vez la posibilidad de realizar una consultoría online de dichos trámites para consulta según sea la necesidad del área a sistematizar.

## **1.2.4 Formulación de Objetivos**

### ❖ **Objetivo General**

Implementar un Software de control documentario para el mejoramiento de la gestión administrativa en la Universidad Autónoma De Ica, Chincha – 2015.

### ❖ **Objetivos Específicos**

- ✓ Automatizar y optimizar la gestión documental en la Universidad Autónoma De Ica.
- ✓ Obtener un seguimiento en tiempo real del estado de los trámites realizados en el Universidad a los estudiantes.

- ✓ Disponer de reportes estadísticos del flujo de documentos en el área.
- ✓ Minimizar el tiempo de los trámites documentarios.
- ✓ Mejorar el control de la gestión administrativa.

### **1.3 Como se manifiesta y qué características tiene**

En la actualidad en la Universidad Autónoma De Ica si bien es cierto el área de mesa de partes cuenta con un módulo de tramite documentario, pero este sistema no es suficiente ya que la gestión de trámite documentario en el área de mesa de partes una parte de trabajo lo realizan de forma manual y la otra parte de trabajo lo realizan en un módulo de tramite documentario que en consecuencia cuando se requiere administrar grandes volúmenes de información y la orientación debida, no se cuenta con las herramientas adecuadas que permita realizar un adecuado servicio de orientación, seguimiento y control de los documentos.

## **1.4 Metodología**

### **1.4.1 Formulación de la Hipótesis**

#### **❖ Hipótesis General**

El software de Seguimiento y el Control de Documentos beneficiará el mejoramiento de la Gestión Administrativa en la Universidad Autónoma De Ica, Chincha - 2015.

#### **❖ Hipótesis Específicas**

- ✓ Automatizar y optimizar la Gestión documental será beneficioso en la Universidad Autónoma De Ica.

- ✓ Realizar un informe en tiempo real del estado de los trámites realizados en el Universidad beneficiara a los estudiantes para tener de conocimiento como va su documento.
- ✓ Se podrá disponer de reportes estadísticos del flujo de los documentos en el área en tiempo real beneficiara a tener un mejor control.
- ✓ Se podrá Minimizar el tiempo de los trámites documentarios en el área.
- ✓ Podrá Mejorar el control de la gestión administrativa.

## 1.4.2 Variables – Operacionalización

### ❖ Variables

**Independiente: Software / Sistema Informático.**

**Dependiente: Mejoramiento de la Gestión  
Administrativa.**

**Enlace: Beneficiará.**

## ❖ Operacionalización

### Variable Independiente: SISTEMA INFORMATICO

CONCEPTO	DIMENSION Ó CATEGORIA	INDICADOR	INDICE	SUBINDICE
Es una aplicación informática capaz de solucionar un conjunto de problemas que exigen un gran conocimiento	Análisis de los Datos.  Desarrollo	Recopilar, examinar la información.  Lenguaje de programación.	Valores y datos necesarios.  Utilización de herramientas basadas en software libre.	Datos tomados de la recopilación y levantamiento de la información dentro de la Universidad.

### Variable Dependiente: MEJORAMIENTO DE LA GESTIÓN ADMINISTRATIVA.

CONCEPTO	DIMENSION Ó CATEGORIA	INDICADOR	INDICE	SUBINDICE
Es fruto de la planeación de la carrera y comprende los aspectos en los que una persona enriquece o mejora sus conocimientos	Sistema de Seguimiento y control de documentos en tiempo real.	Reportes del Sistema	Procesos necesarios para el control y seguimiento de los documentos.	Datos tomados de la recopilación y levantamiento de la información dentro de la Universidad.

## **1.5. Diseño de Metodología**

### **1.5.1. Tipo de estudio y diseño de contrastación de hipótesis**

#### **❖ Tipo de Investigación**

La presente investigación es de tipo aplicada y de nivel experimental, porque se pretenderá estudiar cómo afecta la aplicación de un sistema de gestión de trámite, cuyo indicador es el número de trámites que se tiene registrado.

#### **❖ Diseño de Investigación**

El tipo de diseño es no experimental, por las siguientes razones:

- i. Se puede manipular intencionalmente la variable independiente.
- ii. Se puede medir de manera confiable el efecto que tiene la variable independiente sobre la variable dependiente.
- iii. La variación de la variable dependiente se debe únicamente a la manipulación de la variable independiente.

## 1.5.2. Población, muestra de estudio y muestreo

### ❖ Población

La población son todo el público general y alumnado que realizan trámites documentarios en UNIVERSIDAD AUTONOMA DE ICA, para la cual existe una población de 500 personas.

### ❖ Tamaño de Muestra

Es una herramienta de la investigación científica. Su función básica es determinar que parte de una realidad en estudio (población universo) debe examinarse con la finalidad de hacer diferencias sobre dicha población.

Para determinar el tamaño de la muestra, es decir, el número de encuestas que se realizaron, se tomó en cuenta la siguiente información:

- El cual estuvo conformado por los trabajadores que laboran en las diferentes aéreas que laboran actualmente en UNIVERSIDAD AUTONOMA DE ICA, cual está conformado por una población de 30 personas.
- Un nivel de confianza de 95% y un grado de error de 5% en la fórmula de la muestra.

La fórmula que se aplicará para determinar la muestra es:

$$\text{Dónde: } n = \frac{Z^2 \cdot N \cdot p \cdot q}{(N-1) \cdot e^2 + Z^2 \cdot p \cdot q}$$



# **CAPITULO II: MARCO TEÓRICO**

## 2.1 BASES TEÓRICO-CIENTIFICAS

### 2.1.1 METODOLOGIA ELEGIDA

#### ❖ Metodología UML

La metodología a utilizar es la metodología UML:

UML (UnifiedModelingLanguage) es un lenguaje para especificar, visualizar, construir y documentar las diferentes etapas del desarrollo de software, así como para modelado de procesos de negocio u otros sistemas no-software. UML reúne una colección de las mejores prácticas en la ingeniería que han sido utilizadas con éxito para modelar sistemas grandes y complejos, ya que cubre tanto objetos conceptuales como los procesos de negocio y funciones del sistema, como también objetos concretos como clases en un lenguaje de programación, esquemas de base de datos y componentes reusables de software.

Este lenguaje nos indica cómo crear y leer los modelos, pero no dice como crearlos, por ello entre los objetivos tenemos:

- **Visualizar:** UML permite expresar de una forma gráfica un sistema de forma que otro puede entender.
- **Especificar:** UML permite especificar cuáles son las características de un sistema antes de su construcción
- **Construir:** A partir de los modelos especificados se pueden construir los sistemas diseñados.

- **Documentar:** Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

Las ventajas de la metodología tenemos:

- **Reutilización:** las clases están diseñadas para que se reutilicen en muchos sistemas. Para maximizar, las clases se construyen de manera que se puedan adaptarse los otros sistemas.
- **Integridad:** La estructura de datos (los objetos) solo se pueden utilizar con métodos específicos. Esto tiene particular importancia en los sistemas cliente-servidor y los sistemas distribuidos, en los que usuarios desconocidos podrían intentar el acceso al sistema.
- **Una interfaz de pantalla sugestiva para el usuario:** Hay que utilizar una interfaz de usuario gráfica de modo que el usuario apunte a iconos o elementos de un menú desplegados, relacionados con los objetos. En determinadas ocasiones, el usuario puede ver un objeto en la pantalla. Ver y apuntar es más fácil que recordar y escribir.
- **Independencia del diseño:** Las clases están diseñadas para ser independientes del ambiente de plataformas, hardware y software. Utilizan solicitudes y respuestas con formato estándar. Esto les permite ser utilizadas en múltiples sistemas operativos, controladores de base de datos, controladores de la red, interfaces de usuario gráficas, etc. El creador del software no

tiene que preocuparse por el ambiente o esperar a que este se especifique.

- **Mantenimiento más sencillo:** El programador encargado del mantenimiento cambia un método de clase a la vez. Cada clase efectúa sus funciones independientemente de las demás

Una vez que el portal entre en funcionamiento se pretende establecer una cultura de uso, haciendo del mismo una herramienta diaria de trabajo, información, comunicación e interacción entre los usuarios. De igual forma, para garantizar la confiabilidad de los datos, el acceso a la información contenida en el portal será de acuerdo al tipo de usuario. Con esto se pretende ofrecer información de acuerdo a la necesidad de cada persona



que ingrese al portal, así como también establecer controles de seguridad.

Las metodologías y estándares utilizados en un desarrollo de *software* nos proporcionan las guías para

poder conocer todo el camino a recorrer desde antes de empezar la **implementación**, con lo cual se asegura la calidad del producto final, así como también el cumplimiento en la entrega del mismo en un tiempo estipulado.

Es de suma importancia elegir la metodología adecuada, así como las herramientas de implementación adecuadas, es por ello que la metodología RUP basada en UML nos proporciona todas las bases para llevar al éxito la elaboración de nuestro *software*.

### **Visión General del Proceso Unificado**

El Proceso Unificado es un proceso de desarrollo de software: “conjunto de actividades necesarias para transformar los requisitos del usuario en un sistema software”.

RUP es un marco genérico que puede especializarse para una variedad de tipos de sistemas, diferentes áreas de aplicación, tipos de organizaciones, niveles de aptitud y diferentes tamaños de proyectos.

RUP está basado en componentes. El software está formado por componentes software interconectados a través de interfaces.

RUP está dirigido por casos de uso, centrado en la arquitectura, y es iterativo e incremental.

### **Casos de Uso:**

Un caso de uso es un fragmento de funcionalidad del sistema que proporciona un resultado de valor a un usuario. Los casos de uso modelan los requerimientos funcionales del sistema.

Todos los casos de uso juntos constituyen el modelo de casos de uso.

Los casos de uso también guían el proceso de desarrollo (diseño, implementación, y prueba). Basándose en los casos de uso los desarrolladores crean una serie de modelos de diseño e implementación que llevan a cabo los casos de uso. De este modo los casos de uso no solo inician el proceso de desarrollo sino que le proporcionan un hilo conductor, avanza a través de una serie de flujos de trabajo que parten de los casos de uso.

### **Centrado en la Arquitectura**

La arquitectura de un sistema software se describe mediante diferentes vistas del sistema en construcción.

El concepto de arquitectura software incluye los aspectos estáticos y dinámicos más significativos del sistema.

La arquitectura es una vista del diseño completo con las características más importantes resaltadas, dejando los detalles de lado.

Los casos de uso y la arquitectura están profundamente relacionados. Los casos de uso deben encajar en la arquitectura, y a su vez la arquitectura debe permitir el desarrollo de todos los casos de uso requeridos, actualmente y a futuro

### **Iterativo e Incremental**

Es práctico dividir el esfuerzo de desarrollo de un proyecto de software en partes más pequeñas o mini proyectos.

Cada mini proyecto es una iteración que resulta en un incremento.

Las iteraciones hacen referencia a pasos en el flujo de trabajo, y los incrementos o crecimientos en el producto.

Las iteraciones deben estar controladas. Esto significa que deben seleccionarse y ejecutarse de una forma planificada.

Los desarrolladores basan la selección de lo que implementarán en cada iteración en dos cosas: el conjunto de casos de uso que amplían la funcionalidad, y en los riesgos más importantes que deben mitigarse.

En cada iteración los desarrolladores identifican y especifican los casos de uso relevantes, crean un diseño utilizando la arquitectura seleccionada como guía, para implementar dichos casos de uso. Si la iteración cumple sus objetivos, se continúa con la próxima. Si no deben revisarse las decisiones previas y probar un nuevo enfoque.

### **El Ciclo de Vida del Proceso Unificado**

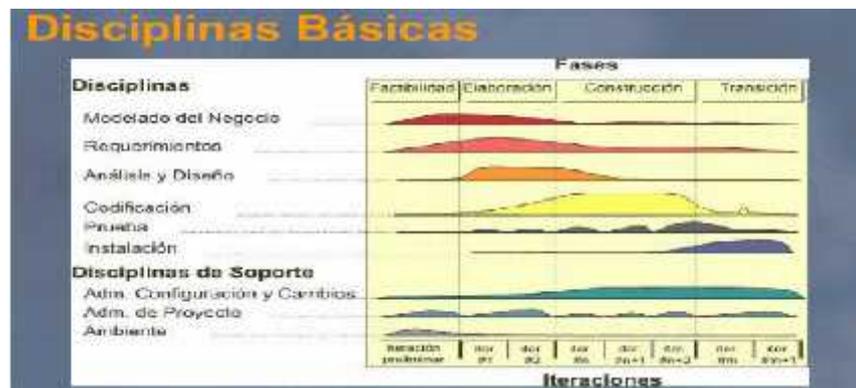
El Proceso Unificado se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema. Cada ciclo constituye una **versión** del sistema.

### **Fases**

Cada ciclo consta de cuatro fases: **inicio**, **elaboración**, **construcción**, y **transición**.



Cada fase se subdivide en **iteraciones**. En cada iteración se desarrolla en secuencia un conjunto de **disciplinas** o flujos de trabajos.



### Disciplinas

Cada disciplina es un conjunto de actividades relacionadas (flujos de trabajo) vinculadas a un área específica dentro del proyecto total. Las más importantes son:

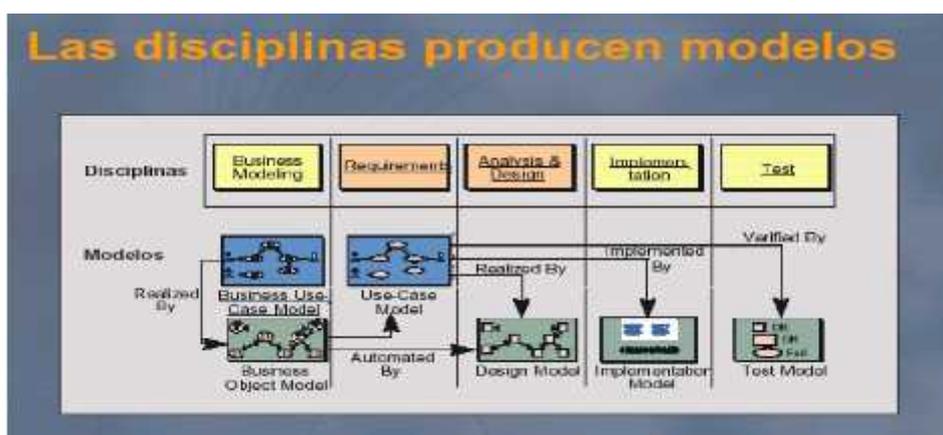
**Requerimientos, Análisis, Diseño, Codificación, y Prueba.**

El agrupamiento de actividades en disciplinas es principalmente una ayuda para comprender el proyecto desde la visión tradicional en cascada.



Cada disciplina está asociada con un conjunto de **modelos** que se desarrollan. Estos modelos están compuestos por **artefactos**. Los artefactos más importantes son los modelos que cada disciplina realiza: **modelo de casos de uso, modelo de diseño, modelo de implementación, y modelo de prueba.**

El Proceso Unificado consiste en una serie de disciplinas o flujos de trabajo que van desde los requisitos hasta las pruebas. Los flujos de trabajo desarrollan modelos desde el modelo de casos de uso hasta el modelo de pruebas.



DISCIPLINA	MODELOS
Requisitos	Modelo de Casos de Uso
Análisis	Modelo de Análisis
Diseño	Modelo de Diseño – Modelo de Despliegue
Implementación	Modelo de Implementación
Prueba	Modelo de Prueba

### **Hitos**

Cada fase finaliza con un **hito**. Cada hito se determina por la disponibilidad de un conjunto de artefactos, es decir un conjunto de modelos o documentos que han sido desarrollados hasta alcanzar un estado predefinido.

Los hitos tienen muchos objetivos. El más crítico es que los directores deben tomar ciertas decisiones antes de que el trabajo continúe con la siguiente fase.

Los hitos también permiten controlar la dirección y progreso del trabajo.

Al final se obtiene un conjunto de datos a partir del seguimiento del tiempo y esfuerzo consumidos en cada fase. Estos datos son útiles para las estimaciones en futuros proyectos.

### **Fase de Inicio**

Durante la fase de inicio se desarrolla una descripción del producto final, y se presenta el análisis del negocio. Esta fase responde las siguientes preguntas:

**¿Cuáles son las principales funciones del sistema para los usuarios más importantes?**

**¿Cómo podría ser la mejor arquitectura del sistema?**

**¿Cuál es el plan del proyecto y cuánto costará desarrollar el producto?**

En esta fase se identifican y priorizan los riesgos más importantes.

El objetivo de esta fase es ayudar al equipo de proyecto a decidir cuáles son los verdaderos objetivos del proyecto.

Las iteraciones exploran diferentes soluciones posibles, y diferentes arquitecturas posibles.

Puede que todo el trabajo físico realizado en esta fase sea descartado. Lo único que normalmente sobrevive a la fase de inicio es el incremento del conocimiento en el equipo.

Los artefactos que típicamente sobreviven a esta fase son:

- *Un enunciado de los mayores requerimientos planteados generalmente como casos de uso.*
- *Un boceto inicial de la arquitectura.*
- *Una descripción de los objetivos del proyecto.*
- *Una versión muy preliminar del plan del proyecto.*
- *Un modelo del negocio.*

La fase de inicio finaliza con el Hito de Objetivos del Ciclo de Vida.

Este hito es alcanzado cuando el equipo de proyectos y los stakeholders llegan a un acuerdo sobre:

- *Cuál es el conjunto de necesidades del negocio, y que conjunto de funciones satisfacen estas necesidades.*
- *Una planificación preliminar de iteraciones.*
- *Una arquitectura preliminar.*

Debe poder responderse las siguientes cuestiones:

¿Se ha determinado con claridad el ámbito del sistema? ¿Se ha determinado lo que va a estar dentro del sistema y fuera del sistema?

¿Se ha llegado a un acuerdo con todas las personas involucradas (stakeholders) sobre los requisitos funcionales del sistema?

¿Se vislumbra una arquitectura que pueda soportar estas características?

¿Se identifican los riesgos críticos? ¿Se prevé forma de mitigarlos?

¿El uso del producto justifica la relación costo-beneficio?

¿Es factible para su organización llevar adelante el proyecto?

¿Están los inversores de acuerdo con los objetivos?

### **Fase de Elaboración**

Durante la fase de **elaboración** se especifican en detalle la mayoría de los casos de uso del producto y se diseña la arquitectura.

Las iteraciones en la fase de elaboración:

- *Establecen una firme comprensión del problema a solucionar.*
- *Establece la fundación arquitectural para el software.*
- *Establece un plan detallado para las siguientes iteraciones.*

- *Elimina los mayores riesgos.*

El resultado de esta fase es la línea base de la arquitectura.

### **Fase de Construcción**

Durante la fase de **construcción** se crea el producto. La línea base de la arquitectura crece hasta convertirse en el sistema completo.

Al final de esta fase, el producto contiene todos los casos de uso implementados, sin embargo puede que no esté libre de defectos.

Los artefactos producidos durante esta fase son:

- *El sistema software*
- *Los casos de prueba*
- *Los manuales de usuario*

La fase de construcción finaliza con el **hito de Capacidad Operativa Inicial**.

Este hito se alcanza cuando el equipo de desarrollo y los stakeholders llegan a un acuerdo sobre:

- El producto es estable para ser usado
- El producto provee alguna funcionalidad de valor
- Todas las partes están listas para comenzar la transición

### **Fase de Transición**

La fase de **transición** cubre el período durante el cual el producto se convierte en la versión beta.

Las iteraciones en esta fase continúan agregando características al sw. Sin embargo las características se agregan a un sistema que el usuario se encuentra utilizando activamente.

Los artefactos construidos en esta fase son los mismos que en la fase de construcción. El equipo se encuentra ocupado fundamentalmente en corregir y extender la funcionalidad del sistema desarrollado en la fase anterior.

La fase de transición finaliza con el **hito de Lanzamiento del Producto**.

Este hito se alcanza cuando el equipo de desarrollo y los **stakeholders** llegan a un acuerdo sobre:

- *Se han alcanzado los objetivos fijados en la fase de Inicio.*
- *El usuario está satisfecho.*

### **Un Proceso conducido por casos de uso**

El Modelo de Caso de Usos representa los requisitos funcionales

La primera disciplina que se desarrolla dentro de cada iteración es la de requerimientos (posiblemente luego de realizar un modelado del dominio o del negocio). El objetivo de esta fase es determinar los requerimientos del sistema. Los requerimientos funcionales son plasmados a través de casos de uso en un Modelo de Casos de Uso.

El modelo de casos de uso ayuda al cliente, a los usuarios, y a los desarrolladores a llegar a un acuerdo sobre cómo utilizar el sistema.

Cada tipo de usuario del sistema se representa mediante un actor que define un rol de utilización del sistema.

Los actores modelan el entorno del sistema, y los casos de uso especifican el sistema.

Un diagrama de casos de uso describe parte del modelo de casos de uso y muestra un conjunto de casos de uso y actores asociados.

### **Un proceso centrado en la arquitectura**

La arquitectura software abarca decisiones importantes sobre:

- *La organización del sistema sw.*
- *Los elementos estructurales que compondrán el sistema y sus interfaces.*
- *La composición de los elementos estructurales y del comportamiento en subsistemas progresivamente más grandes*
- *El estilo de la arquitectura que guía esta organización: los elementos y sus interfaces, sus colaboraciones y su composición.*

La arquitectura se representa mediante vistas del modelo:

- *una vista del modelo de casos de uso*
- *una vista del modelo de análisis*
- *una vista del modelo de diseño*
- *una vista del modelo de despliegue*
- *una vista del modelo de implementación*

Estas vistas solo tienen elementos que son arquitectónicamente significativos. Por Ej. La vista de los casos de uso tiene los actores y casos de uso arquitectónicamente significativos. Lo mismo sucede en los modelos de análisis y diseño.

### **Importancia y necesidad de una arquitectura**

Se necesita una arquitectura para:

- *Comprender el sistema*
- *Organizar el desarrollo*
- *Fomentar la reutilización*
- *Hacer evolucionar el sistema*

### **Desarrollo de la arquitectura**

La arquitectura se desarrolla mediante iteraciones, principalmente en la etapa de elaboración.

El resultado de la fase de elaboración es la línea base de la arquitectura – un esqueleto del sistema con pocos músculos de software.

Los casos de uso que son relevantes para la arquitectura son resumidamente aquellos que mitigan los mayores riesgos del proyecto, aquellos que son más importantes para el usuario, y aquellos que nos ayudan a cubrir todas las funcionalidades significativas.

Al final de la fase de elaboración hemos desarrollado modelos del sistema que representan los casos de uso más importantes y sus realizaciones desde el punto de vista de la arquitectura.

Esta agregación de modelos es la línea base de la arquitectura. Es un sistema pequeño y delgado. Tiene las versiones de todos los modelos que un sistema terminado contiene al final de la fase de construcción. Incluye el mismo esqueleto de subsistemas, componentes y nodos que un sistema definitivo, pero no existe toda la musculatura. Es un sistema ejecutable.

### **Descripción de la arquitectura**

La línea base de la arquitectura, es la versión interna del sistema al final de la fase de elaboración. El conjunto de modelos que describen esta línea base se denomina Descripción de la Arquitectura.

El papel de la descripción de la arquitectura es guiar al equipo de desarrollo a través del ciclo de vida del sistema.

La descripción de la arquitectura puede adoptar diferentes formas. Puede ser un extracto de los modelos que son parte de la línea base de la arquitectura, o puede ser una reescritura de los extractos de forma que sea más fácil leerlos.

La descripción de la arquitectura tiene cinco secciones, una para cada modelo: una vista del modelo de casos de uso, una vista del modelo de análisis (opcional /descartable), una vista del modelo de diseño, una vista del modelo de despliegue, y una vista del modelo de implementación.

### **La vista de la arquitectura del modelo de casos de uso**

Presenta los actores y casos de uso más importantes.

**Ej.** *Vista de la arquitectura del modelo de casos de uso de un sistema de cajero automático (CA)*

*En el ejemplo del CA el caso de uso más importante es Sacar Dinero. Sin él, no tendría sentido el CA.*

*Para definir la arquitectura por tanto, el arquitecto sugiere que el caso de uso Sacar*

*Dinero se implemente en su totalidad durante la fase de elaboración.*

### **La vista de la arquitectura del modelo de diseño**

Presenta los clasificadores más importantes para la arquitectura pertenecientes al modelo de diseño: los subsistemas e interfaces más importantes, así como algunas pocas clases muy importantes, fundamentalmente clases activas.

También presentan como se realizan los casos de uso en términos de esos clasificadores.

**Ej.** *Vista de la arquitectura del modelo de diseño del sistema de cajero automático (CA)*

*Se incluyen las tres clases activas: Gestor de Clientes, Gestor de Transacciones, y Gestor de Cuentas.*

*También se incluyen los subsistemas: Interfaz del CA, Gestión de Transacciones, y Gestión de Cuentas, por ser necesarios para la realización del caso de uso Sacar Dinero.*

### **La vista de la arquitectura del modelo de despliegue**

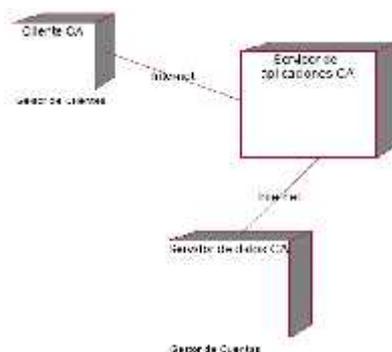
Presenta los nodos interconectados y las clases activas que se ejecutan en ellos identificados durante el diseño.

Esto puede mostrarse por diagramas de despliegue.

**Ej. Vista de la arquitectura del modelo de despliegue del sistema de cajero automático (CA)**

*Se incluyen los siguientes nodos y objetos activos:*

- *Nodo: Cliente CA – Objeto activo: Gestor de Clientes*
- *Nodo: Servidor de aplicaciones CA – Objeto activo: Gestor de transacciones*
- *Nodo: Servidor de datos CA – Objeto activo: Gestor de Cuentas*



**La vista de la arquitectura del modelo de**

### **implementación**

El modelo de implementación es una correspondencia directa de los modelos de diseño y de despliegue.

Cada subsistema de servicio del diseño normalmente termina siendo un componente por cada tipo de nodo en el que deba instalarse.

### **Un proceso iterativo e incremental**

#### **Desarrollo en pequeños pasos**

La tercera clave importante del RUP consiste en desarrollar un producto software en pasos pequeños manejables:

- *Planificar un poco.*
- *Especificar, diseñar, e implementar un poco.*
- *Integrar, probar, y ejecutar un poco en cada iteración.*

Las iteraciones en las primeras fases tratan en su mayor parte con la determinación del ámbito del proyecto, la eliminación de los riesgos críticos, y la creación de la línea base de la arquitectura. Después, a medida que avanzamos a lo largo del proyecto y vamos reduciendo gradualmente los riesgos restantes e implementado los componentes, la forma de las iteraciones cambia, dando incrementos como resultados.

### **Captura de Requisitos**

El propósito fundamental del flujo de trabajo de los requisitos es guiar el desarrollo hacia el sistema correcto.

Hay diferentes puntos de partida para la captura de requisitos. En algunos casos comenzamos haciendo un *modelo del negocio* o partimos de uno ya desarrollado. En otros casos si es un sistema acotado que no da soporte al negocio podemos partir de un modelo de objetos sencillo como un *modelo del dominio*.

En otros casos el cliente puede ya haber desarrollado una especificación completa de requisitos no basada en objetos, de la cual podemos partir.

En forma típica, el flujo de trabajo de requisitos incluye los siguientes pasos:

a) *Enumerar los requisitos candidatos.*

b) *Comprender el contexto del sistema.*

c) *Capturar requisitos funcionales.*

d) *Capturar requisitos no funcionales.*

**a) Enumerar los requisitos candidatos**

La lista de características deseadas del sistema constituyen los requisitos candidatos.

De cada característica se registra:

- Nombre corto
- Descripción
- Estado (propuesto, aprobado, incluido, o validado)
- Coste estimado de implementación (en término de tipos de recursos y horas-hombre)
- Prioridad (crítico, importante, o secundario)
- Nivel de riesgo asociado a la implementación de la característica (crítico, significativo, ordinario)

Estos valores se utilizan para estimar el tamaño del proyecto y decidir cómo dividirlo en secuencia de iteraciones. La prioridad y nivel de riesgo asociados por ejemplo, se utiliza para decidir en qué iteración se implementará la característica.

**b) Comprender el contexto del sistema**

Hay por lo menos dos aproximaciones para expresar el contexto de un sistema: modelado del dominio y modelado del negocio.

Un modelo del dominio describe los conceptos importantes del contexto como objetos del dominio relacionados entre sí.

Un modelo del negocio es más amplio. Describe los procesos con el objetivo de comprenderlos. El modelado del negocio especifica que procesos de negocio soportará el sistema.

### **c) Capturar requisitos funcionales**

Los requisitos funcionales son capturados por medio de casos de uso, que conforman el modelo de casos de uso. Los casos de uso también capturan requisitos no funcionales específicos de un caso de uso determinado.

### **d) Capturar requisitos no funcionales**

Los requisitos no funcionales especifican propiedades del sistema, como restricciones del entorno o de la implementación, rendimientos, etc.

Hay requisitos no funcionales específicos para un caso de uso y otros genéricos para la aplicación. Los que son específicos para un caso de uso, pueden documentarse junto con el caso de uso correspondiente. Los que son más genéricos se documentan por medio de una lista de requisitos adicionales.

### **Modelo del Dominio**

Un modelo del dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las “cosas” que existen o los eventos que suceden en el entorno en el que trabaja el sistema.

Las clases del dominio aparecen en tres formas típicas:

- Objetos del negocio que representan cosas que se manipulan en el negocio, como pedidos, cuentas, contratos, etc.
- Objetos del mundo real y conceptos de los que el sistema debe hacer seguimiento como aviación enemiga, misiles, trayectorias, etc.
- Sucesos que ocurrirán o han ocurrido, como llegada de un avión, su salida, hora de la comida, etc.

El modelo de dominio se representa fundamentalmente por diagramas de clases en UML.

El objetivo del modelado del dominio es comprender y describir las clases más importantes dentro del contexto del sistema.

### **Modelo del Negocio**

El modelado del negocio es una técnica para comprender los procesos de negocio de la organización.

El modelado del negocio está soportado por dos tipos de modelos de UML: el modelado de casos de uso, y modelos de objetos.

Un Modelo de Casos de Uso del Negocio describe los procesos de negocio de una empresa en términos de

casos de uso del negocio y actores del negocio que se corresponden con los procesos del negocio y los clientes respectivamente.

Al igual que el modelo de casos de uso para un sistema software, el modelo de casos de uso del negocio presenta un sistema (en este caso, el negocio) desde la perspectiva de su uso, y esquematiza como proporciona valor a sus usuarios.

El modelo de casos de uso del negocio se describe mediante diagramas de casos de uso.

Un modelo de objetos del negocio describe como cada caso de uso del negocio es llevado a cabo por parte de un conjunto de trabajadores que utilizan un conjunto de entidades del negocio y de unidades de trabajo.

Cada realización de un caso de uso del negocio puede mostrarse en diagramas de interacción y diagramas de actividad.

Una entidad del negocio representa algo que los trabajadores toman, manipulan, inspeccionan, producen o utilizan en un negocio.

Una unidad de trabajo es un conjunto de esas entidades que conforma un todo reconocible para el usuario final.

La técnica de modelado de negocio identifica entidades y trabajadores que participan en la realización de los casos de uso del negocio.

Los trabajadores identificados en el modelo de negocio se utilizan como punto de partida para derivar un primer conjunto de actores y casos de uso del sistema.

### **Búsqueda de Casos de Uso a partir de un modelo del negocio**

En primer lugar se identifica un actor por cada trabajador y por cada actor del negocio (es decir, el cliente).

Cada trabajador y actor del negocio que vaya a ser usuario del sistema de información requerirá un soporte por parte del mismo. El soporte necesario se determina tratando cada uno de los actores uno detrás de otro.

Una vez que hemos encontrado todos los roles de un trabajador o actor del negocio, uno por cada caso de uso del negocio en el que participa, podemos encontrar los casos de uso de los actores del sistema.

La manera más directa de identificar un conjunto tentativo de casos de uso es crear un caso de uso para el actor correspondiente a cada rol de cada trabajador y de cada actor del negocio. Los analistas pueden después ajustar los casos de uso tentativos.

### **Requisitos adicionales**

Los requisitos adicionales, son requerimientos No funcionales que no pueden asociarse a ningún caso de uso en particular. Algunos ejemplos son el rendimiento, las interfaces, y los requisitos de diseño físico, así como las restricciones arquitectónicas.

Los requisitos adicionales se capturan de forma muy parecida a como se hacía en la especificación de requisitos tradicional, es decir con una lista de requisitos. Luego se utilizan durante el análisis junto al modelo de casos de uso.

### **Captura de requisitos como casos de uso**

Los requisitos funcionales se estructuran de forma natural mediante casos de uso que constituyen el Modelo de Casos de Uso.

Los requisitos no funcionales restantes, se modelan como dijimos en el documento de requisitos adicionales.

### **Beneficios del UML**

Los beneficios aportados por el UML son:

- UML promueve el desarrollo de procesos manejados por casos de uso, centrado en arquitectura, iterativos e incrementales.
- Utiliza conceptos de alto nivel de desarrollo tales como colaboraciones, armazones, modelos y componentes, definiendo claramente la semántica de estos conceptos lo cual es esencial para obtener los beneficios de la orientación de objetos, colocando dentro de un contexto completo un lenguaje de modelado único.
- Provee a los desarrolladores un lenguaje de Modelamiento visual listo para utilizar, es así como nosotros podemos desarrollar e intercambiar modelos orientados a objetos significativos. El UML consolida un conjunto de conceptos que son generalmente aceptados por muchos métodos y herramientas de modelado y necesarios en una amplia gama de aplicaciones. Este es uno de los principales beneficios aportados por el UML, permitiendo el avance de la industria del software para construir modelos que puedan ser utilizados por diferentes

herramientas, debido a su aceptación como un estándar de modelado.

- Es independiente de los lenguajes de programación y de métodos y procesos de desarrollo de software. UML puede y debe soportar todos los lenguajes de programación y varios métodos y procesos para construir modelos sin mayor dificultad.
- Proporciona una base formal para entender el lenguaje de modelado. Los usuarios usan la formalidad para ayudarse a comprender el lenguaje, pero el formalismo no debe requerir muchos niveles o capas y uso excesivo de matemáticas. UML provee de una definición formal del modelo estático usando el diagrama de clases. Este diagrama es muy popular y ampliamente aceptado como aproximación formal de un modelo y del intercambio de información, pero además, el UML expresa las restricciones en OCL (ObjectConstraintLenguaje) y las operaciones en un lenguaje natural muy preciso.

### **Metodología UML:**

Es un conjunto de herramientas, que permite modelar (analizar y diseñar) sistemas orientados a objetos.

### **Diagrama de Clases:**

*Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de*

*los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargaran del funcionamiento y la relación entre uno y otro.*

### **Diagrama de Componentes:**

*Representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos componentes.*

*Los componentes físicos incluyen archivos, cabeceras, módulos, ejecutables, o paquetes. 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.*

### **Diagrama de Estado:**

*Un Diagrama de Estados muestra la secuencia de estados por los que pasa bien un caso de uso, bien un objeto a lo largo de su vida, o bien todo el sistema. En él se indican qué eventos hacen que se pase de un estado a otro y cuáles son las respuestas y acciones que genera.*

*En cuanto a la representación, un diagrama de estados es un grafo cuyos nodos son estados y cuyos arcos dirigidos son transiciones etiquetadas con los nombres de los eventos.*

### **Diagrama de Interacción:**

*En los diagramas de interacción se muestra un patrón de interacción entre objetos. Hay dos tipos de diagrama de interacción, ambos basados en la misma información, pero cada uno enfatizando un aspecto*

*particular: Diagramas de Secuencia y Diagramas de Colaboración.*

**Diagrama de Secuencia:**

*Un diagrama de Secuencia muestra una interacción ordenada según la secuencia temporal de eventos. En particular, muestra los objetos participantes en la interacción y los mensajes que intercambian ordenados según su secuencia en el tiempo.*

**Diagrama de Colaboración:**

*Muestra a una serie de objetos con los enlaces entre los mismos, y con los mensajes que se intercambian dichos objetos. Los mensajes son flechas que van junto al enlace por el que “circulan”, y con el nombre del mensaje y los parámetros (si los tiene) entre paréntesis. Cada mensaje lleva un número de secuencia que denota cuál es el mensaje que le precede, excepto el mensaje que inicia el diagrama, que no lleva número de secuencia.*

**Diagrama de Caso de Uso:**

*Un Diagrama de Casos de Uso muestra 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.*

**Diagrama de Actividades:**

*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.*

### **Diagrama de Despliegue:**

*Los Diagramas de Despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria.*

## **2.1.2 HERRAMIENTAS DE SOFTWARE**

### **❖ Microsoft SQL Server**

Microsoft con su software SQL Server 2008 nos ofrece una plataforma de gestión de datos muy óptima, al cual podemos acceder desde cualquier lugar y en cualquier momento. Con SQL Server 2008 se puede almacenar datos estructurados, semi-estructurados, no estructurados y documentos, tales como las imágenes y más; de forma directamente en el base de datos.

SQL Server 2008 posee los más altos niveles de seguridad, fiabilidad y escalabilidad, para obtener los mejores resultados en aplicativos empresariales; ya que con SQL Server 2008 se puede reducir el costo y el tiempo para la gestión de datos y el desarrollo de aplicativos.

Visión multiplataforma de datos de Microsoft.

### **Soluciones que ofrece SQL SERVER 2008**

### **Business Intelligence**

SQL Server 2008 proporciona una plataforma optimizada para la integración de datos aplicables para Business Intelligence, como presentación de informes y análisis, permitiendo a las organizaciones a ofrecer inteligencia donde los usuarios quieren.

### **La virtualización**

Virtualización de servidores, también conocida como la virtualización hardware, es un tema candente en el mundo de TI, ya que puede proporcionar un gran ahorro de costos, mientras aumenta la rapidez de la empresa.

### **Consolidación de Servidores**

SQL Server 2008 optimiza el hardware y reduce los costos de mantenimiento mediante el suministro de una flexible solución de consolidación de servidores con la clase empresarial de gestión y rendimiento.

### **OLTP**

SQL Server 2008 proporciona escalabilidad, de alto rendimiento en su motor de base de datos para aplicaciones de misión crítica que requieren los más altos niveles de disponibilidad y seguridad.

### **Data Warehousing**

SQL Server 2008 proporciona una completa y escalable plataforma de almacenamiento de datos que permite a las organizaciones integrar los datos en el almacén de datos más rápida, y administrar los crecientes volúmenes de datos y los usuarios.

### Desarrollo de Aplicaciones

SQL Server 2008 también es una amplia plataforma para el desarrollo de aplicativos de datos que permite acceder y manipular datos esenciales de una gran variedad de dispositivos, plataformas, y los datos.

- **¿Cómo podría saber si ha copiado la base de datos del servidor de producción de un cliente y lo han instalado en otra base de datos o si están accediendo a la información?**

Con SQL 2008, puede proteger la información con una clave de protección (Encriptación).

**¿Cómo podría saber que datos están siendo leídos y modificados, a qué hora y por quién? SQL 2008 da la opción de Auditora de Datos.**

### **Continuidad del Negocio**

- Si sus clientes necesitan estar siempre en línea con sus sistemas sin caídas, SQL 2008 ofrece mejoras en una técnica llamada “Mirroring”, el cual es una copia o espejo de la base de datos.
- Si el disco se daña, donde reside los datos, SQL 2008 recupera la información de una copia reciente de los datos dañados al otro equipo espejo de manera transparente.
- **Ahorro en espacio en disco**, mediante la técnica de compresión, ahorrando costos en compra de discos si es que el volumen de la información de Base de Datos empieza a crecer en forma rápida.

### **Datos Geo-Espaciales**

- Poder manejar información geográfica, la que hoy en día es de alta importancia en las organizaciones, con todo el tema de globalización.

### **Acceder a la Información desde cualquier lugar en cualquier momento**

Con SQL 2008 podre crear rápidamente aplicaciones conectadas a la base de datos con la funcionalidad de funcionar en forma desconectada y después sincronizarlos con la base de datos central sin perder la línea de negocio y manteniendo los datos validados.

### **Reportes**

Poder acceder a reportes directamente desde Word, mejoras en los tipos de gráficos en los reportes, haciéndolos más entendibles y poder editar los reportes de Microsoft Office, sin saber dónde fue diseñado el reporte.

### **❖ Microsoft Project**

Microsoft Project (o MSP) es un software de administración de proyectos diseñado, desarrollado y comercializado por Microsoft para asistir a administradores de proyectos en el desarrollo de planes, asignación de recursos a tareas, dar seguimiento al progreso, administrar presupuesto y analizar cargas de trabajo.

### **❖ El IBM Rational Rose**

El RationalUnifiedProcess o Proceso Unificado de Racional. Es un proceso de ingeniería de software que suministra un enfoque para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la

producción de software de alta calidad que satisfaga la necesidad del usuario final dentro de un tiempo y presupuesto previsible. Es una metodología de desarrollo iterativo enfocada hacia “los casos de uso, manejo de riesgos y el manejo de la arquitectura”.

El RUP mejora la productividad del equipo ya que permite que cada miembro del grupo sin importar su responsabilidad específica acceda a la misma base de datos de conocimiento.

➤ **Desarrollo Iterativo**

Rational Rose utiliza un proceso de desarrollo iterativo controlado (controlled iterative process development), donde el desarrollo se realiza por una serie de iteraciones. Cada iteración inicia con una primera aproximación del análisis, diseño e implementación con el fin de identificar los riesgos del diseño, los cuales se utilizan para conducir el proceso de iteración, es decir, primero se identifican los riesgos y después se prueba la aplicación para que éstos se hagan mínimos.

Cuando la implementación pasa todas las pruebas que se determinan en el proceso, ésta se revisa y se añaden los elementos modificados al modelo de análisis y diseño. Una vez que la actualización del modelo se ha modificado, se realiza la siguiente iteración.

➤ **Trabajo en Grupo**

La herramienta permite que varias personas trabajen en paralelo en un proceso iterativo controlado, para ello asigna a cada desarrollador un espacio de trabajo privado que contiene el modelo completo y tenga un control exclusivo sobre la propagación de los cambios en ese espacio de trabajo. También es posible descomponer el modelo en unidades controladas e integrarlas con un sistema para realizar el

control de proyectos que permite mantener la integridad de dichas unidades.

➤ **Generador de Código**

Se puede generar código en distintos lenguajes de programación a partir de un diseño en UML.

➤ **Ingeniería Inversa**

Rational Rose proporciona mecanismos para aplicar Ingeniería Inversa a un código fuente soportado con el fin de extraer a partir de tal código el diseño de la aplicación.

## ❖ **ALLFUSION ERWIN 3.4 DATA MODELER**

Es una Herramienta de modelado de datos (datos de análisis de requerimientos, diseño de base de datos etc.) personalizados desarrollados, sistemas de información, incluyendo bases de datos de sistemas transaccionales y data marts.

### **Características de Erwin Data Modeler**

- **Modelado de datos lógico:** Modelos puramente lógicos pueden crearse, de la cual se pueden derivar modelos físicos. También se admiten combinaciones de modelos lógicos y físicos. Apoya el tipo de entidad y lógica nombres de atributos y descripciones, lógicos dominios y tipos de datos, así como relación de nombres.
- **Modelado de datos físico:** Se pueden crear modelos puramente físicos así como combinaciones de modelos lógicos y físicos. Apoya la denominación y descripción de tablas y columnas, tipos de datos definidos por el usuario, claves primarias, llaves foráneas, claves alternativas y el nombramiento y la definición de

restricciones. También se incluye soporte para índices, vistas, procedimientos almacenados y desencadenadores.

- **Transformación de lógico a físico:** Incluye un diccionario de abreviatura llamado “Nombres de Editor de normas” y una asignación de tipo de datos lógico-a-RDBMS llamado “Tipo de datos estándares Editor”.

### ❖ Microsoft Visual Studio 2008

Visual Studio .NET es la Herramienta Rápida de Desarrollo (RAD) de Microsoft para la siguiente generación de Internet que son los Servicios Web XML. Esta herramienta permite la creación de aplicaciones usando el Marco .NET, es decir usando el CLR, la Librería de Clases, ADO .NET, ASP .NET, etc.

Es un software que brinda las herramientas necesarias para crear, distribuir, administrar y dar mantenimiento a aplicaciones Web distribuidas que usan Servicios Web XML, todo esto con una gran facilidad, rapidez y bajo costo.

Se puede crear aplicaciones Web directamente usando el Framework .NET y algún programa editor, por ejemplo el Bloc de Notas, pero el tiempo que llevaría el desarrollo no justificaría el ahorro de costos, en cambio, si se utiliza una herramienta como Visual Studio .NET el tiempo de desarrollo se reduciría enormemente.

Visual Studio .NET permite también la integración y el uso cruzado de lenguajes de programación: Visual Basic .NET, Visual C# .NET, Visual C++ .NET y JScript .NET

A diferencia de la versión anterior no existe Visual Interdev, ni Visual J++, además Visual Foxpro .NET no comparte las características unificadas del Marco .NET

Las nuevas capacidades incluyen:

➤ **Integración para el administrador de la base de datos en el ciclo de vida del software:**

Poner a los profesionales de la base de datos en el ciclo de vida de desarrollo a través de creación de diagramas de representación sin tener una conexión activa a la de base de datos y proyectos de base de datos. Proporcionar un completo conjunto de herramientas de origen de control, pruebas y generación de datos para pruebas, refactoring y una solución implementación que incluye una herramienta visual de diff/merge e implementación de generación de secuencias de comandos.

➤ **Pruebas unitarias de Visual Studio:**

Con esta versión se extienden las capacidades las pruebas unitarias para mejorar el rendimiento y ampliar el alcance: pruebas unitarias ahora se ejecutan con mayor rapidez desde ambos el IDE o desde la línea de comandos, la herencia de prueba permite a los usuarios volver a utilizar los métodos heredados, facilidad de uso mejoras permiten a los usuarios ejecutar una prueba directamente desde su definición, las pruebas unitarias están ahora disponibles para todos los usuarios de Visual Studio Professional Edition y también se pueden utilizar para probar aplicaciones móviles.

➤ **Pruebas de cargas empresariales:**

Visual Studio mejora en las pruebas de capacidad simplificando la interfaz de pruebas de carga y proporcionando una vista de gráfica de varios equipos para ver el rendimiento y el estado de todos los equipos en solamente una prueba. Otras mejoras incluyen mejor administración de los resultados de las pruebas, un

modelo enriquecido de carga y la integración de resultados procedentes de otros orígenes (no PerfMon).

➤ **Optimización del rendimiento y diagnósticos de aplicaciones empresariales a través de pruebas:**

Una nueva área de compatibilidad en Visual Studio es la capacidad de dirigir la optimización del rendimiento del sistema y probar herramientas de diagnóstico a través de Visual Studio. Esto permite a que se ejecute la generación de perfiles durante las pruebas, de modo que se puede ejecutar carga y probar procedimientos contra un sistema, vea cómo se comporta, y uso integrado herramientas una nueva área de compatibilidad en Visual Studio es la capacidad para dirigir optimización del rendimiento del sistema y probar las herramientas de diagnóstico a través de Visual Studio para generar perfiles, depurar y optimizar. También incluimos un historial de rendimiento que los usuarios puedan guardar como perfil de línea de base y, a continuación, si el rendimiento disminuye, comparar para identificar los orígenes de la regresión.

❖ **NET FRAMEWORK 3.5**

Es una plataforma (conjunto de componentes y servicios) de programación, diseñados para simplificar el desarrollo de aplicaciones:

*-Configurable mediante la utilidad mscorcfg.msc.*

*-Las versiones pueden coexistir sin problemas.*

❖ **REDES LAN**

Un sistema de transmisión de datos que permite compartir recursos e información por medio de ordenadores, o redes de ordenadores.

Una definición más completa y actual de Red local sería: Un sistema de comunicaciones capaz de facilitar el intercambio de datos informáticos, voz, facsímil, vídeo conferencias, difusión de vídeo, telemetría y cualquier otra forma de comunicación electrónica.

Existe no obstante una definición oficial, la del Comité IEEE 802, quien define una Red local de la siguiente manera: Una Red local es un sistema de comunicaciones que permite que un número de dispositivos independientes se comuniquen entre sí. Una Red local, como su nombre indica, debe ser local en cuanto al ámbito geográfico, aunque local puede significar cualquier cosa, desde una oficina o un edificio de ocho plantas, hasta un complejo industrial con docenas de edificios con muchos pisos.

El término de red local incluye tanto el software con el hardware necesario para la conexión, gestión y mantenimiento de los dispositivos y para el tratamiento de la información.

Y sus principales características de las redes LAN son:

- ✓ Entornos de pocos Km. (normalmente no suele superar los 3.000 metros)
- ✓ Uso de un medio de comunicación privado.
- ✓ Altas velocidades de transmisión (entre 1 y 5 millones de bits por segundo).
- ✓ La simplicidad del medio de transmisión que utiliza (cable coaxial, cables telefónicos y fibra óptica).
- ✓ La facilidad con que se pueden efectuar cambios en el hardware y el software.
- ✓ Gran variedad y número de dispositivos conectados.
- ✓ Posibilidad de conexión con otras redes.
- ✓ La facilidad de uso.

Componentes para la implementación de una Red LAN:

- ★ **Servidor:** el servidor es aquel o aquellos ordenadores que van a compartir sus recursos hardware y software con los demás equipos de la red. Sus características

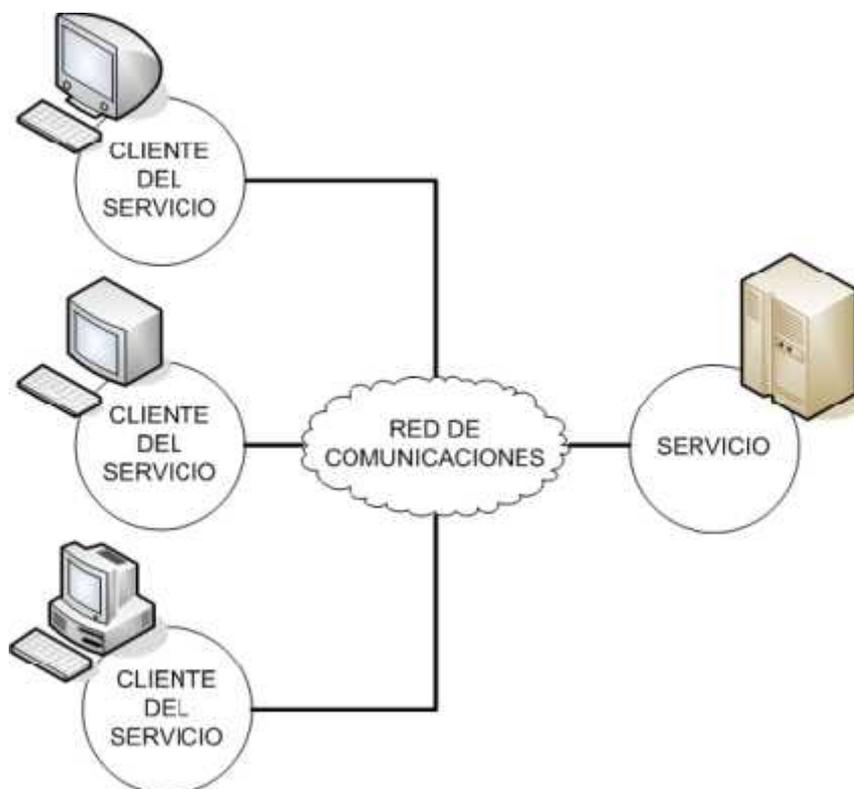
son potencia de cálculo, importancia de la información que almacena y conexión con recursos que se desean compartir.

- ★ **Estación de trabajo:** los ordenadores que toman el papel de estaciones de trabajo aprovechan o tienen a su disposición los recursos que ofrece la red así como los servicios que proporcionan los Servidores a los cuales pueden acceder.
- ★ **Gateways o pasarelas:** es un hardware y software que permite las comunicaciones entre la red local y grandes ordenadores (mainframes). El gateway adapta los protocolos de comunicación del mainframe (X25, SNA, etc.) a los de la red, y viceversa.
- ★ **Bridges o puentes:** es un hardware y software que permite que se conecten dos redes locales entre sí. Un puente interno es el que se instala en un servidor de la red, y un puente externo es el que se hace sobre una estación de trabajo de la misma red. Los puentes también pueden ser locales o remotos. Los puentes locales son los que conectan a redes de un mismo edificio, usando tanto conexiones internas como externas. Los puentes remotos conectan redes distintas entre sí, llevando a cabo la conexión a través de redes públicas, como la red telefónica, RDSI o red de conmutación de paquetes.
- ★ **Tarjeta de red:** también se denominan NIC (Network Interface Card). Básicamente realiza la función de intermediario entre el ordenador y la red de comunicación. En ella se encuentran grabados los protocolos de comunicación de la red. La comunicación con el ordenador se realiza normalmente a través de las ranuras de expansión que éste dispone, ya sea ISA, PCI o PCMCIA. Aunque algunos equipos disponen de este adaptador integrado directamente en la placa base.

- ★ **El medio:** constituido por el cableado y los conectores que enlazan los componentes de la red. Los medios físicos más utilizados son el cable de par trenzado, par de cable, cable coaxial y la fibra óptica (cada vez en más uso esta última).
- ★ **Concentradores de cableado:** una LAN en bus usa solamente tarjetas de red en las estaciones y cableado coaxial para interconectarlas, además de los conectores, sin embargo este método complica el mantenimiento de la red ya que si falla alguna conexión toda la red deja de funcionar
- ★ **Arquitectura Cliente/Servidor.** Forma de computación distribuida en la cual la funcionalidad de la aplicación está dividida entre los recursos de computación de la red. Procesamiento cooperativo de información de negocios mediante un conjunto de procesadores en donde clientes múltiples geográficamente distribuidos inician peticiones que son procesadas por uno o más servidores centrales.
- ★ La arquitectura cliente – servidor proporciona un marco de trabajo para el modelo de diseño lógico, segmenta los componentes de una aplicación en tres capas de servicio. Estas capas no necesariamente corresponden a localizaciones físicas en varias computadoras en una red, sino más bien para el nivel lógico de la aplicación. Como las piezas de una aplicación son distribuidas en una topología física que puede cambiar, dependiendo de los requerimientos del sistema.

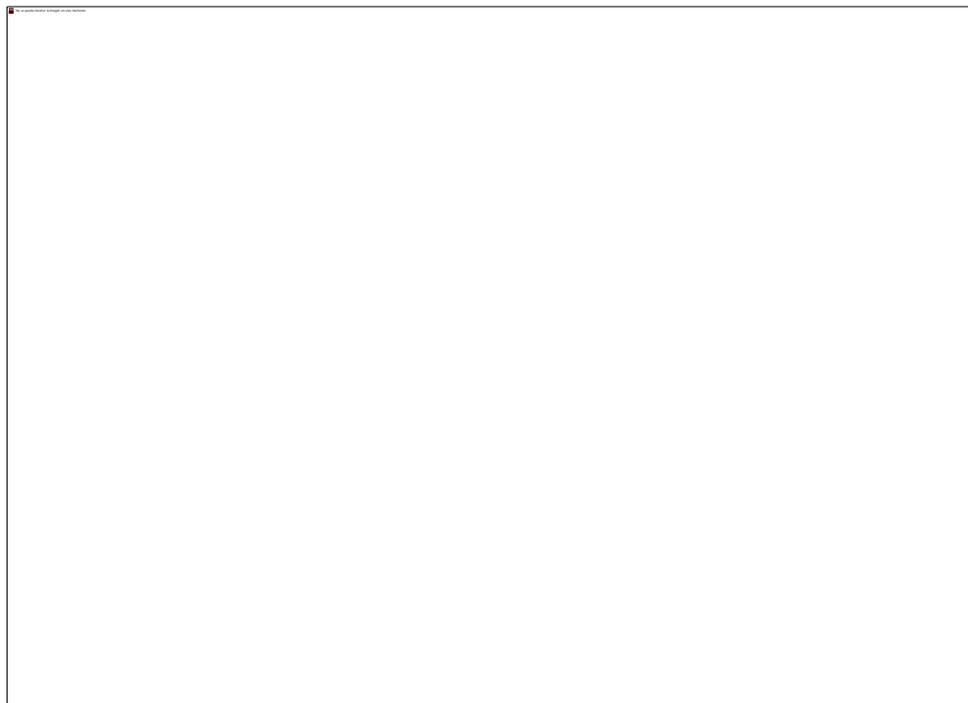
❖ **ARQUITECTURA CLIENTE – SERVIDOR**

La arquitectura cliente-servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras. En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del



sistema. La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma. Una disposición muy común son los sistemas multicapa en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes computadoras aumentando así el grado de distribución del sistema. La arquitectura cliente-servidor sustituye a la arquitectura monolítica en la que no hay distribución, tanto a nivel físico como a nivel lógico. La red cliente-servidor es aquella red de comunicaciones en la que todos los clientes están conectados a un servidor, en el que se centralizan los diversos recursos y aplicaciones con que se cuenta; y que los pone a disposición de los clientes cada vez que estos son solicitados. Esto significa que todas las gestiones que se realizan se concentran en el servidor, de manera que en él se disponen los requerimientos provenientes de los clientes que tienen prioridad, los archivos que son de uso público y los que son de uso restringido, los archivos que son de sólo lectura y los que, por el contrario, pueden ser modificados, etc. Este tipo de red

puede utilizarse conjuntamente en caso de que se esté utilizando en una red mixta.



#### ❖ **WINDOWS XP**

Windows XP (cuyo nombre en clave inicial fue Whistler) es una versión de Microsoft Windows, línea de sistemas operativos desarrollado por Microsoft. Lanzado al mercado el 25 de octubre de 2001, actualmente es el sistema operativo para x86 más utilizado del planeta (con una cuota de mercado del 56.72%) y se considera que existen más de 400 millones de copias funcionando.<sup>4</sup> Las letras "XP" provienen de la palabra eXPeriencia (eXPerience en inglés).

Dispone de versiones para varios entornos informáticos, incluyendo PCs domésticos o de negocios, además de equipos portátiles, "netbooks", "tablet PC" y "centros multimedia". Sucesor de Windows 2000 junto con Windows ME, y antecesor de Windows Vista, es el primer sistema operativo de Microsoft orientado al consumidor que se construye con un núcleo y

arquitectura de Windows NT disponible en versiones para plataformas de 32 y 64 bits.

A diferencia de versiones anteriores de Windows, al estar basado en la arquitectura de Windows NT proveniente del código de Windows 2000, presenta mejoras en la estabilidad y el rendimiento. Tiene una interfaz gráfica de usuario (GUI) perceptiblemente reajustada (denominada Luna), la cual incluye características rediseñadas, algunas de las cuales se asemejan ligeramente a otras GUI de otros sistemas operativos, cambio promovido para un uso más fácil que en las versiones anteriores. Se introdujeron nuevas capacidades de gestión de software para evitar el "DLL Hell" (infierno de las DLLs) que plagó las viejas

Versiones. Es también la primera versión de Windows que utiliza la activación del producto para reducir la piratería del software, una restricción que no sentó bien a algunos usuarios. Ha sido también criticado por las vulnerabilidades de seguridad, integración de Internet Explorer, la inclusión del reproductor Windows Media Player y aspectos de su interfaz.

### **2.1.3 Sistema Operativo**

En esta ocasión describiré el S.O en la cual se desenvolverá el proyecto a implantar:

#### **❖ Windows XP**

Windows XP (cuyo nombre en clave inicial fue Whistler) es una versión de Microsoft Windows, línea de sistemas operativos desarrollado por Microsoft. Lanzado al mercado el 25 de octubre de 2001, actualmente es el sistema operativo para x86 más utilizado del planeta (con una cuota de mercado del 56.72%) y se considera que existen más de 400 millones de copias funcionando.<sup>4</sup> Las letras "XP" provienen de la palabra eXPeriencia (eXPerience en inglés).

Dispone de versiones para varios entornos informáticos, incluyendo PCs domésticos o de negocios, además de equipos portátiles, "netbooks", "tablet PC" y "centros multimedia". Sucesor de Windows 2000 junto con Windows ME, y antecesor de Windows Vista, es el primer sistema operativo de Microsoft orientado al consumidor que se construye con un núcleo y arquitectura de Windows NT disponible en versiones para plataformas de 32 y 64 bits.

A diferencia de versiones anteriores de Windows, al estar basado en la arquitectura de Windows NT proveniente del código de Windows 2000, presenta mejoras en la estabilidad y el rendimiento. Tiene una interfaz gráfica de usuario (GUI) perceptiblemente reajustada (denominada Luna), la cual incluye características rediseñadas, algunas de las cuales se asemejan ligeramente a otras GUI de otros sistemas operativos, cambio promovido para un uso más fácil que en las versiones anteriores. Se introdujeron nuevas capacidades de gestión de software para evitar el "DLL Hell" (infierno de las DLLs) que plagó las viejas

Versiones. Es también la primera versión de Windows que utiliza la activación del producto para reducir la piratería del software, una restricción que no sentó bien a algunos usuarios. Ha sido también criticado por las vulnerabilidades de seguridad, integración de Internet Explorer, la inclusión del reproductor Windows Media Player y aspectos de su interfaz.

## **2.1.4 Redes**

### **❖ ¿Qué es una red?**

Una red, es aquella conexión entre dos o más computadoras, las cuales comparten algunos elementos del hardware, como las impresoras o el CD-ROM, al igual

que información, como diversos archivos de la organización. Al igual que ciertos servicios comunes, como salas de chat y mensajería instantánea, correos electrónicos, etc.

Existen diversas tecnologías de red, siendo las más conocidas las de tipo PPP, la HDLC, ETHERNET, TOKEN RING, entre otras.

Con respecto a los estándares de red, tenemos el IEEE 802.3, IEEE 802.5, IEEE 802.11, IEEE 802.15. El primero se utiliza en el ETHERNET, el segundo en el TOKEN RING, el tercero en el sistema Wi-Fi y el último en el BLUETOOTH.

Ahora, una red, puede llegar a abarcar distintas distancias, por lo que cada una tiene una categorización independiente. Tenemos la PAN, que es una red de área personal que abarca hasta 10 metros cuadrados. Por otra parte tenemos la LAN, que es la más habitual de todas. Ya que se utiliza muchísimo en las empresas. Se le llama área de red local y esta abarca hasta un kilómetro cuadrado. Asimismo, existe la MAN o área de red metropolitana, que llega a abarcar hasta diez kilómetros cuadrados. Por último, existe la red WAN o red de área amplia, la cual se utiliza en todo un país, continente, etc.

Por último, podemos categorizar a una red, según las direcciones que tiene, para compartir la información o transmitirla. Esta aquella en la cual, una terminal transmite y la otra recibe. Por otra parte, esta aquella red que permite que una computadora transmita información y las otras reciban y por último, esta aquella red que permite que varias envíen y reciban información de manera simultánea.

El uso más común de una red lo podemos ver en las oficinas, en donde varios usuarios comparten recursos como una impresora desde sus computadoras. En la actualidad es frecuente incluso que las personas configuren redes internas en el hogar para compartir los recursos informáticos con la familia, como por ejemplo el acceso a Internet y el uso de impresoras y fax.

### **Características de una red**

- **Compartición de archivos:** Fue la razón principal para tener una red. Para que se cumpla se requiere de un directorio compartido que pueda ser accesado por muchos usuarios de la red, junto a toda la lógica asociada para que más de una persona no realice cambios conflictivos a un archivo al mismo tiempo.
- **Compartición de impresoras:** Con esto reducimos el número de impresoras en la organización. Se hace necesario el uso de colas de impresión para que las impresiones se lleven a cabo y de forma automática enviar los trabajos en espera en dicha cola.
- **Servicios de aplicación:** Así como se pueden compartir archivos o carpetas en una red, se pueden compartir aplicaciones, las más comunes son aplicativos de contabilidad. Si se requiere por ejemplo de instalar algún programa en diversas computadoras de la red, en lugar de ir colocando el CD-ROM en cada una, se puede tener una carpeta con el contenido del mismo y ejecutar el instalador desde cada equipo.

- **Correo electrónico:** Es un recurso bastante valioso y que incluso muchas organizaciones no lo aprovechan al máximo. No solamente es útil para las comunicaciones internas sino también para las externas.
- **Acceso remoto:** Se usa principalmente para acceder desde el exterior a los recursos de la red interna. Los usuarios la utilizan para ver sus archivos, correo electrónico ya sea que se encuentren de viaje, desde su hogar, etc.

**En esta ocasión describiré la red en la cual se desenvolverá el proyecto a implantar:**

### ❖ **Internet**

Internet es un conjunto descentralizado de redes de comunicación interconectadas que utilizan la familia de protocolos TCP/IP, garantizando que las redes físicas heterogéneas que la componen funcionen como una red lógica única, de alcance mundial. Sus orígenes se remontan a 1969, cuando se estableció la primera conexión de computadoras, conocida como ARPANET, entre tres universidades en California y una en Utah, Estados Unidos.

Uno de los servicios que más éxito ha tenido en Internet ha sido la World Wide Web (WWW, o "la Web"), hasta tal punto que es habitual la confusión entre ambos términos. La WWW es un conjunto de protocolos que permite, de forma sencilla, la consulta remota de archivos de hipertexto. Ésta

fue un desarrollo posterior (1990) y utiliza Internet como medio de transmisión.

## 2.2 DEFINICIÓN DE TÉRMINOS BÁSICOS

A continuación se define los siguientes conceptos:

### **SOFTWARE**

Se conoce como software *al equipamiento lógico o soporte lógico* de un sistema informático, comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos, que son llamados hardware.

Los componentes lógicos incluyen, entre muchos otros, las aplicaciones informáticas; tales como el procesador de texto, que permite al usuario realizar todas las tareas concernientes a la edición de textos; el software de sistema, tal como el sistema operativo, que, básicamente, permite al resto de los programas funcionar adecuadamente.

### **HARDWARE**

Las computadoras son aparatos electrónicos capaces de interpretar y ejecutar instrucciones programadas y almacenadas en su memoria; consisten básicamente en operaciones aritmético-lógicas y de entrada/salida.<sup>9</sup> Se reciben las entradas (datos), se las procesa y almacena (procesamiento), y finalmente se producen las salidas (resultados del procesamiento). Por ende todo sistema informático tiene, al menos, componentes y dispositivos hardware dedicados a alguna de las funciones

### **BASE DE DATOS**

Una base de datos o banco de datos (en ocasiones abreviada con la sigla *BD* o con la abreviatura *b. d.*) es un conjunto de datos

pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. Actualmente, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital (electrónico), que ofrece un amplio rango de soluciones al problema de almacenar datos.

Existen programas denominados sistemas gestores de bases de datos, abreviados SGBD, que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de estos SGBD, así como su utilización y administración, se estudian dentro del ámbito de la informática.

Las aplicaciones más usuales son para la gestión de empresas e instituciones públicas. También son ampliamente utilizadas en entornos científicos con el objeto de almacenar la información experimental.

### **ACTOR**

Es un tipo con un estereotipo predefinido, que denota una entidad externa al sistema que interactúa con casos de uso.

### **AGREGACIÓN**

Es una forma especial de asociación que especifica una relación total - parte entre el agregado (todo) y una parte que lo compone.

### **ANÁLISIS**

Es la parte del proceso de desarrollo de software cuyo propósito principal es realizar un modelo del dominio del problema. El análisis hace foco en qué hacer, el diseño hace foco en cómo hacerlo.

### **ARQUITECTURA**

Consiste en la estructura organizacional de un sistema. Una arquitectura puede ser descompuesta recursivamente en partes que

interactúan entre sí por medio de interfaces, relaciones que conectan las partes, y restricciones para ensamblar las partes.

### **ATRIBUTO**

Es una parte específica de una clase. Una propiedad de un tipo identificada mediante un nombre.

### **DISEÑO**

Es la parte del proceso de desarrollo de software cuyo propósito principal es decidir cómo se construirá el sistema. Durante el diseño se toman decisiones estratégicas y tácticas para alcanzar los requerimientos funcionales y la calidad esperada.

### **EFICIENCIA**

Es la cantidad de recursos de computadoras y de código requeridos por un programa para llevar a cabo sus funciones.

### **HERENCIA**

Es la propiedad que permite que una subclase herede los atributos y los mensajes de una superclase. Es el mecanismo por el cual elementos más específicos incorporan la estructura y el comportamiento de elementos más generales.

### **CALIDAD DE SOFTWARE**

Es la concordancia con los requerimientos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se esperan de todo software desarrollado profesionalmente.

### **CASOS DE USO**

Es aquello que describe la interacción de los Actores con el sistema para lograr un objetivo.

### **COLABORACIÓN**

Es aquello que describe un conjunto de interacciones entre un conjunto de objetos con el propósito de lograr un objetivo.

### **COMPOSICIÓN**

Es una forma de agregación más fuerte, con una mayor posesión y coincidencia de ciclo de vida entre el todo y las partes. Las partes que no tienen multiplicidad fija pueden ser creadas después del compuesto en sí, pero una vez creadas viven y mueren con él. Estas partes también pueden ser removidas antes de la muerte del compuesto. La composición puede ser recursiva.

### **ADMINISTRACIÓN DE PROYECTO**

Consiste en estructurar el planeamiento, la implementación y el seguimiento de todos los elementos de una solución de sistemas y la iniciación y monitoreo de acciones correctivas.

### **CONFIABILIDAD**

Es la probabilidad de operación libre de fallas de un programa de computadora en un entorno determinado y durante un tiempo específico.

### **CONSISTENCIA**

Es aquello que consiste en el uso de un diseño uniforme de técnicas de documentación a lo largo del proyecto de desarrollo de software.

### **CONTEXTO DEL SISTEMA O AMBIENTE**

Es la parte del mundo que vamos a ignorar, excepto por algunas interacciones importantes entre el sistema y su ambiente cuando se desarrolla un proyecto.

### **DEPENDENCIA**

Es aquello que indica la relación semántica entre elementos del modelo.

### **DIAGRAMA DE ACTIVIDADES**

Es un caso especial de diagrama de estados en el que todos, o la mayoría, son estados activos y en el que todas, o la mayoría, de las transiciones son disparadas por la finalización de las acciones de los estados.

### **DIAGRAMA DE CASOS DE USO**

Es el diagrama que muestra la relación entre los actores y los casos de uso dentro de un sistema.

### **DIAGRAMA DE CLASES**

Es el diagrama que muestra una colección de elementos del modelo tales como las clases, tipos y sus contenidos y relaciones.

### **DIAGRAMA DE COLABORACIONES**

Es un diagrama que muestra interacciones entre objetos organizadas alrededor de los objetos y sus vinculaciones. A diferencia de un diagrama de secuencias, un diagrama de colaboraciones muestra las relaciones entre los objetos. Los diagramas de secuencias y los diagramas de colaboraciones expresan información similar, pero en una forma diferente.

### **DIAGRAMA DE COMPONENTES**

Es un diagrama que muestra la organización de los componentes y sus dependencias.

### **DIAGRAMA DE ENTIDAD / RELACIÓN**

Es una descripción conceptual de las estructuras de datos y sus relaciones.

### **DIAGRAMA DE ESTADO**

Es el diagrama que muestra el estado de la máquina.

### **DIAGRAMA DE FLUJO DE DATOS**

Es una descripción informal del sistema de información.

### **DIAGRAMA DE INTERACCIONES**

Es un término genérico que se aplica a diversos tipos de diagramas que enfatizan la interacción entre objetos. Incluye: diagrama de colaboraciones, diagrama de secuencias, diagrama de actividades.

### **DIAGRAMA DE OBJETOS**

Es un diagrama que contiene los objetos y sus relaciones en un momento dado del tiempo. Un diagrama de objetos puede ser considerado un caso especial de un diagrama de clases o de un diagrama de colaboraciones.

### **DIAGRAMA DE SECUENCIA**

Es el diagrama que muestra los objetos que participan en la interacción y la secuencia de mensajes que intercambian.

### **SEGUIMIENTO**

Observación minuciosa de la evolución y el desarrollo de un proceso.

### **ESPECIFICACIÓN**

Es un informe de acuerdo entre el implementador y el usuario.

### **ESPECIFICACIÓN DE DISEÑO**

Es aquella que establece un acuerdo entre el diseñador y el implementador.

### **ESPECIFICACIÓN DE REQUERIMIENTOS**

Es aquella que establece un acuerdo entre el usuario y el desarrollador del sistema.

### **ESQUEMA (FRAMEWORK)**

Es un micro-arquitectura que provee un molde extensible para aplicaciones de un dominio específico.

### **ESTADO**

Es una condición o situación en la vida de un objeto, durante la cual satisface una condición, realiza una actividad o está esperando un evento.

### **FLEXIBILIDAD**

Es el esfuerzo requerido para modificar un programa operativo.

### **FORMACIÓN**

Es el grado en que el software ayuda para permitir que nuevos usuarios apliquen el sistema.

### **HERENCIA**

Es la propiedad que permite que una subclase herede los atributos y los mensajes de una superclase. Es el mecanismo por el cual elementos más específicos incorporan la estructura y el comportamiento de elementos más generales.

### **IMPLEMENTACIÓN**

Es la definición de cómo está construido o compuesto algo. Por ejemplo: una clase es una implementación de un tipo, un método es una implementación de una operación.

### **INGENIERÍA DE SOFTWARE**

Es una disciplina para el desarrollo de software de alta calidad para sistemas basados en computadora.

### **INSTANCIA DE CASO DE USO**

Es una secuencia de acciones realizadas por el sistema, que producen un resultado observable, valioso para un actor en particular. Usualmente los escenarios ilustran instancias prototípicas de casos de uso.

### **INTERACCIÓN**

Es una especificación de comportamiento cuyo fin es lograr un propósito específico. Abarca un conjunto de intercambios de mensajes entre un conjunto de objetos dentro de un contexto particular. Una interacción puede ilustrarse mediante uno o más escenarios.

### **MÉTODO**

Es un procedimiento o función asociada a un tipo de objeto declarado dentro de un objeto. Es la implementación del mensaje. La implementación de una operación. El algoritmo o procedimiento que permite llegar al resultado de una operación.

### **MODELO DE CASOS DE USO**

Es un modelo que describe los requerimientos funcionales de un sistema en términos de casos de uso.

### **NODO**

Es un objeto físico existente en tiempo de ejecución, que representa un recurso computacional, que generalmente tiene al menos memoria y habitualmente también capacidad computacional. Los objetos de tiempo de ejecución y componentes pueden residir en nodos.

### **OBJETO**

Es un componente del mundo real que tiene una cierta estructura interna y un determinado comportamiento.

### **PARÁMETRO**

Es la especificación de una variable que puede ser pasada, cambiada y/o devuelta. Un parámetro puede incluir nombre, tipo y dirección. Los parámetros son utilizados para operaciones, mensajes y eventos. Sinónimo: parámetro formal.

### **PROCESO**

Es un hilo de ejecución que puede ejecutar concurrentemente con otros hilos.

### **PROCESO DE DESARROLLO**

Es un conjunto parcialmente ordenado de pasos, realizados durante el desarrollo de software con el fin de lograr un objetivo dado; por ejemplo, construir modelos o implementar modelos.

### **RELACIÓN**

Es una conexión semántica entre elementos del modelo. La asociación y la generalización son ejemplos de relaciones.

### **REQUERIMIENTO**

Es una característica, propiedad o comportamiento deseado para un sistema.

### **RESTRICCIÓN**

Es una condición u obligación semántica. Algunas restricciones están predefinidas en el UML, otras pueden ser definidas por el usuario. Las restricciones son uno de los tres mecanismos de extensión del UML.

### **SEGURIDAD**

Es la disponibilidad de mecanismos que controlen o protejan los programas o datos.

### **SIMPLICIDAD**

Es el grado en que un programa puede ser entendido sin dificultades.

### **TIEMPO DE EJECUCIÓN**

Es el período de tiempo durante el cual un programa ejecuta.

## **TRANSICIÓN**

Es una relación entre dos estados que indica que un objeto que está en el primer estado realizará una acción especificada y entrará en el segundo estado cuando un evento especificado ocurra y unas condiciones especificadas sean satisfechas. En dicho cambio de estado se dice que la transición es 'disparada'.

## **CONTROL**

La palabra control tiene muchas connotaciones y su significado depende de la función o del área en que se aplique; puede ser entendida:

- Como la función administrativa que hace parte del proceso administrativo junto con la planeación, organización y dirección, y lo que la precede.
- Como los medios de regulación utilizados por un individuo o empresa, como determinadas tareas reguladoras que un controlador aplica en una empresa para acompañar y avalar su desempeño y orientar las decisiones. También hay casos en que la palabra control sirve para diseñar un sistema automático que mantenga un grado constante de flujo o de funcionamiento del sistema total
- Como la función restrictiva de un sistema para mantener a los participantes dentro de los patrones deseados y evitar cualquier desvío También hay otras connotaciones para la palabra control:
  - Comprobar o verificar;
  - Regular;
  - Comparar con un patrón;
  - Ejercer autoridad sobre alguien (dirigir o mandar);
  - Frenar o impedir.

### **TIEMPO REAL**

Es aquel sistema digital que interactúa activamente con un entorno con dinámica conocida en relación con sus entradas, salidas y restricciones temporales, para darle un correcto funcionamiento de acuerdo con los conceptos de predictibilidad, estabilidad, controlabilidad y alcanzabilidad.

Los sistemas en tiempo real están presentes en nuestra vida diaria, prácticamente en todo lo que nos rodea; en los aviones, trenes y automóviles; en el televisor, la lavadora o el horno de microondas, en los teléfonos celulares y en las centrales telefónicas digitales. Son un elemento imprescindible para garantizar la generación, transmisión y distribución de la energía eléctrica para asegurar la calidad y la seguridad de incontables procesos industriales.

La principal característica que distingue a los STR de otros tipos de sistemas es el tiempo de interacción. Sin embargo, antes de continuar es necesario aclarar el significado de las palabras "tiempo" y "real". La palabra "tiempo" significa que el correcto funcionamiento de un sistema depende no sólo del resultado lógico que devuelve la computadora, también depende del tiempo en que se produce ese resultado. La palabra "real" quiere decir que la reacción de un sistema a eventos externos debe ocurrir durante su evolución. Como una consecuencia, el tiempo del sistema (tiempo interno) debe ser medido usando la misma escala con que se mide el tiempo del ambiente controlado (tiempo externo).

### **GESTION DE LA INFORMACION**

Es un conjunto de procesos por los cuales se controla el ciclo de vida de la información, desde su obtención - por creación o captura, hasta su disposición final - archivada o eliminada. Los procesos también comprenden la extracción, combinación, depuración y distribución de la información a los interesados. Los objetivos de la Gestión de la Información es garantizar la integridad, disponibilidad y confidencialidad de la información.

### **ADMINISTRATIVO**

La Administración, es la ciencia social y técnica encargada de la *planificación, organización, dirección y control* de los recursos (humanos, financieros, materiales, tecnológicos, el conocimiento, etc.) de una organización, con el fin de obtener el máximo beneficio posible; este beneficio puede ser económico o social, dependiendo de los fines perseguidos por la organización.

### **MESA DE PARTES**

Es una unidad organizacional, que es responsable de realizar algunas acciones para cumplir con un procedimiento administrativo determinado. Es decir, se encargará de recepcionar los trámites, registrarlos, darles mantenimiento, derivarlos a las dependencias que corresponden y darles información oportuna a los remitentes cuando hagan consultas.

### **TRAMITE**

Es el objeto que un remitente presenta físicamente (impreso) virtualmente (digitalizado) a una mesa de partes. Este objeto puede tener atributos como el nombre del remitente, el nombre del destinatario (dependencia), la dirección del remitente, la fecha en la que se entrega el trámite, el motivo o contenido del trámite, etc.

# **CAPITULO III: RESULTADOS Y DISCUSIÓN**

## **3.1. ANALISIS E INTERPRETACION DE LOS DATOS**

### **3.1.1. Descripción y Sustentación de la Solución**

El modelo de solución recibe como entrada la variable independiente y luego de ser procesada se obtiene como resultado la variable dependiente.

Cabe señalar que, a un grupo de N trámites se le aplicará el sistema y a otro grupo de N trámites no se le aplicará el sistema, lo cual influirá en el tiempo de respuesta a los remitentes, que se obtendrá a través del registro en una encuesta.

Una vez obtenidos los resultados, se procederá al análisis de los datos y la prueba de hipótesis.

### **3.1.2. Análisis**

El presente capítulo describe la metodología utilizada para el desarrollo de la solución, la identificación de los requerimientos, el análisis de la solución, la definición del sistema, la descripción de los casos de uso y el diagrama de clases.

#### **METODOLOGIA**

El método de estudio que se va usar para este proyecto es un Método Experimental, que consiste en comprobar y demostrar las actividades establecidas en el proyecto que se va a realizar basada en las propias experiencias que nos permite formular posibles soluciones al problema presentado en un área respectiva de la institución.

Así mismo es de suma importancia elegir la metodología adecuada de trabajo, es por ello que la metodología RUP basada en la herramientas UML nos proporciona todas las bases para llevar al éxito la elaboración de nuestro software.

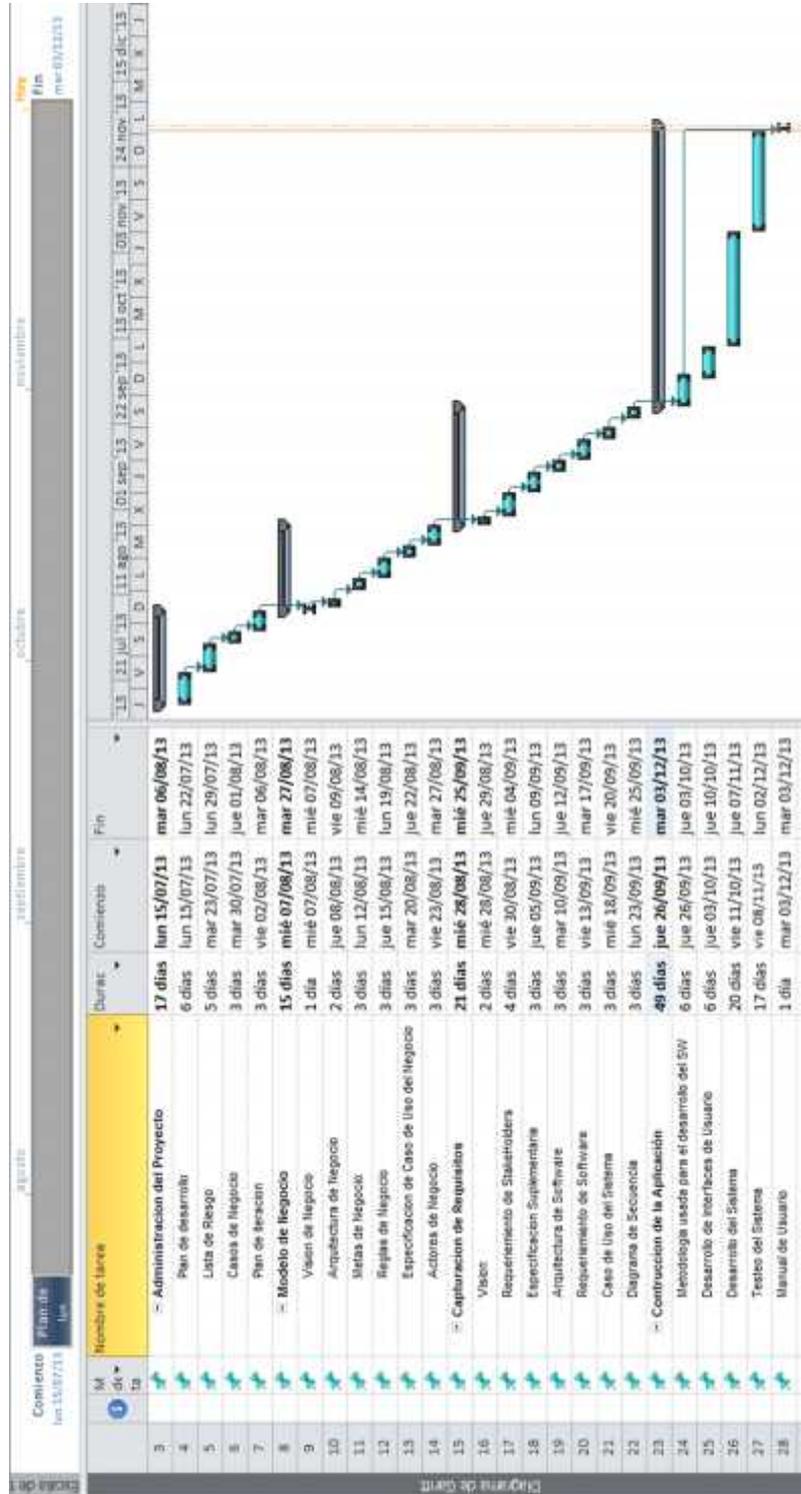
#### **RUP - RATIONAL UNIFIED PROCESS O PROCESO UNIFICADO DE RACIONAL**

Es un proceso de ingeniería de software que suministra un enfoque para asignar tareas y responsabilidades dentro de una

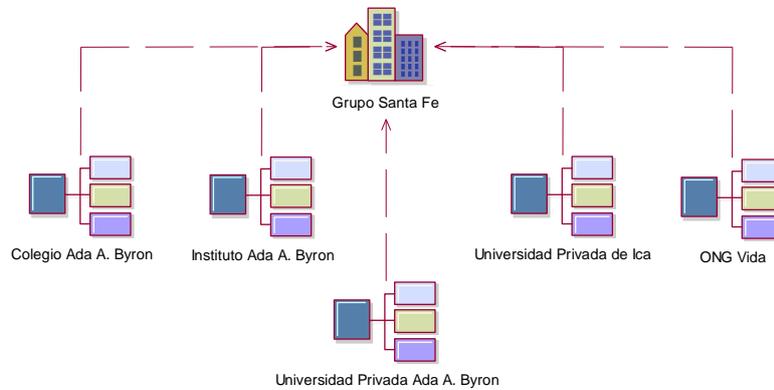
organización de desarrollo. Su objetivo es asegurar la producción de software de alta calidad que satisfaga la necesidad del usuario final dentro de un tiempo y presupuesto previsible.

El RUP mejora la productividad del equipo ya que permite que cada miembro del grupo sin importar su responsabilidad específica acceda a la misma base de datos de conocimiento. Esto hace que todos compartan el mismo lenguaje, la misma visión y el mismo proceso acerca de cómo desarrollar software.

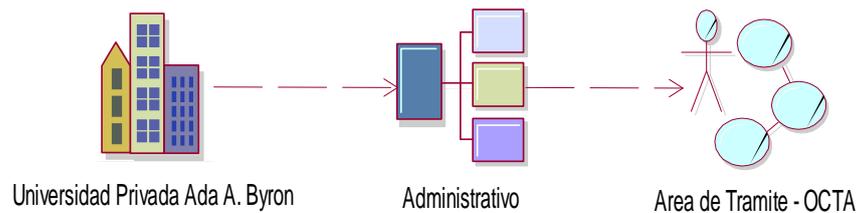
### 3.1.3. Cronograma de Actividades del Proyecto



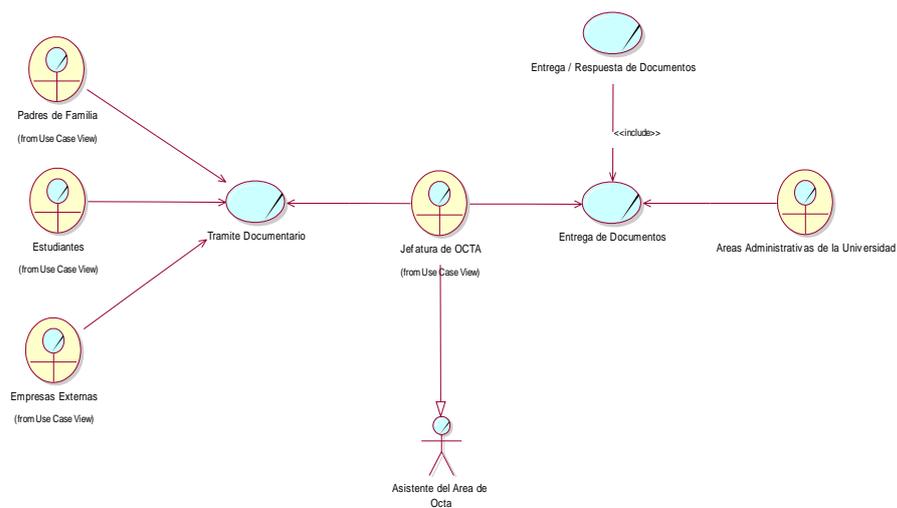
### 3.1.4. Diagrama General de la Empresa



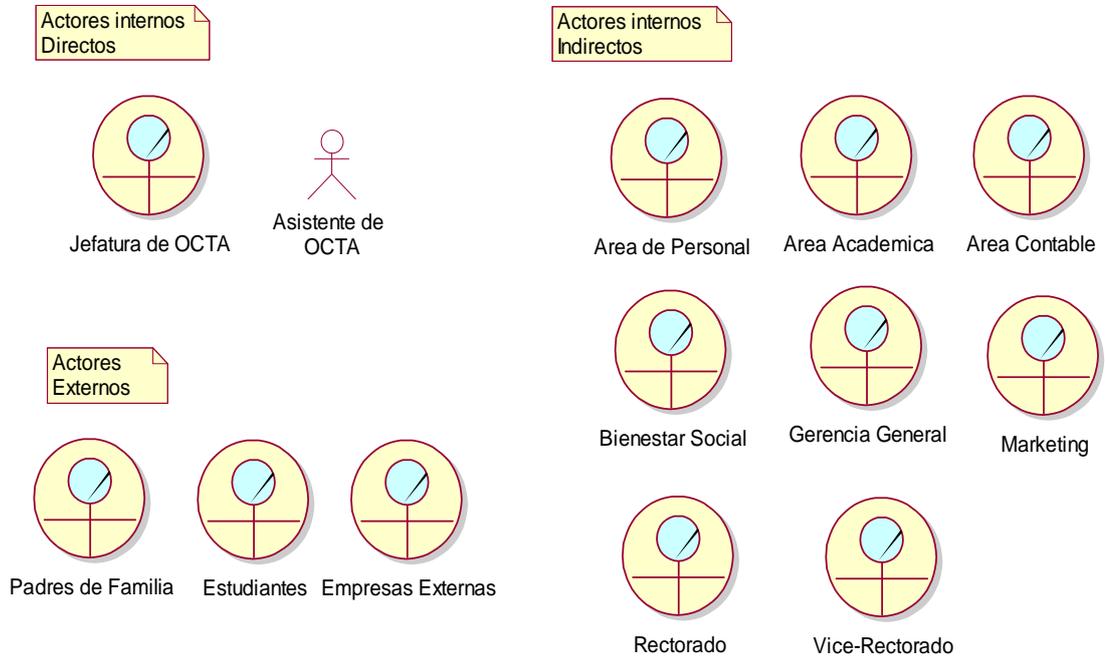
### 3.1.5. Diagrama del Área designada



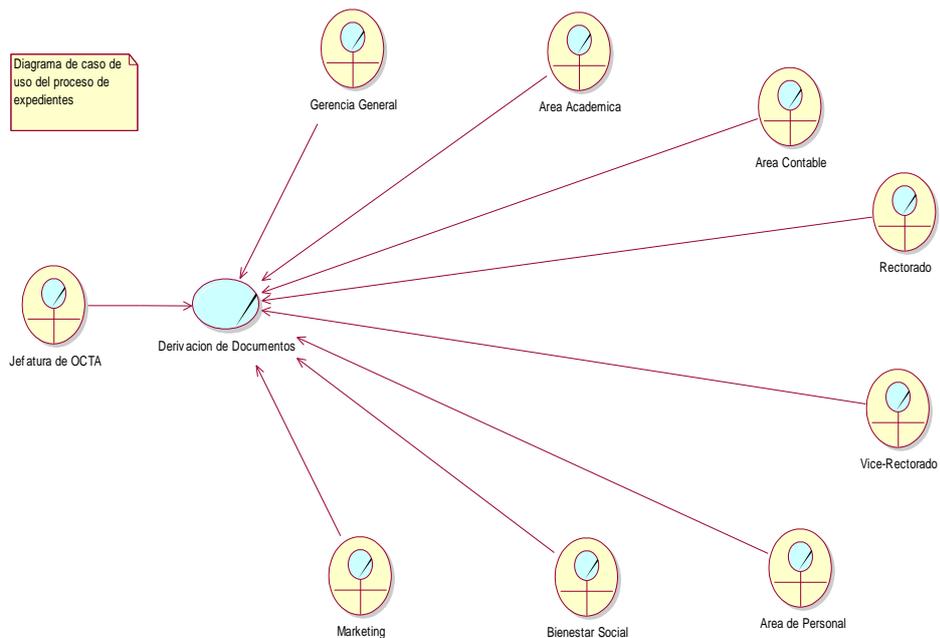
### 3.1.6. Diagrama del Proceso Actual



### 3.1.7. StakeHolder Internos y Externos

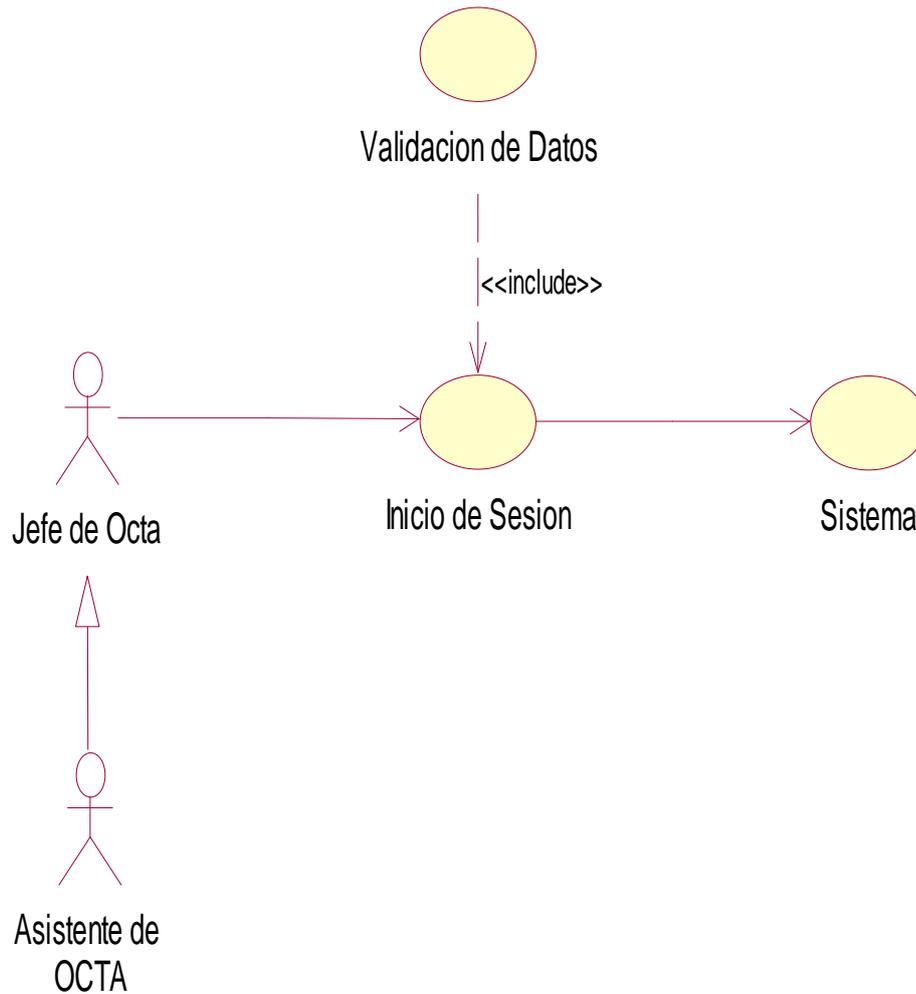


### 3.1.8. Diagrama de Proceso de Expediente Interno

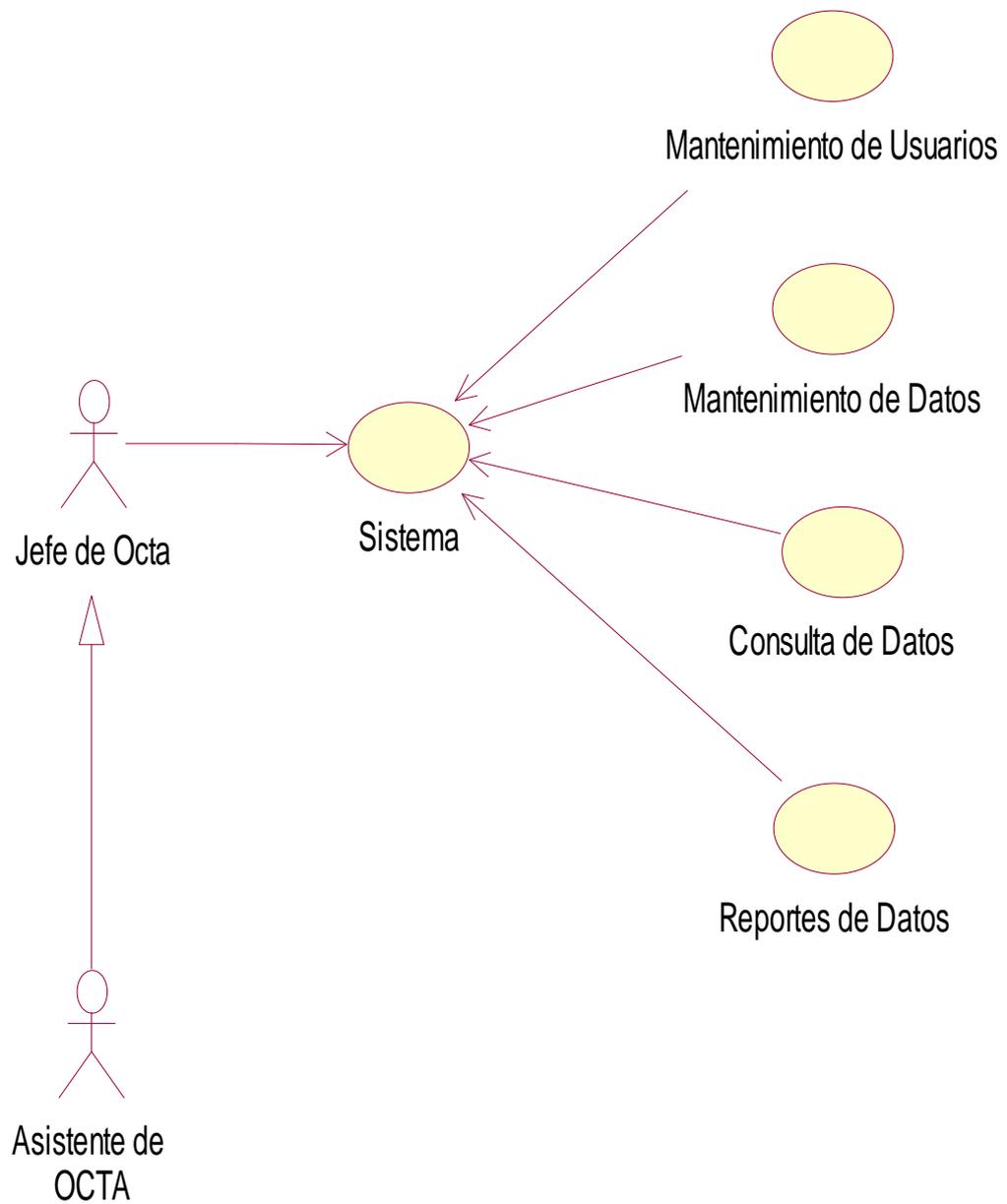


### 3.2. Diagrama de Caso de Uso del Sistema.

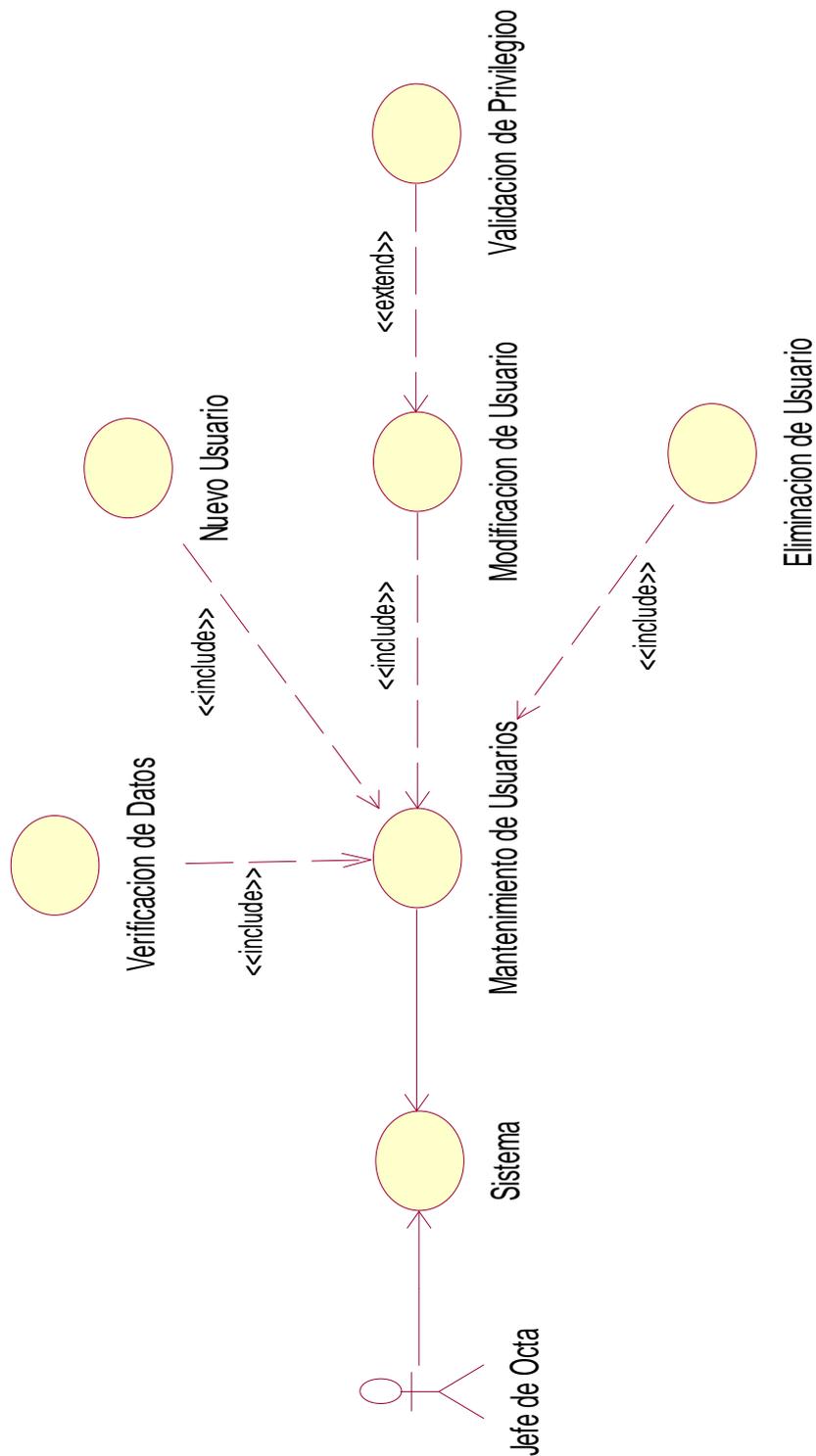
#### 3.2.1. Diagrama de Inicio de Sesión



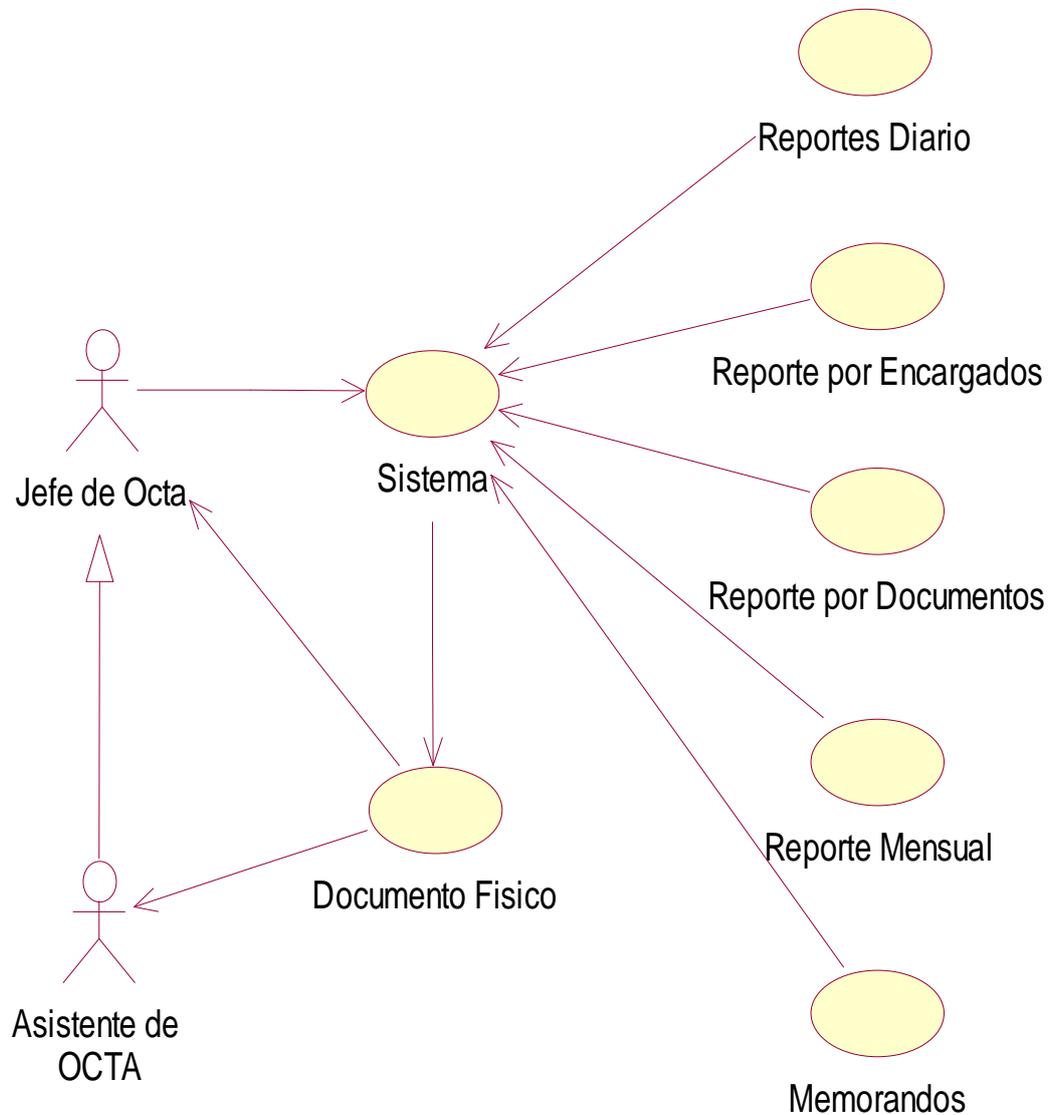
### 3.2.2. Diagrama del Sistema



### 3.2.3. Diagrama de Mantenimiento de Usuario

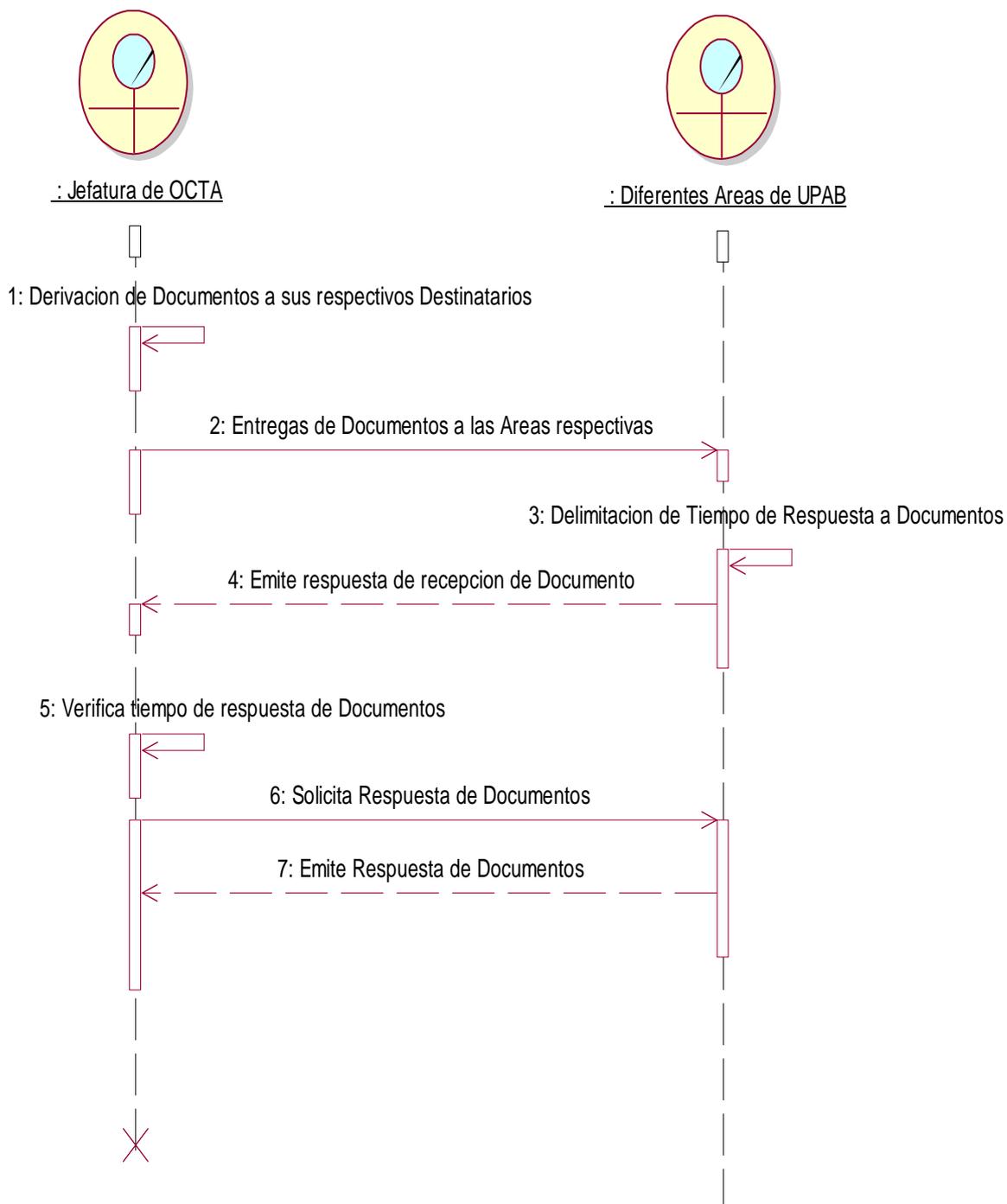


### 3.2.4. Diagrama de Reportes

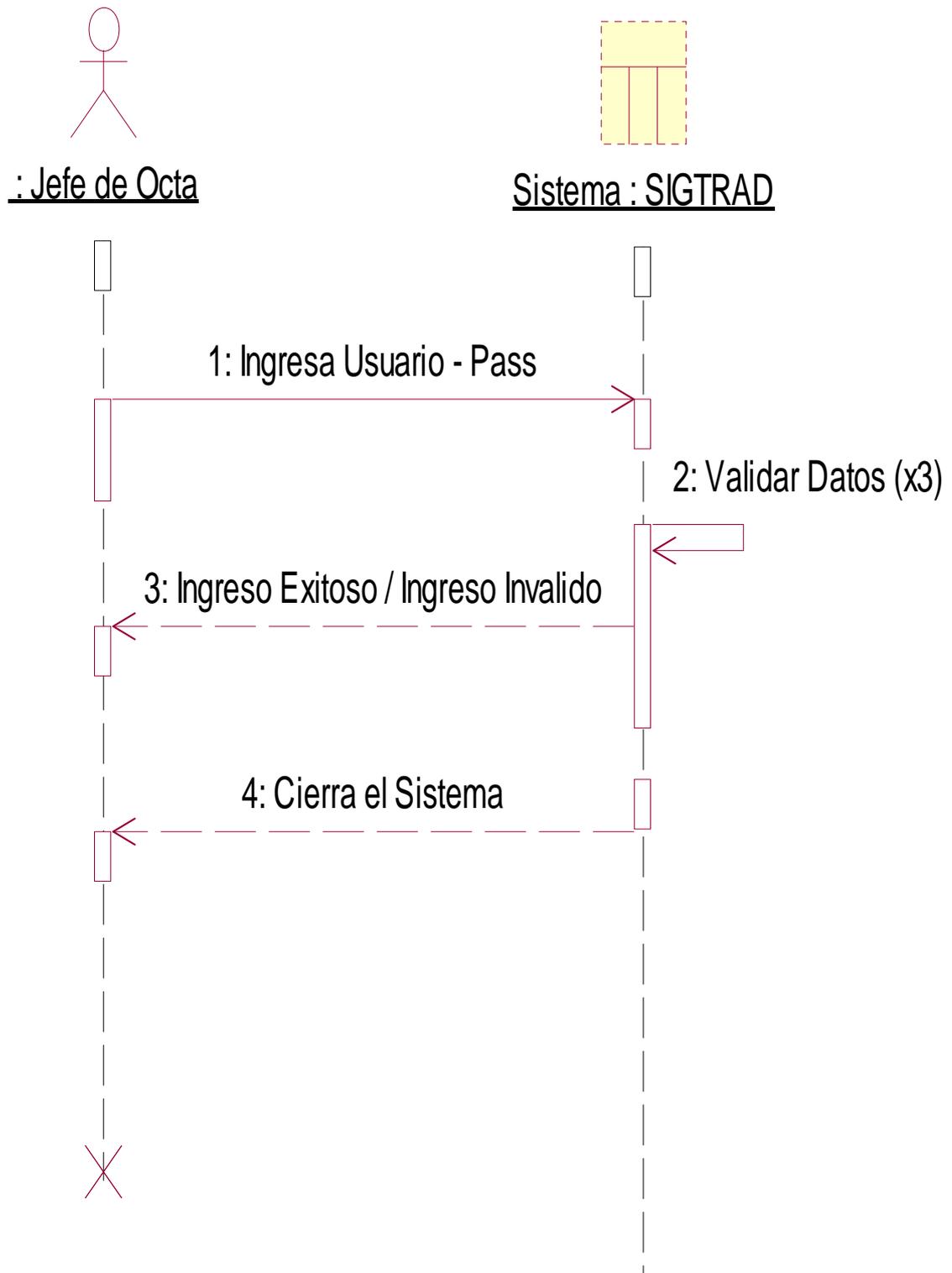


### 3.3. Diagrama de Secuencia del Sistema

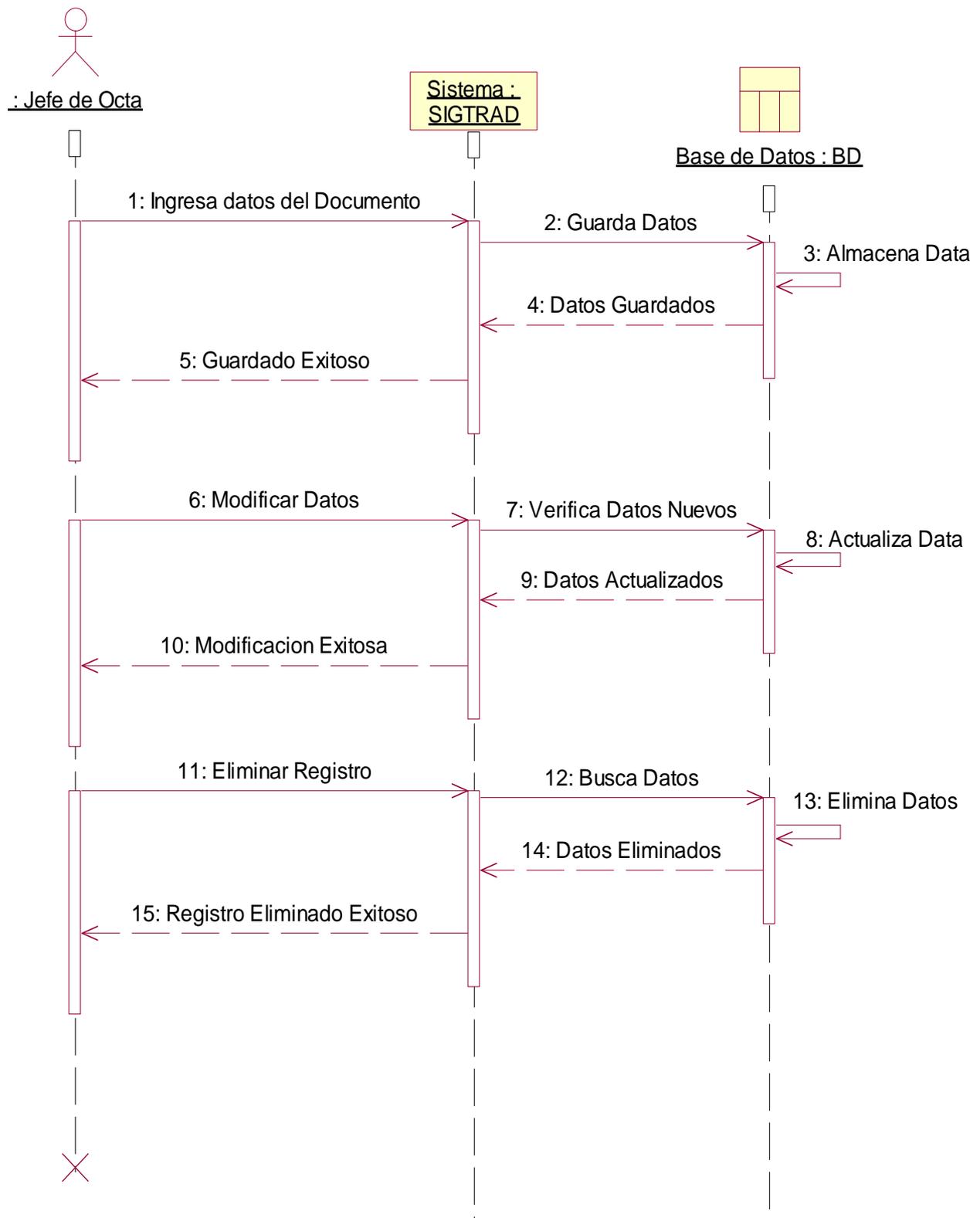
#### 3.3.1. Diagrama de Proceso de Expediente Interno



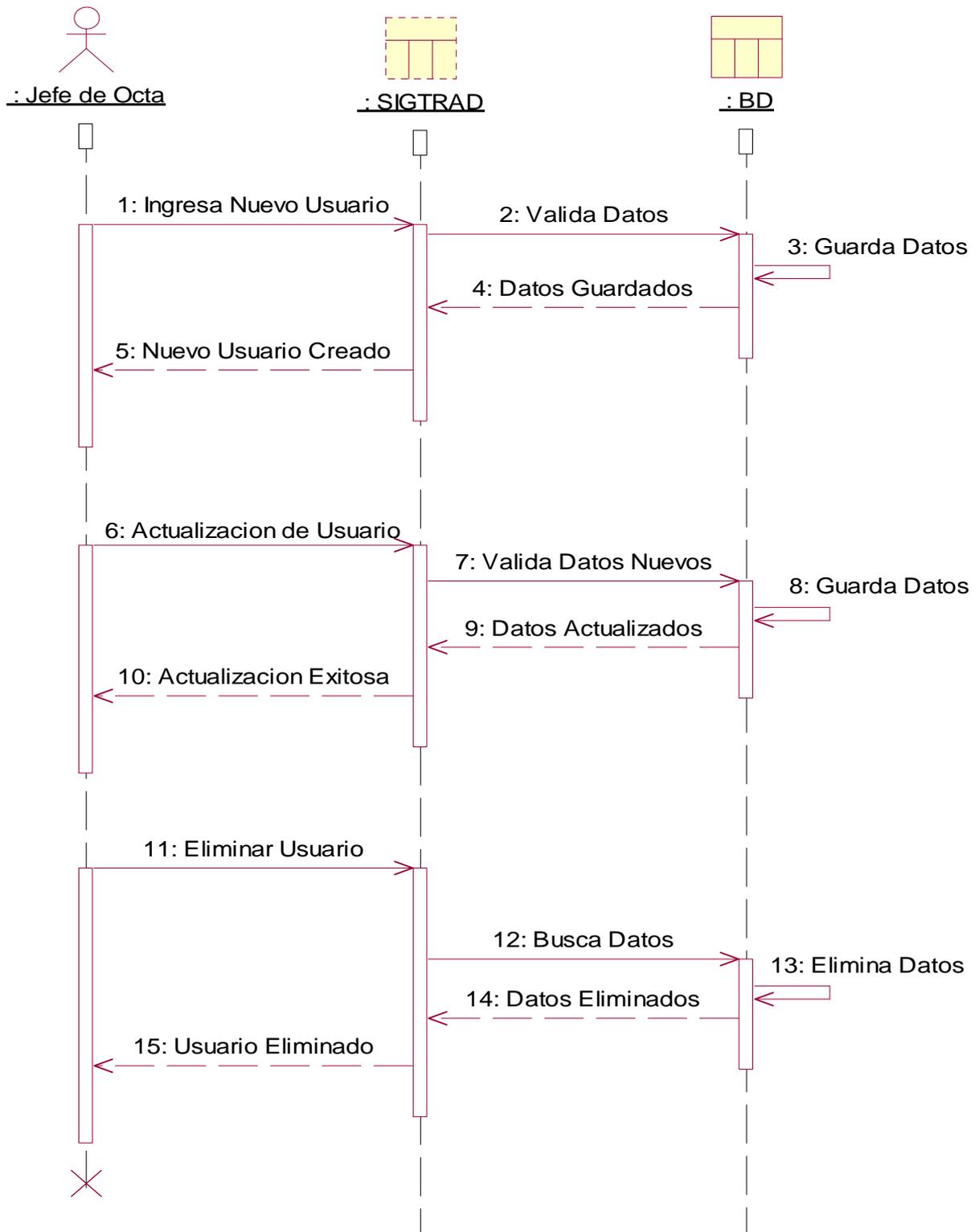
### 3.3.2. Diagrama de Inicio de Sesión



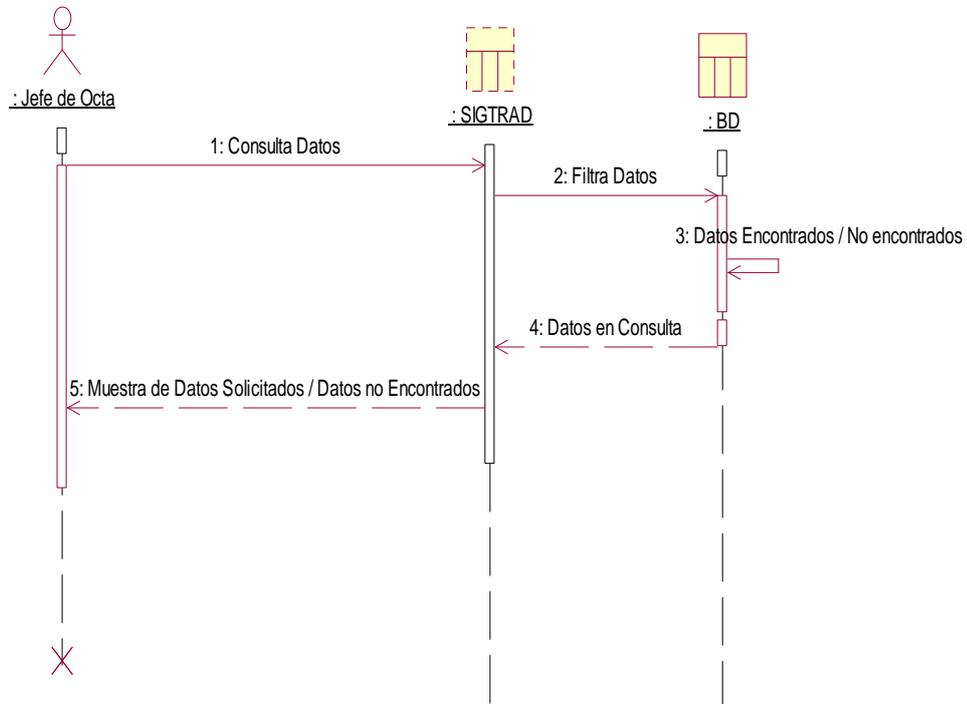
### 3.3.3. Diagrama de Mantenimiento de Datos



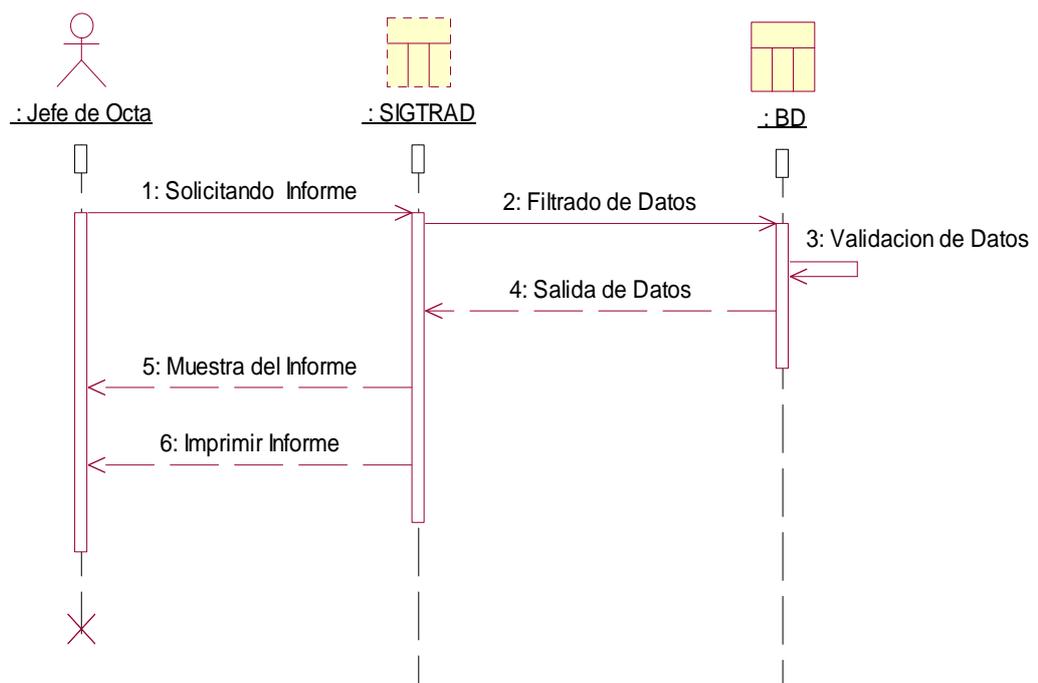
### 3.3.4. Diagrama de Mantenimiento Usuario



### 3.3.5. Diagrama de Consulta de Datos

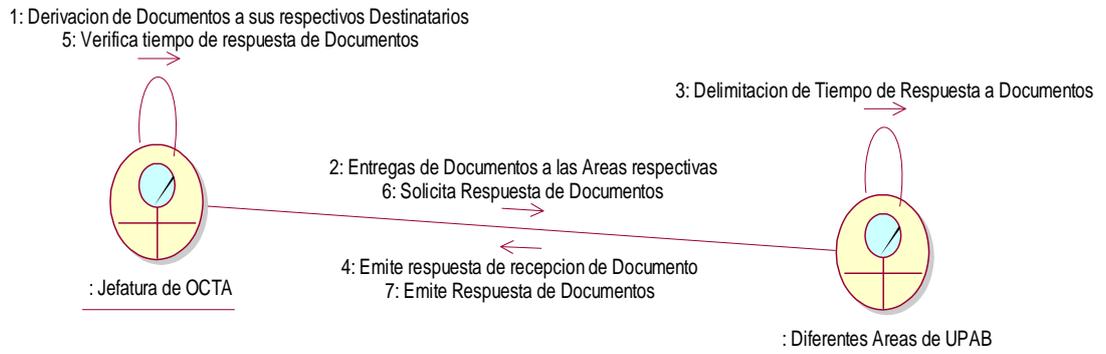


### 3.3.6. Diagrama de Reportes

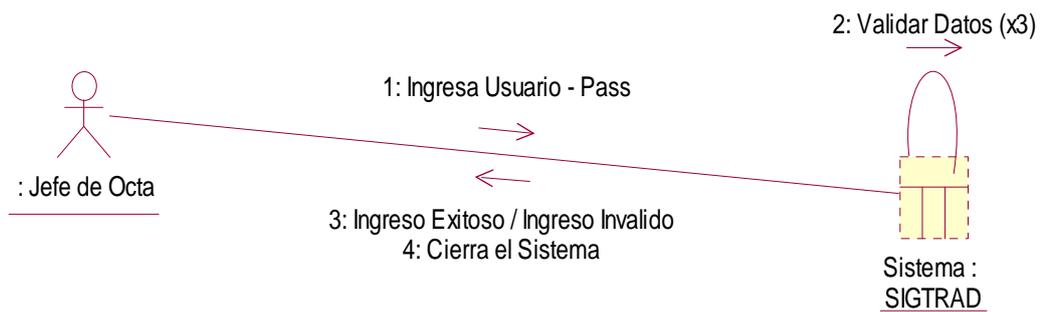


### 3.4. Diagramas de Colaboración del Sistema

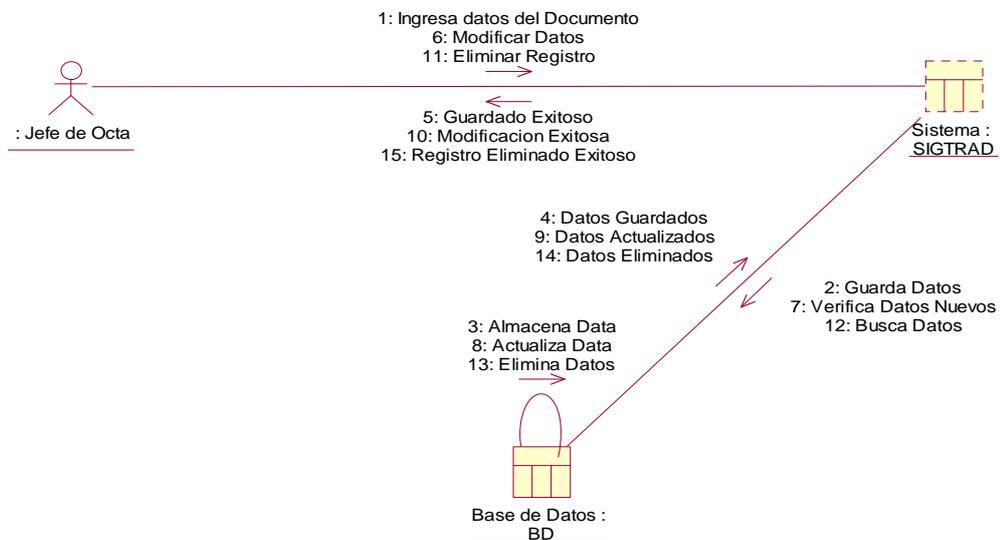
#### 3.4.1. Diagrama de Proceso de Expediente Interno



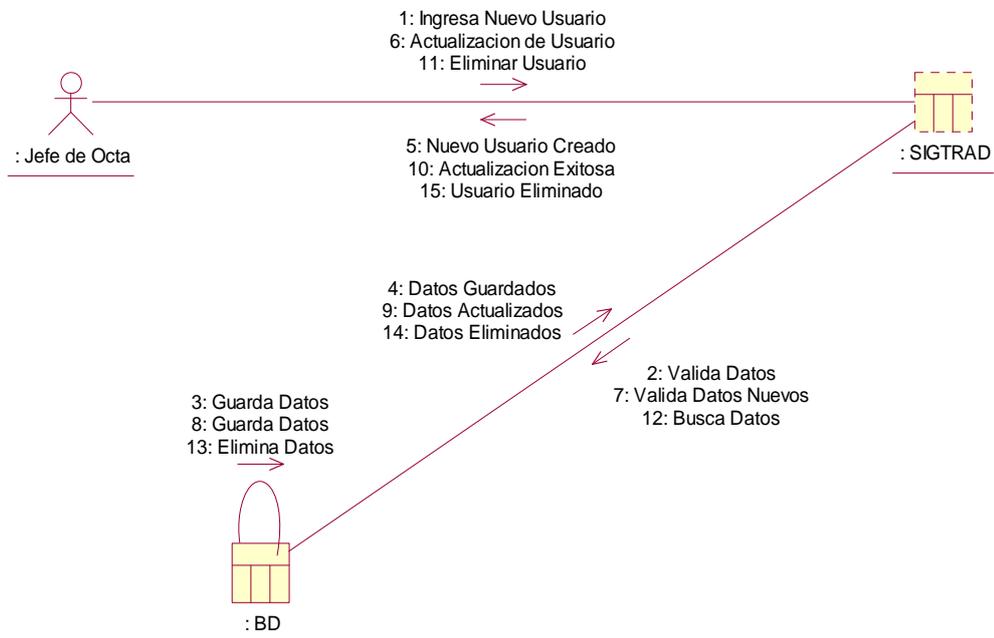
#### 3.4.2. Diagrama de Inicio de Sesión



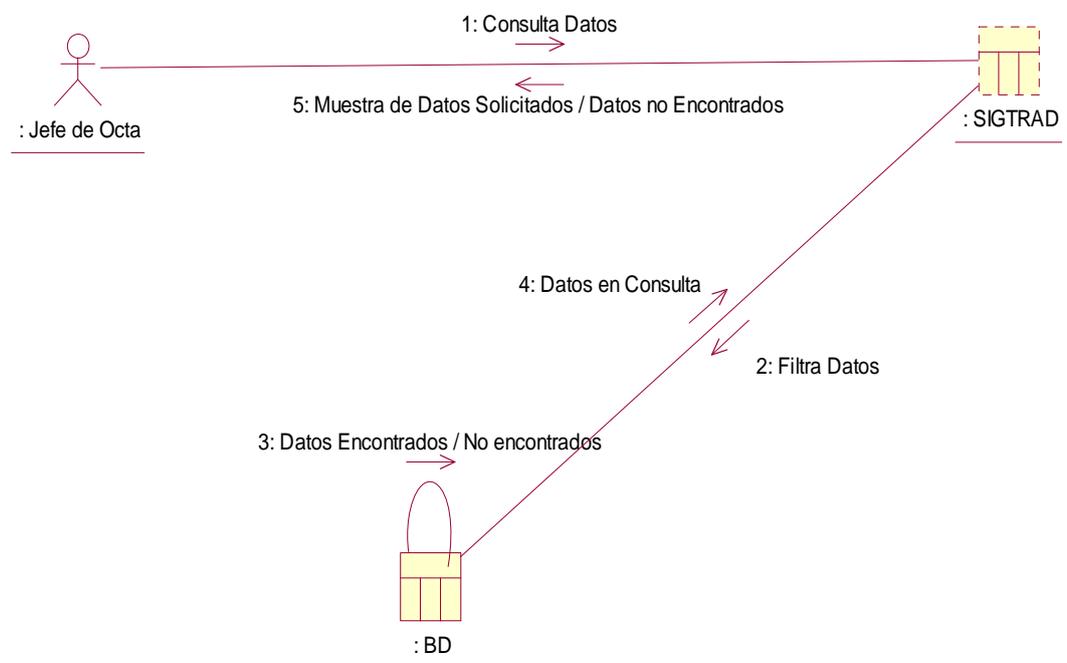
#### 3.4.3. Diagrama de Mantenimiento de Datos



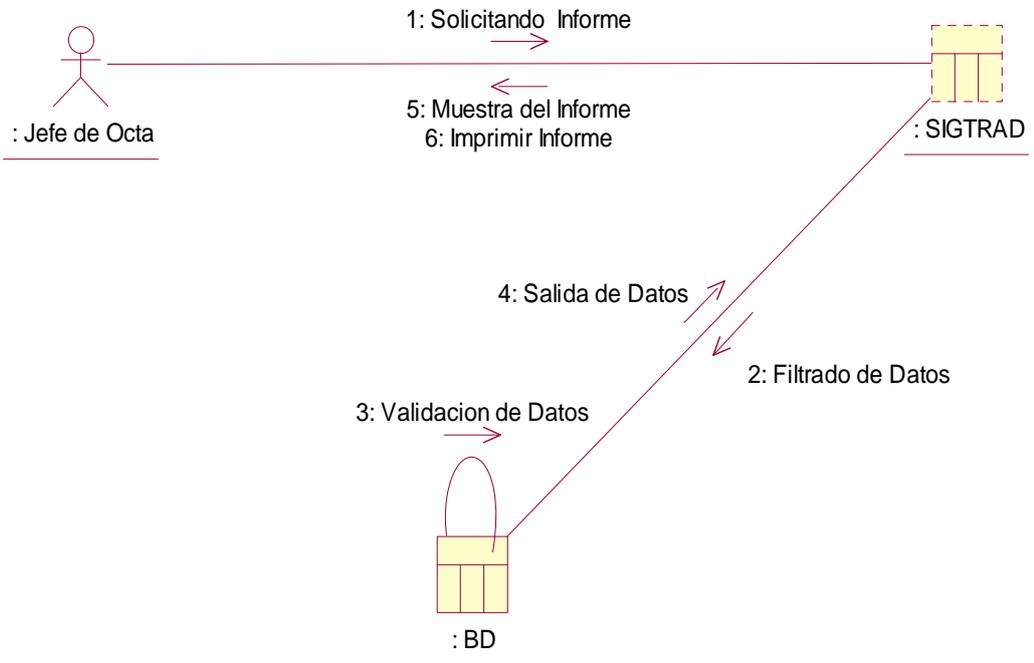
### 3.4.4. Diagrama de Mantenimiento Usuario



### 3.4.5. Diagrama de Consulta de Datos

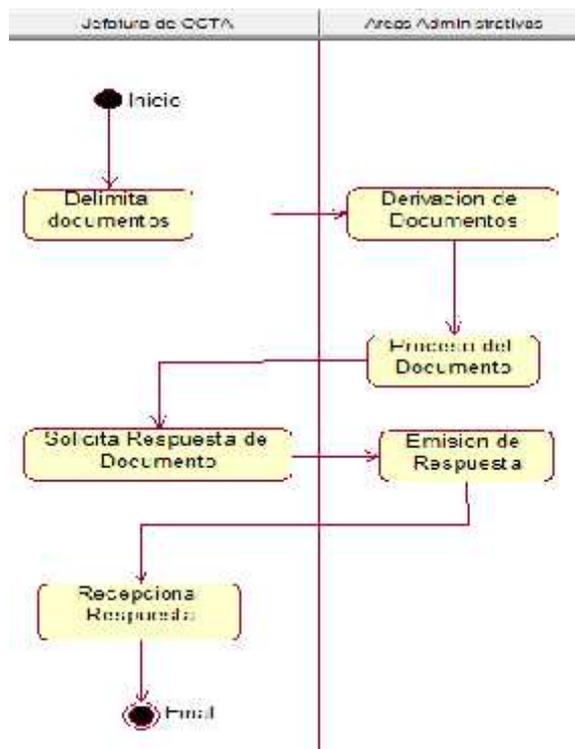


### 3.4.6. Diagrama de Reportes

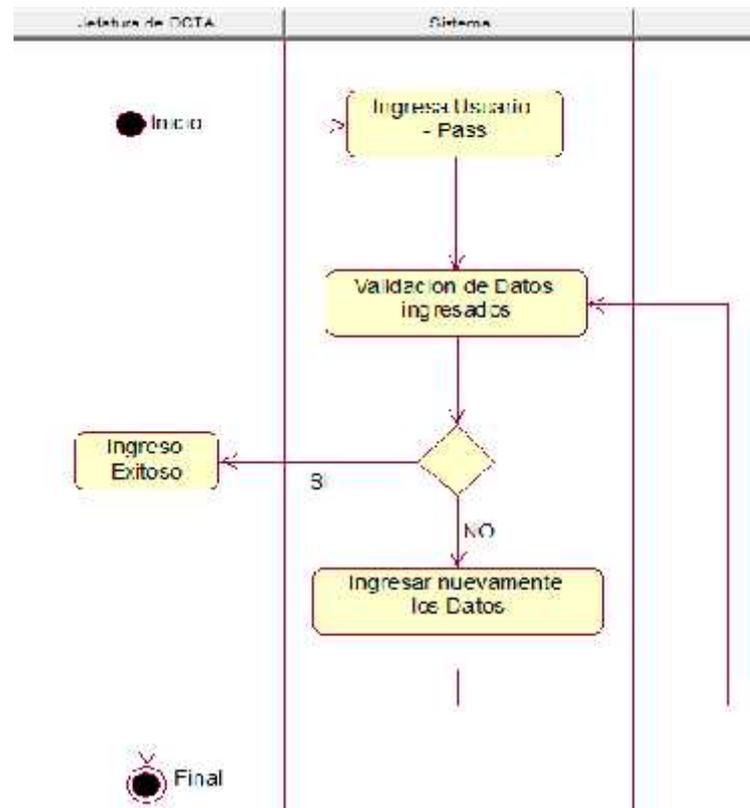


### 3.5. Diagrama de Actividades del Sistema

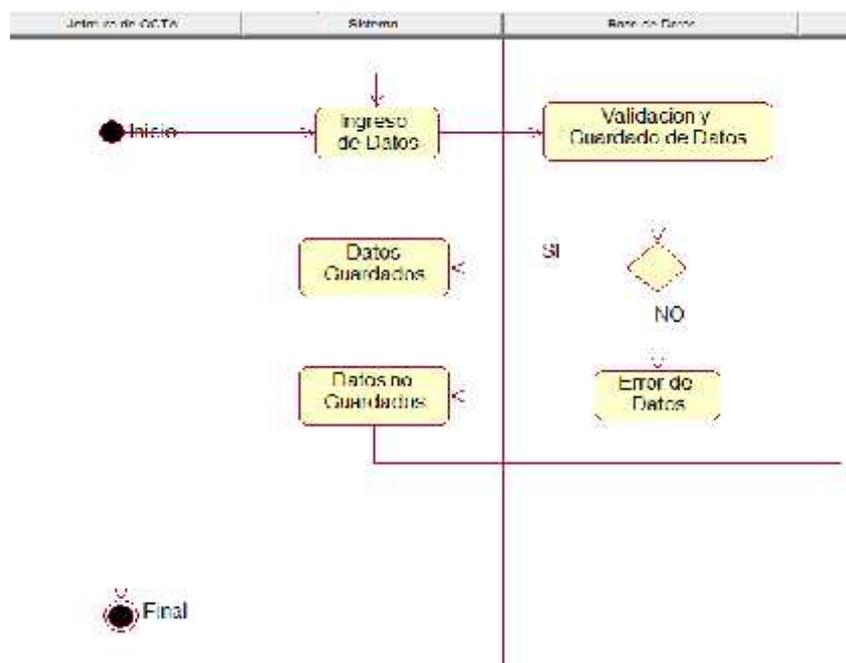
#### 3.5.1. Diagrama de Proceso de Expediente Interno



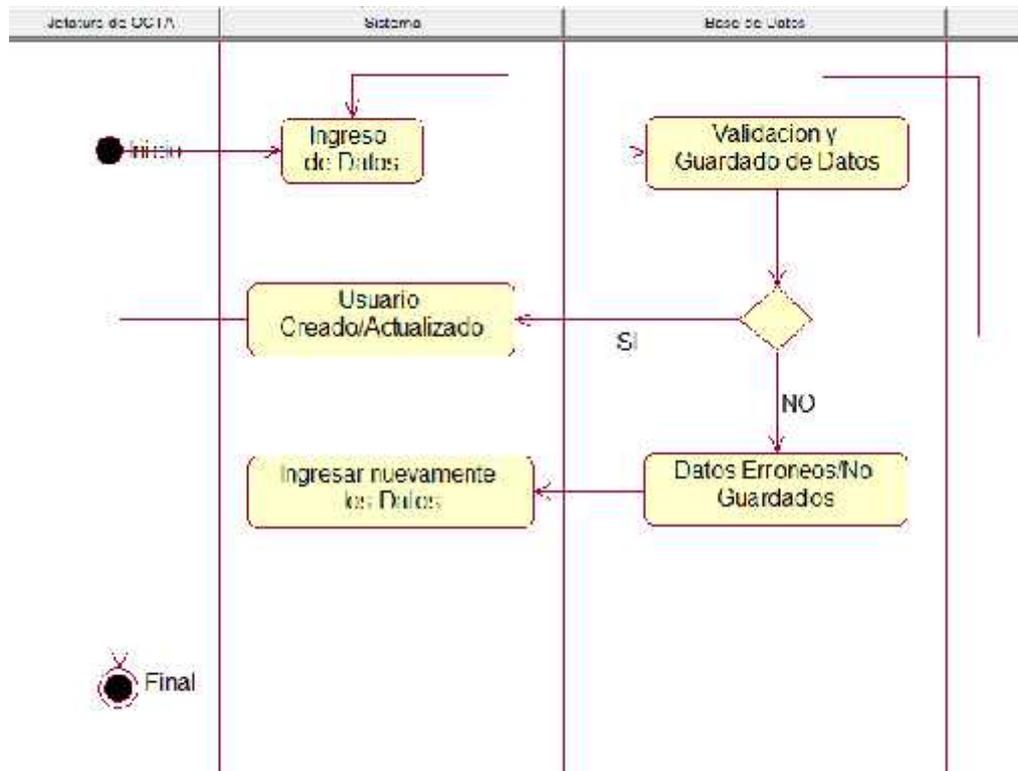
### 3.5.2. Diagrama de Inicio de Sesión



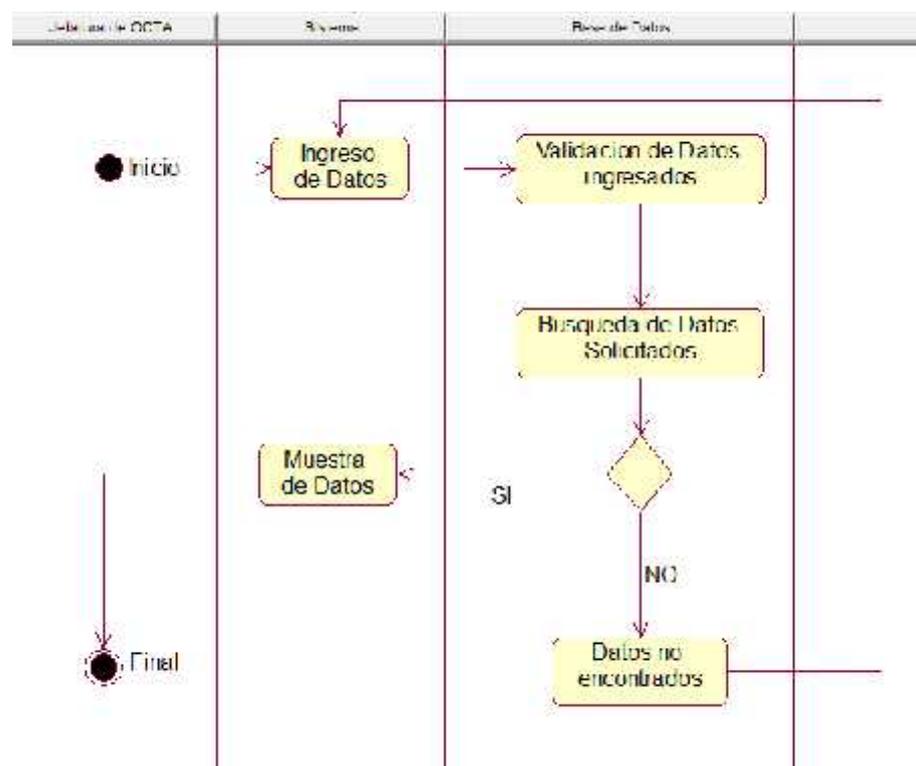
### 3.5.3. Diagrama de Mantenimiento Datos



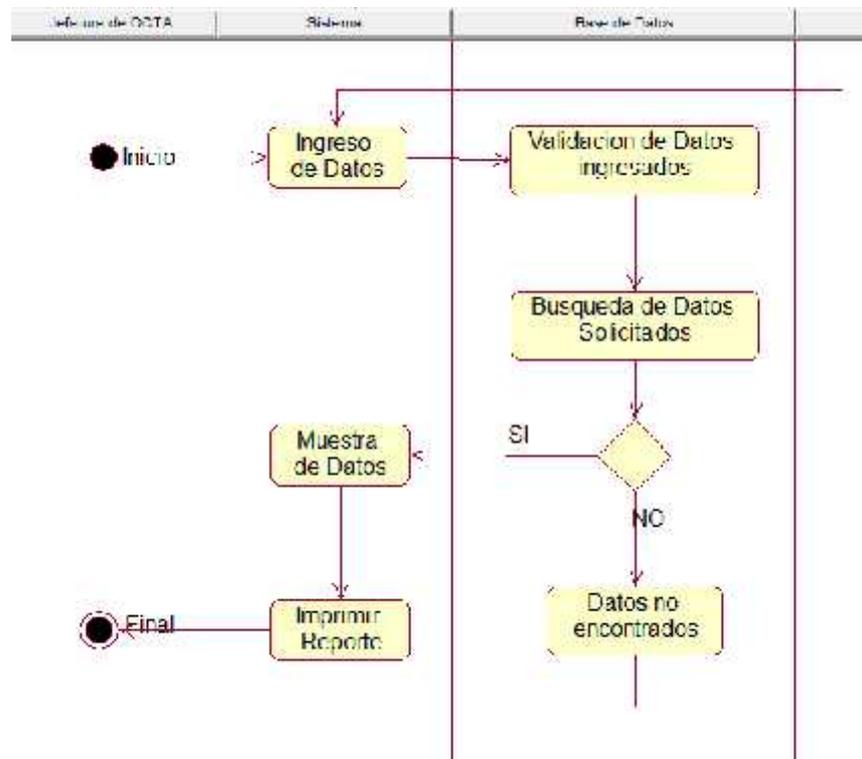
### 3.5.4. Diagrama de Mantenimiento Usuario



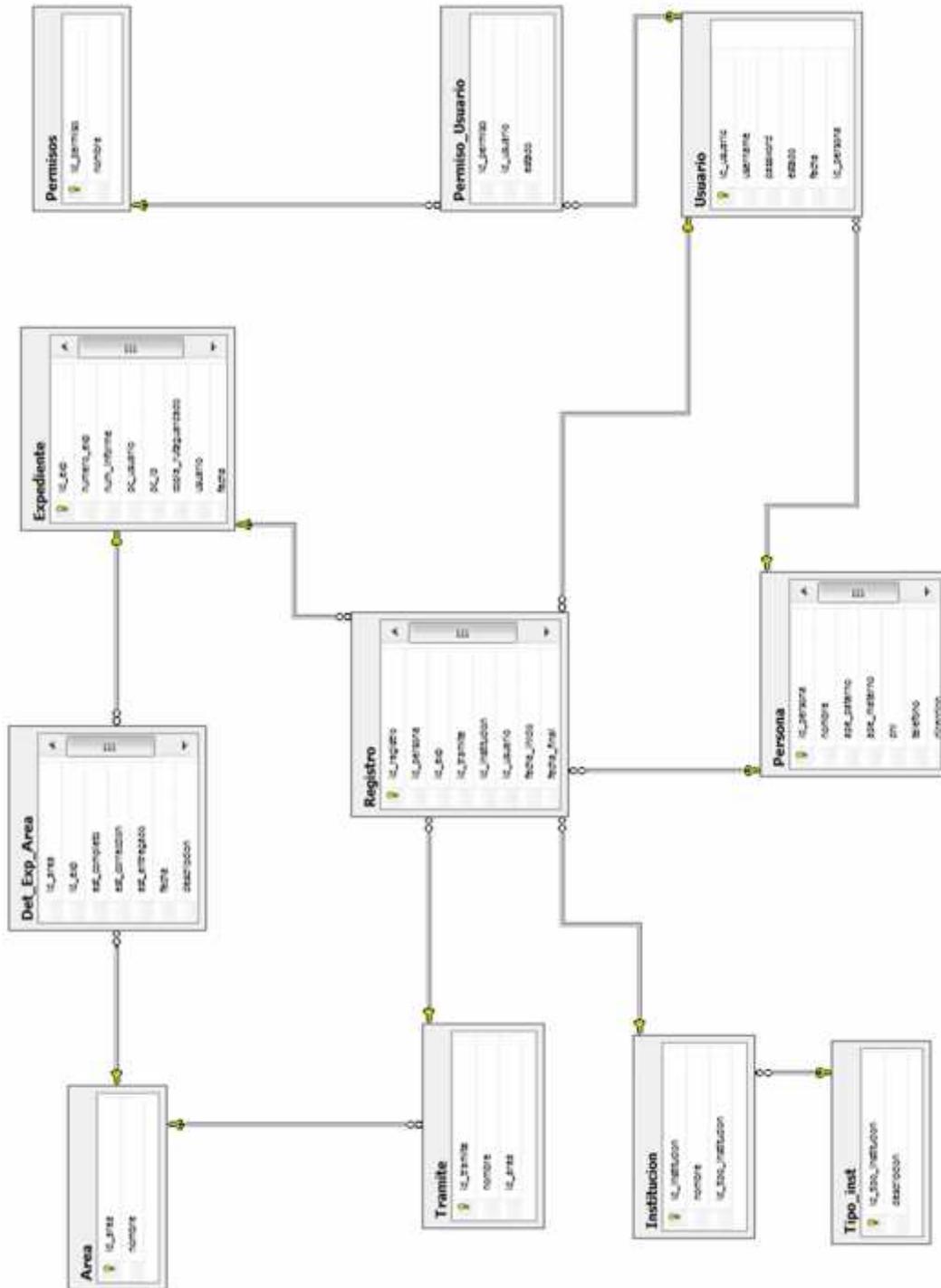
### 3.5.5. Diagrama de Consulta de Datos



### 3.5.6. Diagrama de Reportes



### 3.6. Modelado de la Base de Datos



### 3.7. Interface del Sistema

- Prototipo-Acceso de Seguridad



- Prototipo de Menú Principal:



Prototipo de mantenimiento:

- Registro de usuario

Form1

## REGISTRO USUARIO

Registrar

Codigo  Nombre

Usuario  Apellido Paterno

Clave  Apellido Materno

Fecha  Estado  Activo

NombrePaterno	Estado
Administración	<input type="checkbox"/>
Bibliotecas	<input type="checkbox"/>
Instituto de Estudios	<input type="checkbox"/>
Investigación	<input type="checkbox"/>
Planificación	<input type="checkbox"/>
Proyectos	<input type="checkbox"/>

- Registro de Tramite:

Form1

## REGISTRO DE TRAMITES

**NOMBRE**

Nombre

A. Paterno

A. Materno

D.N.I.

**EXPERIENCIA**

Nº Expediente

**INSTITUCION**

Institucion

**USUARIO**

Usuario

**AREA Y TRAMITE**

Derivado a

Tramite

**RANGO DE TRAMITE**

Fecha Ingreso

Fecha Salida

- Derivación de Documentos:

The screenshot shows a software window titled 'Derivaciones'. The main header is 'DERIVACIONES' with a pencil icon. Below this, there is a section 'Expediente y Tramite' with input fields for 'Nº Expediente' and 'Tramite', and an 'Agregar' button. A 'Derivar a:' dropdown menu is set to 'Lideres'. To the right, there is a 'RUTA DEL INFORME' section with an 'Agregar Informe' button. Below these are three checkboxes: 'Completo', 'Entregado', and 'Correccion'. A 'Descripcion' text area is also present. At the bottom, there are fields for 'Fecha' (showing 'miércoles, 04 de diciembre de 2013') and 'Usuario' (showing 'Heor'). The footer contains three buttons: 'Nuevo', 'Guardar', and 'Salir'.

- Consulta de Expedientes:

The screenshot shows a software window titled 'Busquedaexp'. The main header is 'BUSQUEDA DE EXPEDIENTES'. Below this, there is a 'BUSCAR POR' section with two radio buttons: 'Nº de DNI' (selected) and 'Nombres y Apellidos'. Each radio button is followed by an input field. To the right of these fields is a 'Buscar' button with a magnifying glass icon. Below the search fields is a large, empty grey rectangular area, likely intended for displaying search results.

- Búsqueda por Tramite:

The screenshot shows a software window titled 'Buscuedstram'. The main heading is 'BUSQUEDA DE TRAMITES'. Below the heading, there is a search section with the label 'BUSCAR POR:' and a dropdown menu currently showing 'Nº De Expediente'. To the right of the dropdown is a text input field. Further right is a button with a magnifying glass icon and the text 'Buscar'. Below the search section is a large, empty grey rectangular area, likely a placeholder for search results.

- Actualización de Datos:

The screenshot shows a software window titled 'Actualizarpersonas'. The main heading is 'ACTUALIZACION PERSONAS'. Below the heading, there are several input fields for personal data: 'Nº Registro', 'Nombre', 'D.N.I', 'Teléfono', 'Apellido Paterno', 'Apellido Materno', and 'Dirección'. There is also a field labeled 'Ingresar DNI'. To the right of the 'Teléfono' field is a button with a green checkmark icon and the text 'Actualizar'. Below that is a button with a magnifying glass icon and the text 'Buscar'. Below the form fields is a large, empty grey rectangular area, likely a placeholder for a list of updated records.

- Reportes

04/12/2015 10:20:39am Universidad Privada Ada A. Urquiola

### UBICACION DE EXPEDIENTES POR AREA

**Gerencia**

**Apoyo Economico**

EXPEDIENTE	CLIENTE	FECHA DE ENTREGA	DESCRIPCION
5	Omar Hernandez S. Noriega	17/11/2015 10:31:00am	Modificación de Pago de Cuota
6	Nicolas Hernandez Nolasco	19/11/2015 10:31:00pm	Modificación de Pago de Cuota
5	Omar Hernandez S. Noriega	06/11/2015 12:33:00am	
6	Nicolas Hernandez Nolasco	06/11/2015 12:33:00am	

- Reporte de Historial:

04/12/2015 01:20:39am Universidad Privada Ada A. Urquiola

### HISTORIAL DE EXPEDIENTES

**Biblioteca Universitaria**

CLIENTE	TRAMITE	FECHA DE ENTREGA	DESCRIPCION
Omar Hernandez S. Noriega	Apoyo Economico	05/11/2015 00:00	Subscripción de libros y programas
Nicolas Hernandez Nolasco	Apoyo Economico	05/11/2015 00:00	Subscripción de libros y programas

**Gerencia**

CLIENTE	TRAMITE	FECHA DE ENTREGA	DESCRIPCION
Omar Hernandez S. Noriega	Apoyo Economico	05/11/2015 00:00	
Nicolas Hernandez Nolasco	Apoyo Economico	05/11/2015 00:00	
Omar Hernandez S. Noriega	Apoyo Economico	17/11/2015 10:31	Modificación del Pago de Cuota
Nicolas Hernandez Nolasco	Apoyo Economico	17/11/2015 10:31	Modificación del Pago de Cuota

- Reporte General por Área

**HISTORIAL POR AREAS**

Universidad Privada del A. Tarma

**Admisión**

EXPEDIENTE	CUENTE	FECHA DE INGRESO
1	Cont. Control de Ingreso	2015/01/01 08:00:00

---

**Director Académico**

EXPEDIENTE	CUENTE	FECHA DE INGRESO
2	Cont. Tareas de Ingreso	2015/01/01 08:00:00

---

**Guarida**

EXPEDIENTE	CUENTE	FECHA DE INGRESO
3	Cont. Ingreso de S. E. C.	2015/01/01 12:00:00
4	Ingreso de Ingreso de S. E. C.	2015/01/01 08:00:00

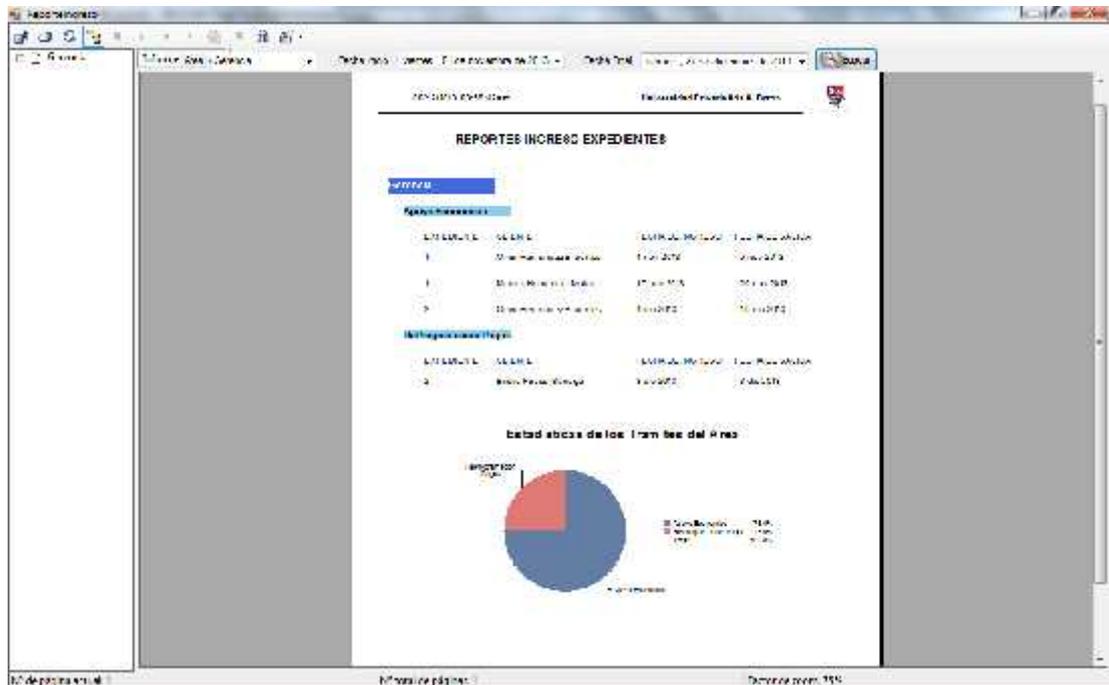
---

**Desarrollo de Pagos**

EXPEDIENTE	CUENTE	FECHA DE INGRESO
5	Cont. Ingreso de Pagos	2015/01/01 08:00:00

Reporte generado el: 2015/01/01 08:00:00

- Reporte Estadístico



### 3.8. Fase de Programación del Sistema

Cada una de las clases y subclases identificadas en el diseño deben ser codificadas para proporcionar una interfaz que tenga relación con el usuario.

Estructura de programar para el acceso del sistema.

Clases: Estructura de las clases empleadas en la programación.

#### CLASE AREA

```
Public Class Area
    Private _nombre As String
    Private _idarea As String
    Public Property idarea() As String
        Get
            Return _idarea
        End Get
        Set(ByVal value As String)
            _idarea = value
        End Set
    End Property
    Public Property nombre() As String
        Get
            Return _nombre
        End Get
        Set(ByVal value As String)
            _nombre = value
        End Set
    End Property
End Class
```

#### CLASE DETEXPAREA

```
Public Class DetExpArea
    Private _area As New Area
    Private _expediente As New Expediente
    Private _estcompleto As String
    Private _estcorreccion As String
    Private _estentregado As String
    Private _fecha As DateTime
    Private _descripcion As String
    Public Property descripcion() As String
        Get
            Return _descripcion
        End Get
        Set(ByVal value As String)
            _descripcion = value
        End Set
    End Property
End Class
```

```
        End Set
    End Property

    Public Property fecha() As DateTime
        Get
            Return _fecha
        End Get
        Set(ByVal value As DateTime)
            _fecha = value
        End Set
    End Property

    Public Property estentregado() As String
        Get
            Return _estentregado
        End Get
        Set(ByVal value As String)
            _estentregado = value
        End Set
    End Property

    Public Property estcorreccion() As String
        Get
            Return _estcorreccion
        End Get
        Set(ByVal value As String)
            _estcorreccion = value
        End Set
    End Property

    Public Property estcompleto() As String
        Get
            Return _estcompleto
        End Get
        Set(ByVal value As String)
            _estcompleto = value
        End Set
    End Property
    Public Property expediente() As Expediente
        Get
            Return _expediente
        End Get
        Set(ByVal value As Expediente)
            _expediente = value
        End Set
    End Property

    Public Property area() As Area
        Get
            Return _area
        End Get
        Set(ByVal value As Area)
            _area = value
        End Set
    End Property
End Class
```

## CLASE EXPEDIENTE

```
Public Class Expediente
    Private _idexpediente As String
    Private _numexpediente As String
    Private _numinforme As String
    Private _numresolucion As String
    Private _copiares As String
    Private _numinflegal As String
    Public Property numinlegal() As String
        Get
            Return _numinflegal
        End Get
        Set(ByVal value As String)
            _numinflegal = value
        End Set
    End Property

    Public Property copiares() As String
        Get
            Return _copiares
        End Get
        Set(ByVal value As String)
            _copiares = value
        End Set
    End Property

    Public Property numresolucion() As String
        Get
            Return _numresolucion
        End Get
        Set(ByVal value As String)
            _numresolucion = value
        End Set
    End Property

    Public Property numinforme() As String
        Get
            Return _numinforme
        End Get
        Set(ByVal value As String)
            _numinforme = value
        End Set
    End Property

    Public Property numexpediente() As String
        Get
            Return _numexpediente
        End Get
        Set(ByVal value As String)
            _numexpediente = value
        End Set
    End Property

    Public Property idexpediente() As String
        Get
            Return _idexpediente
        End Get
        Set(ByVal value As String)
```

```
        _idexpediente = value
    End Set
End Property
End Class
```

## CLASE INSTITUCION

```
Public Class Institucion

    Private _idInstitucion As String
    Private _nombre As String
    Private _tipoInstitucion As New TipoInstitucion
    Public Property tipoInstitucion() As TipoInstitucion
        Get
            Return _tipoInstitucion
        End Get
        Set(ByVal value As TipoInstitucion)
            _tipoInstitucion = value
        End Set
    End Property
    Public Property nombre() As String
        Get
            Return _nombre
        End Get
        Set(ByVal value As String)
            _nombre = value
        End Set
    End Property
    Public Property idInstitucion() As String
        Get
            Return _idInstitucion
        End Get
        Set(ByVal value As String)
            _idInstitucion = value
        End Set
    End Property
End Class
```

## CLASE PERMISO

```
Public Class Permisos
    Private _idpermiso As String
    Private _nombre As String
    Public Property nombre() As String
        Get
            Return _nombre
        End Get
        Set(ByVal value As String)
            _nombre = value
        End Set
    End Property

    Public Property idpermiso() As String
        Get
            Return _idpermiso
        End Get
    End Property
End Class
```

```
        End Get
        Set(ByVal value As String)
            _idpermiso = value
        End Set
    End Property
End Class
```

## CLASE PERMISO USUARIO

```
Public Class PermisoUsuario
    Private _estado As String
    Private _permiso As Permisos
    Private _usuario As Usuario
    Public Property permiso() As Permisos
        Get
            Return _permiso
        End Get
        Set(ByVal value As Permisos)
            _permiso = value
        End Set
    End Property
    Public Property usuario() As Usuario
        Get
            Return _usuario
        End Get
        Set(ByVal value As Usuario)
            _usuario = value
        End Set
    End Property
    Public Property estado() As String
        Get
            Return _estado
        End Get
        Set(ByVal value As String)
            _estado = value
        End Set
    End Property
End Class
```

## CLASE PERSONA

```
Public Class Persona
    Private _idpersona As String
    Private _nombre As String
    Private _apepaterno As String
    Private _apematerno As String
    Private _direccion As String
    Private _telefono As String
    Private _dni As String
    Public Property telefono() As String
        Get
            Return _telefono
        End Get
        Set(ByVal value As String)
            _telefono = value
        End Set
    End Property
End Class
```

```
        End Set
    End Property

    Public Property direccion() As String
        Get
            Return _direccion
        End Get
        Set(ByVal value As String)
            _direccion = value
        End Set
    End Property

    Public Property dni() As String
        Get
            Return _dni
        End Get
        Set(ByVal value As String)
            _dni = value
        End Set
    End Property

    Public Property apematerno() As String
        Get
            Return _apematerno
        End Get
        Set(ByVal value As String)
            _apematerno = value
        End Set
    End Property
    Public Property apepaterno() As String
        Get
            Return _apepaterno
        End Get
        Set(ByVal value As String)
            _apepaterno = value
        End Set
    End Property

    Public Property nombre() As String
        Get
            Return _nombre
        End Get
        Set(ByVal value As String)
            _nombre = value
        End Set
    End Property

    Public Property idpersona() As String
        Get
            Return _idpersona
        End Get
        Set(ByVal value As String)
            _idpersona = value
        End Set
    End Property
End Class
```

## CLASE REGISTRO

```
Public Class Registro
    Private _idregistro As String
    Private _fechainicio As DateTime
    Private _fechafinal As DateTime
    Private _tramite As New Tramite
    Private _institucion As New Institucion
    Private _expediente As New Expediente
    Private _usuario As New Usuario
    Private _persona As New Persona
    Public Property persona() As Persona
        Get
            Return _persona
        End Get
        Set(ByVal value As Persona)
            _persona = value
        End Set
    End Property
    Public Property usuario() As Usuario
        Get
            Return _usuario

        End Get
        Set(ByVal value As Usuario)
            _usuario = value
        End Set
    End Property

    Public Property expediente() As Expediente
        Get
            Return _expediente
        End Get
        Set(ByVal value As Expediente)
            _expediente = value
        End Set
    End Property

    Public Property institucion() As Institucion
        Get
            Return _institucion
        End Get
        Set(ByVal value As Institucion)
            _institucion = value
        End Set
    End Property

    Public Property tramite() As Tramite
        Get
            Return _tramite
        End Get
        Set(ByVal value As Tramite)
            _tramite = value
        End Set
    End Property

    Public Property fechafinal() As DateTime
        Get
```

```
        Return _fechafinal
    End Get
    Set(ByVal value As DateTime)
        _fechafinal = value
    End Set
End Property

Public Property fechainicio() As DateTime
    Get
        Return _fechainicio
    End Get
    Set(ByVal value As DateTime)
        _fechainicio = value
    End Set
End Property

Public Property idregistro() As String
    Get
        Return _idregistro
    End Get
    Set(ByVal value As String)

        _idregistro = value
    End Set
End Property

End Class
```

## CLASE INSTITUCIÓN

```
Public Class TipoInstitucion
    Private _idTipoInstitucion As String
    Private _descripcion As String

    Public Property descripcion() As String
        Get
            Return _descripcion
        End Get
        Set(ByVal value As String)
            _descripcion = value
        End Set
    End Property
    Public Property idTipoInstitucion() As String
        Get
            Return _idTipoInstitucion
        End Get
        Set(ByVal value As String)
            _idTipoInstitucion = value
        End Set
    End Property

End Class
```

## CLASE TRÁMITE

```
Public Class Tramite
    Private _idtramite As String
    Private _nombre As String
    Private _costo As String
    Private _area As New Area
    Public Property area() As Area
        Get
            Return _area
        End Get
        Set(ByVal value As Area)
            _area = value
        End Set
    End Property

    Public Property costo() As String
        Get
            Return _costo
        End Get
        Set(ByVal value As String)
            _costo = value
        End Set
    End Property

    Public Property nombre() As String
        Get
            Return _nombre
        End Get
        Set(ByVal value As String)
            _nombre = value
        End Set
    End Property

    Public Property idtramite() As String
        Get
            Return _idtramite
        End Get
        Set(ByVal value As String)
            _idtramite = value
        End Set
    End Property

End Class
```

## CLASE USUARIO

```
Public Class Usuario
    Private _idusuario As String
    Private _password As String
    Private _usuario As String
    Private _estado As String
    Private _persona As New Persona
    Private _fecha As DateTime
```

```
Public Property fecha() As DateTime
    Get
        Return _fecha
    End Get
    Set(ByVal value As DateTime)
        _fecha = value
    End Set
End Property

Public Property persona() As Persona
    Get
        Return _persona
    End Get
    Set(ByVal value As Persona)
        _persona = value
    End Set
End Property

Public Property estado() As String
    Get

        Return _estado
    End Get
    Set(ByVal value As String)
        _estado = value
    End Set
End Property

Public Property usuario() As String
    Get
        Return _usuario
    End Get
    Set(ByVal value As String)
        _usuario = value
    End Set
End Property

Public Property password() As String
    Get
        Return _password
    End Get
    Set(ByVal value As String)
        _password = value
    End Set
End Property

Public Property idusuario() As String
    Get
        Return _idusuario
    End Get
    Set(ByVal value As String)
        _idusuario = value
    End Set
End Property

    End Class
```

## CONEXION A LA BASE DE DATOS

```
Imports System.Data.SqlClient
Public Class DBAccess
    Private conexion As New SqlConnection _
        ("Data Source =.; DataBase =mesadepartes; Integrated
Security=true")
    Function cs() As SqlConnection
        Return conexion
    End Function
    Sub conectar()
        conexion.Open()
    End Sub
    Sub desconectar()
        conexion.Close()
    End Sub
End Class
```

## CLASES DE PROCESAMIENTO DE DATOS

### CLASE AREA

```
Imports BEANS
Imports DAL
Imports System.Data
Imports System.Data.SqlClient

Public Class AreaDAO

    Dim cnn As New DBAccess
    Dim dts As New DataTable
    Dim comando As New SqlCommand
    Dim dr As SqlDataReader
    Function areaListar() As DataTable

        dts = New DataTable
        cnn.conectar()
        comando.Connection = cnn.cs
        comando.CommandType = CommandType.StoredProcedure
        comando.CommandText = "tramite_listar"
        dr = comando.ExecuteReader()
        dts.Load(dr, LoadOption.OverwriteChanges)
        cnn.desconectar()
        Return dts
    End Function

End Class
```

### CLASE DetExpArea

```
Imports BEANS
Imports DAL
Imports System.Data
Imports System.Data.SqlClient
```

```
Public Class DetExpAreaDAO
    Dim cnn As New DBAccess
    Dim dt As New DataSet
    Dim comando As New SqlCommand
    Dim dap As New SqlDataAdapter
    Dim dts As New DataTable
    Dim dr As SqlDataReader

    Sub insertExpDetArea(ByVal objDetExpArea As DetExpArea)
        'aca se colocara el codigo para registrar a la persona

        cnn.conectar()
        comando.Connection = cnn.cs
        comando.CommandType = CommandType.StoredProcedure
        comando.Parameters.Add(New SqlParameter("@id_area",
objDetExpArea.area.idarea))
        comando.Parameters.Add(New SqlParameter("@id_exp",
objDetExpArea.expediente.idexpediente))
        comando.Parameters.Add(New SqlParameter("@estcompleto",
objDetExpArea.estcompleto))
        comando.Parameters.Add(New SqlParameter("@estcorreccion",
objDetExpArea.estcorreccion))
        comando.Parameters.Add(New SqlParameter("@estentregado",
objDetExpArea.estentregado))
        comando.Parameters.Add(New SqlParameter("@fecha",
objDetExpArea.fecha))
        comando.Parameters.Add(New SqlParameter("@descripcion",
objDetExpArea.descripcion))

        comando.CommandText = "detalleexp_insert"
        comando.ExecuteNonQuery()
        cnn.desconectar()
    End Sub

    Function consultaDetExpArea(ByVal filtros As String) As
DataTable
        cnn.conectar()
        comando.Connection = cnn.cs
        comando.CommandType = CommandType.StoredProcedure
        comando.CommandText = "DetExpArea_listar"
        comando.Parameters.Add(New SqlParameter("@filtro",
filtros))
        dr = comando.ExecuteReader()
        dts.Load(dr, LoadOption.OverwriteChanges)
        cnn.desconectar()
        Return dts
    End Function

    Function reporteexpentrega(ByVal fecha1 As DateTime, ByVal
fecha2 As DateTime, ByVal area As String) As DataTable
        cnn.conectar()
        comando.Connection = cnn.cs
        comando.CommandType = CommandType.StoredProcedure
        comando.CommandText = "reporte_salidas"
        comando.Parameters.Add(New SqlParameter("@area", area))
        comando.Parameters.Add(New SqlParameter("@fecha1",
fecha1.Date))
```

```
        comando.Parameters.Add(New SqlParameter("@fecha2",  
fecha2.Date))  
        dr = comando.ExecuteReader()  
        dts.Load(dr, LoadOption.OverwriteChanges)  
        cnn.desconectar()  
        Return dts  
    End Function  
  
    Function consultaexpingresado(ByVal filtros As String) As  
DataTable  
        cnn.conectar()  
        comando.Connection = cnn.cs  
        comando.CommandType = CommandType.StoredProcedure  
        comando.CommandText = "expediente_listar"  
        comando.Parameters.Add(New SqlParameter("@filtro",  
filtros))  
        dr = comando.ExecuteReader()  
        dts.Load(dr, LoadOption.OverwriteChanges)  
        cnn.desconectar()  
        Return dts  
    End Function  
  
    Function reporte_historial(ByVal exp As String) As DataTable  
        cnn.conectar()  
        comando.Connection = cnn.cs  
        comando.CommandType = CommandType.StoredProcedure  
        comando.CommandText = "reporte_historial"  
        comando.Parameters.Add(New SqlParameter("@exp", exp))  
  
        dr = comando.ExecuteReader()  
        dts.Load(dr, LoadOption.OverwriteChanges)  
        cnn.desconectar()  
        Return dts  
    End Function  
  
End Class
```

#### **CLASE Expediente**

```
Imports BEANS  
Imports DAL  
Imports System.Data  
Imports System.Data.SqlClient  
  
Public Class ExpedienteDAO  
  
    Dim cnn As New DBAccess  
    Dim dt As New DataSet  
    Dim comando As New SqlCommand  
    Dim dap As New SqlDataAdapter  
    Dim dts As New DataTable  
    Dim dr As SqlDataReader  
    Sub insertexpediente(ByVal objExpediente As Expediente)  
        'aca se colocara el codigo para registrar a la persona
```

```
        cnn.conectar()
        comando.Connection = cnn.cs
        comando.CommandType = CommandType.StoredProcedure
        comando.Parameters.Add(New SqlParameter("@numeroexp",
objExpediente.numexpediente))
        comando.CommandText = "expediente_insert"
        comando.ExecuteNonQuery()
        cnn.desconectar()
    End Sub

    Function ExpedientenListar() As DataTable

        dts = New DataTable
        cnn.conectar()
        comando.Connection = cnn.cs
        comando.CommandType = CommandType.StoredProcedure
        comando.CommandText = "expediente_listarregistro"
        dr = comando.ExecuteReader()
        dts.Load(dr, LoadOption.OverwriteChanges)
        cnn.desconectar()
        Return dts
    End Function

    Function consultaExpediente(ByVal filtros As String) As
DataTable
        cnn.conectar()
        comando.Connection = cnn.cs
        comando.CommandType = CommandType.StoredProcedure
        comando.CommandText = "expediente_consulta"
        comando.Parameters.Add(New SqlParameter("@filtro",
filtros))
        dr = comando.ExecuteReader()
        dts.Load(dr, LoadOption.OverwriteChanges)
        cnn.desconectar()
        Return dts
    End Function

    Sub expActualizar(ByVal objExpediente As Expediente)
        cnn.conectar()

        comando.Connection = cnn.cs
        comando.CommandType = CommandType.StoredProcedure
        comando.Parameters.Add(New SqlParameter("@numexp",
objExpediente.numexpediente))
        comando.Parameters.Add(New SqlParameter("@numinforme",
objExpediente.numinforme))
        comando.Parameters.Add(New SqlParameter("@pcusario",
objExpediente.pcusuario))
        comando.Parameters.Add(New SqlParameter("@pcip",
objExpediente.pcip))
        comando.Parameters.Add(New SqlParameter("@copiainforme",
objExpediente.rutainforme))
        comando.Parameters.Add(New SqlParameter("@ususario",
objExpediente.usuario))
```

```
        comando.Parameters.Add(New SqlParameter("@fecha",  
objExpediente.fecha))  
  
        comando.CommandText = "actualizar_exp"  
        comando.ExecuteNonQuery()  
        cnn.desconectar()  
    End Sub  
  
    Function consultaExpedientenombre(ByVal filtros As String)  
As DataTable  
        cnn.conectar()  
        comando.Connection = cnn.cs  
        comando.CommandType = CommandType.StoredProcedure  
        comando.CommandText = "expediente_consultanombre"  
        comando.Parameters.Add(New SqlParameter("@filtro",  
filtros))  
        dr = comando.ExecuteReader()  
        dts.Load(dr, LoadOption.OverwriteChanges)  
        cnn.desconectar()  
        Return dts  
    End Function  
End Class
```

## CLASE Instituto

```
Imports BEANS  
Imports DAL  
Imports System.Data  
Imports System.Data.SqlClient  
Public Class InstitucionDAO  
    Dim cnn As New DBAccess  
    Dim dt As New DataSet  
    Dim comando As New SqlCommand  
    Dim dap As New SqlDataAdapter  
    Dim dts As New DataTable  
    Dim dr As SqlDataReader  
    Sub insertInstitucion(ByVal objInstitucion As Institucion)  
        'aca se colocara el codigo para registrar a la persona  
        cnn.conectar()  
        comando.Connection = cnn.cs  
        comando.CommandType = CommandType.StoredProcedure  
        comando.Parameters.Add(New SqlParameter("@nombre",  
objInstitucion.nombre))  
        comando.Parameters.Add(New SqlParameter("@id_tipo_inst",  
objInstitucion.tipoInstitucion.idTipoInstitucion))  
        comando.CommandText = "institucion_insert"  
        comando.ExecuteNonQuery()  
        cnn.desconectar()  
    End Sub  
  
    Function InstitucionListar() As DataTable  
  
        dts = New DataTable  
        cnn.conectar()  
        comando.Connection = cnn.cs
```

```
        comando.CommandType = CommandType.StoredProcedure
        comando.CommandText = "inst_listar"
        dr = comando.ExecuteReader()
        dts.Load(dr, LoadOption.OverwriteChanges)
        cnn.desconectar()
        Return dts
    End Function
End Class
```

### CLASE Permiso

```
Imports BEANS
Imports DAL
Imports System.Data
Imports System.Data.SqlClient
Public Class PermisoDAO
    Dim cnn As New DBAccess
    Dim dt As New DataSet
    Dim comando As New SqlCommand
    Dim dap As New SqlDataAdapter
    Dim dts As New DataTable
    Dim dr As SqlDataReader

    Function permisos_listar() As DataTable
        cnn.conectar()
        comando.Connection = cnn.cs
        comando.CommandType = CommandType.StoredProcedure
        comando.CommandText = "permisos_listar"
        dr = comando.ExecuteReader()
        dts.Load(dr, LoadOption.OverwriteChanges)
        cnn.desconectar()
        Return dts
    End Function
End Class
```

### CLASE Persona

```
Imports BEANS
Imports DAL
Imports System.Data
Imports System.Data.SqlClient

Public Class PersonaDAO

    Dim cnn As New DBAccess
    Dim dt As New DataSet
    Dim comando As New SqlCommand
    Dim dap As New SqlDataAdapter
    Dim dts As New DataTable
    Dim dr As SqlDataReader
    Sub insertPersona(ByVal objPersona As Persona)
        'aca se colocara el codigo para registrar a la persona
    End Sub
End Class
```

```
        cnn.conectar()
        comando.Connection = cnn.cs
        comando.CommandType = CommandType.StoredProcedure
        comando.Parameters.Add(New SqlParameter("@nombre",
objPersona.nombre))
        comando.Parameters.Add(New SqlParameter("@apellidopat",
objPersona.apepaterno))
        comando.Parameters.Add(New SqlParameter("@apellidomat",
objPersona.apematerno))
        comando.Parameters.Add(New SqlParameter("@dni",
objPersona.dni))
        comando.Parameters.Add(New SqlParameter("@telefono",
objPersona.telefono))
        comando.Parameters.Add(New SqlParameter("@direccion",
objPersona.direccion))

        comando.CommandText = "persona_insert"
        comando.ExecuteNonQuery()
        cnn.desconectar()
    End Sub

    Function consultaPersona(ByVal objPersona As Persona) As
DataTable
        cnn.conectar()
        comando.Connection = cnn.cs
        comando.CommandType = CommandType.StoredProcedure
        comando.CommandText = "persona_consulta"
        comando.Parameters.Add(New SqlParameter("@dni",
objPersona.dni))
        dr = comando.ExecuteReader()
        dts.Load(dr, LoadOption.OverwriteChanges)
        cnn.desconectar()
        Return dts
    End Function

    Sub actualizarPersona(ByVal objPersona As Persona)
        'aca se colocara el codigo para registrar a la persona
        cnn.conectar()
        comando.Connection = cnn.cs
        comando.CommandType = CommandType.StoredProcedure
        comando.Parameters.Add(New SqlParameter("@idpersona",
objPersona.idpersona))
        comando.Parameters.Add(New SqlParameter("@nombre",
objPersona.nombre))
        comando.Parameters.Add(New SqlParameter("@apellidopat",
objPersona.apepaterno))
        comando.Parameters.Add(New SqlParameter("@apellidomat",
objPersona.apematerno))
        comando.Parameters.Add(New SqlParameter("@dni",
objPersona.dni))
        comando.Parameters.Add(New SqlParameter("@telefono",
objPersona.telefono))
        comando.Parameters.Add(New SqlParameter("@direccion",
objPersona.direccion))

        comando.CommandText = "persona_update"
```

```
        comando.ExecuteNonQuery()  
        cnn.desconectar()  
    End Sub  
  
    Function Filtronombres(ByVal filtros As String) As DataTable  
        cnn.conectar()  
        comando.Connection = cnn.cs  
        comando.CommandType = CommandType.StoredProcedure  
        comando.CommandText = "consulta_personal"  
        comando.Parameters.Add(New SqlParameter("@filtro",  
filtros))  
        dr = comando.ExecuteReader()  
        dts.Load(dr, LoadOption.OverwriteChanges)  
        cnn.desconectar()  
        Return dts  
    End Function  
  
    Function Filtrodni(ByVal filtros As String) As DataTable  
        cnn.conectar()  
        comando.Connection = cnn.cs  
        comando.CommandType = CommandType.StoredProcedure  
        comando.CommandText = "consulta_personadni"  
        comando.Parameters.Add(New SqlParameter("@filtro",  
filtros))  
        dr = comando.ExecuteReader()  
        dts.Load(dr, LoadOption.OverwriteChanges)  
        cnn.desconectar()  
        Return dts  
    End Function  
End Class
```

### **CLASE Registro**

```
Imports BEANS  
Imports DAL  
Imports System.Data  
Imports System.Data.SqlClient  
  
Public Class RegistroDAO  
    Dim cnn As New DBAccess  
    Dim dt As New DataSet  
    Dim comando As New SqlCommand  
    Dim dap As New SqlDataAdapter  
    Dim dts As New DataTable  
    Dim dr As SqlDataReader  
    Sub insertRegistro(ByVal objRegistro As Registro)  
        'aca se colocara el codigo para registra  
  
        cnn.conectar()  
        comando.Connection = cnn.cs  
        comando.CommandType = CommandType.StoredProcedure  
        comando.Parameters.Add(New SqlParameter("@id_persona",  
objRegistro.persona.idpersona))  
        comando.Parameters.Add(New SqlParameter("@id_exp",  
objRegistro.expediente.idexpediente))
```

```
        comando.Parameters.Add(New SqlParameter("@id_tramite",
objRegistro.tramite.idtramite))
        comando.Parameters.Add(New
SqlParameter("@id_institucion",
objRegistro.institucion.idInstitucion))
        comando.Parameters.Add(New SqlParameter("@id_usuario",
objRegistro.usuario.idusuario))
        comando.Parameters.Add(New SqlParameter("@fechaini",
objRegistro.fechainicio))
        comando.Parameters.Add(New SqlParameter("@fechafinal",
objRegistro.fechafinal))

        comando.CommandText = "registro_insert"
        comando.ExecuteNonQuery()
        cnn.desconectar()
    End Sub

    Function reporteingreso(ByVal fecha1 As DateTime, ByVal
fecha2 As DateTime, ByVal area As String) As DataTable
        cnn.conectar()
        comando.Connection = cnn.cs
        comando.CommandType = CommandType.StoredProcedure
        comando.CommandText = "reportederivaciones"
        comando.Parameters.Add(New SqlParameter("@area", area))
        comando.Parameters.Add(New SqlParameter("@fecha1",
fecha1.Date))
        comando.Parameters.Add(New SqlParameter("@fecha2",
fecha2.Date))
        dr = comando.ExecuteReader()
        dts.Load(dr, LoadOption.OverwriteChanges)
        cnn.desconectar()
        Return dts
    End Function

    Function reporteingresocaja(ByVal fecha1 As DateTime, ByVal
fecha2 As DateTime, ByVal area As String) As DataTable
        cnn.conectar()
        comando.Connection = cnn.cs
        comando.CommandType = CommandType.StoredProcedure
        comando.CommandText = "reporte_ingresocaja"
        comando.Parameters.Add(New SqlParameter("@area", area))
        comando.Parameters.Add(New SqlParameter("@fecha1",
fecha1.Date))
        comando.Parameters.Add(New SqlParameter("@fecha2",
fecha2.Date))
        dr = comando.ExecuteReader()
        dts.Load(dr, LoadOption.OverwriteChanges)
        cnn.desconectar()
        Return dts
    End Function

    Function reporteestadistica(ByVal fecha1 As DateTime, ByVal
fecha2 As DateTime, ByVal dni As String) As DataTable
        cnn.conectar()
        comando.Connection = cnn.cs
        comando.CommandType = CommandType.StoredProcedure
        comando.CommandText = "reporte_estadisticas"
        comando.Parameters.Add(New SqlParameter("@dni", dni))
```

```
        comando.Parameters.Add(New SqlParameter("@fecha1",
fecha1.Date))
        comando.Parameters.Add(New SqlParameter("@fecha2",
fecha2.Date))
        dr = comando.ExecuteReader()
        dts.Load(dr, LoadOption.OverwriteChanges)
        cnn.desconectar()
        Return dts
    End Function

    Function reporteestadisticageneral(ByVal fecha1 As DateTime,
ByVal fecha2 As DateTime) As DataTable
        cnn.conectar()
        comando.Connection = cnn.cs
        comando.CommandType = CommandType.StoredProcedure
        comando.CommandText = "reporteestaditica_general"
        comando.Parameters.Add(New SqlParameter("@fecha1",
fecha1.Date))
        comando.Parameters.Add(New SqlParameter("@fecha2",
fecha2.Date))
        dr = comando.ExecuteReader()
        dts.Load(dr, LoadOption.OverwriteChanges)
        cnn.desconectar()
        Return dts
    End Function
End Class
```

## CLASE TipInstitucion

```
Imports BEANS
Imports DAL
Imports System.Data
Imports System.Data.SqlClient

Public Class TipoInstitucionDAO

    Dim cnn As New DBAccess
    Dim dts As New DataTable
    Dim comando As New SqlCommand
    Dim dr As SqlDataReader
    Function tipoInstitucionListar() As DataTable

        dts = New DataTable
        cnn.conectar()
        comando.Connection = cnn.cs
        comando.CommandType = CommandType.StoredProcedure
        comando.CommandText = "institucion_listar"
        dr = comando.ExecuteReader()

        dts.Load(dr, LoadOption.OverwriteChanges)
        cnn.desconectar()
        Return dts
    End Function

End Class
```

## CLASE Tramite

```
Imports BEANS
Imports DAL
Imports System.Data
Imports System.Data.SqlClient

Public Class TramiteDAO

    Dim cnn As New DBAccess
    Dim dts As New DataSet
    Dim comando As New SqlCommand
    Dim dap As New SqlDataAdapter

    Sub insertTramite(ByVal objTramite As Tramite)
        'aca se colocara el codigo para registrar a la persona
        cnn.conectar()
        comando.Connection = cnn.cs
        comando.CommandType = CommandType.StoredProcedure
        comando.Parameters.Add(New SqlParameter("@nombre",
objTramite.nombre))
        comando.Parameters.Add(New SqlParameter("@costo",
objTramite.costo))
        comando.Parameters.Add(New SqlParameter("@id_area",
objTramite.area.idarea))
        comando.CommandText = "tramite_insert"
        comando.ExecuteNonQuery()
        cnn.desconectar()
    End Sub

    Public Function tramite_Consulta(ByVal objtramite As
Tramite) As DataTable
        Dim cnn As New DBAccess
        Dim comando As New SqlCommand
        Dim dr As SqlDataReader
        Dim dt As New DataTable
        cnn.conectar()
        comando.Connection = cnn.cs
        comando.CommandType = CommandType.StoredProcedure
        comando.CommandText = "tramite_consulta"
        comando.Parameters.Add(New SqlParameter("@idarea",
objtramite.area.idarea))
        comando.ExecuteNonQuery()
        dr = comando.ExecuteReader
        dt.Load(dr, LoadOption.OverwriteChanges)
        Return dt
        cnn.desconectar()
    End Function

    Public Function tramite_traerCodigo(ByVal objtramite As
Tramite) As DataTable
        Dim cnn As New DBAccess
        Dim comando As New SqlCommand
        Dim dr As SqlDataReader
```

```
        Dim dt As New DataTable
        cnn.conectar()
        comando.Connection = cnn.cs
        comando.CommandType = CommandType.StoredProcedure
        comando.CommandText = "tramite_codigo"
        comando.Parameters.Add(New
SqlParameter("@nombretramite", objtramite.nombre))
        comando.ExecuteNonQuery()
        dr = comando.ExecuteReader
        dt.Load(dr, LoadOption.OverwriteChanges)
        Return dt
    End Function
```

End Class

## CLASE Usuario

```
Imports BEANS
Imports DAL
Imports System.Data
Imports System.Data.SqlClient

Public Class UsuarioDAO
    Dim cnn As New DBAccess
    Dim dt As New DataSet
    Dim comando As New SqlCommand
    Dim dap As New SqlDataAdapter
    Dim dts As New DataTable
    Dim dr As SqlDataReader
    Sub insertUsuario(ByVal objUsuario As Usuario, ByVal
bolMantenimiento As Boolean, ByVal bolRegistro As Boolean, _
    ByVal bolActualizacion As Boolean, ByVal bolReportes As
Boolean, ByVal bolBusqueda As Boolean)
        'aca se colocara el codigo para registrar a la persona

        cnn.conectar()
        comando.Connection = cnn.cs
        comando.CommandType = CommandType.StoredProcedure
        comando.Parameters.Add(New SqlParameter("@usuario",
objUsuario.usuario))
        comando.Parameters.Add(New SqlParameter("@password",
objUsuario.password))
        comando.Parameters.Add(New SqlParameter("@estado",
objUsuario.estado))
        comando.Parameters.Add(New SqlParameter("@fecha",
objUsuario.fecha))
        comando.Parameters.Add(New SqlParameter("@id_persona",
objUsuario.persona.idpersona))
        comando.Parameters.Add(New
SqlParameter("@mantenimiento", bolMantenimiento))
        comando.Parameters.Add(New SqlParameter("@reportes",
bolReportes))
        comando.Parameters.Add(New SqlParameter("@registros",
bolRegistro))
        comando.Parameters.Add(New SqlParameter("@busqueda",
bolBusqueda))
```

```
        comando.Parameters.Add(New
SqlCommandParameter("@actualizacion", bolActualizacion))
        comando.CommandText = "usuario_insert"
        comando.ExecuteNonQuery()
        cnn.desconectar()
    End Sub

    Function validar(ByVal objUsuario As Usuario) As Boolean

        cnn.conectar()
        comando.Connection = cnn.cs
        comando.CommandType = CommandType.StoredProcedure
        comando.Parameters.Add(New SqlParameter("@usuario",
objUsuario.usuario))
        comando.Parameters.Add(New SqlParameter("@password",
objUsuario.password))
        comando.CommandText = "logeo_usuario"
        dr = comando.ExecuteReader()
        If dr.HasRows Then
            Return True
        Else
            Return False
        End If

        cnn.desconectar()

    End Function

    Public Function Usuario_traerCodigo(ByVal objusuario As
Usuario) As DataTable
        Dim cnn As New DBAccess
        Dim comando As New SqlCommand
        Dim dr As SqlDataReader
        Dim dt As New DataTable
        cnn.conectar()
        comando.Connection = cnn.cs
        comando.CommandType = CommandType.StoredProcedure
        comando.CommandText = "usuario_codigo"
        comando.Parameters.Add(New SqlParameter("@username",
objusuario.usuario))
        comando.ExecuteNonQuery()
        dr = comando.ExecuteReader
        dt.Load(dr, LoadOption.OverwriteChanges)
        Return dt
    End Function

End Class
```

## CONCLUSIONES

- ✓ Los Sistemas de Información facilitan y optimizan los procesos en las organizaciones brindándoles una mejor toma de decisiones en sus procesos, con el fin de dar un mejor servicio a los clientes como a sus trabajadores.
- ✓ El diseño y aplicación de un Programa de Gestión Documental en la empresa, mejora sensiblemente el control y organización de los documentos, y contribuye al logro de los objetivos empresariales.
- ✓ El éxito de la implementación de un sistema de información, involucra varios aspectos en los cuales la capacitación previa del personal que va utilizar el programa es un punto crítico para cumplir con los objetivos.
- ✓ La evaluación de los recursos tanto técnicos como humanos es muy importante, ya que, si no se cuenta con ellos no se puede desarrollar el sistema de información o se desarrolla un sistema que no sea funcional en cuanto a los requerimientos y las necesidades de la institución.

## RECOMENDACIONES

- Es recomendable que todas las empresas diseñen y apliquen un Programa de Gestión Documental que les permita controlar y organizar todos los documentos que ingresan y salen de dichas instituciones, ya que ello contribuye al logro de sus objetivos empresariales.
- Es importante que todas las áreas internas de la institución, involucradas en la generación y producción de información, realicen coordinaciones para uniformizar los procedimientos de trámite documentario y evitar la pérdida de tiempo.
- Se recomienda a la organización brindar una capacitación en temas de T.I. para que el personal tenga un mayor conocimiento en su uso y acelerar los beneficios que brinda en su implementación.
- Se debe tener en cuenta un plan o programa de mantenimiento tanto para el equipo de cómputo como para el software. Ya que, los requerimientos como las necesidades de la institución van cambiando, el software, también, debe hacerlo. Y si un equipo no se encuentra en óptimas condiciones repercutirá en el desempeño del software.

## REFERENCIA BIBLIOGRÁFICA

1. Felipe Ramírez. Introducción a la Programación. 2da Edición. Estado de México (Toluca de Lerdo): Alfa omega; 2008.
2. Raghu Ramakrishnan, Johannes Gehrke, Fernando Sáenz Pérez. Sistemas de Gestión de Bases de datos. 3ra Edición. Estado de México: Mc Graw Hill Interamericana; 2006.
3. Sergio Matsukawa Maeda. Análisis y Diseño Orientado a Objetos con UML y Rational Rose. 1ra Edición. Estado de México: Editorial Macro; 2004.
4. Luis Joyanes. Fundamentos de programación. 3ra Edición. Estado de México: Editorial Mac Graw Hill; 2007.
5. Sergio Matsukawa Maeda. Análisis y Diseño Orientado a Objetos con UML y Rational Rose. 5ta Edición. Estado de México: Editorial Macro; 2009.

## **ANEXOS:**

## MATRIZ DE CONSISTENCIA

**TITULO: Software de control documentario para el mejoramiento de la gestión administrativa en la Universidad Autónoma de Ica, chinchá – 2015**

PROBLEMA	OBJETIVOS	HIPÓTESIS	VARIABLES Y SUBVARIABLES	INDICADORES
<p><b>Problema Principal:</b></p> <p>¿El Software de control documentario beneficiara para el mejoramiento de la gestión administrativa en la Universidad Autónoma de Ica de Chinchá - 2015?</p> <p><b>Problemas Secundarios:</b></p> <p><b>P<sub>1</sub>.</b> ¿El Software de control documentario a implantar ayudara a automatizar y optimizar la gestión documental en la Universidad Autónoma de Ica?</p> <p><b>P<sub>2</sub>.</b> ¿Ayudara el Software de control documentario en obtener un informe en tiempo real del estado de sus trámites realizados en el Universidad a los estudiantes?</p> <p><b>P<sub>3</sub>.</b> ¿El Software de control documentario beneficiara para disponer reportes estadísticos del flujo de todos los documentos en el área?</p> <p><b>P<sub>4</sub>.</b> ¿El Software de control documentario ayudara a minimizar el tiempo de los trámites documentarios?</p> <p><b>P<sub>5</sub>.</b> ¿El Software de control documentario ayudara a mejorar el control de la gestión administrativa?</p>	<p><b>Objetivo General:</b></p> <p>Implementar un software de control documentario para el mejoramiento de la gestión administrativa... en la Universidad Autónoma de Ica, Chinchá - 2015.</p> <p><b>Objetivos Específicos:</b></p> <p><b>O<sub>1</sub>.</b> Automatizar y optimizar la Gestión administrativa en la Universidad Autónoma de Ica.</p> <p><b>O<sub>2</sub>.</b> Obtener un seguimiento en tiempo real del estado de los trámites realizados en el Universidad a los estudiantes.</p> <p><b>O<sub>3</sub>.</b> Disponer de reportes estadísticos del flujo de documentos en el área.</p> <p><b>O<sub>4</sub>.</b> Minimizar el tiempo de los trámites documentarios.</p> <p><b>O<sub>5</sub>.</b> Mejorar el control de la gestión administrativa.</p>	<p><b>Hipótesis General:</b></p> <p>El software de control documentario beneficiará en el mejoramiento de la Gestión Administrativa Documentaria en la Universidad Autónoma de Ica, Chinchá - 2015.</p> <p><b>Hipótesis Específicas:</b></p> <p><b>H<sub>1</sub>.</b> Ayudara en automatizar y optimizar la Gestión administrativa en la Universidad Autónoma de Ica.</p> <p><b>H<sub>2</sub>.</b> Realizar un informe en tiempo real del estado de los trámites realizados en el Universidad beneficiara a los estudiantes para tener de conocimiento como va su documento.</p> <p><b>H<sub>3</sub>.</b> Se podrá disponer de reportes estadísticos del flujo de los documentos en el área en tiempo real beneficiara a tener un mejor control.</p> <p><b>H<sub>4</sub>.</b> Se podrá Minimizar el tiempo de los trámites documentarios en el área.</p> <p><b>H<sub>5</sub>.</b> Podrá Mejorar el control de la gestión administrativa.</p>	<p><b>Variables de la Investigación.</b></p> <p><b>Variable Independiente:</b></p> <p>V<sub>1</sub>- Software / Sistema Informático.</p> <p><b>Variable Dependiente:</b></p> <p>V<sub>2</sub>- Mejoramiento de la Gestión Administrativa.</p>	<p><b>Para la Variable Independiente:</b></p> <p>a) Pagos b) Titulación c) Solicitudes d) Cartas</p> <p><b>Para la variable Dependiente</b></p> <p>a) Pagos de Matrícula b) Titulación Grado Universitario. c) Solicitud de Apoyo Económico. d) Cartas de Presentación para Practicas Pre-Profesionales.</p>

**Minimizar tiempo del trámite - Mejor control del trámite documentario – Tener en tiempo real el documento en el área requerida**



Chincha Alta, 15 de Mayo del 2013

Señorita:

**CARMEN INÉS PALOMINO CASTILLA**

**Jefatura de OCTA**

Presente.-

Para informarle de los requerimientos estudiados al ser realizado el primer levantamiento de información por medio de una entrevista escrita y a su vez la explicación del formato de documento realizado en Microsoft Office Excel, en el cual lleva la administración de los trámites documentarios que llegan al Área de OCTA, realizado a la Srta. Carmen Inés Palomino Castilla.

Los requerimientos en estudio son:

General:

- ✓ Indicar en el sistema persona quien emite y recepciona el documento.
- ✓ Indicador límite de tiempo de respuesta.
- ✓ Sistema amigable, de sencillo entendimiento.

Alerta:

- ✓ Indicar de Alerta de vencimiento de plazo de entrega de documento.  
**NOTA:** Tiempo 2 días antes del vencimiento.
- ✓ Extensión de tiempo de entrega de 24 horas después del límite de tiempo para la respuesta. *(con opción a modificar)*

Reportes:

- ✓ Reporte de documentos **NO ENTREGADOS** al área después de finalizar la extensión de 24 horas después del límite de tiempo de respuesta.
- ✓ Reporte de documentos por área/encargado por rango de fecha (Fecha de Inicio y Fin) y muestra del estado del trámite del documento en cuestión.
- ✓ Reporte de lista de documentos por rango de fecha (Fecha de Inicio y Fin).

Si desea agregar algún otro requerimiento visto por su persona o no se halla mencionado, lo puede escribir a continuación:

Reporte de documentos Aprobados; Reporte de documentos  
Pendientes; Reporte de documentos que ya han sido  
entregados al solicitante.

Firma de la Jefa de OCTA

**Entrevista N° 01 - OCTA**

Entrevistado: Evelyn Malagos Escobedo Mamani  
Cargo: Proctuante de Octa - Fecha: 08/05/2013  
Entidad: \_\_\_\_\_

---

---

1. ¿De qué se encarga su Área de trabajo y que funciones desempeña Usted en el área?

*Se encarga de recepcionar los documentos (oficios, Constancias y solicitudes).*

*- Yo me encargo de hacer llegar los documentos a distintos áreas administrativas.*

2. ¿Cómo se está desarrollando actualmente el proceso y control de Trámites Documentario en el Área? Explique

*- De manera inmediata.*

3. ¿Usted maneja algún tipo de formato o documento para llevar el control y seguimiento del Trámite Documentario en el área? Describa y adjunte documento.

*- Cuadernos de campo.*



4. ¿Qué complicaciones Usted encuentra en el proceso de Tramite Documentario como actualmente se está realizando? Explique.

Ninguna, pero si sería posible que brinden un sistema en el área, sería mejor el proceso de trámite.

5. ¿Piensa que es requerido un Sistema de Gestión y Administración para Tramite Documentario y que beneficio usted cree que le brindaría este sistema para su trabajo y área? ¿Por qué?

Si es muy requerido un sistema de gestión y Administración ya que sería más fácil el requerimiento documentario.

6. ¿Cómo le gustaría que fuese el manejo de este Sistema Integrado de Gestión, que tipo de consultas desearía y que tipo de reportes desearía observar? Detalle.

Que detalle todo lo requerido como, datos personales, hacia quien va dirigido el documento, a que área va a ser dirigido. cuando a dicho desarrollo dichos documentos.

Eduardo Muñoz

Señor (es): Lucía Sobrino Mejía cargo N° 1027  
 Remitimos con el portador: Informe Economía N° 71  
original - Laurente Mejía Luis  
Mariano.  
 Entregué:  Fecha: 10/ MAR 2013 Recibi Conforme:   
 Hora: \_\_\_\_\_

Señor (es): Gloria Quispe Torres cargo N° 1048  
 Remitimos con el portador: Reunión de trabajo - consejo  
Directivo - universidad Saludables -  
JUAN Huber Mallme Torres  
 Entregué:  Fecha: 10/ MAR 2013 Recibi Conforme: \_\_\_\_\_  
 Hora: \_\_\_\_\_

Señor (es): Lucía Sobrino Mejía cargo N° 1179  
 Remitimos con el portador: Tramite de título -  
Espinosa Cordova Gabriela - Informe  
Economico  
 Entregué:  Fecha: 10/ MAR 2013 Recibi Conforme:   
 Hora: \_\_\_\_\_

Señor (es): Mesely Rojas Castilla cargo N° 1163  
 Remitimos con el portador: Institución a la ceremonia  
actividades por el 8 marzo internacional de la mujer  
 Entregué:  Fecha: 10/ MAR 2013 Recibi Conforme: \_\_\_\_\_  
 Hora: \_\_\_\_\_

Señor (es): Juice Sobrino Mejía cargo N° 1211  
 Remitimos con el portador: Tramite de título -  
Apestesma Bamacho Natalia - Informe  
Economico  
 Entregué:  Fecha: 10/ MAR 2013 Recibi Conforme:   
 Hora: \_\_\_\_\_

Señor (es): Wilfredo Navarrete cargo N° 1219  
 Remitimos con el portador: Invitaciones a participar en coloquio.  
Emprendimiento y creatividad - Asistir con  
alumnos - Walter Conneso.  
 Entregué:  Fecha: 11/ MAR 2013 Recibi Conforme:   
 Hora: \_\_\_\_\_

Señor (es): Magaly Rojas cargo N° 1214  
 Remitimos con el portador: Invitación a participar en  
coloquio: Emprendimiento y creatividad - Asistir  
con alumnos - Walter Conneso  
 Entregué:  Fecha: 11/ MAR 2013 Recibi Conforme:   
 Hora: \_\_\_\_\_

Señor (es): Ciro Hernandez cargo N° 1228  
 Remitimos con el portador: X Programa de formación de  
Analista de créditos - Difundir a  
egresados  
 Entregué:  Fecha: 11/ MAR 2013 Recibi Conforme:   
 Hora: \_\_\_\_\_



## Entrevista N° 01 - OCTA

Entrevistado: Palomino Castilla Carmen Inés  
Cargo: Jefatura de la OCTA Fecha: 09-05-2013  
Entidad: \_\_\_\_\_

---

---

1. ¿De qué se encarga su Área de trabajo y que funciones desempeña Usted en el área?

Se encarga de recepcionar todo tipo de documento (cartas, informes, oficios, solicitudes, invitaciones, etc) que esté dirigido a la institución sea emitido por público en general o por el personal administrativo.

Funciones:

- \* Realizar la recepción de documentos.
- \* Registrar y escanear los documentos recepcionados.
- \* Emitir los documentos a las respectivas áreas.
- \* Dar seguimiento a los trámites que necesitan respuesta.

2. ¿Cómo se está desarrollando actualmente el proceso y control de Trámites Documentario en el Área? Explique

Se desarrolla de forma manual, ya que para la entrega de los documentos a los diferentes áreas se utiliza cuadernos de cargos. y para el registro de los documentos recepcionados se está utilizando un archivo de excel.

3. ¿Usted maneja algún tipo de formato o documento para llevar el control y seguimiento del Trámite Documentario en el área? Describa y adjunte documento.

Si, se utiliza el programa de Microsoft Office Excel para el registro y seguimiento.  
Se adjunta modelo.



4. ¿Qué complicaciones Usted encuentra en el proceso de Tramite Documentario como actualmente se está realizando? Explique.

Mayormente las complicaciones se presentan cuando un documento requiere respuesta, ya que este depende de otra área, y muchas veces no se cumple en el plazo que se les da.

5. ¿Piensa que es requerido un Sistema de Gestión y Administración para Tramite Documentario y que beneficio usted cree que le brindaría este sistema para su trabajo y área? ¿Por qué?

Si, porque ayudaría a aumentar la eficiencia en el seguimiento de los trámites, y así disminuiría las insatisfacciones e incomodidades de las personas solicitantes.

6. ¿Cómo le gustaría que fuese el manejo de este Sistema Integrado de Gestión, que tipo de consultas desearía y que tipo de reportes desearía observar? Detalle.

- 1º Debe indicar a qué área se emiten los documentos, fecha y proveedor.
- 2º Si algún documento requiere de respuesta, debe de existir el límite de plazo.
- 3º Alertar el vencimiento de plazo para emisión de respuesta, de parte de otras áreas.
- 4º Señalar cuando la respuesta del documento ya fue entregado a la persona solicitante.

Firma del Entrevistado

Adjunto de Pregunta N:3

# RECEPCION DE

1ro

Nº EXPED.	FECHA DE RECEPCION	Apellidos y Nombres / Razón Social Remitente	Tipo de Doc.	Asunto	Dirigida a la Empresa	Destino/ Según documento Recepcionado	Tiempo de Trámite	Limite de Plazo	Tiempo de Demora	se Queda en la OCTA para entrega-fecha	Fecha de Entrega al Area
001140	Lunes, 04 de Marzo de 2013	GUTIERREZ PACHAS JUAN	SOLICITUD	SILABUS	INST	ECON. JORGE CAMPOS MARTINEZ	15	25/03/2013	-	20/03/2013	04/03/2013
001141	Lunes, 04 de Marzo de 2013	ROJAS CASTILLA MAGALY	OFICIO N°007-2013-CO/OBS/UPICA	HORARIO DEL MES DE MARZO	UPICA	DR. MARTIN CAMPOS	-	-	-	-	04/03/2013
001142	Lunes, 04 de Marzo de 2013	ROJAS CASTILLA MAGALY	CARTA	UTILES DE OFICINA	UPICA	MARIA ELENA CLAROS AZCONA	-	-	-	-	04/03/2013
001143	Lunes, 04 de Marzo de 2013	YATACO TORRES JUAN GABRIEL	SOLICITUD	CARTA -CARPETA DE PRACTICAS PRE-PROFESIONALES	UPAB	MAG. ROBERTO BUENDIA	2	06/03/2013	0	06/03/2013	04/03/2013
001144	Lunes, 04 de Marzo de 2013	LLAZACA BRAVO ROSARIO	CARTA	CRONOGRAMA PERIODO 2013-I	UPAB	JAVIER MORA	-	-	-	-	-
001145	Lunes, 04 de Marzo de 2013	CHUMPITAZ MARCO MARIA	SOLICITUD	FEDATEO DE RESOLUCION	UPICA	DR. MARTIN CAMPOS	2	06/03/2013	0	04/03/2013	04/03/2013
001146	Lunes, 04 de Marzo de 2013	MATTOS MARMANILLO CESAR	SOLICITUD	CAMBIO DE CARRERA	UPAB	DR. MARTIN CAMPOS	-	-	-	07/03/2013	04/03/2013
001147	Lunes, 04 de Marzo de 2013	TASAYCO SARAVIA RONALD	SOLICITUD	CARTA-CARPETA DE PRACTICAS PRE-PROFESIONALES	UPAB	MAG. ROBERTO BUENDIA	2	06/03/2013	0	06/03/2013	04/03/2013
001148	Lunes, 04 de Marzo de 2013	CONDE CARDENAS CARLOS	SOLICITUD	CONSTANCIA DE CREDITOS	UPAB	ING. MAGDALENA TALLA	7	13/03/2013	0	07/03/2013	04/03/2013
001149	Lunes, 04 de Marzo de 2013	MUCHAYPIÑA TORRES LUIS	CARTA	ADELANTO DE VACACIONES POSTERIORES	GRUPO SANTA FE	DR. MARTIN CAMPOS	-	-	-	-	04/03/2013
001150	Lunes, 04 de Marzo de 2013	INGA FERNANDEZ STEFFI	SOLICITUD	CARTA-CARPETA DE PRACTICAS PRE-PROFESIONALES	UPAB	MAG. ROBERTO BUENDIA	2	06/03/2013	-	06/03/2013	04/03/2013
001151	Lunes, 04 de Marzo de 2013	CRISOSTOMO GUILLEN VICTOR YOSHI	SOLICITUD	CARTA-CARPETA DE PRACTICAS PRE-PROFESIONALES	UPAB	MAG. ROBERTO BUENDIA	2	06/03/2013	-	06/03/2013	04/03/2013
001152	Lunes, 04 de Marzo de 2013	HERNANDEZ DONAYRE SHEYLA	SOLICITUD	REINCORPORACION A LA UNIVERSIDAD -UPICA	UPICA	DR. MARTIN CAMPOS	-	-	-	-	04/03/2013
001153	Lunes, 04 de Marzo de 2013	LLAZACA BRAVO ROSARIO	CARTA	AUTORIZACION DE LA COMPRA DE 15 DIPLOMAS DE BACHILLER	UPAB	DR. MARTIN CAMPOS	-	-	-	-	04/03/2013
001154	Martes, 05 de Marzo de 2013	BENDEZU PALOMINO OSCAR	SOLICITUD	SILABUS DE LOS CURSOS	UPICA	DR. MARTIN CAMPOS	7	14/03/2013	0	05/03/2013	05/03/2013
001155	Martes, 05 de Marzo de 2013	BENDEZU PALOMINO OSCAR	SOLICITUD	C CERTIFICADO DE ESTUDIOS	UPICA	DR. MARTIN CAMPOS	7	14/03/2013	0	05/03/2013	05/03/2013
001156	Martes, 05 de Marzo de 2013	SIGUAS MENESES VICTOR	SOLICITUD	CONSTANCIA DE CREDITOS	UPAB	ING. MAGDALENA TALLA	7	14/03/2013	0	11/03/2013	06/03/2013
001157	Martes, 05 de Marzo de 2013	GUERRERO CARDENAS MARIA	SOLICITUD	TRASLADO EXTERNO	UPICA	DR. MARTIN CAMPOS	-	-	-	-	05/03/2013





UNIVERSIDAD PRIVADA  
"Ada A. Byron" S.A.C.

RESOLUCIÓN Nº 136-2006-CONAFU

*Mejores Personas, Excelentes Profesionales...!!!*

**REFERENCIA HISTÓRICA DE LA UNIVERSIDAD PRIVADA  
"ADA A . BYRON"**

El Proyecto de Creación de la Universidad Privada "Ada A. Byron" se inicia en el año 2000 Cuando el Ingeniero Martín Campos Martínez promotor de la Organización Educativa "Ada A. Byron" decide complementar una brillante trayectoria de trece años en el campo de la Educación Inicial, Primaria, Secundaria y Superior No Universitaria, con la incorporación Del nivel Universitario, a través de la propuesta de una Universidad Chinchana que responda a las demandas y necesidades propias de la Región Ica.

Luego de más de tres años de reflexión académica, estudios de mercado, reformulación y actualización del proyecto y en estricto cumplimiento de la normatividad del Consejo Nacional para la Autorización y Funcionamiento de Universidades CONAFU, el 03 de Febrero del 2005 se presentó el Proyecto de Desarrollo Institucional ante el CONAFU. Mediante Resolución Nº 092-2005-CONAFU del 15 de Marzo del 2005 se admite a trámite nuestro PDI, con las Carreras Profesionales de Administración y Finanzas; Enfermería; Ingeniería de Sistemas; Psicología e Ingeniería de Industrias Alimentarias.

Con fecha 23 de Diciembre del 2005 se aprueba el Proyecto de Desarrollo Institucional de la Universidad y con fecha 29 de mayo de 2006 se otorga Autorización de Funcionamiento a nuestra casa superior de estudios mediante la resolución Nº 136-2006-CONAFU.

Es así que la Universidad Privada "Ada A. Byron" empieza a organizar sus actividades académicas y administrativas, a efecto de brindar sus servicios educativos a la comunidad Chinchana y a la población de la Región Ica y su zona de influencia.

Luego de llevarse a cabo el Primer Concurso Público de Docentes para Contrato, en el mes de Marzo de 2006. Realizado el primer examen de Admisión, bajo la estricta supervisión del CONAFU y nombrados la integrantes de la Comisión Organizadora, la Universidad Privada "Ada A. Byron" fue inaugurada oficialmente el 14 de Julio de 2006, en acto público presidido por el Presidente del CONAFU Dr. Elio Delgado Azañero y con asistencia de personalidades del mundo Académico Nacional

Es así que en Julio del año 2006 la UPAB S.A.C. empieza desarrollar sus actividades académicas, la primera universidad propia de la ciudad de Chincha.

**VISION DE LA UPAB**

La Universidad Privada "Ada A. Byron" de Chincha, es un centro de estudios superiores Que tiene como finalidad crear, acrecentar y preservar el conocimiento a través de la investigación, transmitir el saber a través del proceso Enseñanza – Aprendizaje, e interactuar con la realidad Regional, Nacional e Internacional, liderando los cambios que requiere el ámbito de influencia de la UPAB. S.A.C.



UNIVERSIDAD PRIVADA  
"Ada A. Byron" S.A.C.

RESOLUCIÓN N° 136-2006-CONAFU

*Mejores Personas, Excelentes Profesionales...!!!*

Busca la formación de profesionales dentro de los alcances de la modernidad basado en la Ciencia, arte, ética y moral; con gran capacidad de abstracción. Aptos para la experimentación, el trabajo en equipo y con gran capacidad de adaptación al cambio; de igual manera con gran sensibilidad social.

**MISIÓN DE LA UPAB :**

La Universidad Privada "Ada A. Byron" tiene el compromiso de una sólida formación esencial humana descubriendo y fortaleciendo el liderazgo personal acompañado de una formación exigentemente académica y tecnológica.

La Universidad tiene el compromiso permanente de aportar soluciones en su entorno próximo así como la región y el país.

La Universidad tiene el compromiso de realizar actividades de proyección social permanente y preferentemente en las zonas marginales y de extrema pobreza en la provincia de Chincha y en la región.

La Universidad está comprometida con el aporte a la Cultura, el deporte, el folklore y toda manifestación popular.

Como toda Universidad en el Perú en sus primeros años está dirigida por una Comisión Organizadora, la nuestra la integran:

**Dr. DAVID VÍCTOR PORTILLO POMALAZA**

**PRESIDENTE**

Médico de Profesión

Doctor en Medicina

Profesor Principal de la Universidad Nacional Mayor de San Marcos

**Dr. DOMINGO PERCY ORTEGA CHACÓN**

**VICEPRESIDENTE**

Sociólogo de Profesión

Doctor en Sociología

Profesor Principal de las Universidades Privadas Ricardo Palma e

Inca Garcilaso de la Vega

Profesor Principal de las Universidades Nacionales Federico Villareal y

Enrique Guzmán y Valle

**Dr. HERNANDO MARTÍN CAMPOS MARTÍNEZ**

**GERENTE GENERAL**

Ingeniero Químico de Profesión

Doctor en Educación

Gerente General de diversas empresas

Profesor Principal de la Universidad Privada "Ada A. Byron"

La Carrera Profesional de **ENFERMERÍA**, que hoy nos representan en esta festividad educativa cuenta con docentes de amplia trayectoria académica y profesional como;

Mg. Gianina Farro Peña Directora de la Facultad de Enfermería de la Universidad Cayetano Heredia

Mg. Tula Espinoza Moreno Directora de la Escuela de Enfermería de la Universidad Nacional Mayor de San Marcos y Presidenta de la Sociedad Peruana de Salud Pública Y Comunitaria.



UNIVERSIDAD PRIVADA

**"Ada A. Byron" SAC**

RESOLUCIÓN Nº 136-2006-CONAFU

*"Mejores personas, Excelentes profesionales"*

### BREVE REFERENCIA HISTÓRICA DE NUESTRA UNIVERSIDAD

El Proyecto de Creación de la Universidad Privada Ada A. Byron se inicia aproximadamente en el año 2000, cuando el empresario Martín Campos Martínez, dueño de la Organización Educativa **Ada A. Byron** decide complementar una brillante trayectoria de trece años en el campo de la Educación Inicial, Primaria, Secundaria y Superior no Universitaria, con la incorporación del Nivel Universitario, a través de la propuesta de una Universidad Chinchana que responda a las demandas y necesidades propias de la Región Ica..

Luego de más de tres años de reflexión académica, estudios de mercado, reformulación y actualización del proyecto, y en estricto cumplimiento a la normatividad del CONAFU, el 03 de febrero del 2005 se presentó el Proyecto de Desarrollo Institucional ante el Consejo Nacional para la Autorización y Funcionamiento de Universidades.

Mediante Resolución NQ092-2005- CONAFU del 15 de Marzo del 2005 se admite a trámite nuestro PDI, con las Carreras Profesionales de Administración y Finanzas, Enfermería, Ingeniería de Sistemas, Psicología e Ingeniería de Industrias Alimentarias.

Con fecha 22 de diciembre del 2005 se aprueba el Proyecto de Desarrollo Institucional de la Universidad con la Resolución Nº 249-2005-CONAFU , y con fecha 29 de mayo del 2006, mediante Resolución N 136-2006-CONAFU se otorga Autorización de Funcionamiento a nuestra casa superior de estudios.

Es así que la Universidad **Ada A. Byron** empieza a organizar sus actividades académicas y administrativas, a efecto de brindar sus servicios educativos a la comunidad chinchana y a la población de la Región Ica y su zona de influencia. Luego de llevarse a cabo el Primer Concurso Público de Docentes para Contrato en el mes de marzo del 2006. realizado el primer examen de Admisión. bajo la estricta supervisión del CONAFU y nombrados los integrantes de la Comisión Organizadora. la Universidad Privada **Ada A. Byron** fue inaugurada el día 14 de julio del 2006. en acto público presidido por el Presidente del CONAFU Dr. Elio Delgado Azañero y con la asistencia de personalidades del mundo Académico Nacional.

Es así que a mediados del año 2006 la UPAB SAC empieza a desarrollar sus actividades académicas. la primera universidad propia de la ciudad de Chincha. En sus tres primeros años de funcionamiento nuestra casa superior de estudios ha logrado la aceptación y reconocimiento de la comunidad Chinchana. gracias al esfuerzo por brindar una educación de calidad y contribuir a la formación integral de la juventud de la Región Ica y su zona de influencia.

#### 1.2 VISIÓN DE LA UPAB

La Universidad Privada **Ada A. Byron** de Chincha. es un centro de estudios superiores en el que se promueve la creatividad y la transmisión del conocimiento para el bien de la humanidad. Se consagra a la enseñanza. la investigación. La proyección social y a la formación integral de los estudiantes. libremente reunidos con sus docentes. animados todos por el mismo amor y compromiso con el saber.

La formación profesional que brinda la UPAB SAC integra el conocimiento científico actualizado con la formación ética y humanística de los estudiantes, distinguiéndose por la calidad de sus servicios educativos y su permanente respuesta a las necesidades y desafíos que plantean el contexto regional. nacional e internacional a las nuevas generaciones.



UNIVERSIDAD PRIVADA

"Ada A. Byron" SAC

RESOLUCIÓN Nº 136-2006-CONAFU

"Mejores personas, Excelentes profesionales"

### 1.3 MISIÓN INSTITUCIONAL

La Universidad Privada Ada A. Byron S.A.C. es una institución de Educación Superior con excelencia académica y administrativa. inspirada en principios éticos. líder en la formación integral de profesionales de calidad. a quienes brindan una preparación que armoniza la visión de conjunto. la especialización. el saber interdisciplinario. el conocimiento de la realidad actual y la práctica de valores.

Nuestra Universidad es una organización educativa que desarrolla las dimensiones: intelectual. física. moral y social de sus alumnos; buscando a través del aprendizaje académico. la investigación y las actividades de proyección social que los estudiantes egresados de esta casa superior de estudios alcancen su realización personal y contribuyan eficazmente a la solución de los problemas fundamentales de su región y al desarrollo de la sociedad peruana.

### 1.4 FINES FUNDAMENTALES

La Universidad Privada Ada A. Byron S.A.C. asume como fines fundamentales:

- ✚ Estar a la vanguardia de la formación universitaria en su zona de influencia y ser reconocida a nivel nacional e internacional por liderar la formación integral de los seres humanos.
- ✚ Formar profesionales que se distinguen por su calidad académica, creatividad. respeto a los valores. solidaridad y compromiso con el desarrollo regional y nacional.
- ✚ Implementar actividades de proyección y extensión social (Seminarios. Talleres. Mesas Redondas. Forums u otros eventos) que permitan difundir los conocimientos generados en el contexto universitario a la población en general y fundamentalmente a las personas que se encuentra en situación de pobreza.
- ✚ Brindar a los estudiantes un conjunto de servicios que complementen su formación básica: deporte. folklore. cursos de desarrollo humano. servicio social. asesoría psicológica. tutoría. entre otros.
- ✚ Poseer una política de pensiones escalonadas. becas y estímulos que permitan que jóvenes de escasos recursos económicos pero de un alto rendimiento académico puedan acceder a nuestra institución.
- ✚ Realizar investigación científica. humanística y tecnológica, cuyos resultados se vinculen tanto a la enseñanza como a la proyección social y se orienten a resolver los problemas locales. regionales y nacionales.
- ✚ Desarrollar el área de producción de bienes y servicios como un espacio de aprendizaje práctico para los estudiantes de nuestra universidad y una fuente de generación de ingresos.
- ✚ Crear. incentivar y difundir cultura. conocimiento y tecnología que respondan a las necesidades de la comunidad y que orienten el desarrollo sostenido de la Región Ica.
- ✚ Desarrollar acciones conjuntas con las diversas instituciones productivas, sociales, religiosas, los gobiernos municipales a efecto de aunar esfuerzos en la solución de problemas sociales como la violencia familiar. La pobreza. los bajos niveles educativos. la salud. la discriminación.

### 1.5 PRINCIPIOS BÁSICOS

Los principios fundamentales que guían a nuestra comunidad universitaria son:

- Búsqueda constante de la verdad.
- Respeto por la dignidad de los seres humanos.
- Compromiso con la solución de los problemas socioeconómicos y el desarrollo nacional.
- Tolerancia y respeto por la diversidad.

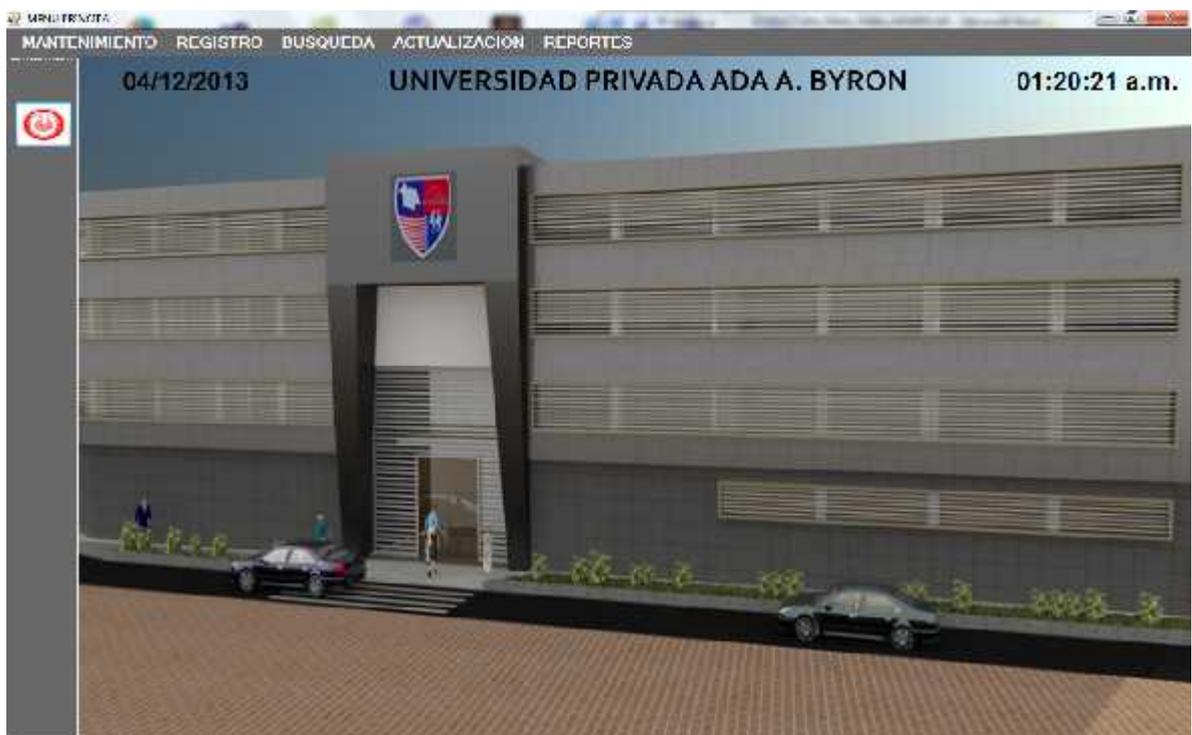
## MANUAL DE USUARIO

Bienvenido al manual de usuario para el manejo del sistema de gestión administrativa SIGTRAD v.1.0. De cómo es el manejo y la interacción con el sistema. A continuación proseguiremos con la explicación del sistema en su manejo:

**1er:** Permite acceder a la aplicación, verificando los usuarios registrados por medio de la interface que puede apreciar a continuación.



**2do:** Ingresado al Sistema (usuario y contraseña) se activara y mostrara la interface principal para el manejo de correspondiente de la persona encargada (mantenimiento, consultas, modificaciones y reportes.)



3er: Al ingresar a la 2da opción de REGISTRO seleccionamos la primera opción y nos mostrara esta interface como podemos apreciar, vamos a realizar el llenado e ingreso de los datos necesarios para el registro del trámite documentario.

The screenshot shows a web application window titled 'REGISTRO DE TRAMITES'. The interface is divided into several sections for data entry:

- CLIENTE:** Includes fields for 'Nombres', 'A. Paterno', 'A. Materno', and 'D.N.I.', each with a green checkmark icon.
- EXPLORACION:** Includes a field for 'Nº Expediente' with a green checkmark icon.
- INSTITUCION:** Includes a field for 'Institucion' with a green checkmark icon.
- USUARIO:** Includes a field for 'Usuario' with the value 'hiana'.
- AREA Y TRAMITE:** Includes a dropdown for 'Derivado a' (set to 'Gerencia') and a field for 'Tramite'.
- RANGO DE TRAMITE:** Includes date fields for 'Fecha Ingreso' (04/12/2011) and 'Fecha Salida' (04/12/2011).

At the bottom, there are three buttons: 'Nuevo', 'Guardar', and 'Salir'.

4to: Al dirigiarnos a MANTENIMIENTO y escogemos USUARIO nos mostrara el siguiente formulario que es para el manejo de las cuentas y los privilegios que tiene cada uno dentro al sistema que se está manejando.

The screenshot shows a web application window titled 'REGISTRO USUARIO'. The interface includes the following elements:

- Form Fields:** 'Codigo', 'Nombre', 'Apellido Paterno', 'Apellido Materno', 'Estado', 'Fecha', 'Clave', and 'Usuario'.
- Buttons:** 'Agregar', 'Nuevo', and 'Permisos'.
- Permissions Table:** A table with columns 'Acciones/Permisos' and 'Estado'. The 'Administración' row is selected and has its checkbox checked. Other rows include 'Reserva', 'Actualización', 'Consulta', and 'Reporte'.

At the bottom, there are three buttons: 'Nuevo', 'Permisos', and 'Salir'.

5to: Aquí podemos ver las derivaciones y lo que registran los procesos en las diferentes áreas para tener un control del expediente y su paso por las diferentes áreas.

Derivaciones

## DERIVACIONES

Expediente y Tramite

Nº Expediente  Tramite  

Derivar a:

Completo

Entregado

Correccion

RUTA DEL INFORME

Agregar Informe 

Descripcion

Fecha  Usuario

 Nuevo  Guardar  Salir

6to: Aquí podemos realizar el filtrado de consulta o búsqueda por medio de 2 opciones por medio de DNI o en caso contrario por Datos del Cliente (apellidos y nombres)

Busquedaexp

## BUSQUEDA DE EXPEDIENTES

BUSCAR POR:

Nº de DNI

Nombres y Apellidos

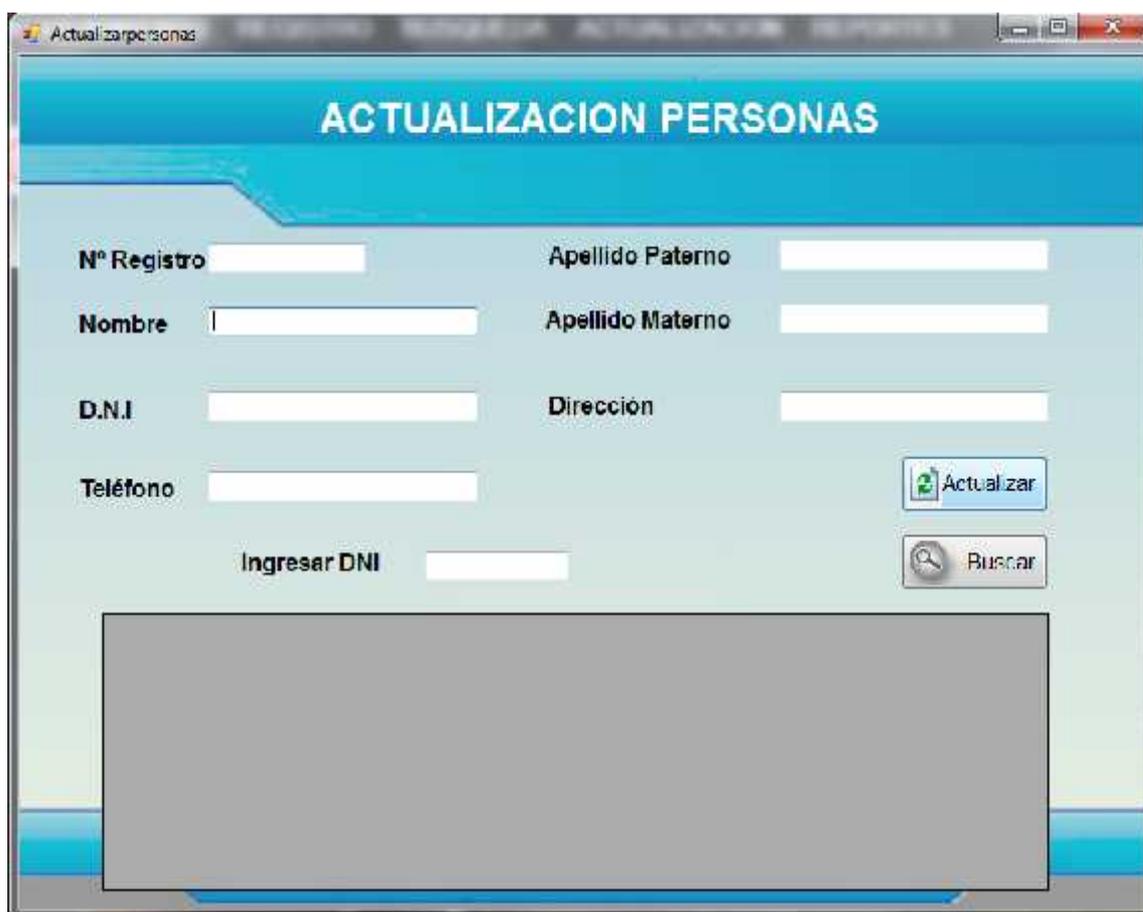
 Buscar

7mo: Aquí podemos realizar el filtrado de consulta o búsqueda por medio de número de expediente del trámite en consulta y nos arrojará o nos mostrará los datos del alumno interesante.



The screenshot shows a web browser window titled "Busquedatram". The main heading is "BUSQUEDA DE TRAMITES". Below the heading, there is a search section with the label "BUSCAR POR:" and a dropdown menu currently set to "Nº De Expediente". To the right of the dropdown is a text input field. Further right is a button with a magnifying glass icon and the text "Buscar". Below the search section is a large, empty grey rectangular area, likely a placeholder for search results.

8vo: Aquí podemos realizar el filtrado para la actualización de los datos de las personas tanto de los encargados como los clientes por medio del filtrado de DNI para actualizar los datos necesarios.



The screenshot shows a web browser window titled "Actualizarpersonas". The main heading is "ACTUALIZACION PERSONAS". The form contains several input fields: "Nº Registro", "Nombre", "D.N.I.", "Teléfono", "Apellido Paterno", "Apellido Materno", "Dirección", and "Ingresar DNI". There are two buttons: "Actualizar" (with a refresh icon) and "Buscar" (with a magnifying glass icon). Below the form is a large, empty grey rectangular area, likely a placeholder for updated data or search results.

9no. Todo lo anteriormente mencionado es parte del sistema pero lo esencial son los reportes que se podrá ver a continuación.

1er. Reporte nos puede mostrar la ubicación del documento actualmente

04/12/2013 01:31:03am Universidad Privada Adz. A. Uxon

### UBICACION DE EXPEDIENTES POR AREA

**Gerencia**

**Apoyo Economico**

FECHA	TRAMITE	FECHA DE ENTREGA	DESCRIPCION
01/11/2012	Ortiz Hernandez Silvana	02/11/2012 00:21:00am	Modificacion de Pago de Cuota
01/11/2012	Morales Hernandez Melzer	02/11/2012 00:21:00am	Modificacion de Pago de Cuota
02/11/2012	Ortiz Hernandez Silvana	02/11/2012 02:00:00am	
02/11/2012	Morales Hernandez Melzer	02/11/2012 02:00:00am	

2do. Reporte nos muestra el historial de documentos que son recepcionado y emitido a las diversas áreas de la universidad.

Ingresar Expediente

04/12/2013 01:31:03am Universidad Privada Adz. A. Uxon

### HISTORIAL DE EXPEDIENTES

**Bienestar Universitario**

CLIENTE	TRAMITE	FECHA DE ENTREGA	DESCRIPCION
Ortiz Hernandez Silvana	Apoyo Economico	02 oct 2012 20:56	Documentacion de ingreso
Morales Hernandez Melzer	Apoyo Economico	02 oct 2012 21:50	Documentacion de ingreso

**Gerencia**

CLIENTE	TRAMITE	FECHA DE ENTREGA	DESCRIPCION
Ortiz Hernandez Silvana	Apoyo Economico	26 nov 2012 0:00	
Morales Hernandez Melzer	Apoyo Economico	02 nov 2012 0:00	
Ortiz Hernandez Silvana	Apoyo Economico	02 nov 2012 0:00	Modificacion de Pago de Cuota
Morales Hernandez Melzer	Apoyo Economico	17 nov 2012 15:21	Modificacion de Pago de Cuota

3er. Reporte nos muestra un historial mucho mas detallado por área y asunto de cual trata el tramite documentario que se requiere.

04/02/2013 04:56:27am Universidad Privada AdA. Titon

### HISTORIAL POR AREAS

**Admisión**  
Cambios Cursum

EXPEDIENTE	CLIENTE	FECHA DE INGRESO
0	OSCAR CARLOS ANDRÉS	03/12/2013 08:52:00a.m.

---

**Oficina Académica**  
Prácticas Profesionales

EXPEDIENTE	CLIENTE	FECHA DE INGRESO
2	Karen Teresa Domínguez	03/12/2013 09:46:00a.m.

**Gerencia**  
Apoyo Personal

EXPEDIENTE	CLIENTE	FECHA DE INGRESO
1	Omar Hernandez Silva	04/12/2013 09:08:00a.m.
1	Marcos Hernandez Silva	07/12/2013 09:59:00a.m.

**Oficina Integración Pagos**

EXPEDIENTE	CLIENTE	FECHA DE INGRESO
2	Freda Patricia Sanchez	04/12/2013 09:08:00a.m.

**Recurrido**

PF: 04/12/2013 09:56:27am 1/1 total de páginas: 1 Fecha de acceso: 2013