

INSTITUTO TECNOLÓGICO SUPERIOR  
CORDILLERA

ESCUELA DE SISTEMAS

Proyecto de Grado, previa obtención del título de:  
Tecnólogo Analista de Sistemas

TEMA

DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA  
BIOMÉTRICO PARA EL CONTROL DE ASISTENCIA EN INDESIC

AUTOR:

JHONI FRANKLIN SUQUI CHIMBO

TUTOR:

ING. ADRIÁN NIETO

2012

QUITO – ECUADOR

## **DECLARACIÓN DE AUTENTICIDAD**

Yo, Jhoni Franklin Suqui Chimbo, declaro bajo juramento que los contenidos y los resultados obtenidos en el presente proyecto, como requerimiento previo para la obtención del Título de Tecnólogo Analista de Sistemas, son absolutamente originales, auténticos y personales y de exclusiva responsabilidad legal y académica del autor.

**Jhoni Franklin Suqui Ch.**

0104654769

## **APROBACIÓN DEL TUTOR**

En mi calidad de Tutor del trabajo sobre el tema: “DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA BIOMÉTRICO PARA EL CONTROL DE ASISTENCIA EN EL INDESIC”, presentado por el ciudadano: Jhoni Franklin Suqui Chimbo, estudiante de la Escuela de Sistemas, considero que dicho informe reúne los requisitos y méritos suficientes para ser sometido a la evaluación por parte del Tribunal de Grado, que el Honorable Consejo de Escuela designe, para su correspondiente estudio y calificación.

**Quito, Abril del 2012**

Ing. Adrián Nieto

**TUTOR**

## **APROBACIÓN DEL TRIBUNAL DE GRADO**

Los miembros del Tribunal de Grado designado por el Honorable Concejo de la Escuela de Sistemas, aprueban el trabajo de investigación de acuerdo con las disposiciones reglamentarias emitidas por el Centro de Investigaciones Tecnológicas y Proyectos del Instituto Tecnológico Superior Cordillera” para proyectos de grado de Tecnólogos Analistas de Sistemas: del Sr: Jhoni Franklin Suqui Chimbo.

**Quito, Abril del 2012**

Para constancia firman:

**PRESIDENTE**

**VOCAL1**

**VOCAL2**

## **AGRADECIMIENTO**

Agradezco en primer lugar a Dios por haberme regalado la vida y por ser mi luz y mi guía, en cada uno de mis proyectos, Él ha puesto brindándome así la oportunidad de crecer espiritual y Profesionalmente.

A mi familia, por su apoyo, por ser mi ejemplo de lucha constante y de superación, por enseñarme que la honradez y la honestidad permiten conseguir sueños altos.

A los profesores de la Escuela de Sistemas que con sus conocimientos guiaron adecuadamente aprendizaje; porque supieron no sólo ser maestros si ni amigos.

Y por supuesto a mi tutor de tesis, Ing. Adrián Nieto, por su paciencia apoyo constante en este proyecto de investigación.

**Gracias**

## **DEDICATORIA**

Dedico este trabajo a mis padres Antonio & Herlinda que desde el cielo miran caminar y alcanzar una pequeña meta y parte de mi vida.

Jhoni Fanklin Suqui Ch.

## ÍNDICE DE CONTENIDO

Página

<b>CAPÍTULO I .....</b>	<b>12</b>
<b>TEMA:</b>	
1.1.    PLANTEAMIENTO DEL PROBLEMA. ....	12
1.2.    DELIMITACIÓN DE LA INVESTIGACIÓN. ....	13
1.3.    FORMULACIÓN DEL PROBLEMA. ....	13
1.4.    OBJETIVO GENERAL. ....	13
1.5.    OBJETIVOS ESPECÍFICOS. ....	13
1.6.    JUSTIFICACIÓN E IMPORTANCIA. ....	14
1.7.    ALCANCE. ....	15
<b>CAPÍTULO II .....</b>	<b>18</b>
<b>MARCO TEÓRICO.....</b>	<b>18</b>
<b>2.1.    LOS ANTECEDENTES. ....</b>	<b>18</b>
<b>2.2.    RESEÑA HISTÓRICA. ....</b>	<b>19</b>
<b>2.3.    MARCO REFERENCIAL.....</b>	<b>21</b>
2.3.1.    SISTEMA BIOMÉTRICO.....	21
2.3.2.    DISEÑO DE LA BASE DE DATOS:.....	21
2.3.3.    ADMINISTRADOR Y ALMACENAMIENTO DE LA BASE DE DATOS. ....	22
2.3.4.    DESARROLLO DE APLICACIÓN.....	24
2.3.5.    HARDWARE. ....	28
2.3.6.    SOFTWARE. ....	30
2.3.7.    KNOW HOW. ....	30
<b>2.4.    MARCO LEGAL. ....</b>	<b>31</b>
<b>CAPÍTULO III .....</b>	<b>37</b>
<b>INVESTIGACIÓN CIENTÍFICA.....</b>	<b>37</b>
<b>3.1.    TIPOS DE INVESTIGACIÓN.....</b>	<b>37</b>
3.1.1.    INVESTIGACIÓN CUANTITATIVA. ....	37
3.1.2.    INVESTIGACIÓN BIBLIOGRÁFICA DOCUMENTAL.....	37
3.1.3.    INVESTIGACIÓN DE CAMPO. ....	38
3.1.4.    INVESTIGACIÓN DESCRIPTIVA.....	38
3.1.5.    INVESTIGACIÓN EXPLICATIVA. ....	39

<b>3.2</b>	<b>MÉTODOS DE INVESTIGACIÓN. ....</b>	<b>39</b>
	MÉTODOS TEÓRICOS. ....	39
3.2.1.	MÉTODO INDUCTIVO. ....	39
3.2.2.	MÉTODO DEDUCTIVO. ....	39
3.2.3.	MÉTODO ANALÍTICO – SINTÉTICO. ....	40
3.2.4.	MÉTODO HIPOTÉTICO- DEDUCTIVO. ....	40
3.2.5.	MÉTODO HISTÓRICO LÓGICO. ....	40
<b>3.3.</b>	<b>TÉCNICAS DE RECOLECCIÓN DE INFORMACIÓN. ....</b>	<b>41</b>
3.3.1.	LA OBSERVACIÓN. ....	41
3.3.2.	LA ENTREVISTA. ....	41
3.3.2.1.	TIPOS DE ENTREVISTA. ....	42
	<b>CAPÍTULO IV. ....</b>	<b>44</b>
	<b>DESARROLLO DE LA PROPUESTA. ....</b>	<b>44</b>
4.1.	DIAGNÓSTICO SITUACIONAL. ....	44
4.2.	ESTRUCTURA ORGANIZACIONAL. ....	44
<b>4.3.</b>	<b>INFRAESTRUCTURA INFORMÁTICA. ....</b>	<b>45</b>
4.3.1.	HARDWARE. ....	46
4.3.2.	SOFTWARE. ....	46
4.3.3.	COMUNICACIONES. ....	47
4.3.4.	RECURSO HUMANO TÉCNICO. ....	48
<b>4.4.</b>	<b>DESCRIPCIÓN DE ALTERNATIVAS. ....</b>	<b>48</b>
4.4.1.	ALTERNATIVA # 1. ....	48
4.4.2.	ALTERNATIVA # 2. ....	50
4.4.3.	ALTERNATIVA # 3. ....	51
<b>4.5.</b>	<b>EVALUACIÓN Y SELECCIÓN DE ALTERNATIVAS. ....</b>	<b>53</b>
4.5.1.	ANÁLISIS E INTERPRETACIÓN DE LOS DE LOS RESULTADOS DE LA EVALUACIÓN. ....	57
<b>4.6.</b>	<b>FACTIBILIDAD TÉCNICA. ....</b>	<b>58</b>
<b>4.7.</b>	<b>DESCRIPCIÓN DE PROCESOS. ....</b>	<b>58</b>
4.7.1.	PROCESOS DEL SISTEMA. ....	58

	<b>Página</b>
<b>4.8. DESCRIPCIÓN DE METODOLOGÍA DE DESARROLLO. ....</b>	<b>76</b>
<b>4.9. MODELO CONCEPTUAL. ....</b>	<b>79</b>
<b>4.10. MODELO FÍSICO Y LÓGICO (DIAGRAMAS UML EN CASO TRES CAPAS).....</b>	<b>82</b>
<b>4.12. ESTÁNDARES DE PROGRAMACIÓN.....</b>	<b>92</b>
4.12.1. ORGANIZACIÓN DE LOS ARCHIVOS.....	92
4.12.2. DECLARACIÓN DE VARIABLES.....	93
4.12.3. SENTENCIAS.....	94
4.12.4. ASIGNACIÓN DE NOMBRES .....	97
4.12.5. DOCUMENTACIÓN.....	101
4.12.6. FORMATOS .....	101
<b>4.12. PANTALLAS Y REPORTE (PROTOTIPO).....</b>	<b>102</b>
<b>4.13. PRUEBAS Y DEPURACIÓN. ....</b>	<b>106</b>
<b>4.15. RECOPIACIÓN Y CARGA DE DATOS.....</b>	<b>114</b>
<b>4.16. PRUEBAS Y DEPURACIÓN FINAL EN FUNCIONAMIENTO. ....</b>	<b>114</b>
<b>4.17. PUESTA EN MARCHA DEL SISTEMA.....</b>	<b>115</b>
<b>4.18. CAPACITACIÓN AL USUARIO FINAL.....</b>	<b>115</b>
<b>4.19. CAPACITACIÓN AL PERSONAL TÉCNICO.....</b>	<b>116</b>
<b>CAPÍTULO V.....</b>	<b>118</b>
<b>IMPACTOS ESPERADOS DEL PROYECTO. ....</b>	<b>118</b>
<b>5.1 CIENTÍFICO EDUCATIVO TÉCNICO TECNOLÓGICO EMPRESARIAL SOCIAL ECONÓMICO. ....</b>	<b>118</b>
5.1.1 EDUCATIVO.....	118
5.1.2 CIENTÍFICO. ....	118
5.1.3 TÉCNICO Y TECNOLÓGICO. ....	118
5.1.4 AMBIENTAL .....	119
5.1.5 ECONÓMICO. ....	119
5.1.6 SOCIAL.....	119
<b>5.8 CONCLUSIONES.....</b>	<b>119</b>
<b>5.9 RECOMENDACIONES.....</b>	<b>120</b>
<b>CAPÍTULO VI.....</b>	<b>122</b>

	<b>Página</b>
<b>ASPECTOS ADMINISTRATIVOS.....</b>	<b>122</b>
6.1    PRESUPUESTO. ....	122
6.2    CRONOGRAMA DE ACTIVIDADES. ....	123
6.3    BIBLIOGRAFÍA.....	124
6.4    WEB- GRAFÍA. ....	124
<b>CAPÍTULO VII. ....</b>	<b>126</b>
<b>7.    ANEXOS.....</b>	<b>126</b>
7.1.    MANUAL DE USUARIO. ....	127
7.2.    MANUAL TÉCNICO. ....	136
7.3.    NOMBRAMIENTO DIRECCIÓN EJECUTIVA. ....	178
7.4.    CEDULA: DIRECTOR. ....	180
7.5.    RUC: INDESIC.....	182

## ÍNDICE DE TABLAS

	<b>Página</b>
TABLA 1 CUADRO DE DISTRIBUCIÓN DE HARDWARE.....	46
TABLA 2 CUADRO DE SOFTWARE INSTALADO .....	47
TABLA 3 CUADRO DE CONECTIVIDAD .....	48
TABLA 4 CUADRO DEL PERSONAL TÉCNICO.....	48
TABLA 5 CUADRO 1 DE ALTERNATIVAS.....	49
TABLA 6 CUADRO 2 DE ALTERNATIVAS.....	51
TABLA 7: CUADRO 3 DE ALTERNATIVAS.....	52
TABLA 8: CRITERIOS DE EVALUACIÓN .....	53
TABLA 9: CUADRO DE EVALUACIÓN TÉCNICA .....	54
TABLA 10: CRITERIO DE EVALUACIÓN ECONÓMICA. ....	54
TABLA 11: CUADRO DE COSTOS ECONÓMICOS. ....	55
TABLA 12: CUADRO DE EVALUACIÓN ECONÓMICA.....	55
TABLA 13: CRITERIOS DE EVALUACIÓN TIEMPO.....	55
TABLA 14: CUADRO DE CALIFICACIÓN SOPORTE .....	56
TABLA 15: CUADRO DE EVALUACIÓN SOPORTE .....	56
TABLA 16: CUADRO DE EVALUACIÓN DE LAS ALTERNATIVAS .....	57
TABLA 17: PRESUPUESTO GENERAL DEL PROYECTO. ....	122

## ÍNDICE DE GRÁFICOS

	<b>Página</b>
GRÁFICO 1: MAPA DE LA UBICACIÓN .....	13
GRÁFICO 2 <i>FUNCIONAMIENTO DEL DISPOSITIVO BIOMÉTRICO.</i> .....	17
GRÁFICO 3 AMIGOS DE LA INSTITUCIÓN .....	18
GRÁFICO 4: ARQUITECTURA DE CAPAS .....	29
GRÁFICO 5 ORGANIGRAMA ORGANIZACIONAL.....	45
GRÁFICO 6 DIAGRAMA DE RED INDESIC.....	47
GRÁFICO 7: PROCESO MANUAL DE CONTROL DE ASISTENCIA. ....	59
GRÁFICO 8: DIAGRAMA DE FLUJO DE PROCESOS.....	59
GRÁFICO 9: DIAGRAMA GENERAL DE CASOS DE USO DEL SISTEMA .....	60
GRÁFICO 10: DIAGRAMA DE CASOS DE USO INGRESO AL SISTEMA.....	60
GRÁFICO 11 CASOS DE USO PARA GENERAR NUEVOS REGISTROS. ....	61
GRÁFICO 12: CASO DE USO PARA MODIFICAR LOS REGISTROS .....	61
GRÁFICO 13: CASO DE USO PARA ELIMINAR UN REGISTRO. ....	62
GRÁFICO 14: CASO DE USO PARA REALIZAR EL CONTROL DIARIO. ....	62
GRÁFICO 15: CASO DE USO PARA IMPRIMIR LOS REPORTES .....	63
GRÁFICO 16: CASO DE USO PARA VERIFICAR LOS REGISTROS.....	63
GRÁFICO 17: DIAGRAMA DE SECUENCIA PARA ELIMINAR REGISTROS .....	70
GRÁFICO 18: DIAGRAMA DE SECUENCIA PARA MODIFICAR REGISTROS.....	70
GRÁFICO 19: DIAGRAMA DE SECUENCIA PARA NUEVOS REGISTROS.....	71
GRÁFICO 20: PROCESO INGRESAR CONTRASEÑA .....	71
GRÁFICO 21: PROCESO INGRESAR EMPLEADO .....	72
GRÁFICO 22: ACTUALIZAR USUARIO.....	72
GRÁFICO 23: ELIMINAR DATOS.....	73
GRÁFICO 24: CONSULTA EMPLEADO .....	73
GRÁFICO 25: CONTROL DIARIO.....	74
GRÁFICO 26: IMPRESIÓN DE DATOS.....	74
GRÁFICO 27: DIAGRAMA DE COLABORACIÓN ELIMINAR REGISTROS .....	75
GRÁFICO 28: DIAGRAMA DE COLABORACIÓN NUEVO EMPLEADO .....	75
GRÁFICO 29: DIAGRAMA DE COLABORACIÓN MODIFICAR DATOS .....	76
GRÁFICO 30: FASES DEL DESARROLLO DEL SISTEMA.....	77
GRÁFICO 31: MODELO DE LA BASE DE DATOS LÓGICO .....	81
GRÁFICO 32: LA BASE DE DATOS MODELO FÍSICO .....	82



## RESUMEN EJECUTIVO

El Proyecto “Sistema Biométrico para el Control de Asistencia en INDESIC” está orientado al estudio y desarrollo de un sistema que permitirá registrar los ingresos y salidas de los empleados, utilizando la tecnología biométrica de la huella dactilar.

En cuanto a su desarrollo, el proyecto comenzará analizando, paso a paso, los distintos procesos de su evolución. Comenzando con el planteamiento del problema, los objetivos que se quieren alcanzar y el desarrollo de la solución.

Desde el segundo capítulo, se presentará los antecedentes de la empresa y la parte teórica de todo el software y hardware que se implementará en el desarrollo. Como se crearán las bases para el desarrollo del Sistema Biométrico, utilizando la huella dactilar.

Posteriormente en el diseño, desarrollo e implementación de la aplicación se encuentran en el tercer y cuarto capítulo. En éste se pone en práctica el desarrollo completo de la metodología de modelado de aplicaciones basado en UML.

Además se emplean los conceptos de modelado de base de datos para el diseño e implementación en donde se almacenará las huellas, la información del empleado y las transacciones realizadas. Adicionalmente, en este capítulo se realiza el dimensionamiento del servidor requerido, el cálculo de espacio en disco de la base de datos y los detalles de la implementación de la aplicación.

Así mismo se describe las pruebas que se realizaron y los escenarios en las que se ejecutaron para comprobar el correcto funcionamiento de la aplicación y las pruebas para validar el grado de confiabilidad y respuesta del producto desarrollado. Se describen también los resultados que se obtuvieron en cada prueba y se presenta el análisis de los mismos.

Finalmente, el Quinto y Sexto Capítulo contiene las conclusiones tanto de las pruebas realizadas así como las conclusiones del proyecto de titulación. Se tienen también las recomendaciones que se realiza para ampliar el presente proyecto y poder brindar nuevas funcionalidades a la aplicación.



## CAPÍTULO I

### TEMA:

Desarrollo e Implementación de un Sistema Biométrico para el Control de Asistencia en el Instituto para el Desarrollo Social de las Investigaciones Científicas (INDESIC).

#### **1.1. Planteamiento del Problema.**

El INDESIC cuenta con 10 empleados fijos (directos) y 15 indirectos (pago por servicios y otros), en donde cada uno labora de acuerdo a un contrato y bajo todas las leyes laborales vigentes.

El problema surge al momento de querer saber el número de horas laboradas de cada uno de los empleados, controlar las faltas, atrasos y permisos; esto con el objetivo de poder presupuestar en los gastos de cada proyecto y poder pagar ordenada y correctamente los respectivos sueldos.

En si no se puede obtener un informe preciso de lo explicado anteriormente, esto se debe a que los registros de los ingresos y salidas de cada uno de los empleados se lo lleva completamente de forma manual, desde el año 1998 año en donde empezó a funcionar la institución, además la persona encargada del área no actualiza constantemente la información de nuevos empleados que ingresan; esta forma de llevar el control se ha realizado desde el año 1998 donde empezó a funcionar la institución.

Así mismo a no disponer de un sistema automatizado para el cálculo de la horas laboradas los pagos de los sueldos es lento y muchas veces hasta es mal calculado, lo que genera pérdida de tiempo, dinero para la empresa y por consiguiente malestar en empleados, inestabilidad laboral y bajo redimiendo en la ejecución de cada uno de los proyectos.

## 1.2. Delimitación de la Investigación.

El presente proyecto se ejecutará en la provincia Pichincha, Cantón Quito, parroquia Santa Prisca en la calles Jorge Juan Nº 31-87 entre Av. Mariana de Jesús y pasaje San Gabriel en donde se encuentran ubicadas las oficinas de la matriz de INDESIC.

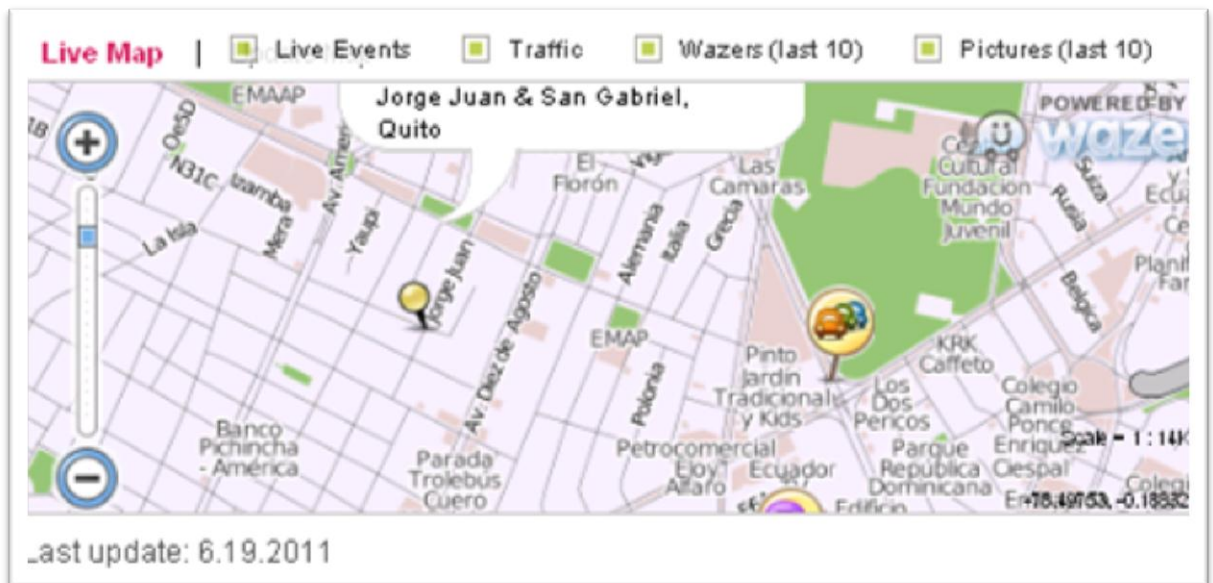


Gráfico 1: Mapa de la Ubicación

Fuente: [www.waze.com](http://www.waze.com)

Autor: Waze.

## 1.3. Formulación del Problema.

¿Con la implementación del Sistema del Control de asistencia Biométrico mejorará la producción laboral y el rendimiento en cada uno de los departamentos del INDESIC?

## 1.4. Objetivo General.

Desarrollar e Implementar de un Sistema Biométrico para el Control de Asistencia en el Instituto para el Desarrollo Social de la Investigaciones Científicas (INDESIC), con el fin de mejorar el control de la asistencia de los empleados, generar sus pagos de sueldos y descuentos.

## 1.5. Objetivos Específicos.

- Analizar la información sobre las técnicas de reconocimiento de patrones utilizados por la biometría y las normativas de calidad en el diseño del sistema.



- Definir todos los requerimientos técnicos y operativos para aplicación del sistema.
- Diseñar los procesos lógicos, base de datos y reglas de la organización sobre el funcionamiento del sistema a implementar en la institución.
- Codificar el sistema para el control de ingreso y salida mediante la autenticación biométrica del usuario (lector de huella dactilar).
- Realizar las pruebas de software respectivo y su respectiva capacitación al usuario final.

### **1.6. Justificación e Importancia.**

El presente proyecto se justifica mediante la creación y desarrollo de un sistema que permita mejorar y modificar las limitaciones del sistema actual (manual), y que a su vez esté diseñado para que se pueda integrar con los proyectos futuros de la organización. Uno de éstos es el módulo de recursos humanos, consiste en que los empleados, profesores, tutores puedan registrar su horas laboradas mediante su código y su huella dactilar.

La principal importancia del desarrollo del proyecto es ver la manera en como ayudará y mejorará la seguridad, los procesos en cálculo de los roles de pago en el sistema informático contable actual que existe. El hecho de firmar manualmente en un registro resulta para muchas personas un proceso, tedioso y largo, pero gracias a la ayuda de la biometría y autenticación, se ha reducido el tiempo de este proceso. El uso de sistemas biométricos de autenticación, es de gran importancia para las empresas que desean tener un nivel de seguridad más controlado y estricto, ya que ofrecen exactitud en la seguridad, con lo que pueden tener un mejor control tanto de la información que maneja como de los accesos a estas.

Actualmente otras instituciones ya cuentan con sistemas que automatizan estos procesos y como el método de control en el INDESIC es completamente obsoleto entonces de aquí surge la necesidad de implantar un sistema que permita automatizar el control de la misma, de esta forma se podrá brindar un mejor servicio a los empleados ya que el tiempo en los procesos de disminuirá considerablemente, se podrá obtener información



real del trabajo de los colaboradores así como también de las multas que tiene, todo esto contribuirá a mejorar la imagen del INDESIC como institución ya que se pondrá al mismo nivel de otras organizaciones públicas y privadas que ya cuentan con este tipo de sistemas.

### **1.7. Alcance.**

El presente proyecto nace el proyecto como una iniciativa para desarrollar un nuevo sistema de control de ingresos y salidas de los empleados basados en tecnología biométrica de huella dactilar, que se integrará al sistema contable utilizado en dicha empresa y entre los módulos a desarrollar son:

#### ✓ **Módulo Seguridad.**

- Registro de Usuarios.
- Registro de Administradores del Sistema.
- Respaldos de Información.
- Identificación mediante la tecnología biométrica

El modulo se crea con la finalidad de generar auditorias de los movimientos dentro del sistema.

#### ✓ **Módulo de Mantenimiento.**

- Modificar.
- Eliminar.
- Nuevo.
- Guardar.

El modulo permitirá realizar el respectivo mantenimiento de cada uno de los datos ingresados al sistema. Esto interactúa dinámicamente con el usuario.

#### ✓ **Módulo de Transaccional.**

- Ingreso de Nuevos empleados.
- Salidas de empleados.



- Descuentos.
- Horas laboradas.
- Vacaciones.
- Migración de datos al sistema contable (SMA).

El modulo permitirá ingresar datos al sistema realizar todos los movimientos para el control de la asistencia del personal, mejorando cada uno de los procesos que actualmente se los realiza de forma manual.

✓ **Módulo de Reportes.**

- Reporte de horas laboradas por empleado.
- Reporte atrasos multas diarias, mensuales.
- Reporte de rol de pagos.

Este módulo mostrará los resultados de cada uno de los procesos, la información será condensada de acuerdo a los parámetros solicitados por el usuario. Así mismo permitirá tomas decisiones en la parte administrativa.

✓ **Módulo de Identificación.**

- Mediante clave.
- Por huella dactilar.

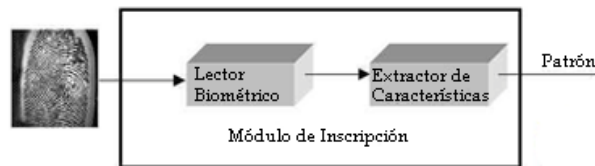
La identificación para los usuarios se podrá realizar mediante las dos formas descritas anteriores de manera que exista mayor confiabilidad.

✓ **Módulo de Inscripción.**

- Mediante clave.
- Por huella dactilar.

El módulo de inscripción se encarga de adquirir y almacenar la información proveniente del indicador biométrico con el objeto de poder contrastar esta información con la que será proporcionada en ingresos posteriores al sistema. Las labores ejecutadas por el módulo de inscripción son posibles gracias a la acción del lector biométrico y del extractor

de características. El primero se encarga de adquirir datos relativos del indicador biométrico elegido y entregar una representación en formato digital de éstos. El segundo extrae, a partir de la salida del lector, características representativas del indicador. Durante este proceso de recopilación de datos, es en donde se presentan los primeros problemas. En primer lugar las muestras deben ser obtenidas mediante un sensor, por lo tanto, están sujetas a la calidad y características técnicas del sensor utilizado, lo que conlleva a que las características del sensor deberán ser estandarizadas, a fin de garantizar que las muestras obtenidas de un usuario en diferentes sistemas sean compatibles. En cuanto al almacenamiento, existen varias formas de guardar los datos previamente recopilados y procesados, que al momento de ser almacenados reciben el nombre de patrón (témplate). La organización de la estructura de los datos debe ser flexible, permitiendo su reestructuración, si fuese necesario.



**Gráfico 2** *Funcionamiento del Dispositivo Biométrico.*

**Fuente:** Arquitectura de un sistema Biométrico para identificación personal

**Autor:** por Ing. César Morales Mejía.

## CAPÍTULO II.

### MARCO TEÓRICO.

#### 2.1. Los Antecedentes.

El INDESIC es una institución sin fines de lucro, sin tendencias político partidistas o religiosas y con gran preferencia por atender los requerimientos de la población joven de escasos económicos, especialmente del sector indígena, con diferentes servicios que contribuyan de manera efectiva en la investigación, profesionalización, capacitación, asistencia técnica y desarrollo económico, para que estos grupos, desde sus ámbitos de gestión promuevan su propio desarrollo y contribuyan a la consolidación de los procesos democráticos y económicos de nuestro país.

Para el INDESIC, la formación de grupos humanos muy bien calificados, ha sido uno de los retos principales gracias a quienes despliega su trabajo en todas las comunidades del Ecuador, ya que, estos grupos hoy en día son entes multiplicadores de las acciones que llevamos adelante.



**Gráfico 3** Amigos de la Institución

**Fuente:** Archivos INDESIC.

**Autor:** Jhonny Suqui



## **2.2. Reseña Histórica.**

### **2.2.1. Historia.**

En 1992, un grupo de estudiantes universitarios de algunas Nacionalidades Indígenas del Ecuador, consolidamos un equipo de profesionales en bien de aportar al desarrollo de las potencialidades individuales y colectivas, generar autoconfianza y elevar el nivel propositivo y crítico de los actores sociales indígenas y afro ecuatorianos, a fin de ser los protagonistas de las transformaciones sociales, políticas y económicas de nuestro país.

Mirando todos los contextos y en medio de los acontecimientos mundiales, visualizamos oportunidades de bienestar individual y colectivo. Preservamos nuestra identidad cultural, garantizado los derechos humanos y colectivos de nuestros pueblos; aportamos a la construcción de una sociedad equitativa y democrática, gracias a lo cual estamos alcanzando la equidad, el reconocimiento y la plena convivencia en una sociedad variadas culturas, y grupos étnicos.

El Instituto para el Desarrollo Social y de las Investigaciones Científicas - INDESIC ubicado en Quito – Ecuador, es una entidad de derecho privado, reconocida mediante Acuerdo No. 1742 del 5 de mayo de 1998 emitido por el Ministerio de Educación.

### **2.2.2. Misión.**

Cooperar con el desarrollo de las Nacionalidades y Pueblos Indígenas del Ecuador y América Latina para consolidar los procesos democráticos de nuestros países, a través de la investigación, profesionalización, capacitación, asistencia técnica y desarrollo comunitario.

### **2.2.3. Visión.**

Líder en la promoción del desarrollo individual y colectivo de las Nacionalidades y Pueblos Indígenas del Ecuador y América Latina, que genere la construcción, la práctica del Estado



Intercultural y plena convivencia en una sociedad libre de discriminación y dogmas raciales.

#### **2.2.4. Nuestros Principios.**

- Proponer la integración de los Pueblos y Nacionalidades Indígenas de Latinoamérica como base del desarrollo, que posibilite la construcción de propuestas encaminadas a mejorar sus condiciones de vida.
- Ser una entidad pluralista, no dogmática, sustentada en la diversidad cultural.
- Impulsar la autodefinición e identidad de las Nacionalidades y Pueblos Indígenas de América Latina, en especial de los jóvenes con grandes capacidades pero con pocas oportunidades.

#### **2.2.5. Servicios.**

1. Auditorios de Capacitación.
2. Biblioteca de Consultas.
3. Prensa Diaria.
4. Hemeroteca.
5. Personal Técnico Capacitado.
6. Préstamo de Ordenadores Portátiles.
7. Prestamos de Proyector.
10. Centro de Cómputo.
11. Sala de Audio Visuales.
12. Internet de Banda Ancha.
13. Servicio de Correo Electrónico.
14. Alojamiento de Sitios Web.
15. Biblioteca virtual.
16. Área de anillados.
17. Cafetería y área de preparación de alimentación.



## **2.3. Marco Referencial.**

### **2.3.1. Sistema Biométrico.**

Con la evolución de las tecnologías asociadas a la información, nuestra sociedad está cada día más conectada electrónicamente. Labores que tradicionalmente eran realizadas por seres humanos hoy, gracias a las mejoras tecnológicas ya son realizadas por los sistemas automatizados. Y dentro de la amplia gama de las posibles actividades que pueden automatizarse; en este proyecto, se estudiará los sistemas biométricos que se ha transformado en un área emergente. La biometría es la ciencia que se dedica a la identificación de individuos a partir de una característica anatómica o un rasgo de su comportamiento.

Una característica anatómica tiene la cualidad de ser relativamente estable en el tiempo, tal como una huella dactilar, la silueta de la mano, patrones de la retina o el iris. No cualquier característica anatómica puede ser utilizada con éxito por un sistema biométrico. Para que esto así sea debe cumplir con las siguientes características: *Universalidad, Unicidad, Permanencia y Cuantificación.*

El sistema biométrico para el control del personal planteado anteriormente se desarrollará con el siguiente software que a continuación se describe cada uno de ellos.

### **2.3.2. Diseño de la Base de Datos:**

El diseño de la base de datos se realizará en el software IBM Rational Rose Enterprise ya proporciona un lenguaje de modelado común que permite crear más deprisa. Este software de calidad nos permite diseñar el modelamiento en tres capas.

Por lo que a continuación se detalla el desarrollo de sus aplicaciones Ada, ANSI C++, C++, CORBA, Java, J2EE, Visual C++ y Visual Basic con código generado a partir de modelos visuales.



- Es compatible con el lenguaje UML (Unified Modeling Language) y es uno de los productos más completos de la familia Rational Rose.
- Soporta patrones de Analysis, ANSI C++, Rose J y Visual C++, Enterprise JavaBeans 2.0, e ingeniería directa e inversa para algunas de las construcciones más comunes de Java 1.5.
- Es capaz de analizar la calidad del código y de generar código gracias a las capacidades de sincronización configurable entre el modelo y el código, además de una gestión más detallada y el uso de modelos con la función de componentes de modelos controlables por separado.
- Se integra con otras herramientas de desarrollo del ciclo vital de IBM Rational, y con cualquier sistema de control de versiones conforme con SCC, incluyendo IBM Rational ClearCase.
- Incluye un complemento de modelado web, que proporciona la capacidad de visualización y el modelado, y herramientas para desarrollar aplicaciones web.
- Permite el modelado UML para diseñar bases de datos, con la posibilidad de representar la integración de los requisitos de datos y aplicaciones mediante diseños lógicos y físicos.
- Crea definiciones de tipos de documentos (DTD) XML para utilizarlas en la aplicación.
- Sistemas operativos admitidos: Windows.

### 2.3.3. Administrador y Almacenamiento de la Base de Datos.

- ✓ **SQL Server 2008.** Permite ejecutar aplicaciones de misión crítica, reduciendo costos de administración de infraestructura de datos y brindando introspectiva e información a todos los usuarios.
  - **Confiable:** Permite a las organizaciones ejecutar sus aplicaciones más críticas con niveles de seguridad, confiabilidad y escalabilidad muy altos.
  - **Productivo:** Permite reducir el tiempo y los costos requeridos para desarrollar y administrar sus infraestructuras de datos.



- **Inteligente:** Ofrece una plataforma integral que brinda introspectiva e información donde sus usuarios lo desean.

Nuevas funcionalidades de SQL Server 2008:

✓ **Protección de la Información.**

- ¿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.

✓ **Acceder a la Información desde cualquier lugar en cualquier momento.**

- Con SQL 2008 podrá 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.

#### 2.3.4. Desarrollo de Aplicación.

**¿Qué es .NET?** Net es la nueva plataforma de desarrollo que ha lanzado al mercado Microsoft, y en la que ha estado trabajado durante los últimos años. Sin duda alguna va a ser uno de los entornos de desarrollo que reinen durante los próximos años.

¿Entonces .NET es un lenguaje de programación?, No. Es mucho más que eso, Microsoft .NET es un conjunto de tecnologías de software, compuesto de varios lenguajes de programación que se ejecutan bajo el .NET Framework. Es además un entorno completamente orientado a objetos y que es capaz de ejecutarse bajo cualquier plataforma.

- **.NET Framework**, que es el entorno de trabajo de la plataforma .NET y que la engloba completamente. Toda la plataforma .NET forma parte de .NET framework.
- **Lenguajes .NET**. Destacan **C#** y **VB.NET**, y recientemente **J#**, aunque existen más lenguajes y con toda seguridad surgirán nuevos.
- El Common Runtime Language **CRL**, que es el motor de ejecución común a todos los lenguajes .NET.
- **MSIL**, Microsoft Intermedia llanguage, es el lenguaje intermedio al que compilan las aplicaciones (Asemmbliés) .NET. Este lenguaje intermedio es interpretado por el CRL en tiempo de ejecución.
- **CLS**, common Language Specification, que engloban las pautas que deben cumplir los lenguajes .NET. Está característica la que va a permitir a otras compañías producir lenguajes compatibles con .NET.
- **ADO.NET**, es la nueva interfaz de bases de datos. No se trata de una evolución de ADO, sino que se trata de una interfaz completamente nueva.



- **ASP.NET**, es la nueva tecnología para páginas web dinámicas completamente integrada dentro del entorno .NET. Representa una auténtica revolución en el desarrollo Web (Internet e Intranet).
- ✓ **Biblioteca de clases .NET**, es el conjunto de clases que componen el .NET framework y que nos permiten realizar casi cualquier tarea de una manera fácil y rápida.
- ✓ **.NET Framework**. Es el corazón de la tecnología .NET. Es el marco de trabajo y ejecución común a toda la tecnología .NET. y por lo tanto un elemento indispensable. Dentro del .NET framework están integrados los lenguajes .NET, el C#, el CLS, ADO.NET, ASP.NET.

Además .NET soporta los siguientes lenguajes de programación:

- **C#**.
- **C++** controlado.
- **VB.NET**.

Todos ellos trabajan perfectamente integrados con un motor común, el .NET framework, siendo **C#** el lenguaje principal, ya que es el único que ha sido desarrollado específicamente para .NET. Gran parte de la biblioteca de clases que componen el framework están escritas en **C#**.

.Net Framework es el conjunto de espacios de trabajo y clases que sirven de base a los lenguajes .NET. Es la evolución de las API de Windows y de la fundación de clases de Microsoft MFC hacia la completa encapsulación. Cualquier programa que desarrollemos utilizando la tecnología .Net, independientemente del lenguaje utilizado, estará basado en .Net Framework.

La mejor forma de entender esto es un ejemplo. Imaginemos los menús de cualquier aplicación Windows que nos permiten acceder a las distintas opciones del programa, pues bien, en el mundo de la tecnología .Net dichos menús son una clase perteneciente al .Net Framework. Esta clase es siempre la misma independientemente del lenguaje y entorno en el que nos encontremos, y la forma de trabajar con ella es la misma desde C#, VB.net o



cualquier lenguaje .Net. Lo mismo ocurre con los botones, las imágenes, la forma de leer los archivos... todo, absolutamente todo lo que hagamos en un lenguaje .NET está basado en el .Net Framework.

✓ **MSIL, CIL y el código controlado.**

Cuando escribimos un programa lo hacemos en un determinado lenguaje que podríamos llamar "humano"(aunque algunos se empeñen en decir que los programadores no somos humanos). Es decir, podemos leer y entender un programa (o al menos intentarlo) a través de un editor de texto, ya que este programa está escrito en lenguaje "humano" (utilizando nuestra letras y esas cosas). El único problema es que un ordenador no es capaz de entender nuestro programa, así que hay que traducirlo a su idioma. A este proceso se le conoce como compilación. Como resultado del proceso de compilación obtenemos el programa ejecutable en código máquina, que entiende el ordenador pero no el "humano"(si alguien lo entiende no es humano, aquí sí). Según la arquitectura del procesador, el sistema operativo, etc... este código es diferente y un programa que se ejecuta correctamente en un entorno **Windows** no funciona en **Macintosh** o **UNIX**. Es decir el programa sólo funciona para la plataforma para la que fue diseñado. Este no es el proceso que ocurre en .NET. Cuando compilamos un programa escrito en cualquiera de los lenguajes .Net no se compila hacia código máquina nativo, sino que se hace hacia MSIL (Microsoft Intermediate Language), este MSIL es un lenguaje intermedio y universal. Cuando compilo un programa escrito en C# o en VB.Net ambos generan MSIL, con ciertas diferencias pero MSIL. Este código será interpretado posteriormente por un intérprete el CLR. De este modo conseguimos que un programa escrito en .NET funcione en cualquier plataforma existente, incluso en plataformas futuras, sólo necesitamos construir el intérprete apropiado.

El MSIL es independiente del procesador, de la plataforma de desarrollo y de la plataforma de ejecución. El MSIL es parte del .NetFramework.

Llegados a este punto tenemos nuestro programa compilado a MSIL, pero el programa no funciona, ya que el procesador sólo entiende su propio código máquina nativo, y MSIL no



lo es. Es entonces cuando aparece el CRL (CommonLanguageRuntime), o motor de ejecución común, que lo que hace es servir de traductor entre el MSIL y el código máquina nativo. Cuando ejecutamos un programa el CRL se encarga de compilar a código nativo dicho programa y ejecutarlo. A este tipo de compiladores se les conoce como compiladores JIT (Just In Time). Si alguno de vosotros conoce **Java** abra gritado ¡plagio, esto es el ByteCode!. No exactamente...cambian... los nombres.

Teóricamente el CRL interpreta MSIL mucho mejor de lo que lo hace Java con el ByteCode y existen ciertas diferencias en la arquitectura interna, el CRL únicamente compila a código nativo la parte necesaria en cada momento durante la ejecución mientras que Java compila el programa completo, pero de este modo podemos ejecutar nuestro programa sobre cualquier máquina, siempre y cuando exista una versión del .Net Framework y del CRL apropiada. Al código que se ejecuta bajo la batuta del CRL se le conoce como código contralado.

Nota técnica. Los lenguajes .Net no son compilados, ya que no generan código nativo, ni interpretados, ya que compilan código MSIL, son lenguajes evaluados.

#### ✓ **Tipos comunes entre lenguajes .NET. CLS y CTS.**

Para que todo esto sea posible Microsoft ha introducido ciertas restricciones a la hora de diseñar los lenguajes de programación y en la forma de programar.

La especificación común de lenguajes **CLS** hace posible que cualquier fabricante de software genere herramientas compatibles con .NET. De esta forma existen versiones .NET de Delphi y otros lenguajes ajenos a Microsoft.

Dentro del CLS destaca especialmente el **CTS (CommonTypeSystem)**, que es una especificación que define como el **CRL** utiliza y administra los tipos de datos. Básicamente es una relación entre los distintos tipos de datos que proporciona cada lenguaje. La consecuencia inmediata del CTS es que dentro de un mismo proyecto .NET podemos utilizar varios lenguajes .NET.



✓ **Assemblies**

Un proyecto .NET no genera un ejecutable tal y como lo conocemos. Un proyecto .NET genera Assemblies. Un assembly es la unidad ejecutable de cualquier programa .NET, pero no se limita al código compilado sino que también incluye lo que se ha dado en llamar el **manifiesto**.

El manifiesto es un listado de las librerías (dll) y los controles (ocx) que son necesarios para que la aplicación funcione. Este listado contiene el número de versión necesario para que la aplicación funcione correctamente, impidiendo que la instalación posterior de un programa afecte a nuestro ejecutable.

Con esta técnica Microsoft pretende poner fin al ya maldito **infierno de las dll**. El tiempo dirá si lo consiguen.<sup>1</sup>

**2.3.5. Hardware.**

**2.3.5.1. Dispositivos:**

En el presente proyecto el dispositivo que permitirá la implementación del sistema será el lector de huellas USB digitales SecuGen un equipo que tiene un 99.9% de efectividad en las capturas de huella digitales.

**2.3.5.2. Tipo de comunicación.**

La comunicación que se utilizará será por conexión directa al internet y mediante una red pública.

**2.3.5.3. Arquitectura de Tres Capas.**

La estrategia tradicional de utilizar aplicaciones compactas causa gran cantidad de problemas de integración en sistemas software complejos como pueden ser los sistemas de gestión de una empresa o los sistemas de información integrados consistentes en más

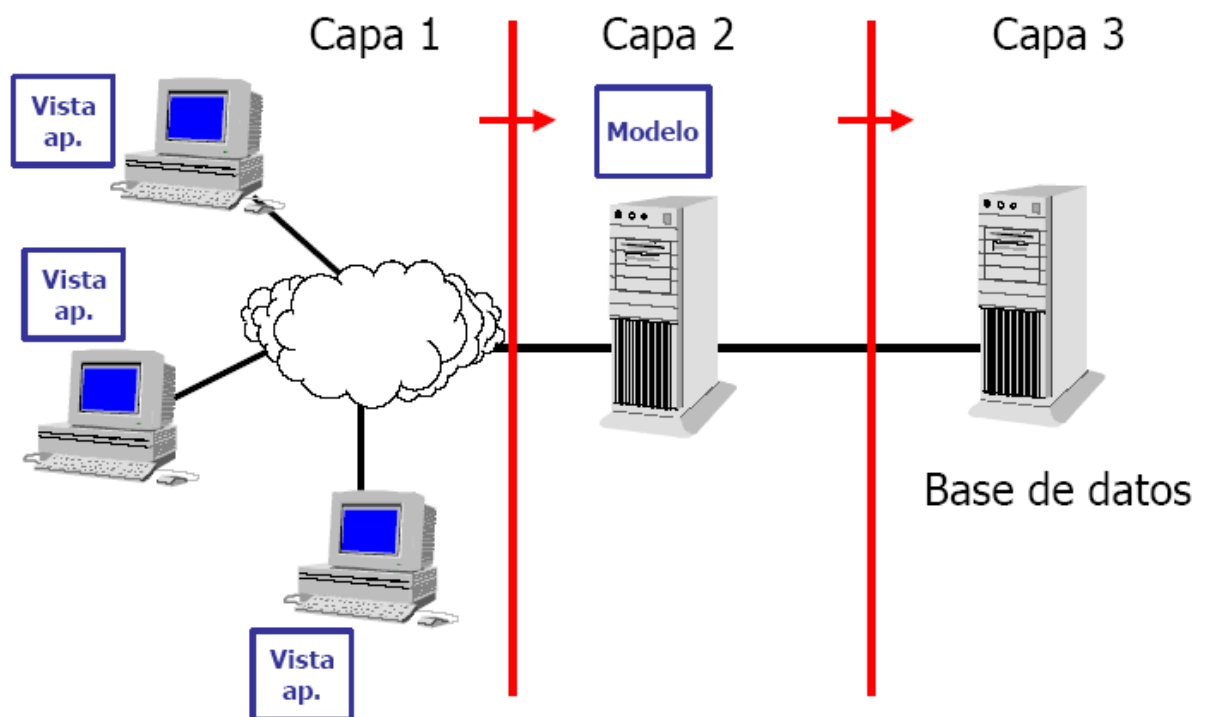
---

<sup>1</sup><http://www.microsoft.com/spain/visualstudio>

de una aplicación. Estas aplicaciones suelen encontrarse con importantes problemas de escalabilidad, disponibilidad, seguridad, integración.

Para solventar estos problemas se ha generalizado la división de las aplicaciones en capas que normalmente serán tres: una capa que servirá para guardar los datos (base de datos), una capa para centralizar la lógica de negocio (modelo) y por último una interfaz gráfica que facilite al usuario el uso del sistema.<sup>2</sup>

### Arquitectura en Tres capas



**Gráfico 4:** Arquitectura de capas

**Fuente:** <http://www.microsoft.com/spain/visualstudio>

**Autor:** Microsoft

<sup>2</sup> <http://oness.sourceforge.net/proyecto/html/ch03s02.html>



### **2.3.6. Software.**

Todas las aplicaciones con lo que se refiere al software descrito anteriormente se utilizarán únicamente las versiones de prueba y con fines académicos más no con fines comerciales. A continuación listamos:

- ***Diseño de Modelamiento: Rational Rose.***
- ***Metodología de Desarrollo: Tres Capas RUP.***
- ***Lenguaje de Programación: .Net.***
- ***Base de Datos: SQL SERVER 2008.***

### **2.3.7. Know How.**

Como parte del ejercicio profesional y la demostración de cada uno de los conocimientos durante los seis semestres en la institución. Estoy capacitado para el desarrollo de sistemas a la medida de los requerimientos de las Empresas y Organizaciones. También la con la formación en administración, costos y gestión de recursos técnicos y humanos, que potencian la capacidad de análisis se amplía la visión de negocios y se desarrolla un pensamiento estratégico. Por esto y como profesionales ya gozamos de una demanda sostenida en empresas industriales, de servicios y comerciales.

#### **Y nuestros campos ocupacionales son:**

- Proyectar, Dirigir y Ejecutar el desarrollo de Sistemas de Información.
- Planificar, Organizar y dirigir el Departamento de Sistemas de una Organización.
- Liderar equipos multidisciplinarios de análisis y programación de sistemas informáticos.
- Elaborar y Ejecutar programas de capacitación en informática.
- Analizar, implementar y administrar redes de área local de PC y bases de datos.
- Administrar los recursos informáticos de una organización



## **2.4. Marco Legal.**

El organismo que controla el funcionamiento del INDESIC es el Ministerio de Educación y Cultura, el mismo que otorgó el permiso el 6 de mayo de 1998 y hasta el momento ha cumplido con todos los requisitos para seguir funcionando.

### **Ver anexos (...).**

De acuerdo a las leyes establecidas en nuestro país y como requisito único para la obtención del título de tecnólogo superior el presente proyecto se enmarcará bajo los siguientes:

#### **Ley de Educación Superior**

##### **Título IV**

##### **Capítulo 2**

##### **De la garantía de la igualdad de oportunidades**

#### **Art. 87.- Requisitos previos a la obtención del título**

Como requisito previo a la obtención del título, los y las estudiantes deberán acreditar servicios a la comunidad mediante prácticas o pasantías pre profesionales, debidamente monitoreadas, en los campos de su especialidad, de conformidad con los lineamientos generales definidos por el Consejo de Educación Superior.

Dichas actividades se realizarán en coordinación con organizaciones comunitarias, empresas e instituciones públicas y privadas relacionadas con la respectiva especialidad.<sup>3</sup>

##### **Título VII**

##### **CAPITULO 2**

##### **De la Tipología de Instituciones, y Régimen Académico**

##### **Sección Primera**

**Art. 118.- Niveles de formación de la educación superior.- Los niveles de formación** que imparten las instituciones del Sistema de Educación Superior son:

a) Nivel técnico o tecnológico superior, orientado al desarrollo de las habilidades y destrezas que permitan al estudiante potenciar el saber hacer. Corresponden a éste los títulos profesionales de técnico o tecnólogo superior, que otorguen los institutos

---

<sup>3</sup> Ley del Ministerio de educación



superiores técnicos, tecnológicos, pedagógicos, de artes y los conservatorios superiores. Las instituciones de educación superior no podrán ofertar títulos intermedios.

Así mismo la entidad en para donde se desarrollara la aplicación esta supervisada por el ministerio de Educación y cultura en donde rige las siguiente leyes.<sup>4</sup>

## **LEY DE EDUCACIÓN**

### **Ley Nº. 127 del 15 de abril de 1983**

#### **Capítulo II**

##### **Principios y Fines**

**Art. 2.-** La educación se rige por los siguientes principios:

- a) La educación es deber primordial del Estado, que lo cumple a través del Ministerio de Educación y de las Universidades y Escuelas Politécnicas del país;
- b) Todos los ecuatorianos tienen el derecho a la educación integral y la obligación de participar activamente en el proceso educativo nacional;
- c) Es deber y derecho primario de los padres, o de quienes los representan, dar a sus hijos la educación que estimen conveniente. El Estado vigilará el cumplimiento de este deber y facilitará el ejercicio de este derecho;
- d) El Estado garantiza la libertad de enseñanza de conformidad con la Ley;
- e) La educación oficial es laica y gratuita en todos sus niveles. El Estado garantiza la educación particular;
- f) La educación tiene sentido moral, histórico y social; se inspira en los principios de nacionalidad, democracia, justicia social, paz, defensa de los derechos humanos y está abierta a todas las corrientes del pensamiento universal;
- g) El Estado garantiza la igualdad de acceso a la educación y la erradicación del analfabetismo;

#### **Título Segundo**

##### **Estructura del Sistema Educativo**

###### **Capítulo I**

###### **Estructura General**

---

<sup>4</sup> Ley del Ministerio de educación.



**Art. 4.-** (Reformado por el Art. 1 de la Ley 150, R.O. 918, 20-IV-92).- El sistema educativo nacional comprende dos subsistemas:

- a) El escolarizado; y,
- b) El no escolarizado.

En el sistema educativo nacional se garantiza la Educación Intercultural Bilingüe que, asimismo, comprenderá dos subsistemas:

- a) El escolarizado; y,
- b) El no escolarizado.

**Art. 5.-** El subsistema escolarizado se emplea en los establecimientos determinados por la Ley y comprende:

- a) La educación regular que se somete a las disposiciones reglamentarias sobre el límite de edad, secuencia de niveles y duración de cursos;
- b) La educación compensatoria, que tiene un régimen especial y se la ofrece a quienes no ingresan a los niveles de educación regular o no los concluyen; y,
- c) La educación especial, destinada a estudiantes excepcionales por razones de orden físico, intelectual, psicológico o social.

**Art. 6.-** El subsistema no escolarizado procura el mejoramiento educacional, cultural y profesional a través de programas especiales de enseñanza-aprendizaje y difusión, mediante los esfuerzos e iniciativas públicos o privados.

**Art. 12.-** El ciclo diversificado procura la preparación interdisciplinaria que permita la integración del alumno a las diversas manifestaciones del trabajo y la continuación de los estudios en el ciclo post-bachillerato o en el nivel superior, atendiendo a los requerimientos del desarrollo social y económico del país y a las diferencias y aspiraciones individuales. Las diversas modalidades se organizan de acuerdo con las necesidades del desarrollo científico, económico y cultural del país y aseguran, con sentido integral, la formación humanística y técnica.

**Art. 13.-** (Derogado por la Disposición Final de la Ley 2000-16, R.O. 77, 15-V-2000).

**Art. 14.-** La duración de los niveles pre-primarios, primarios y medio señalados en los artículos 8, 9, 11, 12 y 13 de esta Ley la fijará el Reglamento.



**Art. 15.-** El Ministerio de Educación y Cultura, en el ámbito de su competencia, podrá autorizar el funcionamiento experimental o piloto de unidades educativas.

**Art. 16.-** De conformidad con las disposiciones de la Constitución de la República y la Ley respectiva, corresponde a las universidades y escuelas politécnicas impartir la educación superior.

**Art. 17.-** El Ministerio de Educación establecerá mecanismos de coordinación con otros organismos del Estado o instituciones ajenas al mismo, que mantengan programas y servicios educativos.

## **Ley de la Propiedad Intelectual**

### **Sección II**

#### **Objeto del Derecho de Autor**

**Art. 8.** La protección del derecho de autor recae sobre todas las obras del ingenio, en el ámbito literario o artístico, cualquiera que sea su género, forma de expresión, mérito o finalidad. Los derechos reconocidos por el presente Título son independientes de la propiedad del objeto material en el cual está incorporada la obra y su goce o ejercicio no están supeditados al requisito del registro o al cumplimiento de cualquier otra formalidad.

Las obras protegidas comprenden, entre otras, las siguientes:

Libros, folletos, impresos, epistolarios, artículos, novelas, cuentos, poemas, crónicas, críticas, ensayos, misivas, guiones para teatro, cinematografía, televisión, conferencias, discursos, lecciones, sermones, alegatos en derecho, memorias y otras obras de similar naturaleza, expresadas en cualquier forma; Colecciones de obras, tales como antologías o compilaciones y bases de datos de toda clase, que por la selección o disposición de las materias constituyan creaciones intelectuales, sin perjuicio de los derechos de autor que subsistan sobre los materiales o datos;

Proyectos, planos, maquetas y diseños de obras arquitectónicas y de ingeniería;

Ilustraciones, gráficos, mapas y diseños relativos a la geografía, la topografía, y en general a la ciencia;



***Programas de ordenador; y,***

Adaptaciones, traducciones, arreglos, revisiones, actualizaciones y anotaciones; compendios, resúmenes y extractos; y, otras transformaciones de una obra, realizadas con expresa autorización de los autores de las obras originales, y sin perjuicio de sus derechos.

**Sección V**

**Disposiciones Especiales sobre ciertas Obras**

**Parágrafo Primero**

**De los Programas de Ordenador**

**Art. 28.** Los programas de ordenador se consideran obras literarias y se protegen como tales. Dicha protección se otorga independientemente de que hayan sido incorporados en un ordenador y cualquiera sea la forma en que estén expresados, ya sea en forma legible por el hombre (código fuente) o en forma legible por máquina (código objeto), ya sean programas operativos y programas aplicativos, incluyendo diagramas de flujo, planos, manuales de uso, y en general, aquellos elementos que conformen la estructura, secuencia y organización del programa.

**Art. 29.** Es titular de un programa de ordenador, el productor, esto es la persona natural o jurídica que toma la iniciativa y responsabilidad de la realización de la obra.

Se considerará titular, salvo prueba en contrario, a la persona cuyo nombre conste en la obra o sus copias de la forma usual.

Dicho titular está además legitimado para ejercer en nombre propio los derechos morales sobre la obra, incluyendo la facultad para decidir sobre su divulgación.

El productor tendrá el derecho exclusivo de realizar, autorizar o prohibir la realización de modificaciones o versiones sucesivas del programa, y de programas derivados del mismo.

Las disposiciones del presente artículo podrán ser modificadas mediante acuerdo entre los autores y el productor.



**Art. 30.** La adquisición de un ejemplar de un programa de ordenador que haya circulado lícitamente, autoriza a su propietario a realizar exclusivamente:

Una copia de la versión del programa legible por máquina (código objeto) con fines de seguridad o resguardo;

Fijar el programa en la memoria interna del aparato, ya sea que dicha fijación desaparezca o no al apagarlo, con el único fin y en la medida necesaria para utilizar el programa; y,

Salvo prohibición expresa, adaptar el programa para su exclusivo uso personal, siempre que se limite al uso normal previsto en la licencia. El adquirente no podrá transferir a ningún título el soporte que contenga el programa así adaptado, ni podrá utilizarlo de ninguna otra forma sin autorización expresa, según las reglas generales.

Se requerirá de autorización del titular de los derechos para cualquier otra utilización, inclusive la reproducción para fines de uso personal o el aprovechamiento del programa por varias personas, a través de redes u otros sistemas análogos, conocidos o por conocerse.

**Art. 31.** No se considerará que existe arrendamiento de un programa de ordenador cuando éste no sea el objeto esencial de dicho contrato. Se considerará que el programa es el objeto esencial cuando la funcionalidad del objeto materia del contrato, dependa directamente del programa de ordenador suministrado con dicho objeto; como cuando se arrienda un ordenador con programas de ordenador instalados previamente.

**Art. 32.** Las excepciones al derecho de autor establecidas en los artículos 30 y 31 son las únicas aplicables respecto a los programas de ordenador.

Las normas contenidas en el presente Parágrafo se interpretarán de manera que su aplicación no perjudique la normal explotación de la obra o los intereses legítimos del titular de los derechos.



## CAPÍTULO III

### INVESTIGACIÓN CIENTÍFICA

#### 3.1. Tipos de Investigación.

##### 3.1.1. Investigación Cuantitativa.

La investigación Cuantitativa es aplicable al proyecto del Sistema de Control de Asistencia INDESIC, ya que al mismo nos permitirá realizar un análisis y una síntesis de la información a través de libros escritos, documentos, manuales técnicos, e incluso la internet, con este proceso analítico sintético se pondrá en práctica los conceptos doctrinarios aprendidos en clase.

En el desarrollo del sistema mencionado anteriormente se investigará en cada uno de los procesos, tanto para la captura como para la verificación de la huella dactilar; las acciones, los eventos que genera el dispositivo.

Mediante el análisis de la teoría nos permitirá conocer, interpretar, comparar y enfocar distintos criterios a fin de obtener de una manera práctica en la programación del sistema.

##### 3.1.2. Investigación Bibliográfica Documental.

Este tipo de investigación se aplica al proyecto ya que mediante él se podrá realizar un tipo de investigación analítica y sintética de libros, manuales técnicos, documentos escritos los mismos que permitirán conocer, interpretar, comparar, y enfocar criterios y opiniones validas en el manejo y estructuración de los datos; realizar todo este proceso lleva consigo el poder materializar los conceptos obtenidos en los documentos anteriormente mencionados y esquematizarlos en el diseño y modelo de la base datos, la



base esencial de la estructuración del modelo del negocio que se debe que construir para éste posteriormente desarrollarlo y obtener un sistema de información.

Igual tratamiento se lo realizará con la información que se adquiriera a través de la Web poniendo mucho énfasis en las conceptualizaciones para construir conclusiones y recomendaciones validas en el desarrollo del proyecto; cabe recalcar que todo esto es con la finalidad de poder orientar de mejor manera el proceso de la teoría con la integración en la práctica a través del desarrollo de sistemas informáticos.

### **3.1.3. Investigación de Campo.**

La necesidad de contar con un del Sistema Biométrico para el Control de Asistencia en INDESIC, y tomando en consideración la demora para la entrega de informes/reportes de asistencia, se concluye en que implementar este proyecto ahorrará a la empresa la inversión de tiempo y recurso humano que hasta la fecha se lo viene realizando por la forma manual de elaborar los informes.

### **3.1.4. Investigación Descriptiva.**

Este tipo de investigación se aplica al proyecto en la utilización de la búsqueda de soluciones a la problemática planteada iniciando con la estructuración de explicaciones lógicas y definidas al árbol de problemas que se tiene que analizar a fin de poder encontrar las causas que originaron los distintos eventos y estos a su vez desencadenaron la problemática general; cabe recalcar que el enfoque general del tema presenta condiciones y características reales las mismas que pueden medirse y cuantificarse en cualquier momento.

La utilización del análisis la síntesis en el proyecto y con este tipo de investigación permitirá a los procesos y procedimientos enmarcados, tener una correcta descripción en los diferentes casos de uso y diagramas, sincronizando con todo el resto de elementos que tendrán que realizarse en el desarrollo del software planteado.



### **3.1.5. Investigación Explicativa.**

Este tipo de investigación se aplica desde el punto de vista del análisis de la problemática que se desarrolla con aristas predominantes en interrogantes que deberán resolver previa investigación analítica y sintética del tema. En este sentido el INDESIC tiene visión progresista y están en la búsqueda constante de mejorar cada día los procedimientos que ayudan a la completa realización de las funciones a cabalidad y dentro de estas herramientas se revisó cada uno de los manuales internos y el respectivo reglamento institucional.

Finalmente con esta investigación se verificará los impactos que puedan causar hacia los usuarios que van a utilizar el sistema. Así mismo se podrá constatar si los datos y documentos que genera el sistema están ordenados lógicamente en los reportes.

## **3.2 Métodos de Investigación.**

### **Métodos Teóricos.**

#### **3.2.1. Método Inductivo.**

Este método va de lo particular a lo general y nos permite transformar los datos en información válida ya que mediante la utilización propia de la investigación científica como la observación, entrevista, encuesta y/o aplicación de formularios se puede llegar a determinar claramente todos los procesos y a materializar un esquema ordenado que permita darnos cuenta cual es la necesidad real de la aplicación de este proyecto; posteriormente ayudará en el desarrollo mismo que mediante la automatización de los procesos el sistema será capaz de poder almacenar la información en una base de datos para que luego la misma sea consultada cuando se la necesite.

#### **3.2.2. Método Deductivo.**

Para el desarrollo e implementación del Sistema Biométrico para el Control de Asistencia en INDESIC. El proceso empieza con la creación de un empleado, la asignación del código, la creación de horarios de empleados. Mediante el acceso a un menú del sistema



contable y escogiendo las opciones necesarias se puede realizar los roles de pago de los empleados. Extrayendo información de la base de datos podremos obtener los reportes requeridos para la presentación de informes estadísticos sobre las entradas y salidas de cada uno de los empleados en un periodo determinado.

### **3.2.3. Método Analítico – Sintético.**

Este método de investigación científica es aplicable en el proyecto ya que el mismo permitirá realizar un análisis detallado de la información obtenida y discernir al mismo tiempo; de tal manera que para la creación y asignación de los respectivos horarios ya se obtenga las pautas necesarias a fin de ir construyendo paso a paso los hechos o fenómenos ocurridos, cabe recalcar que también reflejan las causas que originaron la problemática estudiada anteriormente y además permitirá que toda la información mediante el análisis y la síntesis sea depurada (los registros de asistencia son únicos con la huella dactilar) a fin de que la misma se vuelva veraz, confiable.

### **3.2.4. Método Hipotético- Deductivo.**

Este método tiene su aplicación en la generación de nuevos conocimientos a través de otros establecidos y se da especialmente en la aplicación de los lenguajes de programación ya que al momento de generar código fuente se establecerán errores de los cuales obtendremos nuevas ideas en el establecimiento de un sistema o programa.

Igualmente es de mucha aplicación en el ambiente de Networking por el surgimiento de ataques a equipos por parte de virus y personal mal intencionado, los mismos que me orientarán a desarrollar nuevas tácticas y estrategias de protección perimetral.

### **3.2.5. Método Histórico Lógico.**

Está vinculado al conocimiento en las distintas etapas del desarrollo del sistema (modelación, el análisis, Diseño y la programación del software), mediante su secuencia cronológica. Con el método histórico se analiza la evolución concreta de los sistemas biométricos y sus ventajas en los diferentes períodos de la historia. El estudio final determinará cuál de los siguientes dispositivos se instalará en la empresa, su



configuración y como se programará la aplicación. Así mismo La estructura lógica del objeto implica la migración de datos al sistema contable SMA e integración a otros sistemas.

### **3.3. Técnicas de Recolección de Información.**

Para la recolección de datos primarios en donde se va a implementar el Sistema Biométrico para el Control de Asistencia en INDESIC se procede básicamente por la observación, por las encuestas o las entrevistas a los sujetos de estudio y por experimentación.

#### **3.3.1. La Observación.**

Esta técnica de investigación se aplica al proyecto con el fin de obtener datos principales acerca de la organización los mismos que me permitirán realizar una observación de la infraestructura, ambiente de trabajo, departamentos existentes, y en fin datos iniciales que determinen el tipo de organización tecnológica en la que se va a desarrollar el sistema, para la ejecución de esta técnica de recolección de información es necesario estar presente en el lugar donde se va efectuar el levantamiento de información por tanto se hizo un análisis visual de todas las áreas: “Sistemas”, “Departamento Financiero”, “Dirección Ejecutiva”, “Comunicación”, y “Desarrollo de Proyectos”, áreas en donde se va a obtener la información y por las relaciones que existen entre cada departamento.

#### **3.3.2. La Entrevista.**

La entrevista es un instrumento importante de la investigación, misma que fue realizada INDESIC de la ciudad de Quito, para obtener la información por parte del encuestado acerca de existe la posibilidad de desarrollar el Sistema desarrollo e implementación de un sistema biométrico para el control de asistencia, así mismo al ser parte de la institución ya se conoce la realidad de la empresa, las políticas y procesos a seguir para el buen desenvolvimiento del proyecto.



### 3.3.2.1. Tipos de Entrevista.

- **Nivel ejecutivo.**

**¿Estaría usted dispuesto a auspiciar nuestro proyecto en su empresa?**

**R:** Efectivamente al ser una organización y disponer de varios proyectos en donde algunos colaboradores y pasantes están bajo nuestra tutela se requiere este sistema para que nos ayude con este control.

**Análisis:** después de tener la reunión se miró el interés por el auspicio en dicho proyecto que se solicitó es así que se inició con todos los procesos en el desarrollo.

**¿Está la empresa dispuesta a brindarnos la información requerida para el desarrollo de este sistema?**

**R:** La información es confidencial pero al ser parte del personal interno obviamente se le brindará el apoyo necesario.

**Análisis:** las facilidades que se ofrece la institución por ser parte de ella nos facilita el trabajo de la recolección de datos.

**¿La empresa autorizaría a utilizar sus instalaciones para el desarrollo de este proyecto?.**

**R:** Esta es una institución de privada en donde se puede hacer prácticas y nuestra infraestructura si nos permite habilitar un lugar para desarrollo.

**Análisis:** El desarrollo del proyecto se realizara como un anexo de las labores diarias de nuestro trabajo.

**¿Existe la facilidad de la empresa en reuniones para poder socializar el proyecto?**

**R:** En la próxima reunión del directorio se le solicita este presente para que exponga los beneficios del proyecto planteado con la finalidad informar.

**Análisis:** Se prepara un diseño prototipo de la aplicación y la presentación respectiva para dicha exposición.

**¿Si en la implementación de este sistema se requiere el cambio de tecnología estaría usted dispuesto a invertir en dicho cambio?**



Como conoce estamos adquiriendo nuevos equipos y si para mejorar estos procesos requiere inversión no tenemos ningún problema en adquirir nuevos equipos tecnológicos.

**Análisis:** Hay total apertura presupuestaria para la adquisición de la tecnología y equipos.

- **Nivel Operativo.**

**¿Cree usted que al implantar un sistema para el control de asistencia mejorará el rendimiento del trabajo en su departamento?**

Efectivamente en la actualidad los cálculos de la horas laboradas son tediosas y lentas lo que dificulta muchas veces el pago de los roles a su debido tiempo.

- **Nivel Técnico.**

**¿Existe la infraestructura para instalar el sistema planteado?**

No totalmente siempre creo que faltara algún dispositivo para el sistema pero la apertura por parte de la administración en la adquisición de nuevos equipos es muy importante.

**¿Existe personal capacitado para el manejo de la información generada por el sistema?**

Creo que después de la capacitación respectiva se manejará la información de ingreso y salida de los datos.



## CAPÍTULO IV

### DESARROLLO DE LA PROPUESTA

#### 4.1. Diagnóstico Situacional

En la Institución existe varias líneas de trabajo, para un uso básico, en un 100% son clones ensamblado, de arquitectura CISC, con Procesadores Intel / Dual Core de 32 Bits y 64 Bits. Respecto a software se establece un equipo de office, para el procesar la enseñanza – aprendizaje. Para navegar por la web se utiliza en un 80% Internet Explorer 8, el restante utiliza Mozilla Firework.

Actualmente existe una red LAN y WAN, la topología es en forma de estrella. El servicio de Internet se lo da a conocer, mediante configuraciones específicas de un router, conectado hacia un switch y este distribuye a todas las estaciones de trabajo. El proveedor de servicios es CNT.

Por ser una institución educativa de carácter social, no existen seguridades periféricas estrictas para la DMZ. El Firewall utilizado es el propio del sistema operativo Windows y los Antivirus como Avira, Nod32 y el acceso a ciertas páginas web se las restringe con el firmware del router.

No se asegura un plan informático creado, que asegure o preserve la información transmitida, por los diferentes procesos detallados.

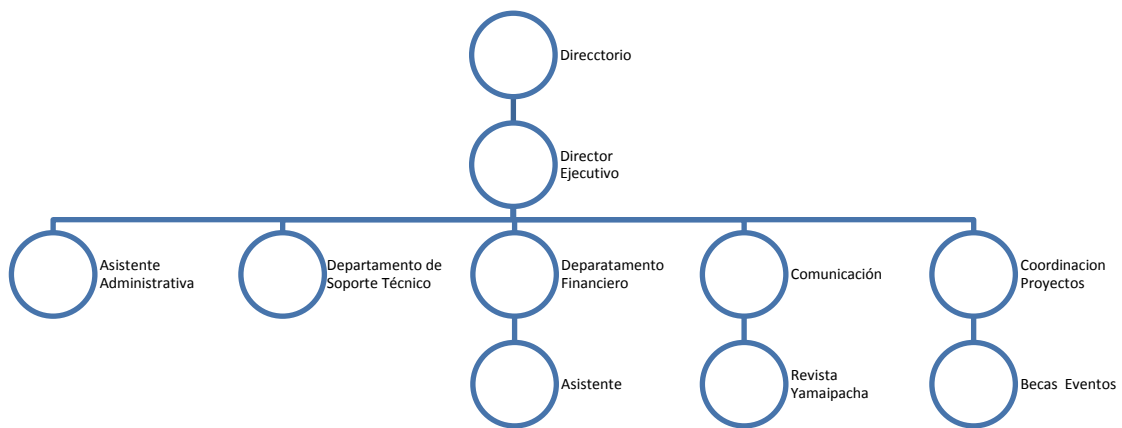
#### 4.2. Estructura Organizacional.

Es de vital importancia pues a través del que se dan a conocer los diversos puestos que existen en la institución, el nombre de cada uno de ellos, así como también el nombre de

quienes ocupan cada cargo. De esta forma se determina como son las líneas de mando en cualquier organización.

El organigrama presentado en el Instituto para el Desarrollo Social y de las Investigaciones Científicas "INDESIC", mantiene una estructura jerárquica, basada en funciones. Existen departamentos definidos y también áreas en las cuales el personal esta puede cambiar de actividad por ser una organización indígena que promueve la igualdad de trabajo.

En el siguiente diagrama se detalla la Organización Institucional.



**Gráfico 5** Organigrama Organizacional

**Fuente:** INDESIC

**Autor:** Dirección Ejecutiva INDESIC

### 4.3. Infraestructura Informática.

La institución ha invertido buena cantidad de tiempo y dinero para tener de infraestructura nueva y acorde para el desarrollo de sus actividades. Es por eso que el departamento de Tecnología Informática ha podido optimizar todos los recursos para que trabajen con mayor eficiencia y agregando funcionalidad a la infraestructura sin tener que desbaratar lo que ya está instalado.



#### 4.3.1. Hardware.

Cada uno de los equipos que se encuentran instalados en los diferentes departamentos tiene las siguientes características:

Características	Ubicación	Arquitectura
Intel Dual Core I3	Financiero	CISC
Intel Dual Core I3	Asistente Administrativa	CISC
Intel Aton	Dirección Ejecutiva	CISC
Intel Pentium IV	Laboratorio	CISC
Intel Pentium IV	Laboratorio	CISC
Intel Pentium IV	Laboratorio	CISC
Intel Pentium IV	Laboratorio	CISC
Intel Pentium IV	Laboratorio	CISC
Intel Core I7	Servidor Aplicaciones	CISC
Intel Pentium IV	Servidor Internet	CISC
Intel Core I3	Coordinadora proyectos	CISC
Intel Pentium IV	Asistente Proyectos	CISC
IMac Core i3	Comunicación	IBM

**Tabla 1** Cuadro de Distribución de Hardware

**Autor:** Jhonny Suqui

**Fuente:** Estudio y Verificación del Desarrollador.

#### 4.3.2. Software

Una de las deficiencias grandes de la organización es no poseer licencias del software debido que próximamente se migrará al software libre como política de todas las empresas públicas.

Software	Ubicación
Office 2007	Laboratorio, Asistente Proyectos

Office 2010	Estaciones de Trabajo, Secretaria, Dirección Ejecutiva, Servidores
Windows XP sp3	Estaciones de Trabajo, Asistente Proyectos, Dirección
Windows 7	Personal, financiero, Asistente Proyectos
Mozilla Firefox	Todos los departamentos
Internet explore	Todos los departamentos
SMA	Servidor, Financiero, Asistente
Adobe	CC3 y CS4, CS5, Comunicación, Soporte

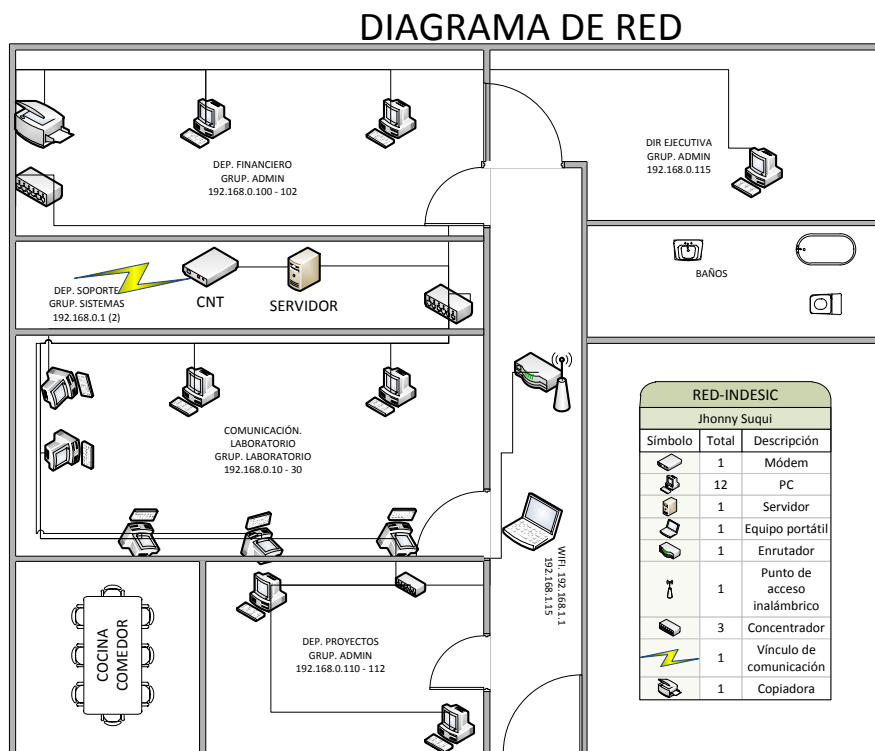
**Tabla 2** Cuadro de Software Instalado

**Autor:** Jhonny Suqui

**Fuente:** Estudio y Verificación del Desarrollador.

### 4.3.3. Comunicaciones.

El INDESIC, dispone de tres tipos de redes LAN, WAN y WLAN con topología tipo estrella y las clases de redes privadas y configuradas son tipo C mientras que la pública tipo A. Así mismo los equipos se detallan a continuación. **Ver diagrama (6)**



**Gráfico 6** Diagrama de Red INDESIC

**Autor:** Jhonny Suqui

**Fuente:** Estudio y Verificación del Desarrollador.



Equipo	Ubicación
Router	Redes
Switch	Redes
Cable UTP Cat 5	Redes
Conectores RJ45	Redes
Tarjetas de Red	Redes

**Tabla 3** Cuadro de Conectividad

**Autor:** Jhonny Suqui

**Fuente:** Estudio y Verificación del Desarrollador.

#### 4.3.4. Recurso Humano Técnico.

Las siguientes personas que intervienen el desarrollo están involucradas en el trabajo de la empresa y por lo tanto conocen sobre los procesos de la organización y la parte técnica.

Nombre	Descripción
Tlgo: Jhonny Suqui	Autor de Proyecto de grado
Ing. Kleber Naula	Director Ejecutivo INDESIC

**Tabla 4** Cuadro del personal Técnico

**Autor:** Jhonny Suqui

**Fuente:** Estudio y Verificación del Desarrollador.

#### 4.4. Descripción de Alternativas.

A continuación se realiza la respectiva descripción de cada una de las alternativas propuestas para este proyecto, mismas que serán analizadas por el personal técnico de la empresa.

##### 4.4.1. Alternativa # 1.

En la siguiente alternativa enviada por la empresa *A TIEMPO.COM Ecuador S.A.* se analiza los siguientes puntos mencionados a continuación:

**Ventajas:** Tiempo de entrega y la solvencia en el mercado con grandes empresas públicas y privadas.



**Desventajas:** El costo del sistema por instalación mantenimiento y garantía son muy elevados ya que tiene varios empleados.

ÍTEMS	CUMPLE	NO CUMPLE
Plataforma: 2 Capas	X	
Metodología : RUP		X
Modelos: Lógico, Físico, Script	X	
Front End: Visual C#	X	
Back End : MySql	X	
<b>Pruebas:</b>		
Unidad	X	
Integración	X	
Validación	X	
Sistema	X	
Seguridad	X	
Interface	X	
<b>Estándares</b>		
Diseño		X
Programación	X	
Bases de datos	X	
Elaboración del Sistema biométrico.	X	
Implementación E- Learning Moodle	X	
Implementación Sistema biométrico	X	
Capacitación Técnica	X	
Capacitación Usuario	X	
Manuales Técnicos	X	
Documentación	X	
Garantía Técnica	X	

**Tabla 5** Cuadro 1 de Alternativas

**Autor:** Jhonny Suqui

**Fuente:** Proforma Empresa A tiempo.

**Proforma:**

- **Costo:** USD. \$ 1500.00 + IVA(no incluye dispositivo USB- TC/IP)
- **Tiempo:** 4 meses.
- **Garantía:** 6 meses.
- **Entrega:** Solo software con aplicación.
- **Con Fuentes:** Adicional USD. \$ 800.00 + IVA.

**4.4.2. Alternativa # 2.**

En la siguiente propuesta enviada por la empresa **Esuman Cia. Ltda** tiene similares características pero las más relevantes tenemos.

**Ventajas:** Se ajusta a todos los requerimientos (módulos) de quien solicita, ya sea para para grandes o pequeñas empresas.

**Desventajas:** No dispone de garantía de los equipos así mismo los pagos por el servicio no tienen crédito.

ÍTEMS	CUMPLE	NO CUMPLE
Plataforma: 2 Capas	X	
Metodología : RUP		X
Modelos: Lógico, Físico, Script	X	
Front End: .NET	X	
Back End : MySql	X	
<b>Pruebas:</b>		
Unidad		X
Integración	X	
Validación	X	
Sistema	X	
Seguridad	X	
Interface	X	
<b>Estándares</b>		



Diseño	X	
Programación	X	
Bases de datos	X	
Elaboración del Sistema biométrico		X
Implementación E- Learning Moodle	X	
Implementación Sistema Biométrico	X	
Capacitación Técnica	X	
Capacitación Usuario	X	
Manuales Técnicos	X	
Documentación	X	
Garantía Técnica	X	

**Tabla 6** Cuadro 2 de Alternativas**Autor:** Jhonny Suqui**Fuente:** Proforma Empresa A tiempo.**Proforma:**

- **Costo:** USD. \$ 1350.00 (Dispositivos no incluyen).
- **Tiempo:** 3 meses.
- **Garantía:** 6 meses.
- **Entrega:** Solo programa instalado y funcionando.
- **Con Fuentes:** No incluye.

**4.4.3. Alternativa # 3.**

En la última propuesta presentada por Jhonny Suqui como proponente y desarrollador del proyecto de destaca algunos datos:

**Ventajas:** Al ser un sistema desarrollado por parte de un proyecto y auspiciado por la empresa los costos son mínimos, la garantía, el código fuente y, la capacitación entre otros no tienen ningún costo.

**Desventajas:** La experiencia en el desarrollo de sistemas frente a las alternativas 1 y 2 es muy larga y por último los dispositivos (biométricos) utilizados son muy diferentes.



ÍTEMS	CUMPLE	NO CUMPLE
Plataforma: 3 Capas.	X	
Metodología: RUP.	X	
Modelos: Lógico, Físico, Script.	X	
Front End: .Net. C#	X	
Back End :SQL Server 2008	X	
Pruebas:		
Unidad		X
Integración	X	
Validación	X	
Sistema	X	
Seguridad	X	
Interface.	X	
Estándares.		
Diseño.	X	
Programación.	X	
Bases de datos.	X	
Elaboración del Sistema biométrico.	X	
Implementación Sistema biométrico.	X	
Capacitación Técnica.	X	
Capacitación Usuario.	X	
Manuales Técnicos.	X	
Documentación.	X	
Garantía Técnica.	X	

**Tabla 7:** Cuadro 3 de Alternativas

**Autor:** Jhonny Suqui

**Fuente:** Desarrollador.

**Proforma:**

- **Costo:** USD. \$ 500 (solo por Dispositivos)
- **Tiempo:** 6 meses.



- **Garantía:** 1 Año.
- **Entrega:** Con fuentes y sin ningún valor adicional.

#### 4.5. EVALUACIÓN Y SELECCIÓN DE ALTERNATIVAS.

Con la finalidad de poder determinar la alternativa que ofrece mayores beneficios para la Institución se ha definido un sistema de calificación con coeficientes matemáticos que están determinados de la siguiente manera:

<b>Técnico.</b>	70%.
<b>Económico.</b>	20%.
<b>Soporte Técnico.</b>	10%.

Así mismo en criterio para la evaluación de las alternativas sobre cada uno de los valores tendrán una escala del 1 al 10 y su equivalente como se describe a continuación:

CRITERIO DE EVALUACIÓN	
Valor	Calificación
10	Excelente
8	Muy Bueno
5	Bueno
3	Regular
2	Malo

**Tabla 8:** Criterios de Evaluación

**Autor:** Jhonny Suqui

**Fuente:** Desarrollador.

CALIFICACIÓN TÉCNICA				
CRITERIO	PESO	ALTERNATIVA	ALTERNATIVA	ALTERNATIVA
		1	2	3
<b>Técnico.</b>	<b>70</b>	<b>66</b>	<b>57</b>	<b>68</b>
Plataforma: 3 Capas.	10	8	8	10
Metodología: RUP.	10	9	7	10
Modelamiento: Físico, Lógico, Script.	10	10	9	10



Pruebas, Validación, Seguridad.	10	10	7	10
Diseño.	10	9	8	10
Programación.	10	10	9	8
Bases de datos.	10	10	9	10

**Tabla 9:** Cuadro de Evaluación Técnica

**Autor:** Jhonny Suqui

**Fuente:** Desarrollador.

Realizando la evaluación definitiva de las 3 propuestas presentadas se puede concluir claramente que:

La empresa 1 alcanzo los siguientes resultados: En la parte técnica sumo 66 puntos por lo que se le asigna un puntaje de 66%.

La empresa 2, alcanzo los siguientes resultados: En la parte técnica sumo 57 puntos por lo que se le asigna un puntaje de 57%.

La empresa 3, alcanzo los siguientes resultados: En la parte técnica sumo 68 puntos por lo que se le asigna un puntaje de 68%.

Para la evaluación económica se tomara en cuenta los siguientes parámetros

CRITERIO DE EVALUACIÓN				Inversión de %
PESO	Equivalente	Costo	%	> 10
5	Máximo	\$1,500.00	20%	< 16
0	Mínimo	\$ 0	0%	<>12

**Tabla 10:** Criterio de Evaluación Económica.

**Autor:** Jhonny Suqui.

**Fuente:** Desarrollador.

CALIFICACIÓN ECONÓMICA				
ÍTEMS	PESO	ALTERNATIVA 1	ALTERNATIVA 2	ALTERNATIVA 3
<b>ECONÓMICO</b>				
Costo por sistema	5.00	\$ 1,500.00	\$ 1,350.00	\$ -
Costo Dispositivos	5.00	\$ -	\$ 500.00	\$ 300.00
Costo Fuentes	5.00	\$ 800.00	\$ 600.00	\$ -



Costos Capacitación	5.00	0	\$ -	\$ -
<b>TOTAL</b>	20.00			

**Tabla 11:** Cuadro de costos Económicos.

**Autor:** Jhonny Suqui.

**Fuente:** Desarrollador.

<b>EVALUACIÓN ECONÓMICA</b>				
ÍTEMS	PESO	ALTERNATIVA 1	ALTERNATIVA 2	ALTERNATIVA 3
<b>ECONÓMICO</b>	<b>20.00%</b>	<b>30.60%</b>	<b>32.66%</b>	<b>4.00%</b>
Costo por sistema	5.00%	20.00%	18.00%	0.00%
Costo Dispositivos	5.00%	0.00%	6.66%	4.00%
Costo Fuentes	5.00%	10.60%	8.00%	0.00%
Costos Capacitación	5.00%	0.00%	0.00%	0.00%
Inversión de Criterios	Max 0 Min 5	12%	10%	16%

**Tabla 12:** Cuadro de Evaluación Económica.

**Autor:** Jhonny Suqui.

**Fuente:** Desarrollador.

Realizando la evaluación económica de las 3 propuestas y tomando en cuenta al menor valor como máximo porcentaje y dentro de los criterios se concluye que:

La empresa 1, alcanzo 30.6 puntos y se le asigna un puntaje de 12%.

La empresa 2, alcanzo 32.66 puntos y se asigna un puntaje de 10%.

La empresa 3, alcanzo 4 puntos y se le asigna un puntaje de 16%.

En la última evaluación sobre el soporte técnico se evaluara con los siguientes parámetros.

<b>CRITERIOS DE EVALUACIÓN</b>							
Valor	# Visitas	Garantía/ Tiempo	Equivalente	Valor	Criterio	Equivalente	%
Máximo	9	12 meses	2.5	1	Disponible	2.5	2%
Mínimo	2	6 meses	1.25	0	No Disponible	0	0%

**Tabla 13:** Criterios de Evaluación Tiempo.



**Autor:** Jhonny Suqui.  
**Fuente:** Desarrollador.

CALIFICACIÓN DE SOPORTE				
ÍTEMS	PESO	ALTERNATIVA 1	ALTERNATIVA 2	ALTERNATIVA 3
<b>SOPORTE TÉCNICO</b>				
Manuales Técnicos	2.50	1.00	1.00	1.00
N° de Visitas	2.50	7.00	9.00	2.00
Garantía Meses	2.50	12.00	6.00	12.00
Capacitación	2.50	1.00	1.00	1.00
<b>TOTAL</b>	<b>10.00</b>			

**Tabla 14:** Cuadro de Calificación Soporte  
**Autor:** Jhonny Suqui  
**Fuente:** Desarrollador.

EVALUACIÓN DE SOPORTE				
ÍTEMS	PESO	ALTERNATIVA 1	ALTERNATIVA 2	ALTERNATIVA 3
<b>SOPORTE TÉCNICO.</b>				
Manuales Técnicos.	2.50%	2.5	2.5	2.5
N° de Visitas	2.50%	1.95	2.5	0.56
Garantía.	2.50%	2.5	1.05	2.5
Capacitación.	2.50%	2.5	2.5	2.5
<b>TOTAL</b>	<b>10.00%</b>	<b>9.45</b>	<b>8.55</b>	<b>8.06</b>

**Tabla 15:** Cuadro de Evaluación Soporte.  
**Autor:** Jhonny Suqui.  
**Fuente:** Desarrollador.

Realizando la evaluación de sobre la garantía y soporte técnico de las 3 propuestas presentadas se puede concluir claramente que:

La empresa 1, alcanzo los siguientes resultados: En la parte de soporte sumo 9.45 puntos por lo que se le asigna un puntaje de 9.45%.

La empresa 2, alcanzo los siguientes resultados: En la parte soporte sumo 8.55 puntos por lo que se le asigna un puntaje de 8.55%.



La empresa 3, alcanzo los siguientes resultados: En la parte técnica sumo 8.06 puntos por lo que se le asigna un puntaje de 8.06%.

Después de analizar cada uno de los resultados en los cuadros anteriores se concluye los siguientes datos:

EVALUACIÓN DE ALTERNATIVAS				
		A	B	C
FACTOR RELEVANTE	PESO ASIGNADO	CALIFICACIÓN	CALIFICACIÓN	CALIFICACIÓN
Técnico	70%	66.00%	57.00%	68.00%
Económico	20%	12.00%	10.00%	16.00%
Soporte Técnico	10%	9.45%	8.55%	8.06%
<b>Suma</b>	<b>100%</b>	<b>87%</b>	<b>76%</b>	<b>92%</b>

Tabla 16: Cuadro de Evaluación de las Alternativas

**Autor:** Jhonny Suqui

**Fuente:** Desarrollador.

#### 4.5.1. ANÁLISIS E INTERPRETACIÓN DE LOS DE LOS RESULTADOS DE LA EVALUACIÓN.

**Técnico:** Las especificaciones técnicas del software representan la parte más importante de todo el proceso de selección de la alternativa más idónea. Se ha diseñado una matriz que contiene pesos cuantitativos de acuerdo a la importancia de cada uno de los elementos que intervienen en la mencionada alternativa, estos pesos sumarán una totalidad de 100 puntos lo que corresponderá al 70% de la parte técnica.

**Económico:** Para determinar el puntaje respectivo que le corresponde a la parte económica se define de la siguiente manera: según los criterios analizados con sus respectivos pesos cuantitativos la oferta más económica obtendrá 5 puntos, que corresponde al 20% de la parte económica.

**Soporte Técnico:** Está determinado por la calidad de técnicos que posea la empresa y además por la lista de clientes que esta tengan, esto se lo realiza para saber el nivel de conocimiento de la empresa y su porcentaje de aceptación en el medio se calificará del total del 10 puntos equivalente a 10% .

Interfaz.

Por lo tanto de las tres alternativas presentadas se selecciona con el mayor porcentaje dando como resultado la tercera alternativa.



#### **4.6. FACTIBILIDAD TÉCNICA.**

De lo expuesto anteriormente se desprende que la opción de desarrollo propuesta como proyecto de grado es la más conveniente en la parte técnica, económica, garantía y soporte técnico.

La misma establece que el aspecto técnico es la principal para poder realizar una calificación coherente y acertada, por otro lado el aspecto económico favorece la ejecución del proyecto, igual tratamiento nos indica lo referente al soporte técnico y garantía técnica.

Por consiguiente es factible la realización del proyecto con la alternativa de desarrollo propio, lo que implica que su soporte y ejecución será estrictamente con apoyo de la institución en todos los géneros que se pueda realizar (técnico, económico).

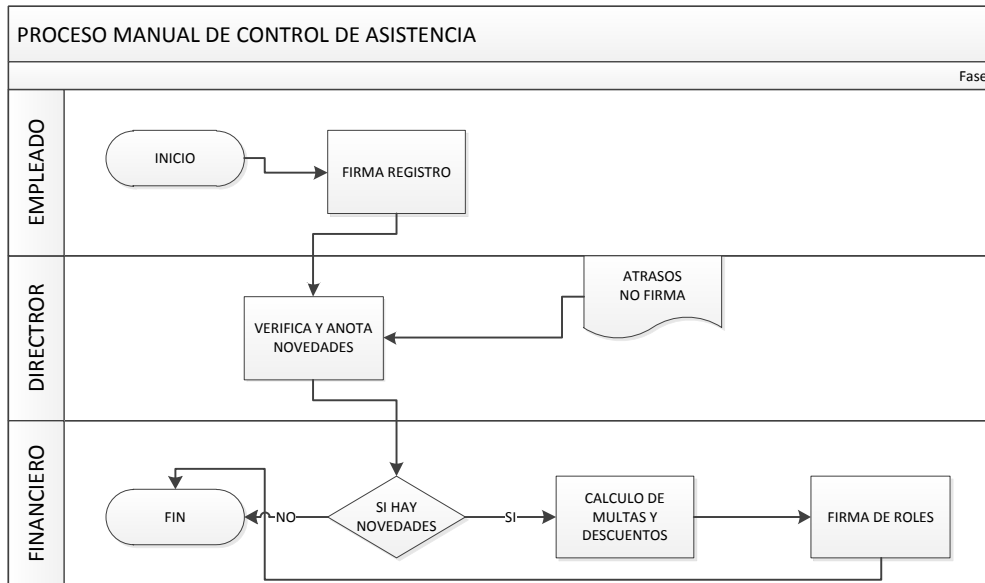
#### **4.7. DESCRIPCIÓN DE PROCESOS.**

**Actualmente se lo hace manual.**

1. El empleado ingresa y firma el al hoja de registro. Ingreso –Salida.
2. El director verifica diariamente el registro y anota novedades.
3. El departamento financiero al final del mes verifica los atrasos y novedades para sus pagos.

##### **4.7.1. Procesos del Sistema.**

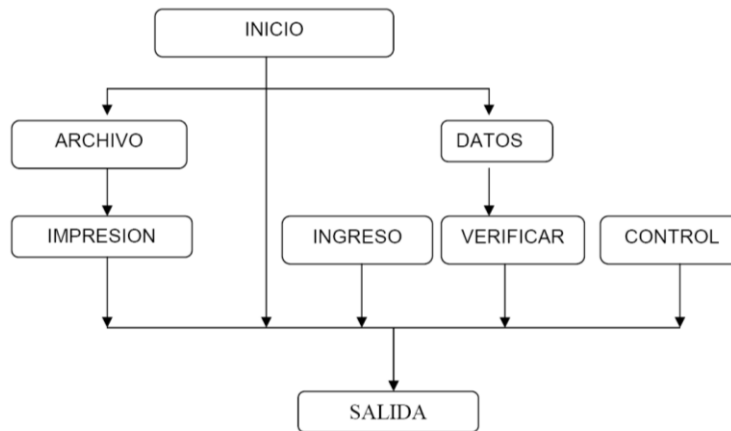
- a) Ingreso acceso al Sistema “Login”.
- b) Creación de Datos del Sistema Nombre de empresa, Cargos, Sueldos y otros.
- c) Creación de horarios.
- d) Asignación de horarios a empleados.
- e) Configurar dispositivo (Interno).
- f) Registro y enrolamiento de empleado con huella dactilar.
- g) Marcaciones de entrada y salida por huella dactilar.
- h) Reportes de asistencia.
- i) Migración de datos al sistema contable.



**Gráfico 7:** Proceso manual de control de Asistencia.

**Autor:** Jhonny Suqui

**Fuente:** Archivos Desarrollador e INDESIC.



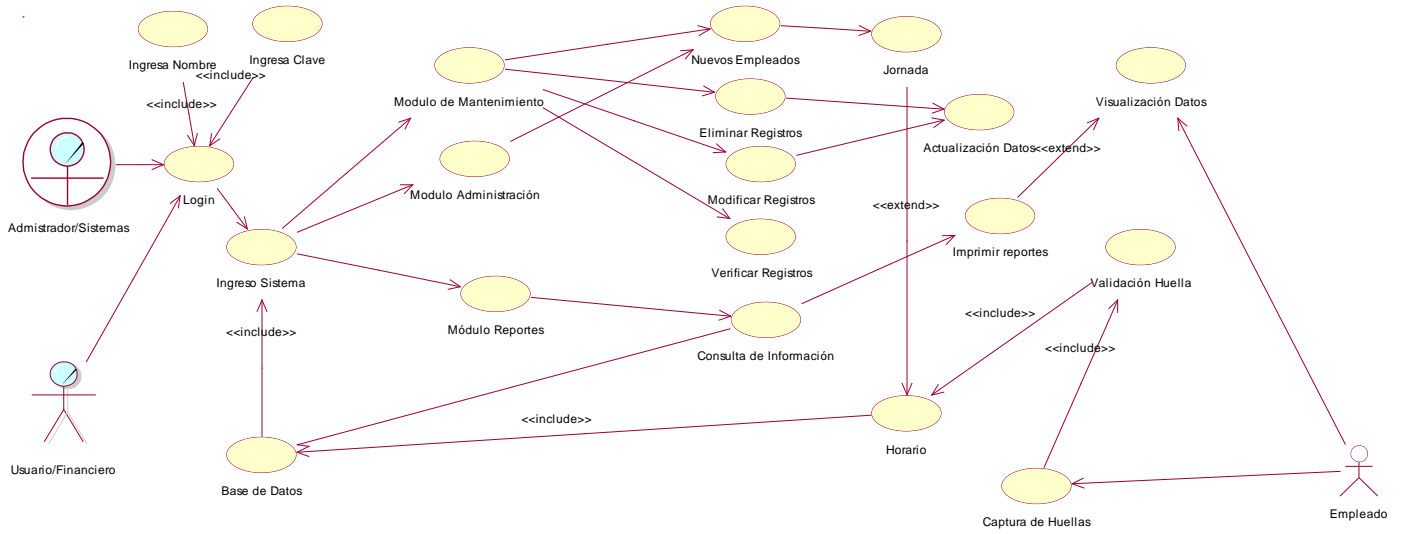
**Gráfico 8:** Diagrama de Flujo de procesos

**Autor:** Jhonny Suqui

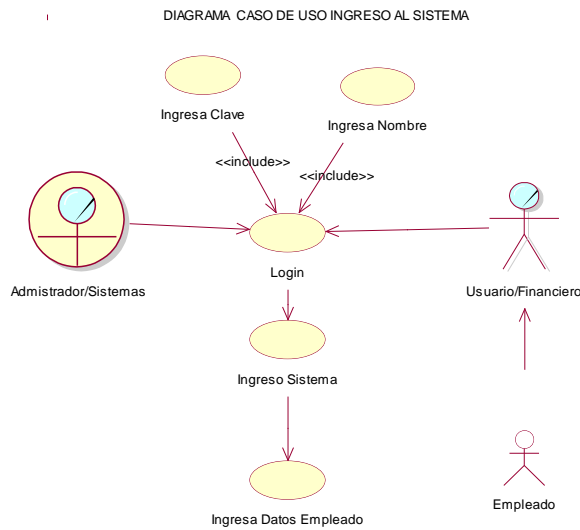
**Fuente:** Archivos Desarrollador e INDESIC.

### DIAGRAMAS DE CASO DE USO DIAGRAMAS DE CASO DE USO PARA EL USUARIO ADMINISTRADOR

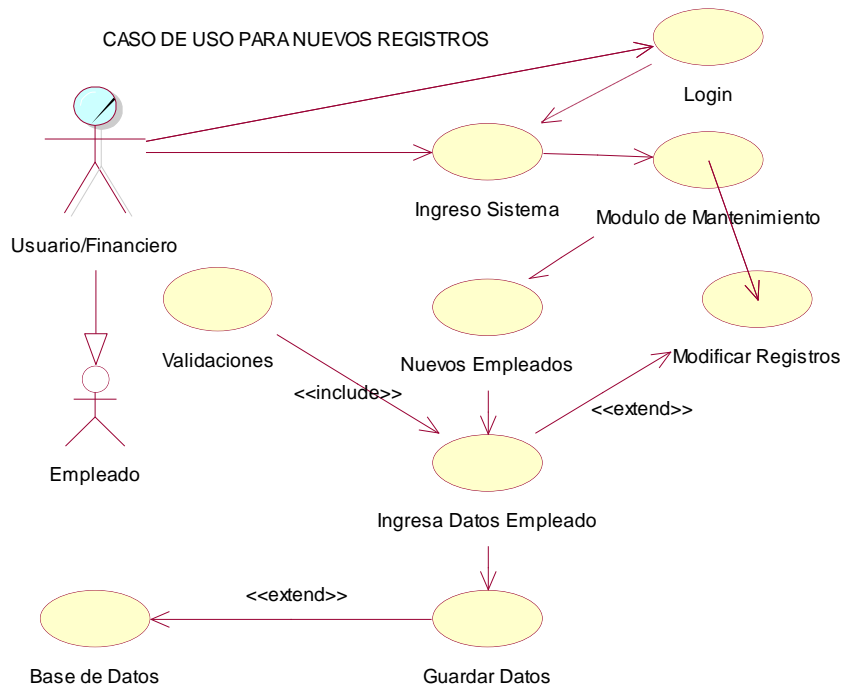
De acuerdo al siguiente esquema se destaca que el Administrador podrá realizar todas las actividades previstas en el sistema interno y contará con la posibilidad de consulta.



**Gráfico 9:** Diagrama General de Casos de Uso del Sistema  
**Autor:** Jhonny Suqui  
**Fuente:** Diseño de Base de Datos Desarrollador.



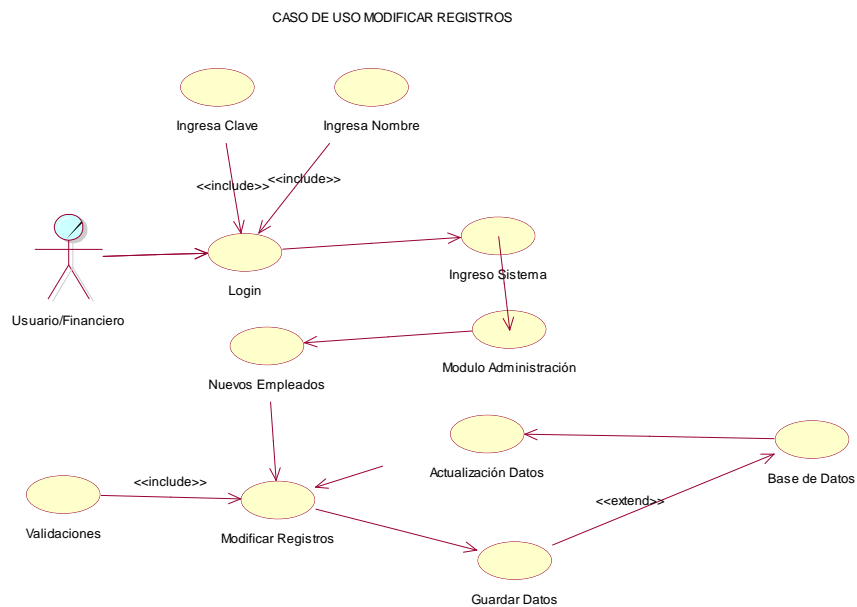
**Gráfico 10:** Diagrama de Casos de Uso Ingreso al Sistema.  
**Autor:** Jhonny Suqui.  
**Fuente:** Desarrollador del sistema.



**Gráfico 11:** Casos de Uso para Generar nuevos registros.

**Autor:** Jhonny Suqui.

**Fuente:** Diseño Base de Datos Diseñador.

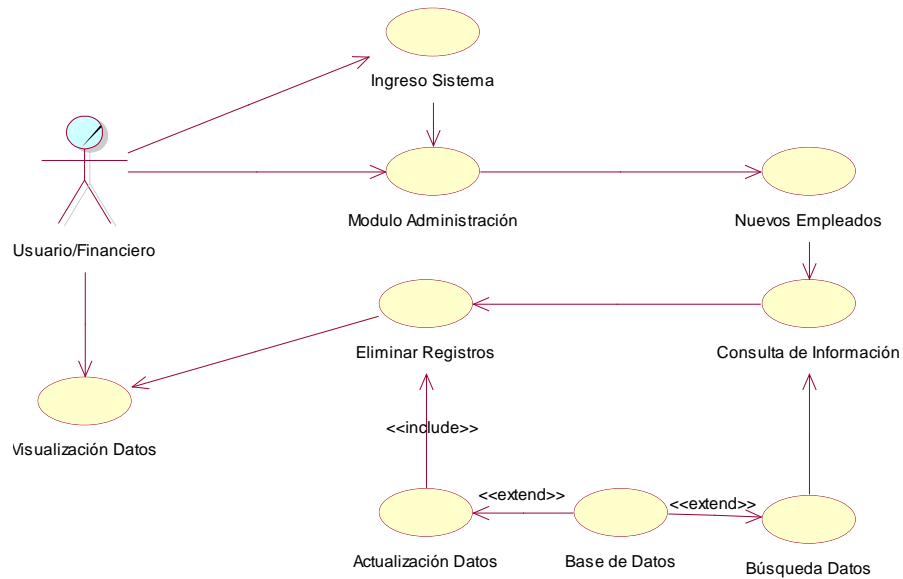


**Gráfico 12:** Caso de Uso para modificar los registros.

**Autor:** Jhonny Suqui.

**Fuente:** Diseño Base de Datos Diseñador.

CASO DE USO ELIMINAR REGISTROS

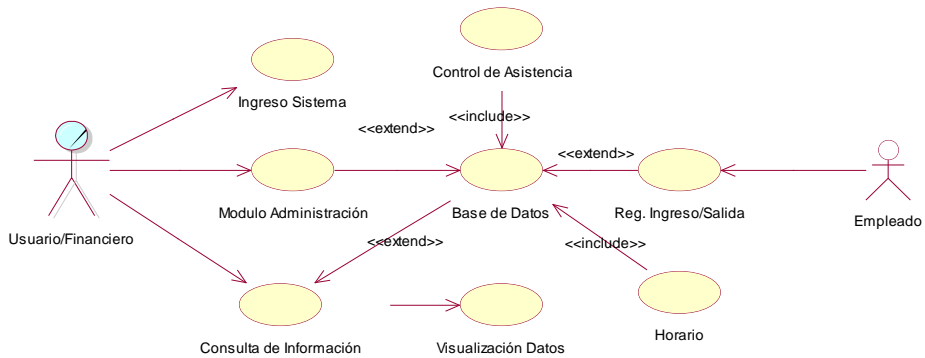


**Gráfico 13:** Caso de Uso para eliminar un registro.

**Autor:** Jhonny Suqui.

**Fuente:** Diseño Base de Datos Diseñador.

CASO DE USO CONTROL DIARIO

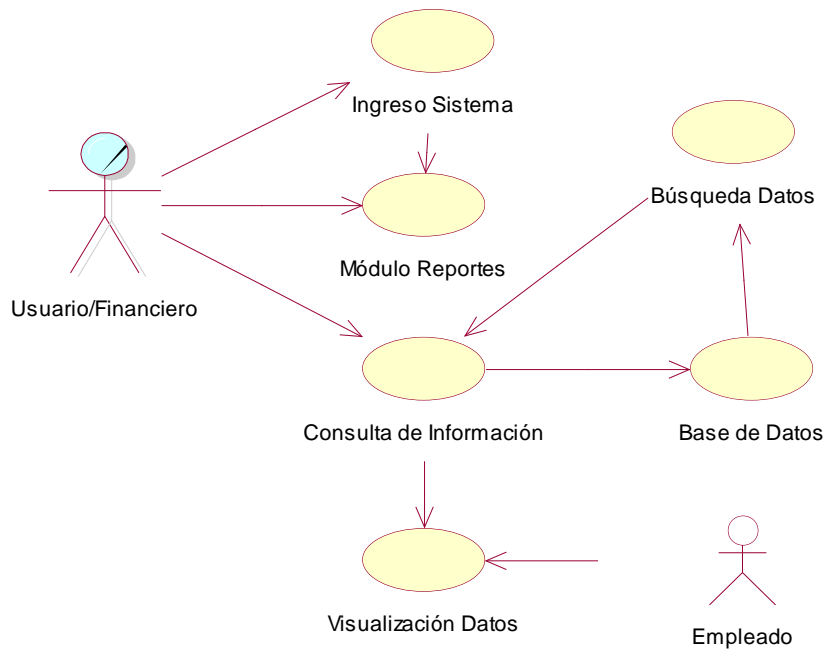


**Gráfico 14:** Caso de Uso para realizar el Control Diario.

**Autor:** Jhonny Suqui.

**Fuente:** Diseño Base de Datos Diseñador.

CASO DE USO IMPRIMIR REPORTES

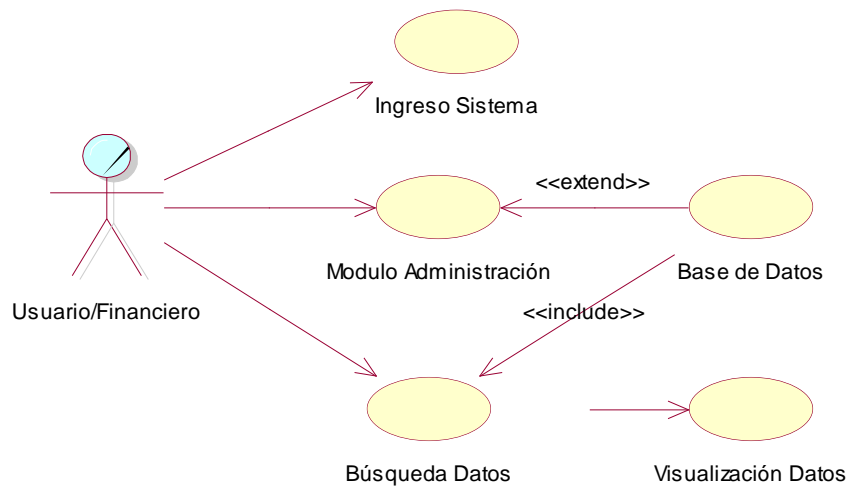


**Gráfico 15:** Caso de Uso para Imprimir los Reportes

**Autor:** Jhonny Suqui

**Fuente:** Diseño Base de Datos Diseñador

CASO DE USO VERIFICAR REGISTROS



**Gráfico 16:** Caso de Uso Para verificar los Registros

**Autor:** Jhonny Suqui

**Fuente:** Diseño Base de Datos Diseñador



<b>Caso de uso:</b> Iniciar Sesión	
<b>Actor:</b> Sistemas/Usuario	
<b>Descripción:</b> El responsable del Departamento ingresa el nombre de usuario y contraseña para acceder al sistema. Se comprueba si el usuario y contraseña son válidos.	
<b>Activación:</b> Si el usuario ingresa a los menús y utiliza la opción de <b>nuevo</b> le permitirá realizar la creación de un nuevo registro de información en la base de datos general.	
<b>Curso Normal</b>	<b>Alternativas</b>
1. El administrador hace clic en el menú buscando la pestaña respectiva del registro que desea ingresar.	<b>1.1 Sistema:</b> Pide escribir de nuevo el nombre y la contraseña. <b>1.2 Usuario:</b> vuelve a escribir el nombre y la contraseña
2. Se le muestra en pantalla el formulario correspondiente a la selección hecha.	<b>2.1 Sistema:</b> Pide escribir de nuevo el nombre y la contraseña con un máximo de 3 errores. <b>2.2</b> Vuelve escribir el nombre y contraseña
3. Una vez guardada la información, se retorna al menú principal y el registro puede ser utilizado en otro formulario.	
<b>Precondiciones:</b> que el sistema no este iniciado.	
<b>Pos condiciones:</b> Formulario de modificación.	
<b>Puntos de extensión:</b> En el caso de modificación, se requiere previamente haber creado los registros.	
<b>Observaciones y datos:</b> Tomar muy en cuenta el ingreso de datos correctos.	

### CASO DE USO CREACIÓN DE NUEVOS REGISTROS PARA USUARIO ADMINISTRADOR.

<b>Caso de uso:</b> Creación de nuevos registros
<b>Actor:</b> Recursos Humanos



<b>Descripción:</b> Permite la creación de nuevos registros de información en la base de datos para los empleados.	
<b>Activación:</b> Si el usuario ingresa a los menús y utiliza la opción de <b>nuevo</b> le permitirá realizar la creación de un nuevo registro de información en la base de datos general.	
<b>Curso Normal</b>	<b>Alternativas</b>
1. El administrador hace clic en el menú buscando la pestaña respectiva del registro que desea modificar.	1.1 Puede escoger entre nuevo, eliminar, refrescar y cerrar.
2. Se le muestra en pantalla el formulario correspondiente a la selección hecha.	2.1 Se pueden generar mensajes de error por datos incorrectos que no se rigen a las reglas preestablecidas al momento de guardar la información.
3. Una vez guardada la información, se retorna al menú principal y el registro puede ser utilizado en otro formulario.	
<b>Precondiciones:</b> Haber ingresado al menú principal de datos del empleado.	
<b>Pos condiciones:</b> Formulario de modificación.	
<b>Puntos de extensión:</b> En el caso de modificación, se requiere previamente haber creado los registros.	
<b>Observaciones y datos:</b> Tomar muy en cuenta el ingreso de datos correctos.	

#### CASO DE USO MODIFICAR REGISTROS PARA USUARIO ADMINISTRADOR.

<b>Caso de uso:</b> Modificar registros
<b>Actor:</b> Recursos Humanos
<b>Descripción:</b> Permite la creación de nuevos registros de información en la base de datos para los empleados.
<b>Activación:</b> Permite la modificación registros de información en la base de datos para los empleados.
<b>Activación:</b> Si el usuario ingresa a los menús y utiliza la opción <b>modificar</b> le permitirá



realizar la corrección de registros de información en la base de datos general.	
<b>Curso Normal</b>	<b>Alternativas</b>
1. El administrador hace clic en el menú buscando la pestaña respectiva del registro que desea modificar.	1.1 Puede escoger entre nuevo, eliminar, refrescar y cerrar.
2. Se le muestra en pantalla el formulario correspondiente a la selección hecha.	2.1 Se pueden generar mensajes de error por datos incorrectos que no se rigen a las reglas preestablecidas al momento de guardar la información.
3. Una vez guardada la información, se retorna al menú principal y el registro puede ser utilizado en otro formulario.	
<b>Precondiciones:</b> Haber ingresado al menú principal de datos del empleado.	
<b>Postcondiciones:</b> Formulario de modificación.	
<b>Puntos de extensión:</b> En el caso de modificación, se requiere previamente haber creado los registros.	
<b>Observaciones y datos:</b> Tomar muy en cuenta el ingreso de datos correctos.	

**CASO DE USO ELIMINAR REGISTROS PARA USUARIO ADMINISTRADOR.**

<b>Caso de uso:</b> Eliminar registros	
<b>Actor:</b> Recursos Humanos	
<b>Descripción:</b> Permite eliminar registros de información en la base de datos para los empleados.	
<b>Activación:</b> Si el usuario ingresa a los menús y utiliza la opción de <i>eliminar</i> le permitirá realizar la acción se ñ ala p ara la información en la base de datos general.	
<b>Curso Normal</b>	<b>Alternativas</b>
1. El administrador hace clic en el menú buscando la pestaña respectiva del registro que desea eliminar.	1.1 Puede escoger entre nuevo, modificar, refrescar y cerrar.



2. Se le muestra en pantalla el formulario correspondiente a la selección hecha.	2.1 Se pueden generar mensajes de error por datos incorrectos que no se rigen a las reglas preestablecidas al momento de guardar la información.
3. Una vez guardada la información, se retorna al menú principal y el registro puede ser utilizado en otro formulario.	
<b>Precondiciones:</b> Haber ingresado al menú principal de Datos del empleado	
<b>Postcondiciones:</b> Formulario de eliminar.	
<b>Puntos de extensión:</b> En el caso de eliminar, se requiere previamente haber creado los registros.	
<b>Observaciones y datos:</b> Tomar muy en cuenta la eliminación de datos.	

**CASO DE USO VERIFICAR REGISTROS PARA USUARIO ADMINISTRADOR.**

<b>Caso de uso:</b> Verificar registros	
<b>Actor:</b> Recursos Humanos	
<b>Descripción:</b> Permite la consulta o verificación de registros de información en la base de datos para los empleados, mediante la huella digital.	
<b>Activación:</b> Si el usuario ingresa a los menús y utiliza la opción <b>verificar</b> le permitirá realizar la consulta de registros de información en la base de datos general.	
<b>Curso Normal</b>	<b>Alternativas</b>
1. El administrador hace clic en el menú buscando la pestaña respectiva del registro que desea verificar.	1.1 Puede escoger entre ingreso y control.
2. Se le muestra en pantalla el formulario correspondiente a la selección hecha.	2.1 Se pueden generar mensajes de error por datos incorrectos que no se rigen a las reglas preestablecidas al momento de guardar la información.
3. Una vez consultada la información, se	



retorna al menú principal y el registro puede ser utilizado en otro formulario.	
<b>Precondiciones:</b> Haber ingresado al menú principal de Datos del empleado	
<b>Postcondiciones:</b> Formulario de verificación.	
<b>Puntos de extensión:</b> En el caso de verificar o consultar se requiere previamente haber ingresado los registros.	
<b>Observaciones y datos:</b> Tomar muy en cuenta la consulta de datos.	

**CASO DE USO CONTROL DIARIO PARA USUARIO ADMINISTRADOR.**

<b>Caso de uso:</b> Control Diario	
<b>Actor:</b> Recursos Humanos y/o Empleado	
<b>Descripción:</b> Permite el ingreso de la hora y salida del empleado de la información en la base de datos para los empleados, mediante la huella digital.	
<b>Activación:</b> Si el usuario ingresa a los menús y utiliza la opción <b>control diario</b> le permitirá ingresar a la base de datos la hora de entrada y salida.	
<b>Curso Normal</b>	<b>Alternativas</b>
1. El administrador hace clic en el menú buscando la pestaña respectiva para la entrada y salida del empleado.	1.1 Puede escoger entre ingreso y verificar.
2. Se le muestra en pantalla el formulario correspondiente a la selección hecha	2.1 Se pueden generar mensajes de error por datos incorrectos que no se rigen a las reglas preestablecidas al momento de guardar la información.
3. Una vez hecho el ingreso de la información, se retorna al menú principal y el registro puede ser utilizado en otro formulario.	
<b>Precondiciones:</b> Haber ingresado al menú principal de Datos del empleado.	



<b>Postcondiciones:</b> Formulario de control diario.
<b>Puntos de extensión:</b> En el caso del ingreso y salida del control diario se requiere previamente haber ingresado los registros.
<b>Observaciones y datos:</b> Tomar muy en cuenta el control diario.

**CASO DE USO IMPRIMIR REPORTES PARA USUARIO ADMINISTRADOR.**

<b>Caso de uso:</b> Imprimir Reportes	
<b>Actor:</b> Recursos Humanos	
<b>Descripción:</b> Permite el reporte de la asistencia de los empleados.	
<b>Activación:</b> Si el administrador ingresa a los menús y utiliza la opción <i>imprimir</i> le permitirá ingresar a la base de datos e imprimir el reporte.	
<b>Curso Normal</b>	<b>Alternativas</b>
1. El administrador hace clic en el menú buscando la pestaña respectiva para el reporte.	1.1 Puede escoger entre ingreso, Control y verificar.
2. Se le muestra en pantalla el formulario correspondiente a la selección hecha.	2.1 Se pueden generar mensajes de error por datos incorrectos que no se rigen a las reglas preestablecidas al momento de guardar la información.
3. Una vez hecho el ingreso de la información, se retorna al menú principal y el registro puede ser utilizado en otro formulario.	
<b>Precondiciones:</b> Haber ingresado al menú principal de Datos del empleado	
<b>Postcondiciones:</b> Formulario de impresión o guardar html, para abrir en Word, Excel, etc.	
<b>Puntos de extensión:</b> En el caso de impresión se requiere previamente haber ingresado los registros.	
<b>Observaciones y datos:</b> Tomar muy en cuenta el control diario.	

DIAGRAMAS DE SECUENCIAS

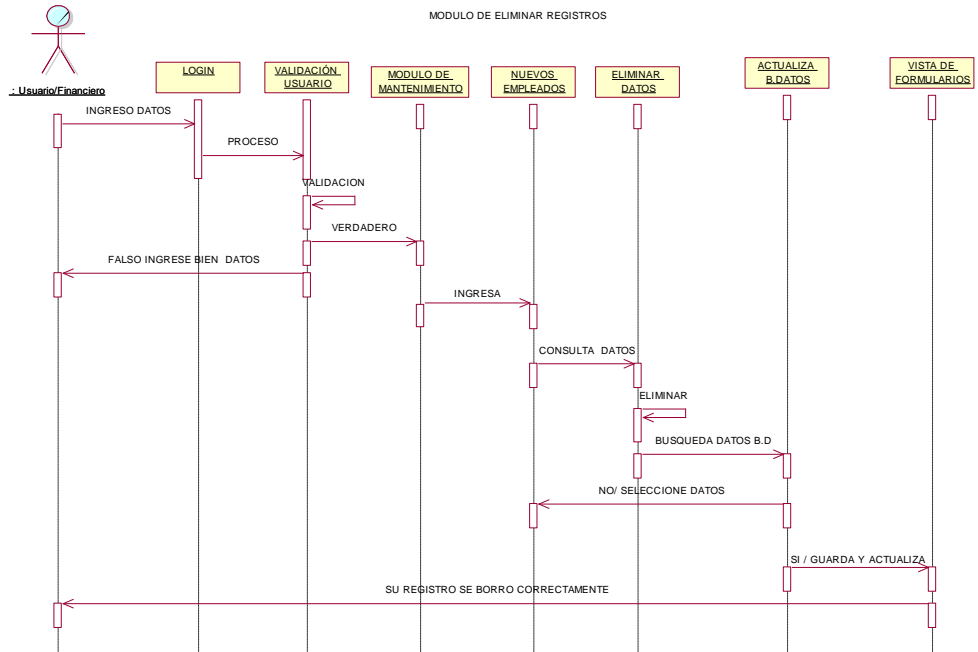


Gráfico 17: Diagrama de secuencia para Eliminar Registros

Autor: Jhonny Suqui

Fuente: Diseño de Base de Datos Diseñador

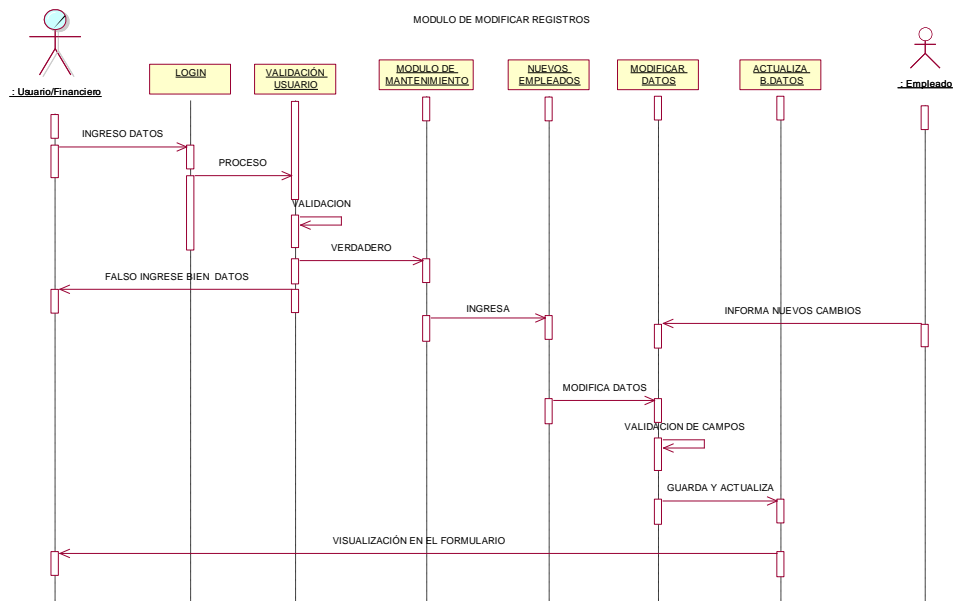


Gráfico 18: Diagrama de Secuencia para Modificar Registros

Autor: Jhonny Suqui

Fuente: Diseño de Base de Datos Diseñador

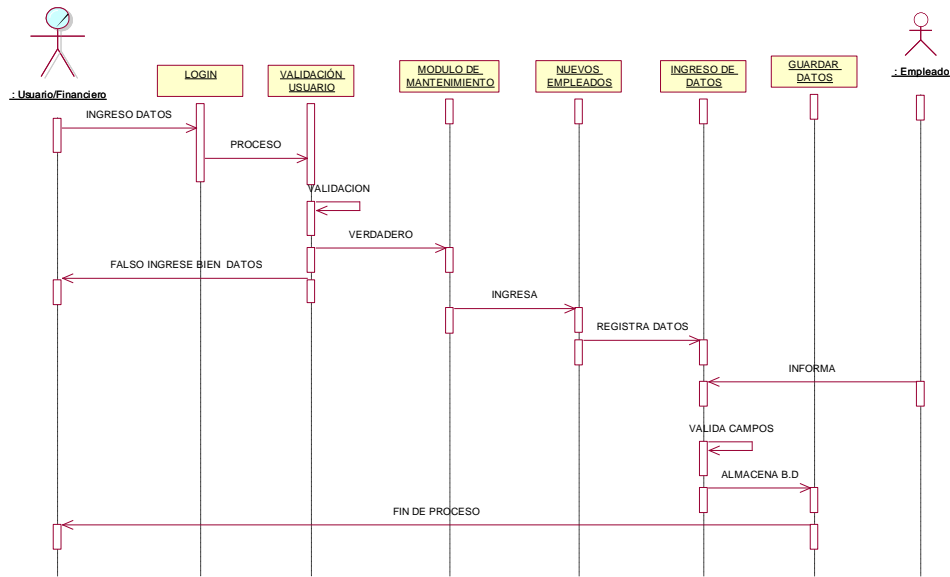


Gráfico 19: Diagrama de Secuencia para Nuevos Registros

Autor: Jhonny Suqui

Fuente: Diseño de Base de Datos Diseñador

### DIAGRAMAS DE SECUENCIA PARA USUARIO ADMINISTRADOR

#### Guión #1 "Validar Usuario":

- 1.- La persona designada como administrador del sistema, es el encargado del control de los registros (administrador), entra al sistema.
- 2.- El ingresa su perfil, especificando el tipo de usuario que será (administrador, usuario, invitado).
- 3.- Después ingresa la CLAVE.
- 4.- Con esta información, si es la correcta logra ingresar a la interfaz gráfica.



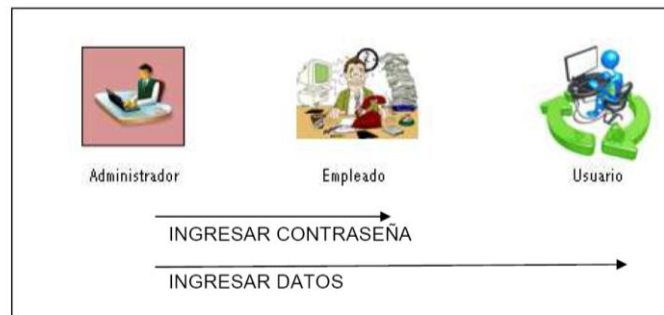
Gráfico 20: Proceso Ingresar contraseña

Autor: Jhonny Suqui

Fuente: Diseño de Base de Datos Diseñador

**DIAGRAMA DE SECUENCIA VALIDAR USUARIO PARA USUARIO ADMINISTRADOR****Guión #2 “Ingresar Empleado”:**

1. Este tipo de usuarios son los encargados del ingreso de nuevos empleados, para lo cual ingresan los datos de cada uno, en el formulario correspondiente a su interfaz.
2. Una vez lleno el formulario, elige la función guardar en el sistema, mediante su botón correspondiente.

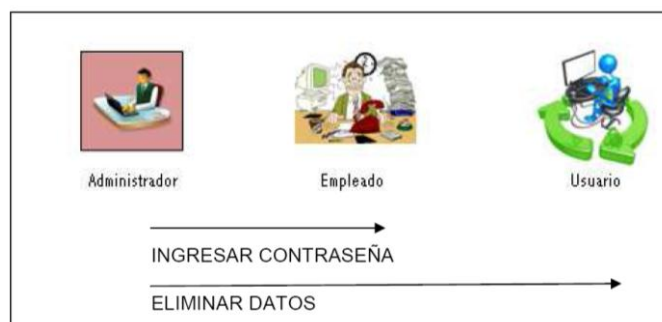
**Gráfico 21:** Proceso Ingresar Empleado**Autor:** Jhonny Suqui**Fuente:** Diseño de Base de Datos Diseñador**DIAGRAMA DE SECUENCIA INGRESAR USUARIO PARA USUARIO ADMINISTRADOR.****Guión #3 “Actualizar Empleado”:**

1. Son los encargados de la modificación de la información a lo que actualizará cada uno de los datos, en el formulario correspondiente a su interfaz.
2. Una vez lleno el formulario, elige la función guardar en el sistema, mediante su botón correspondiente.

**Gráfico 22:** Actualizar Usuario.**Autor:** Jhonny Suqui.**Fuente:** Diseño de Base de Datos Diseñador.

**DIAGRAMA DE SECUENCIA ACTUALIZAR EMPLEADO PARA USUARIO ADMINISTRADOR.****Guión #4 “Eliminar Empleado”:**

1. Se encarga de borrar la información de la base de datos, en el formulario correspondiente a su interfaz.

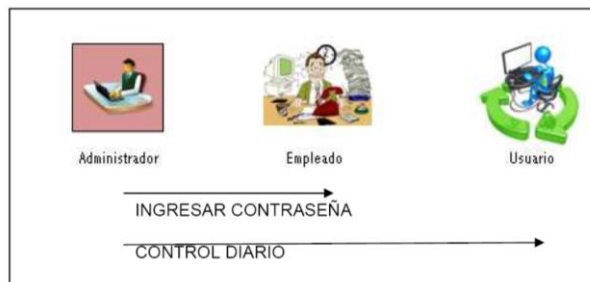
**Gráfico 23: Eliminar Datos.****Autor:** Jhonny Suqui.**Fuente:** Diseño de Base de Datos Diseñador**DIAGRAMA DE SECUENCIA ELIMINAR EMPLEADO PARA USUARIO ADMINISTRADOR.****Guión #5 “Consulta Empleado”**

1. Se encarga de consultar la información de la base de datos, utilizando el lector de huellas dactilares, en el formulario correspondiente a su interfaz.

**Gráfico 24: Consulta empleado****Autor:** Jhonny Suqui**Fuente:** Diseño de Base de Datos Diseñador

**DIAGRAMA DE SECUENCIA CONSULTA EMPLEADO PARA USUARIO ADMINISTRADOR.****Guión #6 "Control Diario Empleado":**

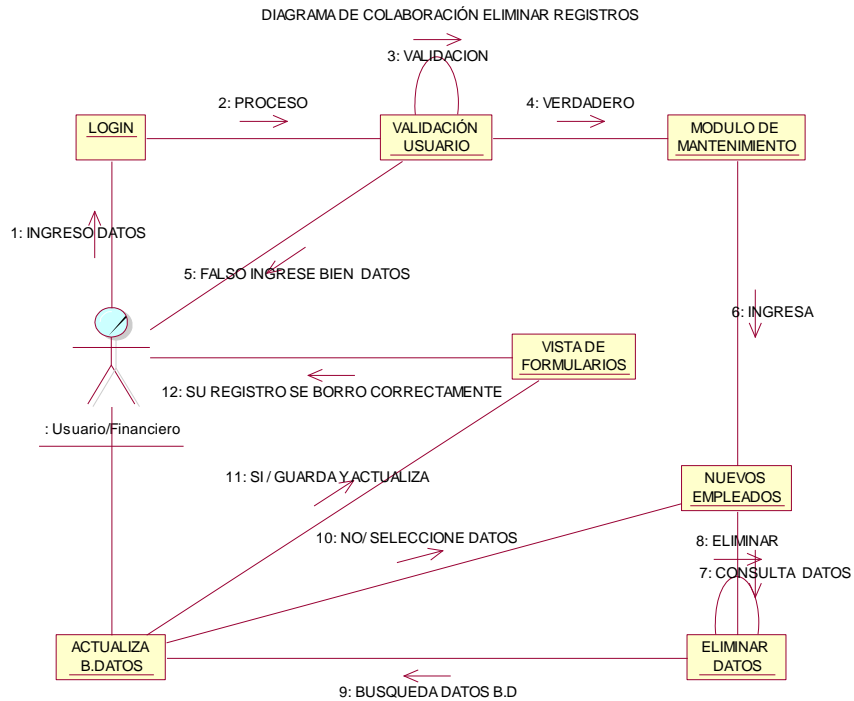
1. Se encarga de llevar el control del empleado mediante la huella dactilar y el computador le asignará la hora de entrada o salida, según corresponda.

**Gráfico 25:** Control Diario**Autor:** Jhonny Suqui**Fuente:** Diseño de Base de Datos Diseñador**DIAGRAMA DE SECUENCIA CONTROL DIARIO EMPLEADO PARA USUARIO ADMINISTRADOR.****Guión #7 "Impresión":**

1. Permite sacar el reporte de la asistencia de los empleados tomando en cuenta su departamento y la hora de entrada y salida.

**Gráfico 26:** Impresión de Datos**Autor:** Jhonny Suqui**Fuente:** Diseño de Base de Datos Diseñador

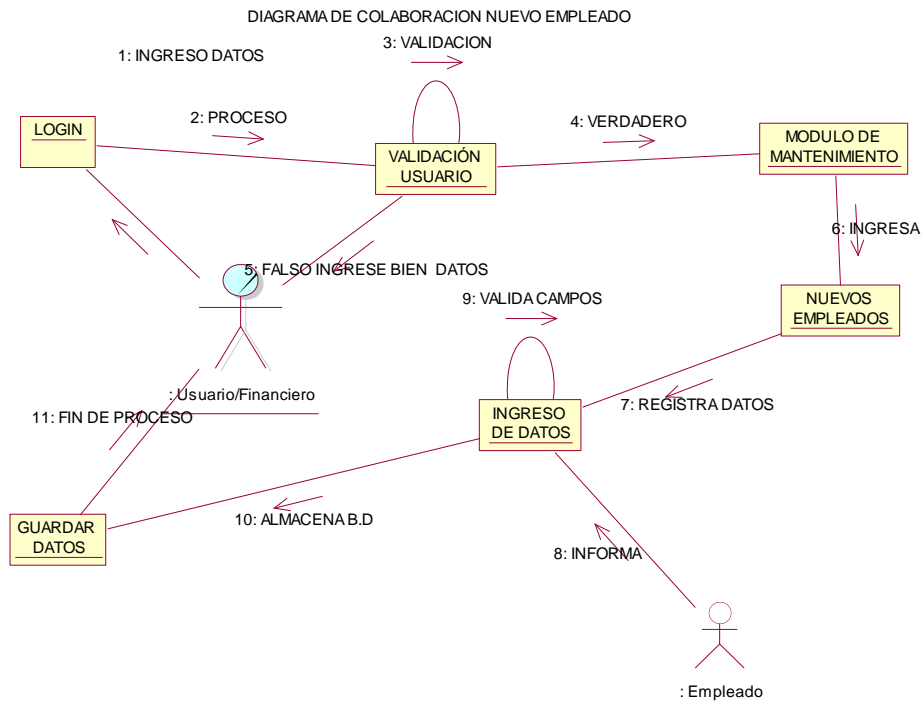
**DIAGRAMA DE COLABORACIÓN**



**Gráfico 27:** Diagrama de Colaboración Eliminar Registros

**Autor:** Jhonny Suqui

**Fuente:** Diseño de Base de Datos Diseñador



**Gráfico 28:** Diagrama de Colaboración Nuevo empleado

**Autor:** Jhonny Suqui

**Fuente:** Diseño de Base de Datos Diseñador

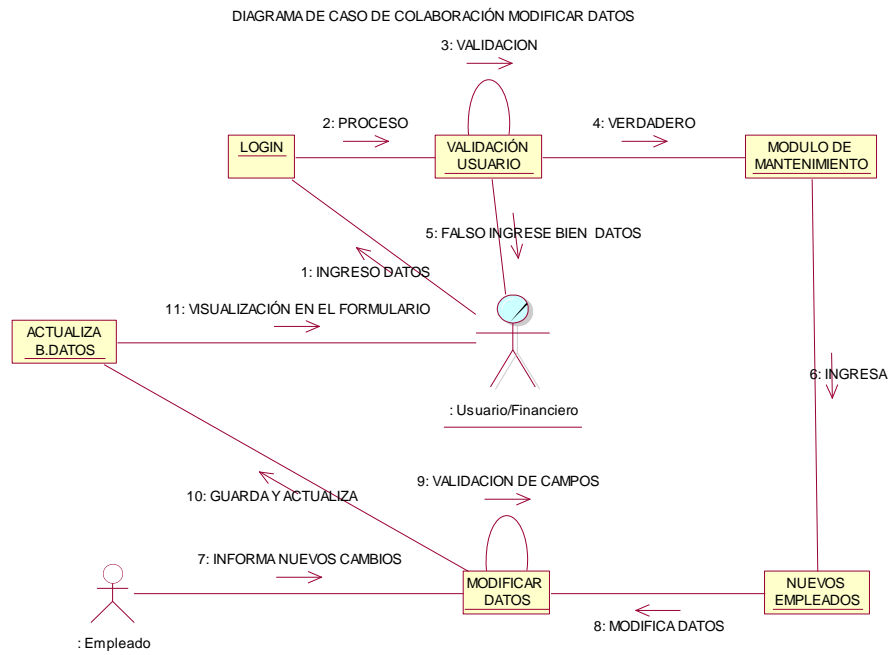


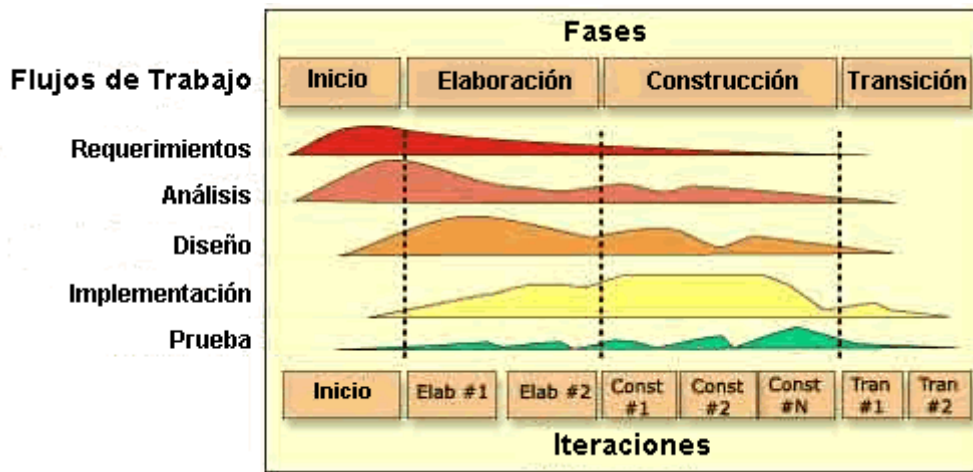
Gráfico 29: Diagrama de Colaboración Modificar Datos

Autor: Jhonny Suqui

Fuente: Diseño de Base de Datos Diseñador

#### 4.8. DESCRIPCIÓN DE METODOLOGÍA DE DESARROLLO.

La arquitectura en donde se desarrollara son la denominada de **tres capas** se basa en la división en el nivel de **acceso a datos**, nivel de **lógica de negocio** y nivel de **presentación o aplicación**. Se pretende desarrollar en esta arquitectura con el lenguaje de programación .NET, en donde podemos decir que en el primer nivel creamos la clase de acceso a datos con la conexión a la base de datos y los métodos de selección, modificación y eliminación de datos. En el nivel medio, implementaremos los métodos que gestionará la comunicación de datos entre las capa de presentación y de acceso a datos. Por último, en la capa de presentación, nos preocuparemos únicamente de hacer la petición de datos a negocio y de mostrarlos en los controles que queramos.



**Gráfico 30:** Fases del Desarrollo del sistema

**Autor:** Jhonny Suqui

**Fuente:** Logica de Negocios

Todas las etapas mostradas en la Fig 22, son un reflejo del proceso que se sigue a la hora de resolver cualquier tipo de problema. Es por ello que para poder resolver el problema, se deberá ir desarrollando parte por parte para no cometer los errores típicos, así:

- Utilizar diferentes técnicas de recolección de información para tratar de comprender a cabalidad el problema (análisis).
- Luego plantear una posible solución, la cual se irá mejorando conforme se vaya teniendo más conocimiento (diseño).
- Una vez claro, empezar a desarrollar el sistema de control de entrada y salida del personal utilizando por supuesto el lector de huellas digital (implementación – construcción).
- Al finalizar la programación del sistema constatar que todo este correcto, y si hay errores corregirlos y así mejorar dicho sistema (pruebas).
- Después de haber corregido proceder a la instalación y a la migración de los datos de la empresa hacia el nuevo sistema (implantación).

A continuación se describe brevemente lo que se deberá realizar en cada una de las etapas de desarrollo del sistema de Control de Entrada y Salida del Personal mediante la administración de un lector de huellas digital:

✓ **Estudio Preliminar.**

Es la realización de tareas previas que influirán decisivamente en la finalización del proyecto, estas tareas se conocen explícitamente como el fuzzy front-end del proyecto.



Aquí se determinará si el proyecto planteado es viable y factible, y si es así se realizará el levantamiento de la información necesaria para poder conocer los procesos de registro de entrada y salida del personal que llevan a cabo dentro de la empresa y desarrollar el sistema de control de entrada y salida del personal utilizando un lector de huellas digital.

✓ **Análisis.**

Aquí se averiguará exactamente la situación actual de la empresa, cuáles son los procesos manuales que se llevan a cabo, qué es lo que registra, cuándo y cómo hacen ese control de la entrada y salida de su personal, y así descubrir qué es lo que realmente se necesita y llegar a una comprensión adecuada de los requerimientos del sistema (las características que el sistema debe poseer); así mismo aquí determinaremos qué herramientas de hardware y software se utilizarán para el desarrollo del mismo.

✓ **Diseño.**

Se representarán las características del sistema que nos permitirán implementarlo de forma efectiva (el cómo). Se tratará de que el diseño sea lo más adecuado posible para que cuando toque la construcción se faciliten las tareas.

✓ **Implementación.**

Luego de saber exactamente las funciones que debe desempeñar el sistema de información (análisis) y tener claro cómo se va a organizar sus distintos componentes (diseño), hay que pasar a la etapa de implementación, pero nunca antes. Antes de escribir una sola línea de código (o de crear una tabla en nuestra base de datos) tendremos que comprender bien el problema a resolver y haber aplicado principios básicos de diseño que nos permitan construir un sistema de información de calidad.

La implementación, comprende la programación de la aplicación y el diseño de la base de datos, una vez claras las herramientas a utilizar se iniciará manipulando el dispositivo biométrico mediante el SDK3 que viene con él y así tener claro cómo funciona tanto el lector como el SDK, lo cual llevará a conocer mejor cómo se podría administrar el lector e incorporarlo fácilmente al sistema de control de asistencia; luego, se creará la base de datos con los atributos y procesos encontrados en las etapas anteriores haciendo uso de una herramienta CASE; después se determinará y programará la interfaz gráfica del sistema para así cumplir uno de los objetivos planteados que es crear una interfaz gráfica



agradable para el usuario, se realizará la programación de los módulos, la conexión con la base de datos, el desarrollo de los reportes y con ello lograr el funcionamiento adecuado del sistema total.

✓ **Pruebas.**

La etapa de pruebas se realizará para detectar los errores que se hayan podido cometer en las etapas anteriores del proyecto, y eventualmente corregirlos, antes de que la Agencia los tenga que sufrir. De hecho, una prueba es un éxito cuando se detecta un error (y no al revés, como nos gustaría pensar).

Realizar un plan de pruebas de cada módulo y del sistema en total nos ayudará a ver su funcionamiento real.

✓ **Implementación.**

Terminada la primera versión del sistema, se procederá a implementarlo siguiendo el proceso de instalación y verificando los requerimientos de hardware y software necesarios para un correcto funcionamiento.

Se considerará que debería empezar con la creación de usuarios y la generación de períodos laborales; parametrizar el sistema: ingresando las profesiones de los empleados, los departamentos y los horarios de turno que serán dependientes para la ficha personal de cada empleado; llenar la ficha de cada empleado y realizar el registro de su huella digital, la cual servirá para el posterior registro de la hora de entrada y salida del trabajo.

#### **4.9. MODELO CONCEPTUAL.**

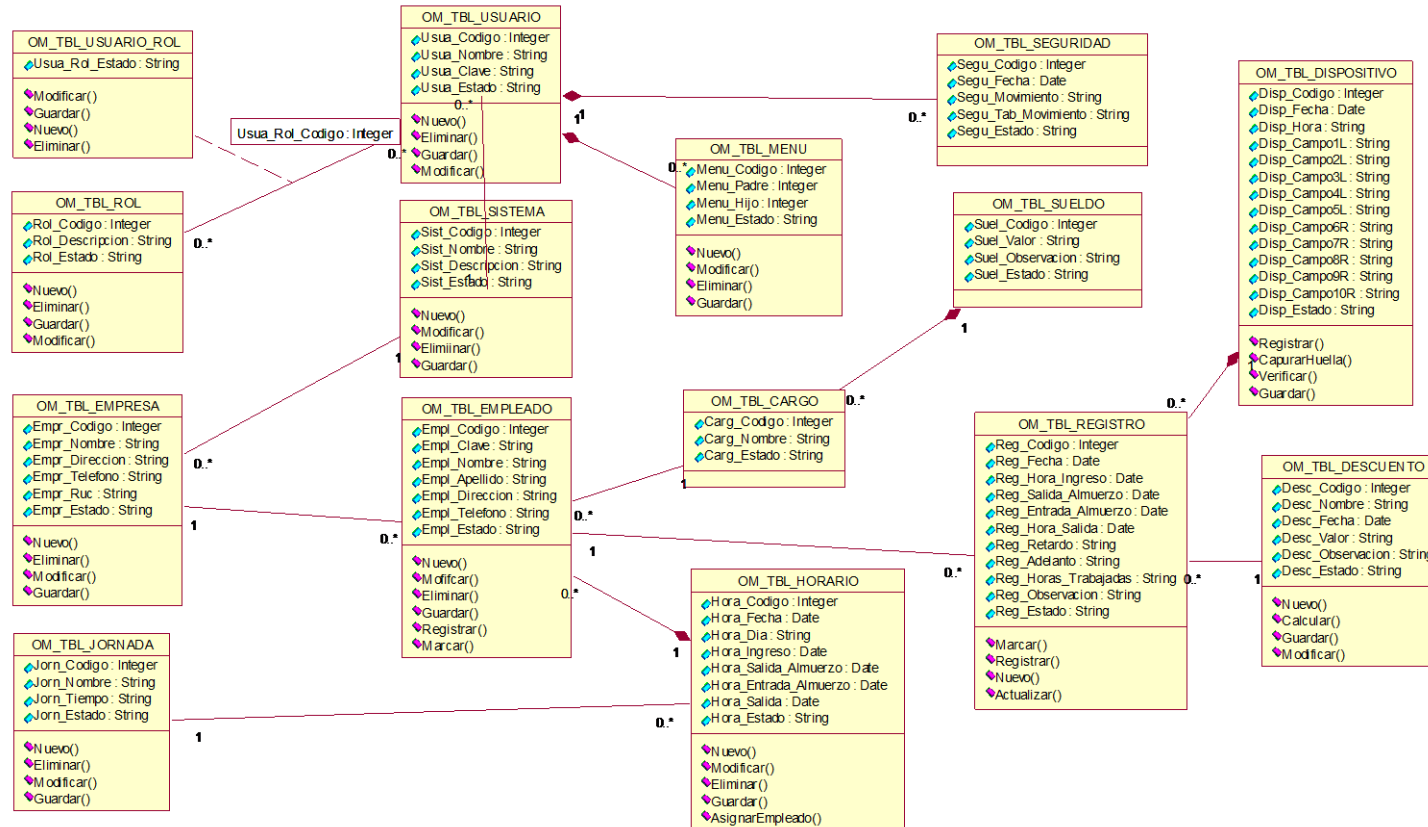
Se creó el modelo conceptual de datos referente a la información obtenida en la institución, y basado en estándares de programación (relacionado con la tecnología biométrica), esto a partir de un mapa conceptual (organigrama funcional) y de las actividades que se realiza manualmente se diseñó los nuevos procesos. El modelo propuesto permitirá un cambio en cada uno de las tareas, ya que al considerar de manera efectiva y convertir los procesos en objetos y los mismos caracteres la base de datos funcionara de una manera eficiente.



Es empezó realizando las entidades que forman parte de dicho modelo son: empleado, departamento, profesión, dirección, teléfono, sexo, horario, huella, registro, multa, y usuarios, cabe indicar que las dos últimas entidades no tienen una relación directa con la entidad empleado a la cual se relacionan las demás ya que alrededor de ella funcionará el sistema, porque a esas entidades (días festivos y usuarios) se las crea con la finalidad de que la una (días festivos) almacene y presente en alguna parte del sistema que tal o cual día es una fecha importante y la otra (usuarios) es necesaria para que se almacene la información del administrador de todo el sistema.



DIADRAMALÓGICO SISTEMA



### 4.10. MODELO FÍSICO Y LÓGICO (DIAGRAMAS UML EN CASO TRES CAPAS)

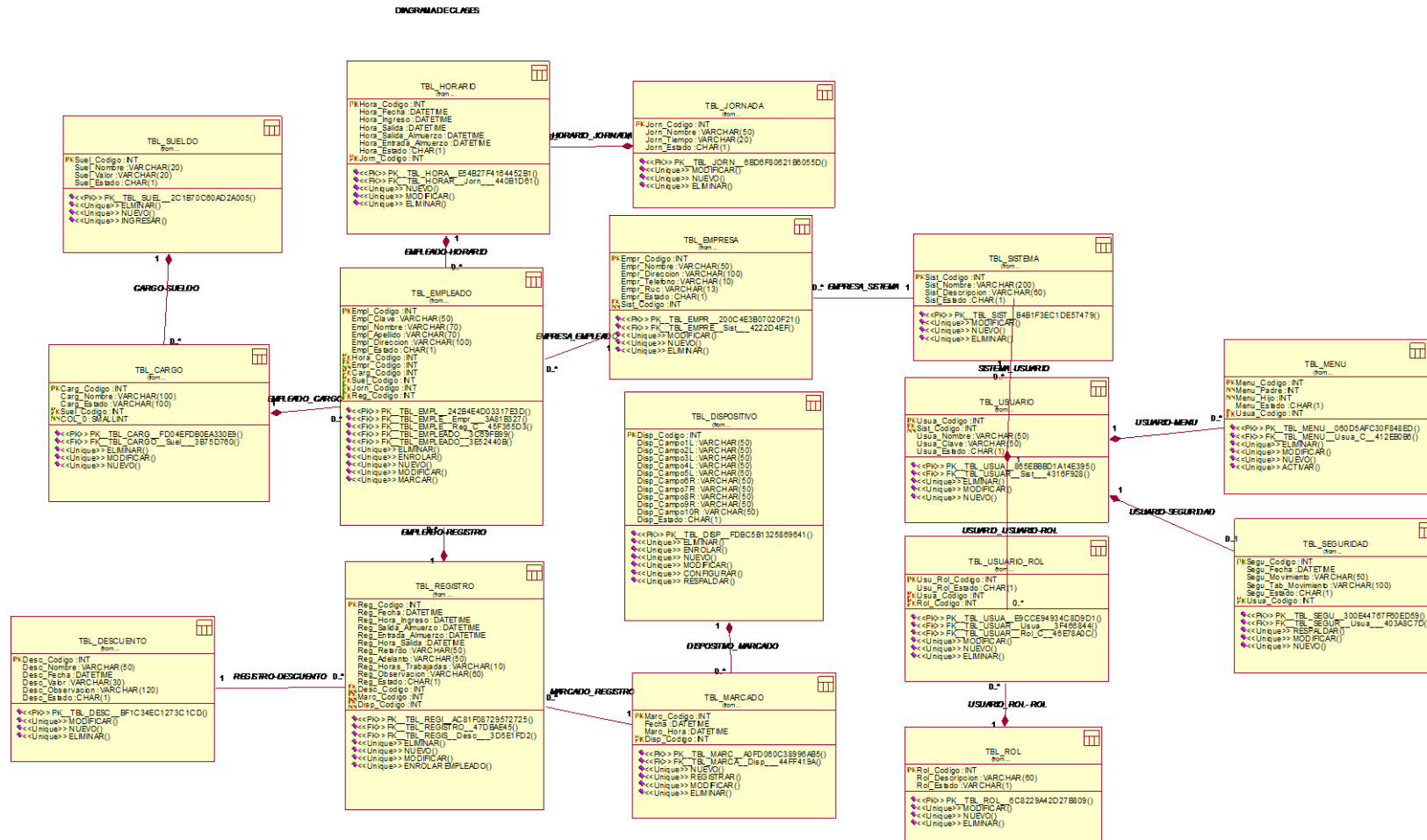


Gráfico 32: La Base de Datos Modelo Físico

Autor: Jhonny Suqui



#### 4.11. DICCIONARIO DE DATOS

Para la obtención del diccionario de datos se realizó la siguiente consulta en SQL 2008.

Generando de esta forma los siguientes campos del diccionario de datos.

```
select
    d.object_id,
    a.name [table],
    --schema_name(a.schema_id) schema_name,
    --a.create_date,
    b.name [column],
    c.name [type],
    CASE
        WHEN c.name = 'numeric' OR c.name = 'decimal' OR c.name =
'float' THEN b.precision
        ELSE null
    END [Precision],
    b.max_length,
    CASE
        WHEN b.is_nullable = 0 THEN 'NO'
        ELSE 'SI'
    END [Permite Nulls],
    CASE
        WHEN b.is_identity = 0 THEN 'NO'
        ELSE 'SI'
    END [Es Autonumerico],
    ep.value [Descripcion],
    f.ForeignKey,
    f.ReferenceTableName,
    f.ReferenceColumnName
from sys.tables a
    inner join sys.columns b on a.object_id= b.object_id
    inner join sys.systypes c on b.system_type_id= c.xtype
    inner join sys.objects d on a.object_id= d.object_id
    LEFT JOIN sys.extended_properties ep ON d.object_id = ep.major_id
AND b.column_id = ep.minor_id
    LEFT JOIN (SELECT
        f.name AS ForeignKey,
        OBJECT_NAME(f.parent_object_id) AS TableName,
```



```
                COL_NAME(fc.parent_object_id,fc.parent_column_id)
AS ColumnName,
                OBJECT_NAME (f.referenced_object_id) AS
ReferenceTableName,

                COL_NAME(fc.referenced_object_id,fc.referenced_column_id) AS
ReferenceColumnName
                FROM sys.foreign_keys AS f
                INNER JOIN sys.foreign_key_columns AS fc ON
f.OBJECT_ID = fc.constraint_object_id)    f ON f.TableName =a.name AND
f.ColumnName =b.name
WHERE a.name <> 'sysdiagrams'
ORDER BY a.name,b.column_Id
```



ID	TABLA	COLUMNA	TIPO	PRECISIÓN	MAX-LENG	PERMITR NUL	ES AUTONUM ERICO	DESCRIPCIÓN	FORENKEY	REFERENCE TABLE NAME	REFERENCE TABLE COLUMNA
213575799	TBL_CARGO	Carg_Codigo	int	NULL	4	NO	NO	NULL	NULL	NULL	NULL
213575799	TBL_CARGO	Carg_Nombre	varchar	NULL	100	SI	NO	NULL	NULL	NULL	NULL
213575799	TBL_CARGO	Carg_Estado	varchar	NULL	100	SI	NO	NULL	NULL	NULL	NULL
213575799	TBL_CARGO	Suel_Codigo	int	NULL	4	NO	NO	NULL	FK_TBL_CARGO__Suel__3B75D760	TBL_SUELDO	Suel_Codigo
277576027	TBL_DESCUENTO	Desc_Codigo	int	NULL	4	NO	NO	NULL	NULL	NULL	NULL
277576027	TBL_DESCUENTO	Desc_Nombre	varchar	NULL	50	SI	NO	NULL	NULL	NULL	NULL
277576027	TBL_DESCUENTO	Desc_Fecha	datetime	NULL	8	SI	NO	NULL	NULL	NULL	NULL
277576027	TBL_DESCUENTO	Desc_Valor	varchar	NULL	30	SI	NO	NULL	NULL	NULL	NULL
277576027	TBL_DESCUENTO	Desc_Observacion	varchar	NULL	120	SI	NO	NULL	NULL	NULL	NULL
277576027	TBL_DESCUENTO	Desc_Estado	char	NULL	1	SI	NO	NULL	NULL	NULL	NULL
597577167	TBL_DISPOSITIVO	Disp_Codigo	int	NULL	4	NO	NO	NULL	NULL	NULL	NULL
597577167	TBL_DISPOSITIVO	Disp_Campo 1L	varchar	NULL	50	SI	NO	NULL	NULL	NULL	NULL
597577167	TBL_DISPOSITIVO	Disp_Campo 2L	varchar	NULL	50	SI	NO	NULL	NULL	NULL	NULL
597577167	TBL_DISPOSITIVO	Disp_Campo 3L	varchar	NULL	50	SI	NO	NULL	NULL	NULL	NULL
597577167	TBL_DISPOSITIVO	Disp_Campo 4L	varchar	NULL	50	SI	NO	NULL	NULL	NULL	NULL



597577167	TBL_DISPOSITIVO	Disp_Campo	5L	varchar	NULL	50	SI	NO	NULL	NULL	NULL	NULL
597577167	TBL_DISPOSITIVO	Disp_Campo	6R	varchar	NULL	50	SI	NO	NULL	NULL	NULL	NULL
597577167	TBL_DISPOSITIVO	Disp_Campo	7R	varchar	NULL	50	SI	NO	NULL	NULL	NULL	NULL
597577167	TBL_DISPOSITIVO	Disp_Campo	8R	varchar	NULL	50	SI	NO	NULL	NULL	NULL	NULL
597577167	TBL_DISPOSITIVO	Disp_Campo	9R	varchar	NULL	50	SI	NO	NULL	NULL	NULL	NULL
597577167	TBL_DISPOSITIVO	Disp_Campo	10R	varchar	NULL	50	SI	NO	NULL	NULL	NULL	NULL
597577167	TBL_DISPOSITIVO	Disp_Estado		char	NULL	1	SI	NO	NULL	NULL	NULL	NULL
21575115	TBL_EMPLEADO	Empl_Codigo		int	NULL	4	NO	NO	NULL	NULL	NULL	NULL
21575115	TBL_EMPLEADO	Empl_Clave		varchar	NULL	50	SI	NO	NULL	NULL	NULL	NULL
21575115	TBL_EMPLEADO	Empl_Nombre		varchar	NULL	70	SI	NO	NULL	NULL	NULL	NULL
21575115	TBL_EMPLEADO	Empl_Apellido		varchar	NULL	70	SI	NO	NULL	NULL	NULL	NULL
21575115	TBL_EMPLEADO	Empl_Direccion		varchar	NULL	100	SI	NO	NULL	NULL	NULL	NULL
21575115	TBL_EMPLEADO	Empl_Estado		char	NULL	1	SI	NO	NULL	NULL	NULL	NULL
21575115	TBL_EMPLEADO	Hora_Codigo		int	NULL	4	NO	NO	NULL	FK_TBL_EMPLEADO_3E52440B	TBL_HORARIO	Hora_Codigo
21575115	TBL_EMPLEADO	Empr_Codigo		int	NULL	4	NO	NO	NULL	FK_TBL_EMPLE_Empr_3A81B327	TBL_EMPRESA	Empr_Codigo



21575115	TBL_EMPLEADO	Carg_Codigo	int	NULL	4	NO	NO	NULL	FK_TBL_EMPLEADO_ _3C69FB99	TBL_CARGO	Carg_Codigo
21575115	TBL_EMPLEADO	Suel_Codigo	int	NULL	4	NO	NO	NULL	FK_TBL_EMPLEADO_ _3C69FB99	TBL_CARGO	Suel_Codigo
21575115	TBL_EMPLEADO	Jorn_Codigo	int	NULL	4	NO	NO	NULL	FK_TBL_EMPLEADO_ _3E52440B	TBL_HORARIO	Jorn_Codigo
21575115	TBL_EMPLEADO	Reg_Codigo	int	NULL	4	NO	NO	NULL	FK_TBL_EMPLE_Reg _C__45F365D3	TBL_REGISTRO	Reg_Codigo
85575343	TBL_EMPRESA	Empr_Codigo	int	NULL	4	NO	NO	NULL	NULL	NULL	NULL
85575343	TBL_EMPRESA	Empr_Nombr e	varchar	NULL	50	SI	NO	NULL	NULL	NULL	NULL
85575343	TBL_EMPRESA	Empr_Direcci on	varchar	NULL	100	SI	NO	NULL	NULL	NULL	NULL
85575343	TBL_EMPRESA	Empr_Telefo no	varchar	NULL	10	SI	NO	NULL	NULL	NULL	NULL
85575343	TBL_EMPRESA	Empr_Ruc	varchar	NULL	13	SI	NO	NULL	NULL	NULL	NULL
85575343	TBL_EMPRESA	Empr_Estado	char	NULL	1	SI	NO	NULL	NULL	NULL	NULL
85575343	TBL_EMPRESA	Sist_Codigo	int	NULL	4	NO	NO	NULL	FK_TBL_EMPRE__Sist __4222D4EF	TBL_SISTEMA	Sist_Codigo
341576255	TBL_HORARIO	Hora_Codigo	int	NULL	4	NO	NO	NULL	NULL	NULL	NULL
341576255	TBL_HORARIO	Hora_Fecha	datetime	NULL	8	SI	NO	NULL	NULL	NULL	NULL
341576255	TBL_HORARIO	Hora_Ingreso	datetime	NULL	8	SI	NO	NULL	NULL	NULL	NULL
341576255	TBL_HORARIO	Hora_Salida	datetime	NULL	8	SI	NO	NULL	NULL	NULL	NULL
341576255	TBL_HORARIO	Hora_Salida_ Almuerzo	datetime	NULL	8	SI	NO	NULL	NULL	NULL	NULL
341576255	TBL_HORARIO	Hora_Entrad	datetime	NULL	8	SI	NO	NULL	NULL	NULL	NULL



		a_Almuerso									
341576255	TBL_HORARIO	Hora_Estado	char	NULL	1	SI	NO	NULL	NULL	NULL	NULL
341576255	TBL_HORARIO	Jorn_Codigo	int	NULL	4	NO	NO	NULL	FK_TBL_HORAR_Jor n__440B1D61	TBL_JORNADA	Jorn_Codigo
533576939	TBL_JORNADA	Jorn_Codigo	int	NULL	4	NO	NO	NULL	NULL	NULL	NULL
533576939	TBL_JORNADA	Jorn_Nombr e	varchar	NULL	50	SI	NO	NULL	NULL	NULL	NULL
533576939	TBL_JORNADA	Jorn_Tiempo	varchar	NULL	20	SI	NO	NULL	NULL	NULL	NULL
533576939	TBL_JORNADA	Jorn_Estado	char	NULL	1	SI	NO	NULL	NULL	NULL	NULL
917578307	TBL_MARCADO	Marc_Codigo	int	NULL	4	NO	NO	NULL	NULL	NULL	NULL
917578307	TBL_MARCADO	Fecha	datetime	NULL	8	SI	NO	NULL	NULL	NULL	NULL
917578307	TBL_MARCADO	Marc_Hora	datetime	NULL	8	SI	NO	NULL	NULL	NULL	NULL
917578307	TBL_MARCADO	Disp_Codigo	int	NULL	4	NO	NO	NULL	FK_TBL_MARCA_Dis p__44FF419A	TBL_DISPOSITIVO	Disp_Codigo
789577851	TBL_MENU	Menu_Codig o	int	NULL	4	NO	NO	NULL	NULL	NULL	NULL
789577851	TBL_MENU	Menu_Padre	int	NULL	4	NO	NO	NULL	NULL	NULL	NULL
789577851	TBL_MENU	Menu_Hijo	int	NULL	4	NO	NO	NULL	NULL	NULL	NULL
789577851	TBL_MENU	Menu_Estad o	char	NULL	1	SI	NO	NULL	NULL	NULL	NULL
789577851	TBL_MENU	Usua_Codigo	int	NULL	4	NO	NO	NULL	FK_TBL_MENU_Usu a_C__412EB0B6	TBL_USUARIO	Usua_Codig o
661577395	TBL_REGISTRO	Reg_Codigo	int	NULL	4	NO	NO	NULL	NULL	NULL	NULL
661577395	TBL_REGISTRO	Reg_Fecha	datetime	NULL	8	SI	NO	NULL	NULL	NULL	NULL
661577395	TBL_REGISTRO	Reg_Hora_In	datetime	NULL	8	SI	NO	NULL	NULL	NULL	NULL



		greso									
661577395	TBL_REGISTRO	Reg_Salida_A Imuerzo	datetime	NULL	8	SI	NO	NULL	NULL	NULL	NULL
661577395	TBL_REGISTRO	Reg_Entrada _Almuerzo	datetime	NULL	8	SI	NO	NULL	NULL	NULL	NULL
661577395	TBL_REGISTRO	Reg_Hora_Sa lida	datetime	NULL	8	SI	NO	NULL	NULL	NULL	NULL
661577395	TBL_REGISTRO	Reg_Retardo	varchar	NULL	50	SI	NO	NULL	NULL	NULL	NULL
661577395	TBL_REGISTRO	Reg_Adelant o	varchar	NULL	50	SI	NO	NULL	NULL	NULL	NULL
661577395	TBL_REGISTRO	Reg_Horas_T rabajadas	varchar	NULL	10	SI	NO	NULL	NULL	NULL	NULL
661577395	TBL_REGISTRO	Reg_Observa cion	varchar	NULL	60	SI	NO	NULL	NULL	NULL	NULL
661577395	TBL_REGISTRO	Reg_Estado	char	NULL	1	SI	NO	NULL	NULL	NULL	NULL
661577395	TBL_REGISTRO	Desc_Codigo	int	NULL	4	NO	NO	NULL	FK__TBL_REGIS__Desc __3D5E1FD2	TBL_DESCUENTO	Desc_Codigo
661577395	TBL_REGISTRO	Marc_Codigo	int	NULL	4	NO	NO	NULL	FK__TBL_REGISTRO__ 47DBAE45	TBL_MARCADO	Marc_Codig o
661577395	TBL_REGISTRO	Disp_Codigo	int	NULL	4	NO	NO	NULL	FK__TBL_REGISTRO__ 47DBAE45	TBL_MARCADO	Disp_Codigo
725577623	TBL_ROL	Rol_Codigo	int	NULL	4	NO	NO	NULL	NULL	NULL	NULL
725577623	TBL_ROL	Rol_Descripci on	varchar	NULL	60	SI	NO	NULL	NULL	NULL	NULL
725577623	TBL_ROL	Rol_Estado	varchar	NULL	1	SI	NO	NULL	NULL	NULL	NULL
2105058535	TBL_SEGURIDAD	Segu_Codigo	int	NULL	4	NO	NO	NULL	NULL	NULL	NULL



2105058535	TBL_SEGURIDAD	Segu_Fecha	datetime	NULL	8	SI	NO	NULL	NULL	NULL	NULL
2105058535	TBL_SEGURIDAD	Segu_Movimiento	varchar	NULL	50	SI	NO	NULL	NULL	NULL	NULL
2105058535	TBL_SEGURIDAD	Segu_Tab_Movimiento	varchar	NULL	100	SI	NO	NULL	NULL	NULL	NULL
2105058535	TBL_SEGURIDAD	Segu_Estado	char	NULL	1	SI	NO	NULL	NULL	NULL	NULL
2105058535	TBL_SEGURIDAD	Usua_Codigo	int	NULL	4	NO	NO	NULL	FK_TBL_SEGUR_Usua a__403A8C7D	TBL_USUARIO	Usua_Codigo
469576711	TBL_SISTEMA	Sist_Codigo	int	NULL	4	NO	NO	NULL	NULL	NULL	NULL
469576711	TBL_SISTEMA	Sist_Nombre	varchar	NULL	200	SI	NO	NULL	NULL	NULL	NULL
469576711	TBL_SISTEMA	Sist_Descripcion	varchar	NULL	60	SI	NO	NULL	NULL	NULL	NULL
469576711	TBL_SISTEMA	Sist_Estado	char	NULL	1	SI	NO	NULL	NULL	NULL	NULL
149575571	TBL_SUELDO	Suel_Codigo	int	NULL	4	NO	NO	NULL	NULL	NULL	NULL
149575571	TBL_SUELDO	Suel_Nombre	varchar	NULL	20	SI	NO	NULL	NULL	NULL	NULL
149575571	TBL_SUELDO	Suel_Valor	varchar	NULL	20	SI	NO	NULL	NULL	NULL	NULL
149575571	TBL_SUELDO	Suel_Estado	char	NULL	1	SI	NO	NULL	NULL	NULL	NULL
405576483	TBL_USUARIO	Usua_Codigo	int	NULL	4	NO	NO	NULL	NULL	NULL	NULL
405576483	TBL_USUARIO	Sist_Codigo	int	NULL	4	NO	NO	NULL	FK_TBL_USUAR_Sist __4316F928	TBL_SISTEMA	Sist_Codigo
405576483	TBL_USUARIO	Usua_Nombre	varchar	NULL	50	SI	NO	NULL	NULL	NULL	NULL
405576483	TBL_USUARIO	Usua_Clave	varchar	NULL	50	SI	NO	NULL	NULL	NULL	NULL
405576483	TBL_USUARIO	Usua_Estado	char	NULL	1	SI	NO	NULL	NULL	NULL	NULL



853578079	TBL_USUARIO_ROL	Usu_Rol_Cod igo	int	NULL	4	NO	NO	NULL	NULL	NULL	NULL
853578079	TBL_USUARIO_ROL	Usu_Rol_Esta do	char	NULL	1	SI	NO	NULL	NULL	NULL	NULL
853578079	TBL_USUARIO_ROL	Usua_Codigo	int	NULL	4	NO	NO	NULL	FK_TBL_USUAR_Usu a__3F466844	TBL_USUARIO	Usua_Codig o
853578079	TBL_USUARIO_ROL	Rol_Codigo	int	NULL	4	NO	NO	NULL	FK_TBL_USUAR_Rol _C__46E78A0C	TBL_ROL	Rol_Codigo



## 4.12. ESTÁNDARES DE PROGRAMACIÓN

### 4.12.1. Organización de los Archivos

#### **Nombres de proyecto**

Para aplicaciones .NET, no es necesario nombrar un proyecto, ya que este toma el nombre de la carpeta que lo contiene. Para estos efectos el nombre del proyecto será el nombre de la carpeta. El nombre del proyecto debe indicar el modulo que contiene. Es permitido el uso de espacios en blanco. Ejemplo:

Modulo Seguridad.

Módulo de Registro.

#### **Archivos Fuente**

Los archivos fuente estarán contenidos dentro del directorio del proyecto. Los archivos fuente pertenecientes a las clases deben situarse dentro de la carpeta App\_Code, reservada por .NET, que se encuentra dentro del proyecto.

#### **Estructura del directorio.**

Las aplicaciones del sistema biométrico desarrollado en .NET generan por defecto las siguientes carpetas:

#### **Carpeta de Aplicaciones**

Archivos de clase que no son pantallas visuales y son necesarios para todo el proyecto.

#### **Carpeta App, MainWindow, BiometricoAplicacion,**

Archivos skinfiles o de presentación visual.

#### **Carpeta Bin**

Archivos dll de las aplicaciones o extensiones que se utilizan en el proyecto.

Asimismo, por convención se deben respetar la siguiente estructura de directorios:

#### **Carpeta Gráficos**

Contiene todos los gráficos necesarios para las páginas de formularios del sistema.

#### **Carpeta Reportes/controles**

Contiene las clases de controles personalizados.

#### **Carpeta Inicial**

Carpeta de inicio que debe contener al menos los archivos default del proyecto de solución en .net. Contiene las todas las otras carpetas mencionadas.



## Creación de Carpetas

Se pueden crear carpetas para contener los archivos .net que sean similares, asociadas por funcionalidad o por acceso de perfiles de usuarios.

### 4.12.2. Declaración de Variables

#### Número de declaraciones por línea

Se debe realizar una declaración por línea, debido a que esto permite la documentación.

Ejemplo:

```
Private String strCod;
```

Solo está permitida la declaración de múltiples variables en una línea cuando se trata de variables contadoras utilizadas para sentencias de iteración.

```
int i, j;  
for (int i = 1; i < 12; i++)  
{  
    for (int j = 1; j < 12; j++)  
    {  
  
    }  
}
```

#### Inicialización de variables

La inicialización de variables locales se debe realizar al declarar la variable. Ejemplo:

```
Private GMPPlan oGMPP = New GMPPlan();
```

#### Alcance de las variables

Las variables a nivel de clase se declaran como “private”, mientras que las variables dentro de los procedimientos y funciones se declaran de acuerdo a su tipo. Ejemplo:

```
Private GMPPlan oGMPP = New GMPPrograma();  
Int iCantidad = 0;
```



### **Declaración de Clases**

Se debe realizar una declaración por línea, debido a que esto permite la documentación. El nombre debe comenzar con el prefijo “o” seguido del nombre de la clase de manera abreviado. Ejemplo:

```
Private GMPPlan oGMPP = New GMPPlan();
```

### **4.12.3. Sentencias.**

#### **Sentencias simples**

Cada línea debe contener solo una sentencia.

#### **Return statements**

Las sentencias de retorno no deben contener paréntesis externos.

#### **Sentencias condicionales**

If

```
if (ValidarSemana)
{
    ListarTareas(2, ddlTarea.SelectedValue, 0);
}
```

If - else

```
if (ValidarSemana)
{
    ListarTareas (2, ddlTarea.SelectedValue, 0);
}
else
{
    lblMensaje.Text = "Datos No Validos."
}
```

If – elseif - else



```
if (ValidarSemana)
{
    ListarTareas (2, ddlTarea.SelectedValue, 0);
}
else if (ValidarMes)
{
    ListarTareas (1, ddlTarea.SelectedValue, 0);
}
else
{
    lblMensaje.Text = "Datos No Validos.";
}
```

### **Sentencias de iteración**

For

```
for (int i = 1 ; i < 31; i++)
{
}
```

### **Sentencias de selección**

Case

```
swiath (intOpcion)
{
    case 1:
        VerPlan(strCodigo);
        break;
    Case 2:
        VerPrograma(strCodigo);
        Break;
    case default:
```



```
MostrarMensaje("Selecciones una opcion valida.");  
}
```

### **Sentencia de manipulación de errores**

Try - catch

```
try  
{  
    cmd.Connection = oConexion;  
    oDataAdapter = New SqlDataAdapter(cmd);  
    oDataAdapter.Fill(oDataSet, sTabla);  
    tabla = oDataSet.Tables(sTabla);  
}  
catch (ex As Exception)  
{  
    MostrarMensaje(ex.Message.ToString());  
}
```

Try - catch - finally

```
try  
{  
    cmd.Connection = oConexion;  
    oDataAdapter = New SqlDataAdapter(cmd);  
    oDataAdapter.Fill(oDataSet, sTabla);  
    tabla = oDataSet.Tables(sTabla);  
}  
catch (ex As Exception)  
{  
    MostrarMensaje(ex.Message.ToString());  
}  
finally
```



```
{  
    cmd.close();  
}
```

#### 4.12.4. Asignación de Nombres

##### Nombres de variables

##### Variables generales

No deben contener underline ni espacios en blanco.

Deben tener un prefijo que indique el tipo de dato, de acuerdo a la siguiente tabla:

Tipo de dato	Prefijo
Boolean	Bln
Byte	Byt
Currency	Cur
Double	Dbl
Decimal	Dec
Date	Dtm
Float/Single	Sng
Integer	Int
Long	Lng
Object	Obj
String	Str
Variant	Vnt

##### Variables de propiedades

Los nombres de las variables que almacenan la información de propiedad de las clases deben ser en minúscula y tener el prefijo “p\_”, de preferencia con el mismo nombre de la propiedad de la clase Objeto de Dominio de la que forma parte. Ejemplo:       private  
int p\_itareapreventiva ;

```
private String p_ndescripcion ;
```

##### Variables de contadores



Se debe utilizar i, j, k, l, m, n para los nombres para las variables de contadores utilizados.

**Ejemplo:** private int i;

### **Variables especiales**

Existen algunas variables especiales que siempre deben nombrarse de la misma manera, de acuerdo a la siguiente tabla:

Nombre	Clase	Descripción
Ex	Exception	Manejador de errores de C#.NET
Sex	SQLException	Manejador de errores de SQL
Cex	ControladorExcepciones	Manejador propio de errores
Dial	Dialogos	Muestra mensajes de tipo dialogo
Itm	ListItem	Item de un control dropdownlist
oConexion	SqlConnection	Conexión a Base de datos
oDataAdapter	SqlDataAdapter	Adaptador de datos
oDataSet	DataSet	Contenedor de datos
oDataTable	DataTable	Tabla de datos
oComando	SqlCommand	Comando de datos
oCon	ConexionBD	Procesa las solicitudes a la base de datos
doS	DomainObject	Referencia a objetos de tipo Domain Object
oS	Entity	Referencia clases Entidad
Value	Cualquier tipo de dato	Parámetros del método Set las propiedades de los DomainObject.
resDial	DialogResult	Resultado devuelto por un cuadro de dialogo

### **Ejemplo:**

```
private doPlanMantenimiento doS As New doPlanMantenimiento();
```

En caso que se requiera declarar más de un tipo de estas variables especiales, se podrá preceder un número para enumerar si en caso se deba declarar más de una clase entidad.

Ejemplo:



```
private doPlanMantenimiento doS As New doPlanMantenimiento();  
private doPrograma doS1 As New doPrograma();
```

### **Nombres de constantes**

No deben contener underline ni espacios en blanco.

Deben estar en mayúsculas.

### **Nombres de clases**

#### **Clases Interfaz**

Deben tener el nombre del objeto seguido de underline y el formato que vayan a tener, no deben contener espacios en blanco. Los nombres deben tener la primera letra en mayúscula, de la misma manera si los nombres son compuestos y los nombres del formato deben ir en minúscula.

Ejemplo:

*PlanMantenimiento\_consulta*

*PlanMantenimiento\_listado*

#### **Clases Negocio**

No deben contener underline ni espacios en blanco. Los nombres deben comenzar con el prefijo "GF" seguido de la letra en mayúscula correspondiente al módulo y el nombre de la clase con la primera letra en mayúscula. Ejemplo:

GS : Gestión de Seguridad

E : Ejecución

Expediente : Nombre

GFExpediente

#### **Clases Entidad**

No deben contener underline ni espacios en blanco.

Los nombres de las clases deben ser sustantivos. **Ejemplo:** *TareaPreventiva.cs*

#### **Clases de Objetos de Dominio**

No deben contener underline ni espacios en blanco.

Los nombres de las clases deben ser sustantivos con el prefijo "do", de preferencia con el mismo nombre de la clase Entidad que la utiliza. Ejemplo: *doTareaPreventiva.cs*

### **Nombres de Controles**



Los nombres de los controles van precedidos por un prefijo de acuerdo al siguiente cuadro:

Control	Sufijo	Grupo de Controles	Ejemplo
TextBox	txt	Estándar	txtBuscar
Label	lbl	Estándar	lblBuscar
Button	btn	Estándar	btnBuscar
DropDownList	ddl	Estándar	ddlCategoria
CheckBox	chk	Estándar	chkPreferencia
CheckBoxList	cbl	Estándar	cblPreferencia
RadioButton	rbt	Estándar	rbtSexo
RadioButtonList	rbl	Estándar	rblSexo
Calendar	cal	Estándar	calFechaInicio
FileUpload	fup	Estándar	fupArchivo
Image	img	Estándar	imgFoto
LinkButton	lbt	Estándar	lbtSiguiente
ImageButton	ibt	Estándar	ibtDetalle
GridView	gdv	Datos	gdvLista
DataList	dls	Datos	dlsLista
RequiredFieldValidator	rfv	Validación	rfvNombre
RangeValidator	rav	Validación	ravEdad
RegularExpressionValidator	rev	Validación	revDato
CompareValidator	cov	Validación	covEmail
CustomValidator	cuv	Validación	cuvDato
SiteMapPath	smp	Exploración	smpMaster
Menu	mnu	Exploración	mnuMaster
TreeView	twv	Exploración	twvMaster
ScriptManager	scm	Extensiones AJAX	scmPagina
CalendarExtender	cae	Extensiones AJAX	caeFechaInicio
ConfirmButtonExtender	cbe	Extensiones AJAX	cbeEliminar



### **Nombre de Botones**

Los nombres de los botones deben tener la primera letra en mayúscula. Ejemplo:

- Grabar
- Imprimir
- Nuevo

### **Nombre de Etiquetas**

#### **Ingreso de datos**

Los nombres de las etiquetas para el caso de ingreso de datos deben tener la primera letra en mayúscula, en caso que los nombres estén separados por la palabra “de” estas deben ir en minúscula y la primera letra de los siguientes nombres deben empezar con mayúscula. Ejemplo:

- Tipo de Objeto
- Fecha de Inicio

#### **Columnas de Listado**

Los nombres de las etiquetas para el caso columnas de listado deben tener subrayado y la primera letra en mayúscula.

#### **4.12.5. Documentación**

##### **Documentación de Métodos.**

Los métodos (subrutinas y funciones) se documentan mediante comentarios antes de la declaración del método. Es indispensable que se documente todos los métodos propios creados en las clases. Ejemplo:

```
//Procedimiento para ir a la página principal del modulo  
private void Salir()  
{  
    Response.Redirect("~/default.aspx", True);  
}
```

Es opcional que se documente los métodos derivados de los controles.

#### **4.12.6. Formatos**

### Formato de Fecha

Los formatos de fechas deben ser dd/MM/yyyy.

### Formato de Mensajes

Mensajes de validación de campos requeridos

Los mensajes de validación de campos requeridos deben ir al costado de los campos y deben ser de color rojo. **Ejemplo:** *Ingrese Descripción*

### Mensajes de validación de fechas

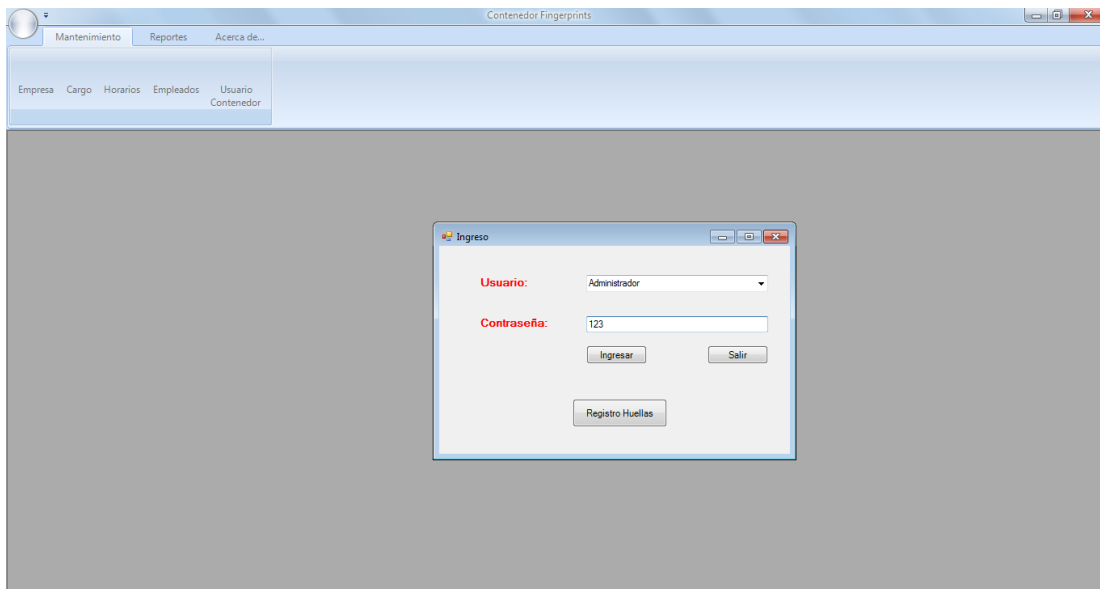
Los mensajes de validación de fechas deben ir al costado de los campos y deben ser de color rojo. **Ejemplo:** *Fecha (dd/mm/yyyy) entre: 1/1/2000-31/12/2078*

### Mensajes de validación de búsqueda de información.

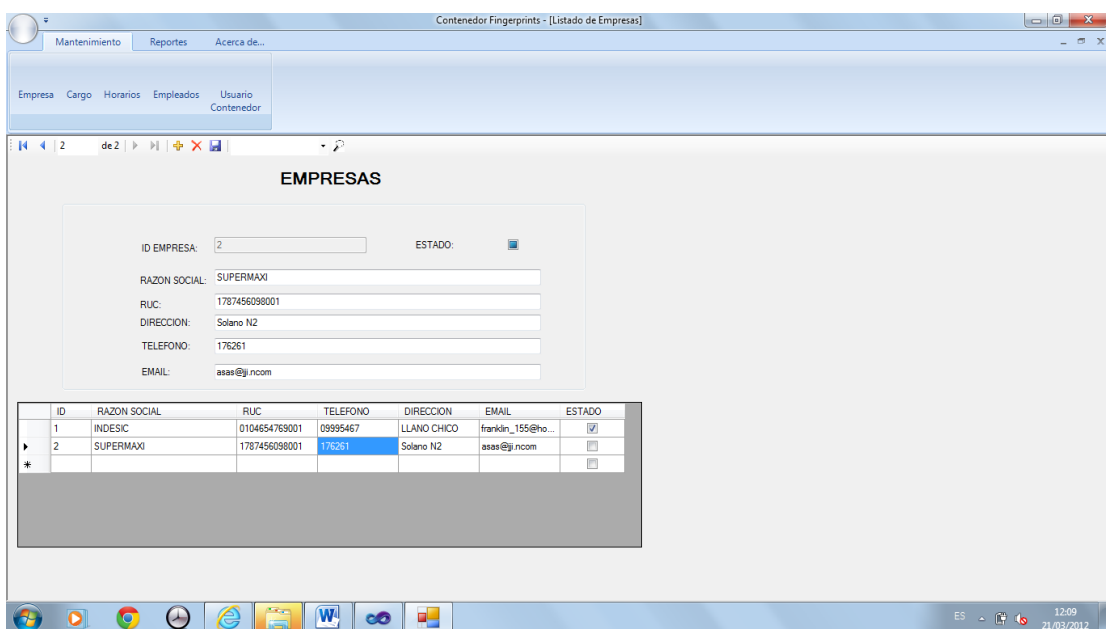
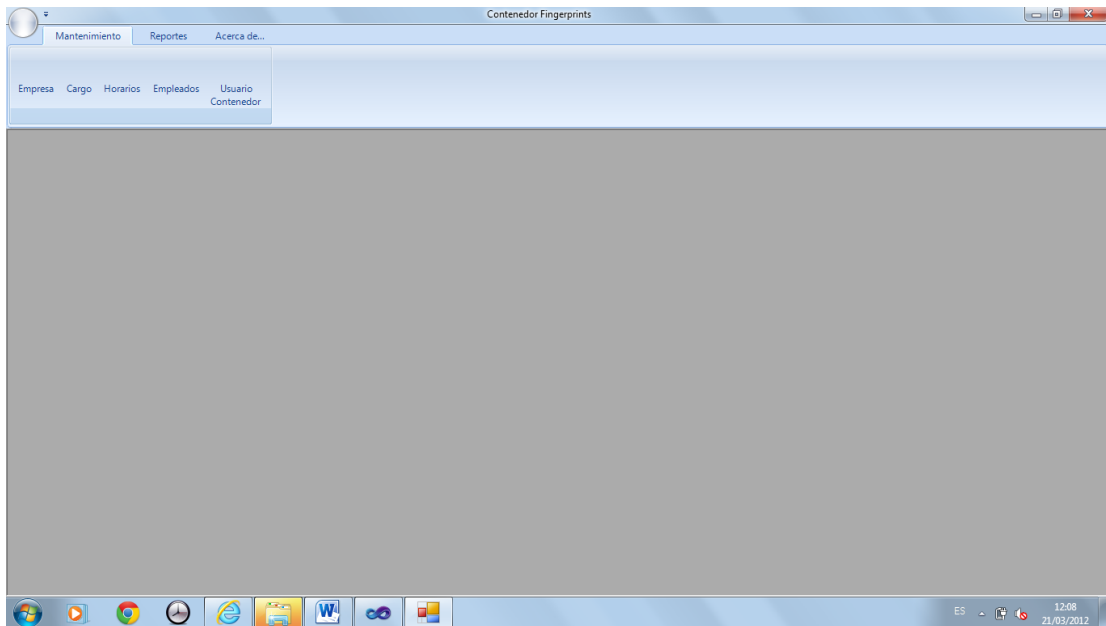
Los mensajes de validación de búsqueda de información deben ir en la parte inferior de la pantalla y deben ser “No se encuentran registros”.

## 4.12. PANTALLAS Y REPORTE (PROTOTIPO).

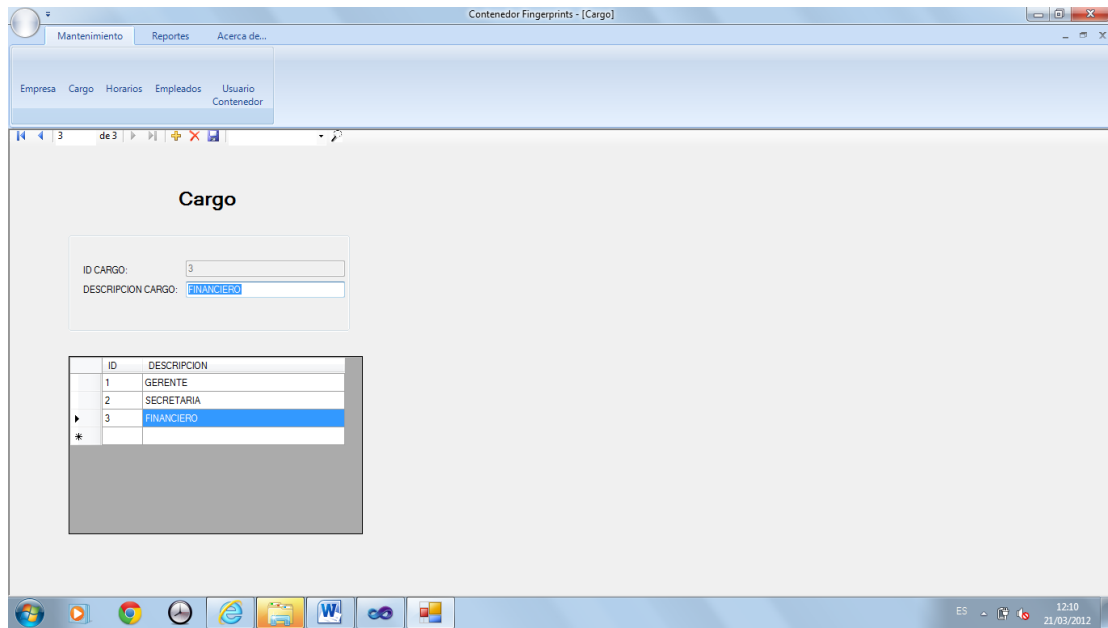
El sistema biométrico tiene la siguiente presentación.



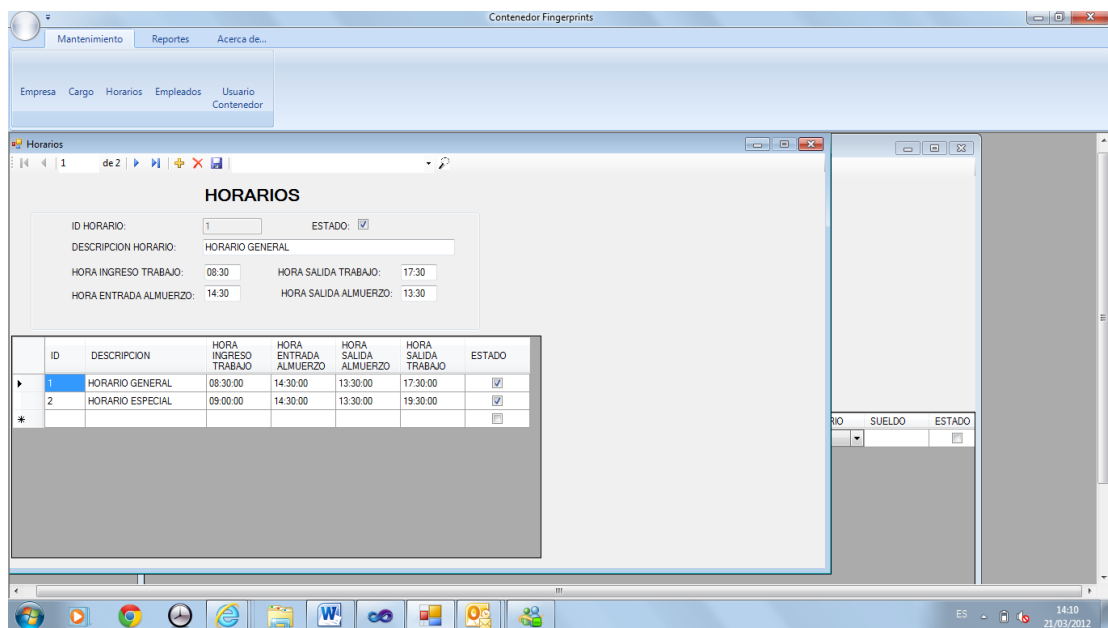
Pantalla de inicio debe registrar el cómo usuario o administrador.



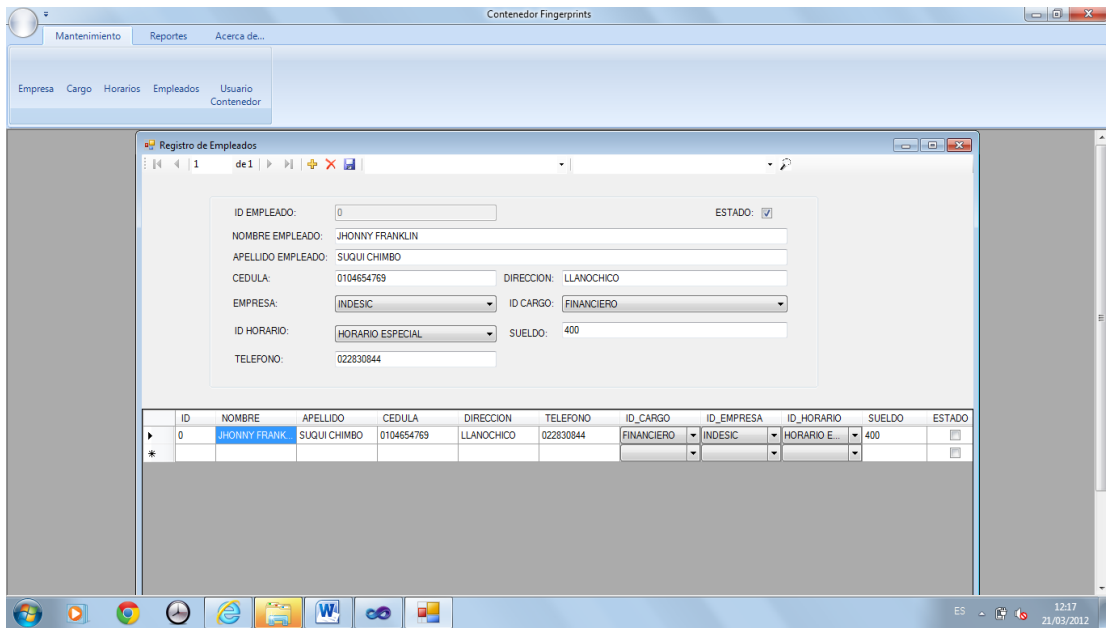
Para la inicialización del sistema debe generar el nombre de la empresa y sus respectivos datos.



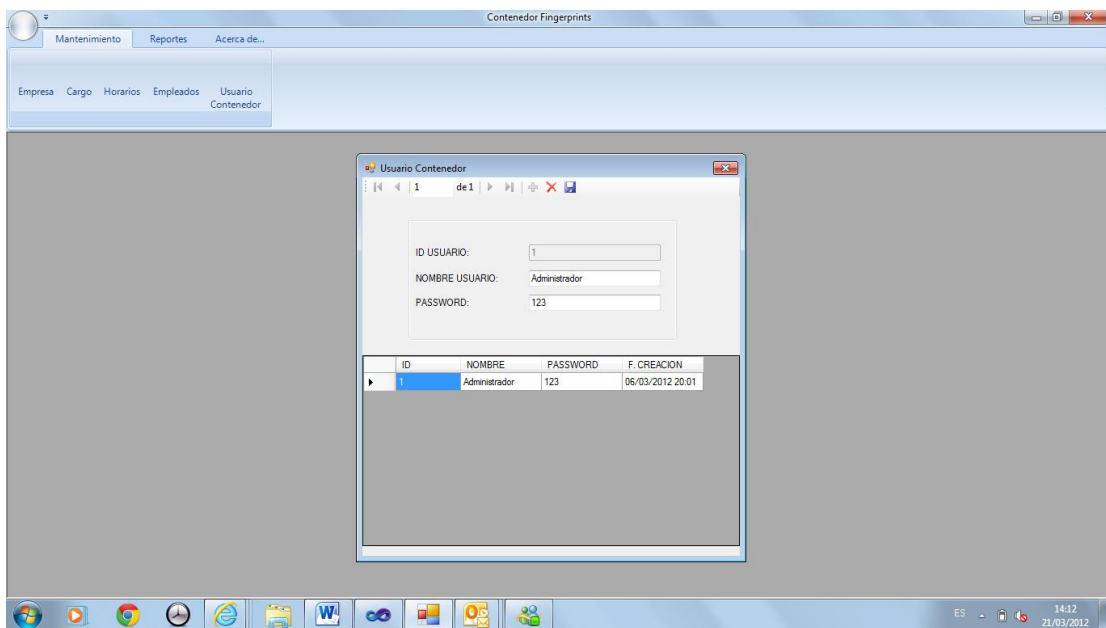
Para poder ingresar los datos del empleado primero se debe crear los cargos, y sus respectivos sueldos.



Así mismo debe crear los respectivos horarios para cada empleado.



Para el registro del empleado se debe ingresar los datos en el formulario.



Así mismo la seguridad de cada uno de los movimientos del sistema.



Calculos de Asistencia

Dep: INDESIC  
Nombre: Johanna Vilaña

Rango Tiempo: De 01/03/2012 a 14/03/2012

Calculador, Reporte, Exporta Datos, Diseñar Reporte

Orden:  Dpto.  UsuarioNo.  Nombre  Temp.  Descend

Emp No.	AC-No.	Nombre	Schedule	Dia	Horario	HoraEnt	HoraSal	Macro-Ent	Macro-Sal	Normal	TiemReal	No C/Ent	SaloTemp	Falta	HoraExtra	TiemTrabajado	Excepción	Simb
9	7	Johanna Vilaña	01/03/2012	Mañana	08:30	13:30					0.5							
9	7	Johanna Vilaña	01/03/2012	Tarde	14:30	17:30					0.5							
9	7	Johanna Vilaña	02/03/2012	Mañana	08:30	13:30					0.5							
9	7	Johanna Vilaña	02/03/2012	Tarde	14:30	17:30					0.5							
9	7	Johanna Vilaña	05/03/2012	Mañana	08:30	13:30					0.5							
9	7	Johanna Vilaña	05/03/2012	Tarde	14:30	17:30					0.5							
9	7	Johanna Vilaña	06/03/2012	Mañana	08:30	13:30					0.5							
9	7	Johanna Vilaña	06/03/2012	Tarde	14:30	17:30					0.5							
9	7	Johanna Vilaña	07/03/2012	Mañana	08:30	13:30					0.5							
9	7	Johanna Vilaña	07/03/2012	Tarde	14:30	17:30					0.5							
9	7	Johanna Vilaña	08/03/2012	Mañana	08:30	13:30					0.5							
9	7	Johanna Vilaña	08/03/2012	Tarde	14:30	17:30					0.5							
9	7	Johanna Vilaña	09/03/2012	Mañana	08:30	13:30					0.5							
9	7	Johanna Vilaña	09/03/2012	Tarde	14:30	17:30					0.5							
9	7	Johanna Vilaña	12/03/2012	Mañana	08:30	13:30					0.5							
9	7	Johanna Vilaña	12/03/2012	Tarde	14:30	17:30					0.5							

### Pantalla de visualización de los reportes

Reporte General Estadístico de Asistencia (INDESIC) 2012-03-01 to 2012-03-14

Name	AC-No.	Duty	Actual	Absent	Late	Early	OT	AFL	BL	NIn	NOut	Clock	Sick	Vaca	Other	WorkTme	Work
		WDia	WDia	WDia	Min.	Min.	Hora	Hora	WDia	Temp	Temp	Temp	Hora	Hora	Hora	Hora	%
INDESIC																	
Johanna Vilaña	7		8		8					16	16	32					0
Totot	1		8		8					16	16	32					0
Soporte Técnico																	
Jenny Suqui	1		10	2	7.5	90	120		3	17	18	40	3			12.63	20
Totot	1		10	2	7.5	90	120		3	17	18	40	3			12.63	20
Financiero																	
Anita Bastidas	6		10		10					20	20	40					0
Totot	1		10		10					20	20	40					0
Comunicación																	
Tatiana Guaman	5		10		10					20	20	40					0
Totot	1		10		10					20	20	40					0
Coordinación Becas																	
Narcisca Farinango	4		10	0.5	9.5		60			19	20	40				3.5	5
Totot	1		10	0.5	9.5		60			19	20	40				3.5	5
Grand Totot	5		48	2.5	45	90	180		3	92	94	192	3			16.13	5.2

### Reporte General de los empleados por asistencia

#### 4.13. PRUEBAS Y DEPURACIÓN.

Las siguientes pruebas que se ejecutara son:

Ejemplo de las validaciones.

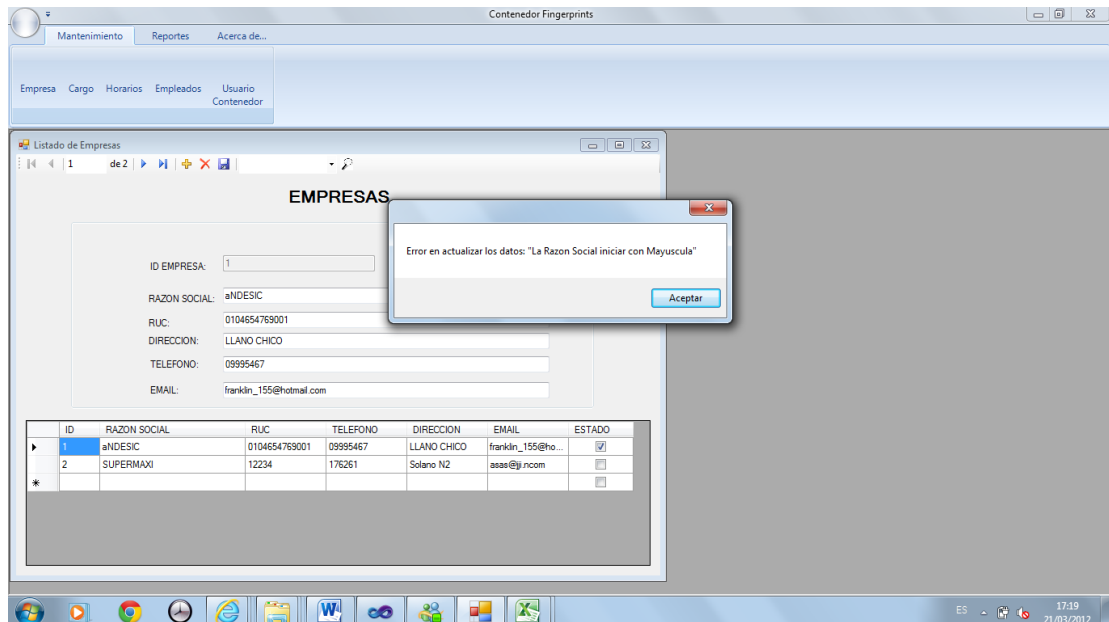
##### 1. La validación mediante la huella digital.

En esta prueba la no coincidió el código de la huella debido a que se debía convertir en un formato diferente ejemplo:

```
if (err == BSPError.ERROR_NONE)
{
    MaxFingersForEnrollText.Text =
Convert.ToString(initInfo.MaxFingersForEnroll);
    SamplesPerFingerText.Text =
Convert.ToString(initInfo.SamplesPerFinger);
    DefaultTimeoutText.Text =
Convert.ToString(initInfo.DefaultTimeout);
    EnrollImageQualityText.Text =
Convert.ToString(initInfo.EnrollImageQuality);
    VerifyImageQualityText.Text =
Convert.ToString(initInfo.VerifyImageQuality);
    SecurityLevelText.Text =
Convert.ToString(initInfo.SecurityLevel);
}
```

## 2. Ingreso de la información a la base de datos.

Para el ingreso de los datos la validación de cada uno de los campos es muy importante ya que evito ingresar datos basura a la base de datos.



## 3. Cálculos de horas laboradas.



En la prueba para el cálculo de las horas laboradas procedio a generar los reportes en donde no calculaba algunos registros por los diferentes horarios de los empleados.

#### 4. Impresión de Reportes.

En esta prueba únicamente se parametrizo algunos campos como:

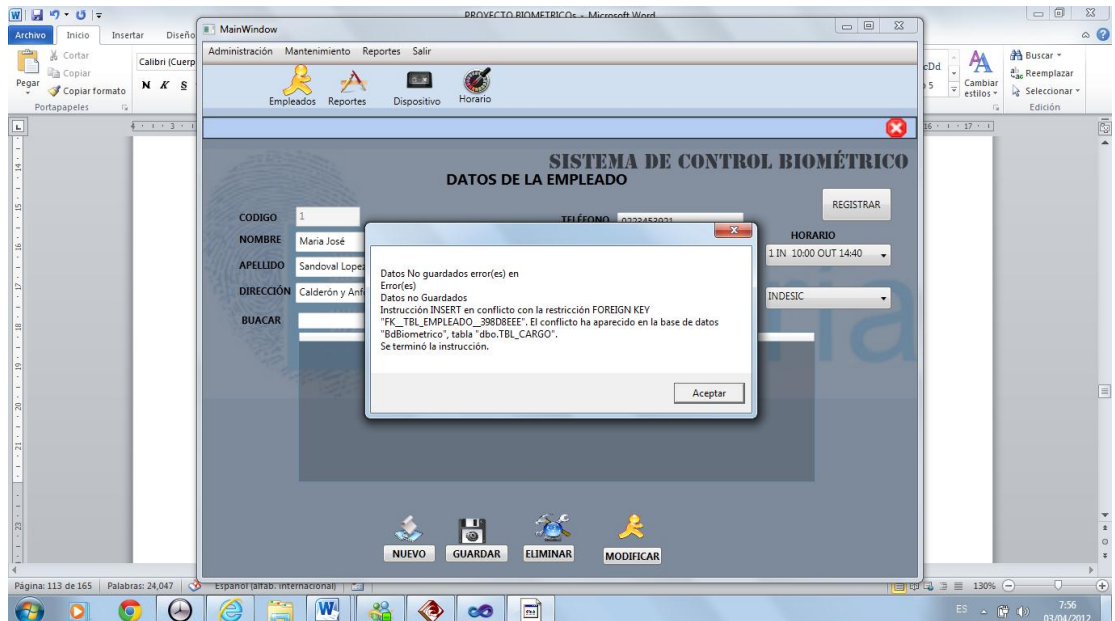
Fecha inicio: fecha final y mediante consultas de sql se puede visualizar los datos solicitados asi mismo no cincidia algunas filas del reporte.

The screenshot shows a software interface for generating reports. At the top, there are dropdown menus for 'Dep.' (set to INDESIC) and 'Nombre' (set to Johanna Vilaña). A 'Rango Tiempo' section has 'De' (01/03/2012) and 'a' (14/03/2012) date pickers. Buttons for 'Calcular', 'Reporte', and 'Exporta Datos' are visible. Below the form is a table with columns: Emp No., AC-No., Nombre, Schedule, Dia, Horario, HoraEnt, HoraSal, Marc-Ent, Marc-Sal, Normal, TiemReal, No C/Ent, Salio Tempr, Faltas, HoraExtra, TiemTrabajado, and Exce. The table contains 14 rows of data for employee Johanna Vilaña, showing work schedules from 01/03/2012 to 12/03/2012.

Emp No.	AC-No.	Nombre	Schedule	Dia	Horario	HoraEnt	HoraSal	Marc-Ent	Marc-Sal	Normal	TiemReal	No C/Ent	Salio Tempr	Faltas	HoraExtra	TiemTrabajado	Exce
9	7	Johanna Vilaña	Johanna Vilaña	01/03/2012	Mañana	08:30	13:30				0.5						
9	7	Johanna Vilaña	Johanna Vilaña	01/03/2012	Tarde	14:30	17:30				0.5						
9	7	Johanna Vilaña	Johanna Vilaña	02/03/2012	Mañana	08:30	13:30				0.5						
9	7	Johanna Vilaña	Johanna Vilaña	02/03/2012	Tarde	14:30	17:30				0.5						
9	7	Johanna Vilaña	Johanna Vilaña	05/03/2012	Mañana	08:30	13:30				0.5						
9	7	Johanna Vilaña	Johanna Vilaña	05/03/2012	Tarde	14:30	17:30				0.5						
9	7	Johanna Vilaña	Johanna Vilaña	06/03/2012	Mañana	08:30	13:30				0.5						
9	7	Johanna Vilaña	Johanna Vilaña	06/03/2012	Tarde	14:30	17:30				0.5						
9	7	Johanna Vilaña	Johanna Vilaña	07/03/2012	Mañana	08:30	13:30				0.5						
9	7	Johanna Vilaña	Johanna Vilaña	07/03/2012	Tarde	14:30	17:30				0.5						
9	7	Johanna Vilaña	Johanna Vilaña	08/03/2012	Mañana	08:30	13:30				0.5						
9	7	Johanna Vilaña	Johanna Vilaña	08/03/2012	Tarde	14:30	17:30				0.5						
9	7	Johanna Vilaña	Johanna Vilaña	09/03/2012	Mañana	08:30	13:30				0.5						
9	7	Johanna Vilaña	Johanna Vilaña	09/03/2012	Tarde	14:30	17:30				0.5						
9	7	Johanna Vilaña	Johanna Vilaña	12/03/2012	Mañana	08:30	13:30				0.5						
9	7	Johanna Vilaña	Johanna Vilaña	12/03/2012	Tarde	14:30	17:30				0.5						

#### 5. Verificación de duplicidad de registros.

Los además los campos que contenían relaciones duplicaban los códigos generando inconsistencia en la base de datos.



#### Otras pruebas.



Estas pruebas son propias del biométrico y no tuvieron ninguna novedad al momento de ejecutar con su respectivo software.

- **Prueba 1 - Precisión maximizada.** Líneas **rojas** de la carta ROC.
- **Prueba 2 - Velocidad maximizada.** Líneas **verdes** de las cartas ROC.
- **Prueba 3 - Tamaño mínimo de plantillas.** Líneas **azules** de las cartas ROC.

La comparación de plantillas fue realizada utilizando los **4 cores** del procesador.

Las curvas de características de operación del algoritmo **(ROC)** se utilizan típicamente para graficar la calidad de reconocimiento de un algoritmo. Las curvas ROC muestran la dependencia de la tasa de rechazo falso **(FRR)** sobre la tasa de reconocimiento falso **(FAR)**.

Pruebas del Algoritmo con escáner DigitalPersona U.are.U 4000 Detalle	Prueba 1	Prueba 2	Prueba 3
Tamaño promedio de la plantilla (bytes)	3865	631	238
Velocidad promedio de extracción (milisegundos)	150		
Velocidad de comparación (huellas por segundo)	32672	51884	44032
FRR con 0.001% de FAR	0.56%	1.18%	1.49%

#### 4.14. Instalación del Sistema.

El sistema biométrico consiste en el registro automático que funciona con la huella digital de los empleados, para agilizar los procesos de cálculo y evitar tiempos muertos en la organización.

##### ✓ Requerimientos del Sistema



- **PC** con procesador compatible **x86 (32bit)** o **x86-64 (64bit)** o **Mac** con procesador **x86** o **PowerPC**. Se recomienda procesador de 2.66 GHz o superior.
- Al menos **128 MB RAM** disponibles para la aplicación. Se requiere RAM adicional para aplicaciones que ejecutan comparación 1-a-muchos pues todas las plantillas deben ser cargadas en memoria. Por ejemplo, **25.000** plantillas (2 huellas cada una) requiere cerca de **50 MB de RAM** adicional
- ✓ **Espacio libre en disco duro (HDD):**
  - Al menos 1 GB para el desarrollo.
  - Se requieren 100 MB para el despliegue de componentes VeriFinger.
  - Se podría requerir espacio adicional en los siguientes casos:
    - VeriFinger no requiere almacenar la imagen original de la huella para comparar; las plantillas son suficientes. Sin embargo, se recomienda almacenar esta información para su uso potencial en el futuro.
    - Típicamente un motor de base de datos se ejecuta en servidores back-end (computador separado). Pero, el motor de base de datos se puede instalar en el mismo equipo para aplicaciones “standalone”. En este caso debe haber más espacio disponible en HDD para almacenar las plantillas. Por ejemplo, 25.000 plantillas (2 huellas cada una) almacenadas en una base de datos relacional requieren 60 MB de espacio libre. Adicionalmente la base de datos requiere espacio para ejecutarse, por favor consulte con su proveedor de BD el espacio requerido.
- ✓ **Escáner de huellas.** VeriFinger SDK incluye soporte para más de 50 escáneres de huellas bajo diferentes plataformas (Vea la sección anterior para más detalles).
- ✓ **Motor de base de datos** o conexión con él. Las plantillas VeriFinger pueden ser almacenadas en cualquier base de datos (incluso archivos) que soporte datos binarios. VeriFinger Extended SDK contiene módulos de soporte para:
  - Microsoft SQL Server (MS Windows).
  - PostgreSQL (Windows y Linux).



- MySQL (Windows y Linux).
- Oracle (Windows y Linux).
- SQLite (para todas las plataformas).
- ✓ **Conexión de red LAN (TCP/IP)** para aplicaciones cliente-servidor También se requiere conexión de red para VeriFinger Matching Server (incluido en la versión Extended del SDK). La comunicación no está encriptada, si se requiere seguridad, recomendamos el uso de una red dedicada (no accesible desde fuera) o una red segura (como un túnel VPN; que debe ser configurado utilizando las herramientas provistas por el sistema operativo o software de terceros).
- ✓ **Específico para Microsoft Windows:**
  - Microsoft Windows 2000/XP/2003/2008/Vista/7, 32-bit o 64-bit. Se recomienda la plataforma 32-bit para aplicaciones que utilicen escáner de huellas, pues la mayoría de los equipos operan bajo esta plataforma.
  - Microsoft .NET Framework o superior (para uso de componentes .NET)
  - Uno de los siguientes entornos de desarrollo:
    1. Microsoft Visual Studio 2005 SP1 o superior (para desarrollo bajo C/C++, C#, Visual Basic .NET).
    2. Sun Java 1.5 SDK o superior.
    3. Microsoft Visual Basic 6.
    4. Delphi 7.
- ✓ **Específico para Linux:**
  - Linux kernel 2.6 o superior, 32-bit o 64-bit. Se recomienda la plataforma 32-bit para aplicaciones que utilicen escáner de huellas, pues la mayoría de los equipos operan bajo esta plataforma.
  - glibc 2.3.6 o superior
  - libs GTK+ 2.10.x o superior y paquetes dev (para ejecutar los ejemplos del SDK y las aplicaciones basadas en ellos).
  - GCC-4.0.x o superior (para desarrollo de aplicaciones).
  - GNU Compilación 3.81 o superior (para desarrollo de aplicaciones).
  - Sun Java 1.5 SDK o superior (para desarrollo de aplicaciones con Java).



- pkg-config-0.21 o superior (opcional; sólo para compilar módulos de soporte para bases de datos con VeriFinger Matching).

✓ **Específico para Mac:**

- Mac OS X (versión 10.4 o superior)
- XCode 2.4 o superior (para desarrollo de aplicaciones)

## Contenido

El kit del sistema biométrico que usted adquirió incluye:

Scanner biométrico secuaster.

1. Hardware para el control de licencia HASP.

CD con:

- Instaladores del kit biométrico.
- Manual de usuario e instalación.
- Soporte técnico durante los 30 primeros días.

## Recomendación de uso de lector biométrico.

- El dispositivo puede utilizarse en temperaturas de 0 a 55 grados centígrados.
- No debe dejarse caer ni darle golpes con objetos contundentes.
- No debe limpiarse el cristal con productos detergentes ni corrosivos.
- No debe sumergirse en ningún tipo de líquidos.
- No se debe derramar ningún tipo de líquidos sobre el lector.
- El equipo al cual está conectado el lector, debe estar conectado a energía regulada.
- El lector debe estar conectado sólo a puertos USB estándares sin ninguna modificación.
- No se debe tratar de leer imágenes de dedos húmedos o con suciedad.
- Se debe ejercer una presión moderada al momento de colocar un dedo sobre el lector, para su correcto funcionamiento.
- Se debe colocar sólo la primera falange del dedo utilizado sobre el lector.
- No se debe rayar con ningún tipo de objeto corto punzante el cristal del lector.
- No se debe pintar con ningún tipo de tinta o pintura el cristal del lector.



- No se debe ubicar el lector en zonas con luz solar directa.
- Se debe seguir estrictamente el procedimiento de instalación proporcionado en el manual de instalación.
- Se deben seguir las recomendaciones del instructivo de matrícula para obtener una imagen de la mejor calidad posible.

### **Importancia de la Calidad de las Tomas en el Proceso de Registro (Enrolar)**

La captura de las huellas digitales para su registro en un sistema biométrico es fundamental para el buen funcionamiento de este, por tal motivo se debe tener todo el cuidado y control en varios aspectos al momento de tomar una huella digital desde un sistema automatizado, aunque dicho sistema ofrezca sistemas de control de calidad el factor humano sigue jugando un papel importantísimo en la calidad de las huellas capturadas. Una registro excelente garantizará un falso rechazo muy bajo, una mala matrícula incrementará el falso rechazo a niveles no tolerantes. Se debe tener en cuenta a la hora de tomar huellas digitales lo siguiente:

- El lector debe estar en línea con la mano, la persona no debe hacer ningún esfuerzo torciendo la mano para ubicar el dedo con el lector.
- La mano debe quedar apoyada sobre alguna superficie
- La mano debe estar extendida, sin ningún dedo empuñado.
- La punta del dedo debe quedar con la esquina superior del lector.
- No se debe colocar el dedo en punta.
- No se debe tomar la huella de forma rodada, es decir, como cuando se está tomando la huella en un papel.
- Se debe hacer presión sobre el lector.



#### **4.15. Recopilación y Carga de Datos.**

Para el ingreso de los datos al sistema se ingresara en base con los documentos legales de la organización; RUC- Empresa, Contrato laboral- Horarios de labores, Cédula ID- Empleado.

Mientras que para la carga de los datos se registrará a las personas que laboran y otras agentes particulares laboran indirectamente y que colaboran con dicha organización.

La persona encargada de hacer el proceso de registro de los empleados será únicamente el usuario administrador previa autorización de la dirección ejecutiva del INDESIC.

Para la él envío de los datos al sistema contable implantado (**SMA**) únicamente generará los archivos planos con los campos solicitados por el otro sistema para generar los roles de pago.

#### **4.16. Pruebas y Depuración Final en Funcionamiento.**

Una vez concluido el desarrollo se realizó una presentación del sistema a la Dirección Ejecutiva de la empresa y a algunos de los empleados del área, los cuales luego de ello brindaron sus observaciones y recomendaciones con el fin de hacer que el sistema mejore.

Antes de las pruebas con los usuarios, se ingresaron datos iniciales al sistema tales como son: las profesiones, los departamentos y los horarios de turno que serán dependientes para la ficha personal de cada empleado. Luego se hizo el llenado del formulario del empleado donde se colocaron datos personales de cada empleado y el registro de la huella digital.



La mayoría de módulos probados pasaron con éxito dicha prueba, aunque sí se hicieron algunas sugerencias sobre todo en la forma de la presentación de los reportes debido a que se había incluido la presentación del logotipo de INDESIC en el inicio de las hojas pero había que omitirlo porque se cuenta con hojas pre impresas con dicho logotipo, sólo irá el nombre de la empresa al inicio de cada reporte pero no el logotipo.

#### **4.17. Puesta en Marcha del Sistema.**

Una vez realizadas todas las pruebas, se procedió a la instalación definitiva del sistema en el computador de la empresa, para ello se revisó que el equipo cumpla con las características mencionadas en uno de los capítulos anteriores. Luego se procedió a instalar en el computador: Microsoft SQL Server 2008, Microsoft Visual Studio 2010, el Framework 2.2. de .Net, el SDK GrFinger 2007 junto con su licencia y el sistema de control de personal. Por último se conectó el lector biométrico, procediendo así mismo a la instalación del software necesario para que cada uno de ellos pueda ser reconocido por el computador y utilizado en el sistema.

Se ingresó al sistema y se realizó la verificación de cada uno de los datos de todos los empleados de la empresa para desde este momento comenzar a registrar su hora de entrada y salida utilizando el lector de huellas digital.

#### **4.18. Capacitación al Usuario Final.**

##### **Metodología de Capacitación aplicada para el usuario final.**

**Lugar:** El sitio para la capacitación a los usuarios es la calle Jorge Juan N31-87 y Pasaje San Gabriel “Aula de estudios de la Biblioteca del INDESIC”.

**Tiempo:** La Duración de la capacitación es de 2 horas.

**Temática:** Los temas a impartir son:

- Ventajas y desventajas del sistema.
- Funcionamiento del sistema.
- Como registrar la horas de entradas y salidas.
- Como calcula el sistema sus horas de atrasos y faltas.
- Qué hacer si no hay registra su huella.



- Otros.

**Estrategia.** La estrategia a aplicar es la confianza, el compañerismo y que existe para impartir dicha capacitación.

**Ayudas:** Las ayudas que se utilizará son los siguientes materiales:

Infocus.

Presentación de un video y la demostración del sistema.

Un volante del uso del dispositivo.

Presentación Power Point.

**Evaluación:** La evaluación contiene el siguiente formato:

NOMBRE				
El tema tratado fue importante. (Explique).				
Esta de acuerdo la operación con el sistema.	Si	No	Porque.	
Puede hacer utilizar sistema.	Si		No	
Escriba algunas ventajas de sistema. (Describa).	Si	No	Porque	
La capacitación impartida UD, entendió.	Si	No	Sugerencias:	
Califique al instructor:	Regular	Bueno	Muy Bueno	Excelente

#### 4.19. Capacitación al Personal Técnico.

**Lugar:** El sitio para la capacitación a los usuarios es la calle Jorge Juan N31-87 y Pasaje San Gabriel "Aula de estudios de la Biblioteca del INDESIC".

**Tiempo:** La Duración de la capacitación es de 2 horas.

**Temática:** Los temas a impartir son:

- Ventajas y desventajas del sistema.
- Funcionamiento del sistema.
- Como Crear nuevos empleados, sueldos, horarios, administradores.
- Como Generar los reportes de los empleados.
- Como registrar la huella del empleado.



- Otros.

**Estrategia.** La estrategia a aplicar es la confianza, el compañerismo y que existe para impartir dicha capacitación.

**Ayudas:** Las ayudas que se utilizará son los siguientes materiales:

Infocus.

Presentación de un video y la demostración del sistema.

Un volante del uso del dispositivo.

Presentación Power Point.

**Evaluación:** La evaluación contiene el siguiente formato:

NOMBRE				
El tema tratado fue importante. (Explique).				
Esta de acuerdo la operación con el sistema.	Si	No	Porque.	
Puede hacer utilizar sistema	Si		No	
Escriba algunas ventajas de sistema. (Describa).	Si	No	Porque	
La capacitación impartida UD, entendió	Si	No	Sugerencias:	
Califique al instructor:	Regular	Bueno	Muy Bueno	Excelente



## **CAPÍTULO V**

### **Impactos Esperados del Proyecto.**

#### **5.1 Científico Educativo Técnico Tecnológico Empresarial Social Económico.**

Uno de los elementos claves para la ejecución del proyecto en todos los aspectos, radica en el mejoramiento substancial de la competitividad de cada una de las empresas es por eso que el INDESIC como su mejor estrategia está consolidando el tiempo mediante un eficiente y efectivo control de asistencia.

##### **5.1.1 Educativo.**

El presente proyecto contribuirá a la sociedad actual, como base del conocimiento y del aprendizaje, para los alumnos de las áreas de sistemas o las carreras afines ya que se describe paso a paso como se desarrolló la programación del dispositivo biométrico. Mismas que estarán disponibles como fuente de consultas en las bibliotecas del INDESIC y de ITSCO.

##### **5.1.2 Científico.**

Después de las investigaciones realizadas se determina que el sistema desarrollado generó un gran impacto en los conocimientos científicos del desarrollador y que servirán como experiencia e inicio para poder integrar el presente software con otros sistemas que se manejan en la institución u otras.

##### **5.1.3 Técnico y Tecnológico.**

Con la utilización de este tipo de tecnología biométrica la empresa pasa a tener mayor prestigio institucionalmente y a competir a nivel nacional con las organizaciones estatales.



#### **5.1.4 Ambiental**

De los antecedentes recogidos en los sistemas biométricos diseñados el dispositivo para el control de asistencia no causaría ningún tipo de daños al medioambiente. Ya que el equipo es diseñado con material biodegradable.

#### **5.1.5 Económico.**

El impacto económico para el Instituto para el Desarrollo Social y de las Investigaciones Científicas INDESIC representa positivamente debido costos generales que se planteó en las propuestas anteriores 1 y 2 no representa gastos institucionales si no por el contrario permitirá pagar solamente las horas laboradas y aumentar la productividad del personal.

#### **5.1.6 Social.**

Con la implementación del sistema de control de asistencia cada empleado ya conoce las novedades acerca de su tiempo laborado, de tal manera que ya no se genera la inconformidad entre los compañeros del departamento.

### **5.8 Conclusiones.**

Luego de haber llegado a la finalización de la investigación se obtienen las siguientes conclusiones y recomendaciones.

1. Para la fase de levantamiento de información, el uso de técnicas de recolección de información como las entrevistas y las encuestas permitieron conocer mejor los procesos que la empresa llevaba para realizar el control de asistencia del personal, y luego a través de ello se pudo hacer un diseño adecuado del sistema para mejorar dichos procesos.
2. Mantener comunicación constante con cada una de las personas que utilizarán el sistema en el INDESIC durante todas las fases de desarrollo del sistema ayudó a obtener información relevante que permitió desarrollar un sistema acorde a las necesidades que dicha agencia tenía.



3. El lector biométrico Secugen Hámster III, por su capacidad para capturar imágenes de alta calidad libres de distorsión y en tiempo mínimo ayuda al control de asistencia del personal de INDESIC, no permitiendo la suplantación de la identidad de ninguno de los empleado y logrando que cada día (a partir de su implementación) ellos mejoren su puntualidad y cumplan así con toda su jornada de trabajo.
4. La base de datos creada en SQL Server se relaciona fácilmente con el sistema de control de personal, debido a que dicha herramienta forma parte de Visual Studio 2008 lenguaje en el que fue desarrollado dicho sistema.
5. El sistema es fiable gracias a que el acceso a las diferentes funcionalidades del sistema se lo hace a través de permisos asignados al usuario-administrador mediante un login y una contraseña, manteniendo así la integridad de la información.
6. Visual Studio 2008 al ser una plataforma de desarrollo muy potente permitió crear un sistema funcional y de fácil manejo, puesto que el ingreso de datos es cómodo, el cálculo de las horas trabajadas es el adecuado y porque varios de los controles de DevExpress utilizados tales como: skins, ribbons y navbars hacen que la interfaz gráfica del mismo sea agradable a la vista del usuario.
7. El Sistema de Control de Personal almacena la información en una base de datos relacional, permitiendo que el acceso a la información relevante de cada empleado sea rápido y la obtención de los reportes sea oportuno.

### **5.9 Recomendaciones.**

1. Debe realizarse una capacitación inicial a los usuarios del sistema para que no tengan ningún problema al momento de utilizar cada uno de los módulos del sistema.
2. Hay que notificar al desarrollador del sistema acerca de los errores que se presenten en la manipulación de la información, si es que eso sucediera el desarrollador tendrá que solucionarlo lo más rápido posible.
3. Se debe registrar las horas de entrada y salida a través del lector biométrico en los horarios establecidos, ya que de lo contrario no se podrá tener información adecuada



sobre dichos ingresos y habrá errores al momento de obtener los informes de horas trabajadas.

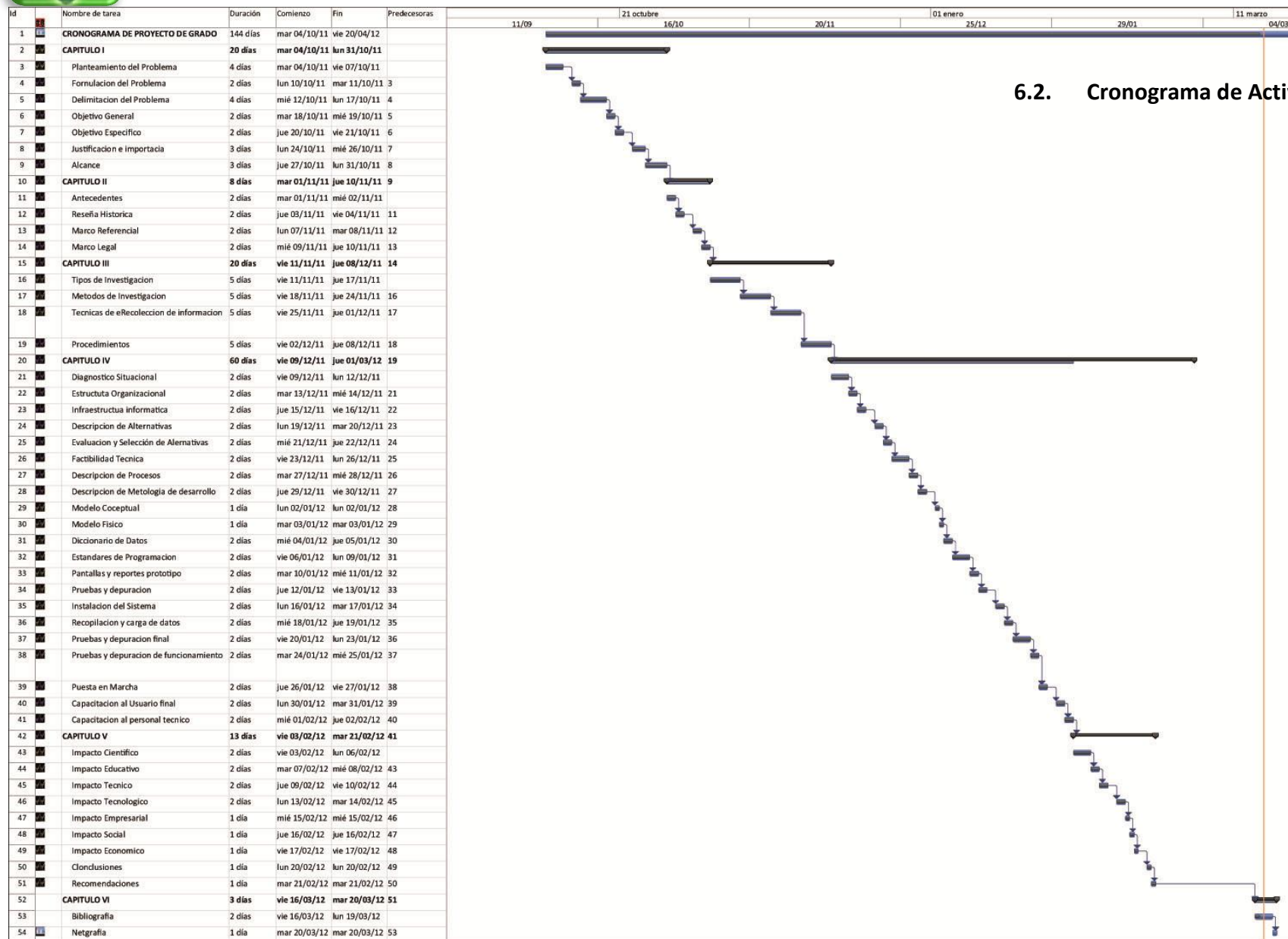
4. No existe excusa para que el usuario no cheque asistencia, pues no necesita contraseña, tarjeta u otro tipo de identificación, basta su huella digital.
5. Realizar un mantenimiento periódico del lector biométrico para que permanezca limpio, libre de grasa, polvo o cualquier otra impureza que lleve a obtener una imagen de mala calidad.
6. Cuando se vaya a hacer el registro inicial de la huella de cada empleado, se recomienda realizarlo un par de veces con la finalidad de tener una mejor imagen para que cuando el lector haga la identificación/validación no haya mucho retardo en ello.
7. El computador donde vaya a estar instalado el sistema debe contar con el abastecimiento de energía eléctrica suficiente para que cuando haya una suspensión del fluido eléctrico este no se apague y se produzca pérdida de datos.
8. Cambiar la clave de usuario periódicamente y realizar frecuentemente respaldo de la información, para de esta manera mantener la seguridad de la información de la empresa.
9. Para obtener una interfaz de sistema agradable a la vista del usuario que cada día es más exigente, una buena alternativa es utilizar varios de los controladores de Microsoft Expression Blend 4, ya que dicha aplicación facilitarán el diseño de interfaces impactantes.

**CAPÍTULO VI****Aspectos Administrativos****6.1 Presupuesto.**

<b>ANEXO</b>			
<b>DETALLE</b>	<b>CANTIDAD</b>	<b>COSTO UNITARIO</b>	<b>COSTO TOTAL</b>
<b>IMPRESIÓN</b>		12.50	12.50
Presentación Tesis	10	5.00	50.00
Consultas y Diseños	5	2.50	12.50
Manuales	5	5.00	25.00
<b>ESTACIÓN DE TRABAJO</b>	<b>1</b>	800.00	800.00
<b>ÚTILES DE ESCRITORIO</b>	<b>1</b>	500.00	500.00
<b>INTERNET</b>	<b>6</b>	25.00	150.00
<b>VARIOS</b>	<b>6</b>	30.00	180.00
<b>EQUIPOS ESPECIAL</b>	<b>1</b>	500.00	500.00
<b>ALIMENTACIÓN</b>	<b>180</b>	1.50	270.00
<b>TRANSPORTE</b>	<b>30</b>	1.50	45.00
<b>CAPACITACIÓN</b>	<b>10</b>	20.00	200.00
		<b>TOTAL</b>	<b>\$ 2,745.00</b>

**Tabla 17:** Presupuesto General del Proyecto.

Autor: Jhonny Suqui



6.2. Cronograma de Actividades.

Proyecto: CRONOGRAMA FINAL  
Fecha: dom 18/03/12

Tarea	Resumen	Hito externo	Resumen inactivo	Informe de resumen manual	Sólo fin
División	..... Resumen del proyecto	Tarea inactiva	Tarea manual	Resumen manual	Fecha limite
Hito	◆ Tareas externas	Hito inactivo	Sólo duración	Sólo el comienzo	Progreso

Página 1



### 6.3. Bibliografía.

- [L1] PRESSMAN, Roger. Ingeniería del Software – Un enfoque práctico, McGraw-Hill/Interamericana Editores, México, 2005, ISBN 970-10-5473-3
- [L2] LARMAN, Craig. UML y Patrones. Introducción al análisis y diseño orientado a objetos. Prentice Hall, México 1999. ISBN: 970-17-0261-1
- [L3] CONNOLLY, Thomas M.; BEGG, Carolyn E. Sistemas de Bases de Datos. Un enfoque práctico para diseño, implementación y gestión. Person Educación S.A., Madrid, 2005, Cuarta Edición. ISBN: 84-7829-075-3
- [L4] KENDALL, Kenneth; KENDALL, Julie. Análisis y Diseño de Sistemas. Prentice Hall, México 1997. Tercera Edición.
- [L5] McCONNELL, Steven. Desarrollo y gestión de Proyectos Informáticos. McGRAW-HILL, Madrid 1997.
- [L6] PIATTINI, Mario G; CALVO-MANZANO, José A.; CERVERA, Joaquín; FERNÁNDEZ, Luis. Análisis y diseño de Aplicaciones Informáticas de Gestión. Una perspectiva de Ingeniería del Software. RA-MA Editorial, Madrid – España; ALFAOMEGA Grupo Editor, México 2004.

### 6.4. Web- Grafía.

- [I01] GALVIS TRASLAVIÑA, Carlos Mauricio. Introducción a la Biometría. Disponible en: <http://www.monografias.com>
- [I02] WIKIPEDIA. Software Development Kit. Disponible en: [http://es.wikipedia.org/wiki/Software\\_development\\_kit](http://es.wikipedia.org/wiki/Software_development_kit)
- [I03] El Reconocedor Digital de Huellas, Biopod de APC recibió el Primer Premio como “Mejor solución de Seguridad”, 2005. Disponible en: [http://www.bureaudeprensa.com/es/view.php?bn=bureaudeprensa\\_hardware&key=1123012467](http://www.bureaudeprensa.com/es/view.php?bn=bureaudeprensa_hardware&key=1123012467)
- [I04] AUDIOMETRICS. Lector de Huellas Digitales. Disponible en: <http://www.audiotronics.es/product.aspx?productid=24922>
- [I05] AMERICAN POWER CONVERSION CORPORATION. HELP BIOPOD APC. Disponible en: <http://www.apc.com>
- [I06] SECUGEN. SecuGen Hámster Plus. Disponible en: <http://www.secugen.com/products/php.htm>
- [I07] SECUGEN. Top 5 Reasons for Choosing SecuGen. Disponible en: <http://www.secugen.com>
- [I08] SECUGEN CORPORATION. SecuGen® Hamster III. Disponible en: <http://www.secugen.com>
- [I09] SYBASE. PowerDesigner. Disponible en:



<http://www.sybase.es/products/modelingdevelopment/powerdesigner/powerdesignerbenefits>

[I10] WIKIPEDIA. Microsoft Visual Studio. Disponible en:

[http://es.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](http://es.wikipedia.org/wiki/Microsoft_Visual_Studio)

[I11] CORPORACIÓN SYBVEN. Arquitectura Empresarial con SYBASE® POWERDESIGNER®. Disponible en:

[http://www.corporacionsybven.com/portal/index.php?option=com\\_content&view=article&id=108:sybase-powerdesigner&catid=87:modelladomultidimensional&Itemid=89](http://www.corporacionsybven.com/portal/index.php?option=com_content&view=article&id=108:sybase-powerdesigner&catid=87:modelladomultidimensional&Itemid=89)

[I12] WIKIPEDIA. Microsoft SQL Server. Disponible en:

[http://es.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](http://es.wikipedia.org/wiki/Microsoft_SQL_Server)

[I13] MICROSOFT. ¿Qué es SQL Server 2005? Disponible en:

<http://www.microsoft.com/spain/sql/productinfo/overview/what-is-sqlserver.msp>

[I14] MICROSOFT. Microsoft SQL Server. 12 veces más conveniente: 12 modos en que ahorra costos. Disponible en:

<http://www.microsoft.com/latam/sqlserver/ahorro.aspx>

[I15] GRIAULE. GRFINGER SDK 2007. Disponible en: <http://www.griaule.com>

[I16] GRIAULEBIOMETRICS. FINGERPRINT SDK. Disponible en:

[http://www.griaulebiometrics.com/page/es/fingerprint\\_sdk](http://www.griaulebiometrics.com/page/es/fingerprint_sdk)

[I17] WIKIPEDIA. Crystal Reports. Disponible en:

[http://es.wikipedia.org/wiki/Crystal\\_Reports](http://es.wikipedia.org/wiki/Crystal_Reports)<http://www.udem.cl/cyt/derecho/firma.html>

[I18] MSDN MICROSOFT. ¿Qué es Crystal Reports? Disponible en:

[http://msdn.microsoft.com/es-es/library/ms225593\(VS.80\).aspx](http://msdn.microsoft.com/es-es/library/ms225593(VS.80).aspx)

[I19] DEVELOPER EXPRESS V2009. Demo Center DevExpress – WinForms. Disponible en:

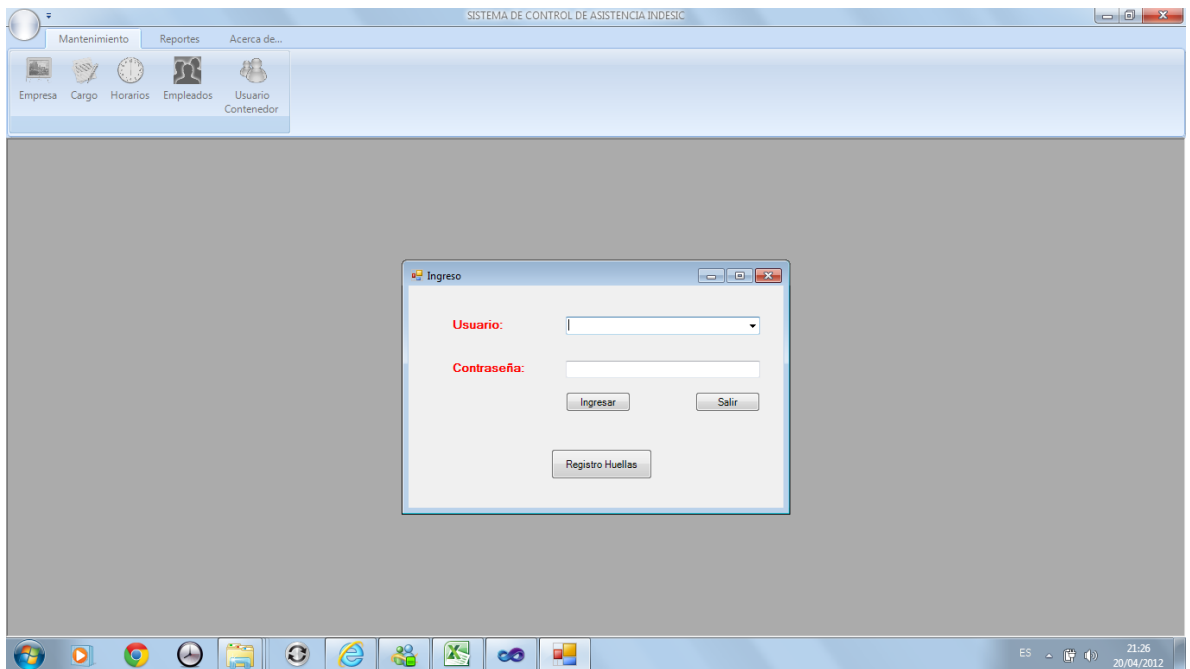
<http://www.devexpress.com>



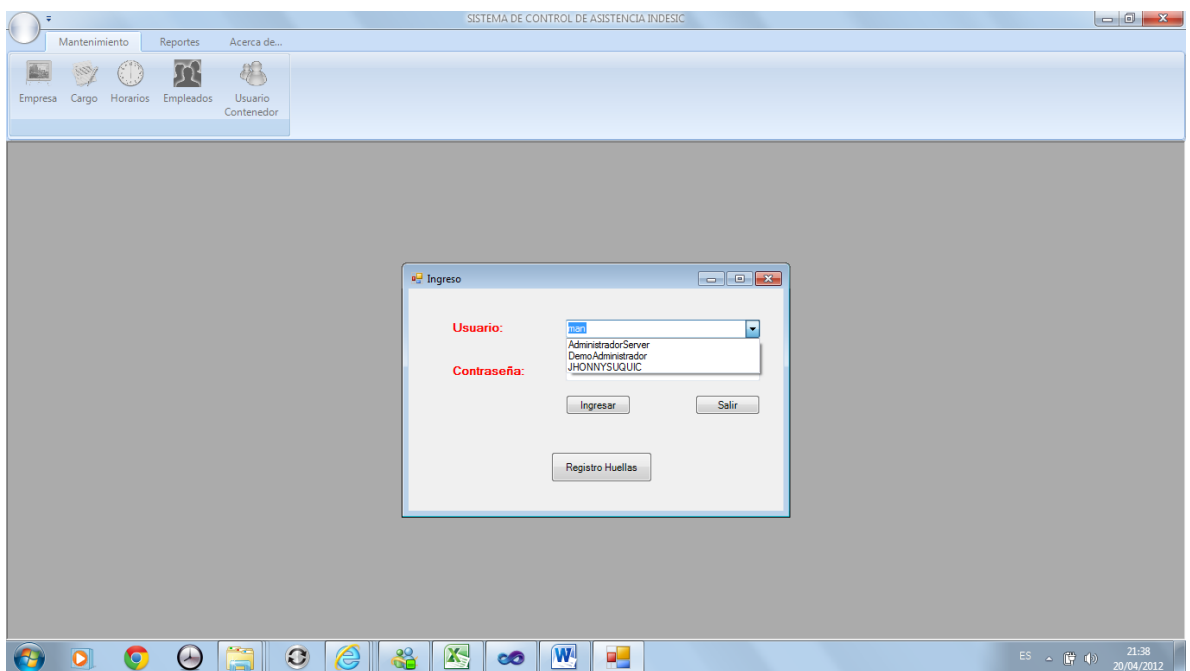
## **CAPÍTULO VII.**

### **7. ANEXOS.**

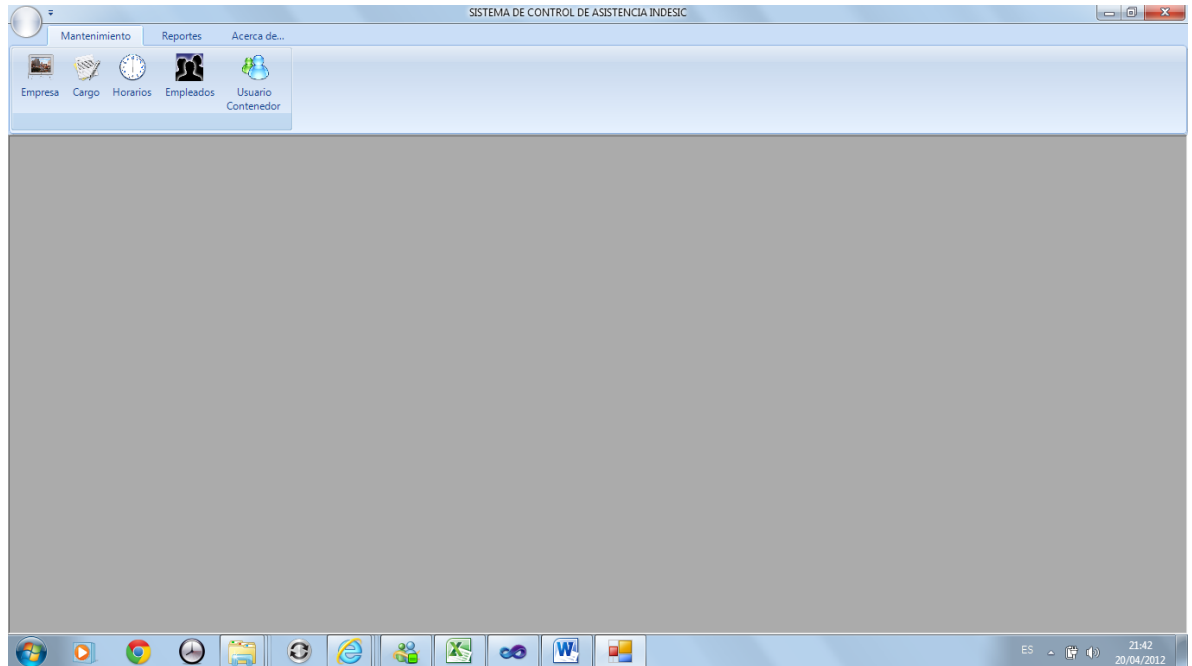
## 7.1. Manual de Usuario.



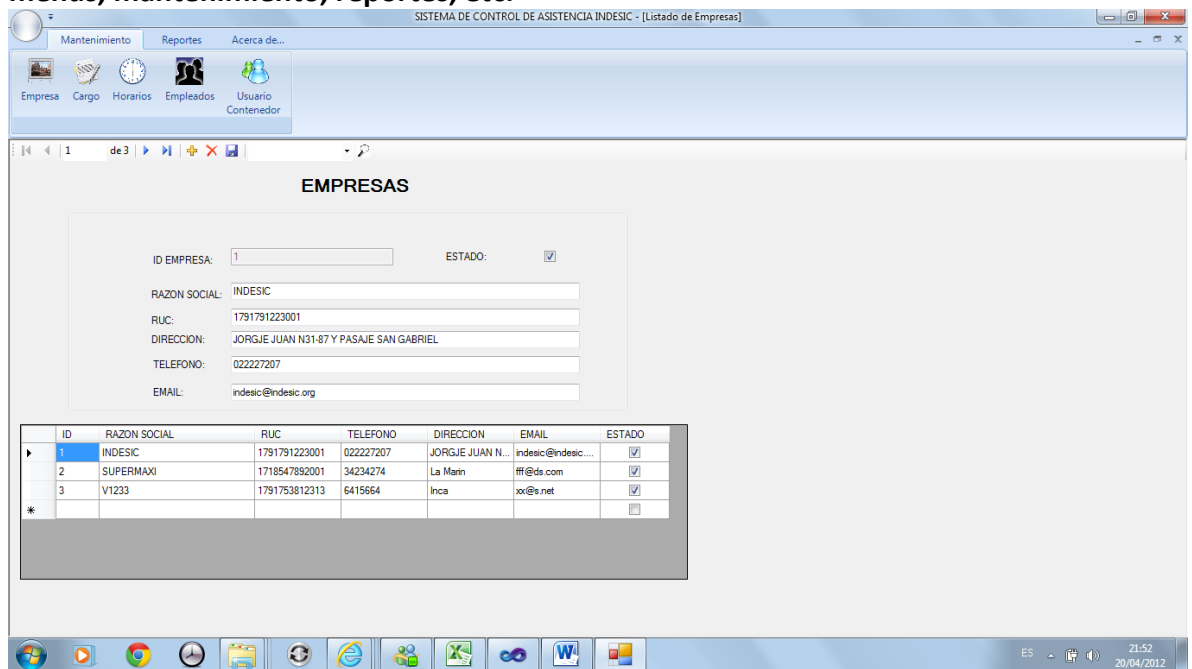
### 1 Pantalla de Bienvenida.



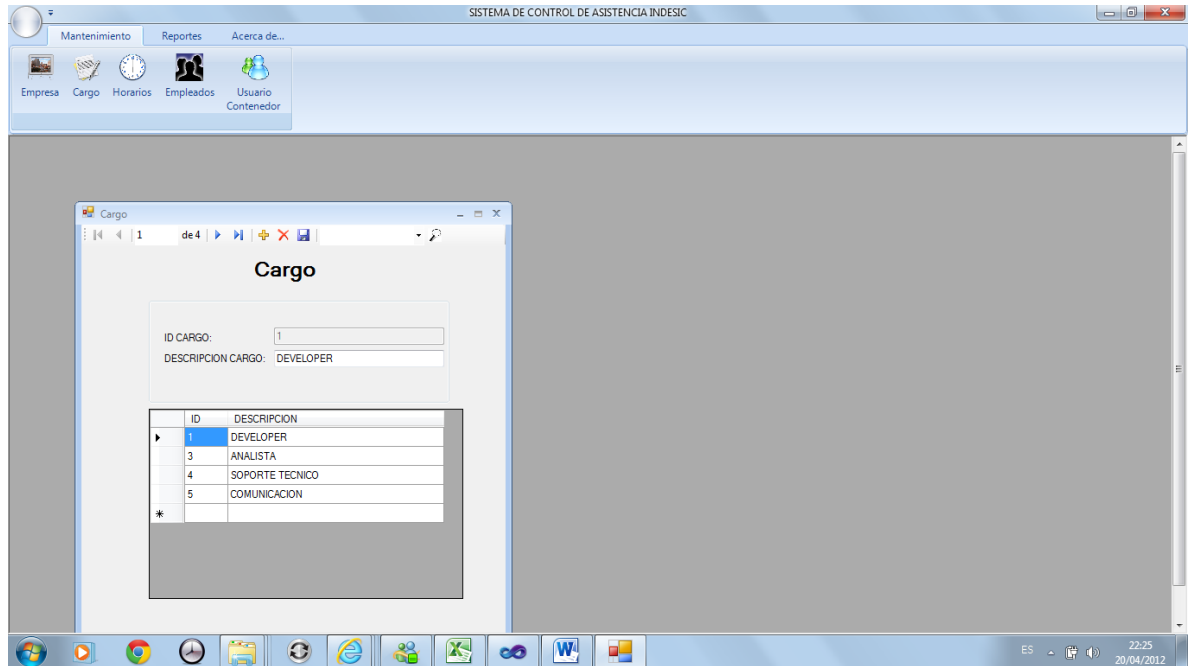
2. Para iniciar el programa debemos seleccionar el usuario que vamos iniciar. Ejemplo Administrador Server, luego digitamos la clave. Presionamos Enter o ingresar.



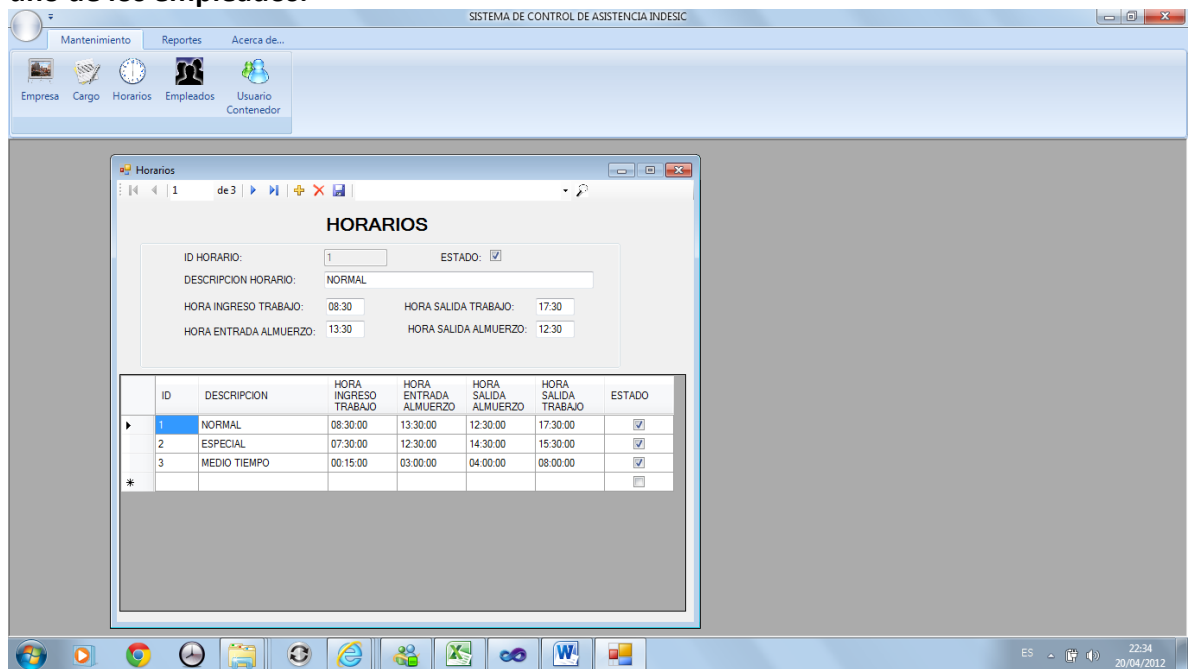
3. En esta pantalla disponemos de las siguientes opciones con sus respectivos menús, mantenimiento, reportes, etc.



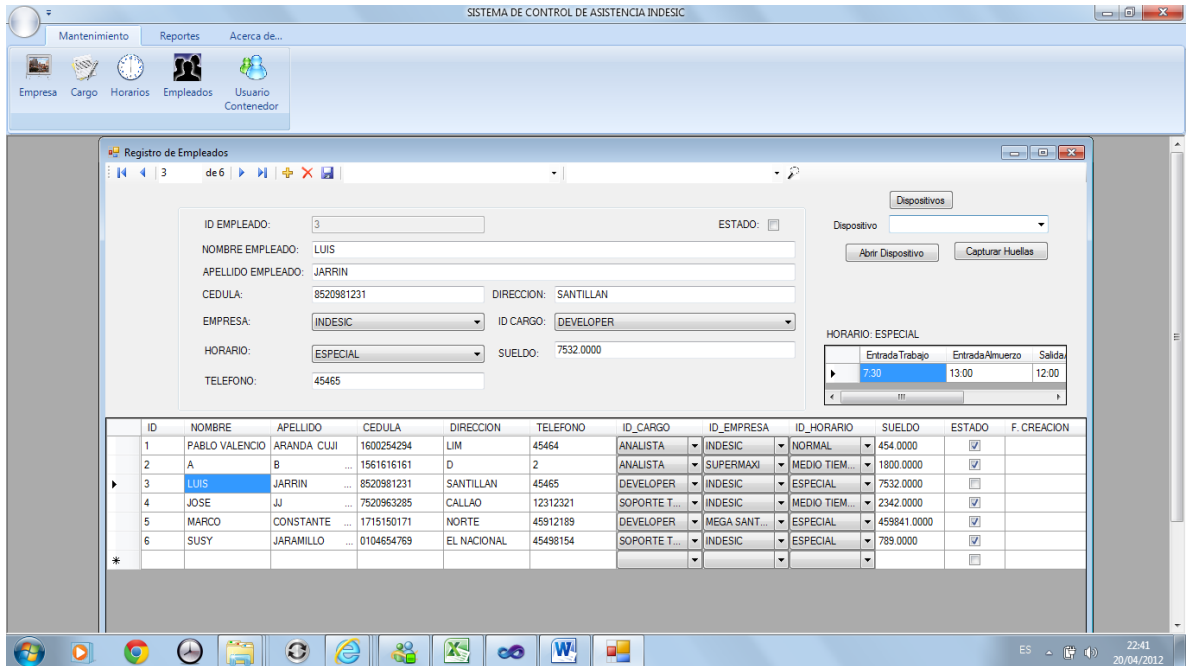
4. En esta pantalla podemos crear nuevas empresas los campos requeridos son : Razón Social, Ruc, Dirección, Teléfono, y finalmente email.



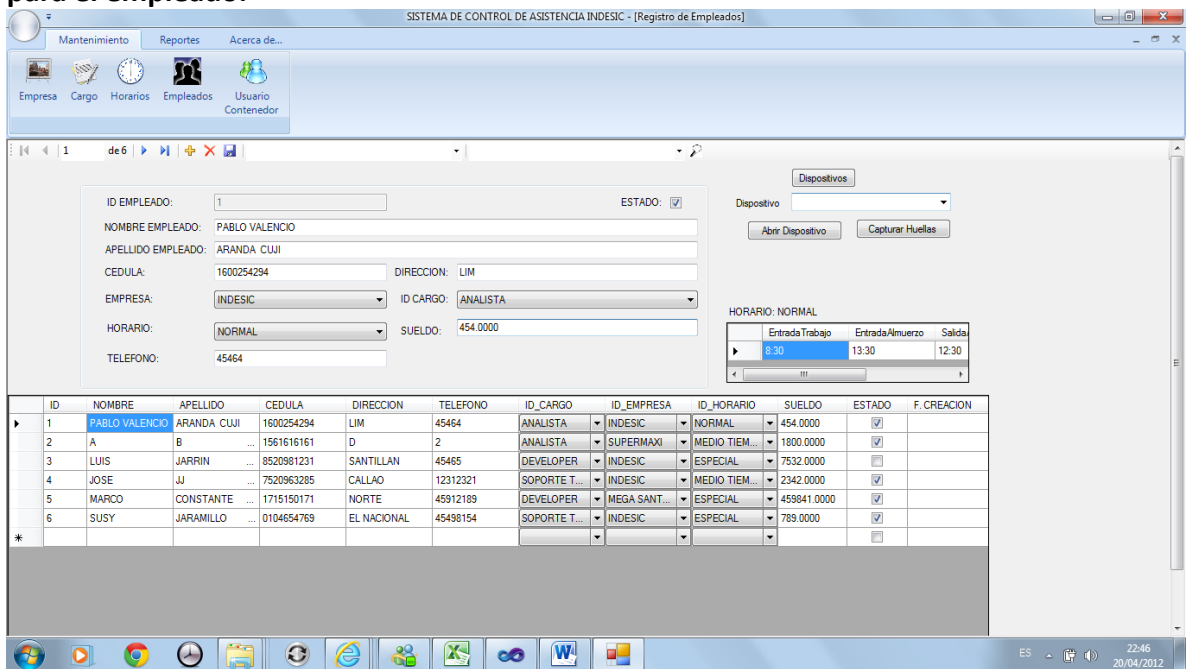
5. En esta pantalla podemos crear todos los Cargos para luego poder asignar a cada uno de los empleados.



6. En esta pantalla creamos los horarios de cada empleado. Importante se debe digitar las horas en formato de 24 horas



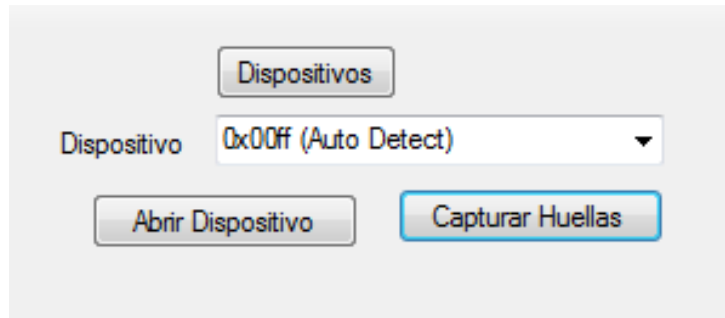
7. En esta podemos ingresar los empleados. Importante se debe asignar horario para el empleado.



8. Para enrolar las huellas del empleado se debe:

1. Dar clic en el dispositivo para que se cargue.
2. Una vez cargado se debe seleccionar en el combo.
3. Seleccionamos en el Botón abrir dispositivo. (En la parte inferior izquierda se

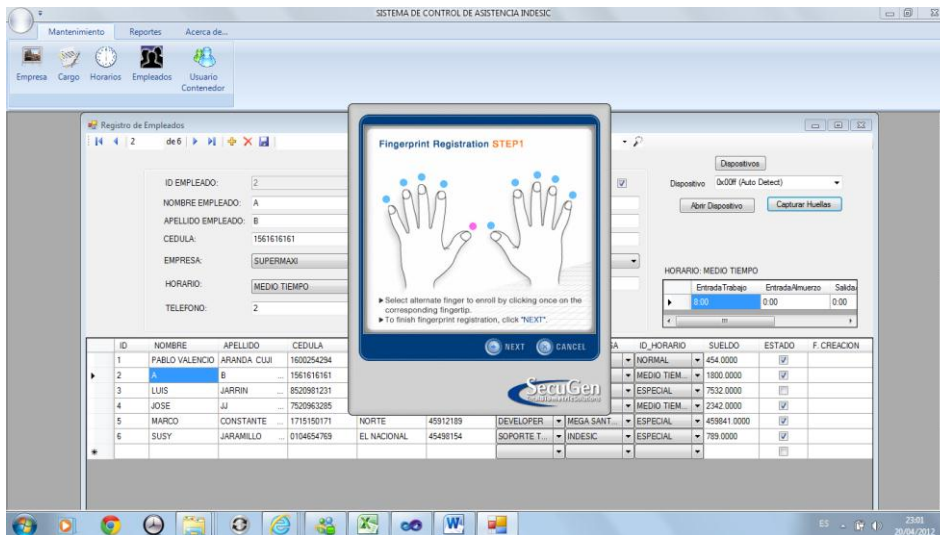
mostrará un mensaje Dispositivo Abierto



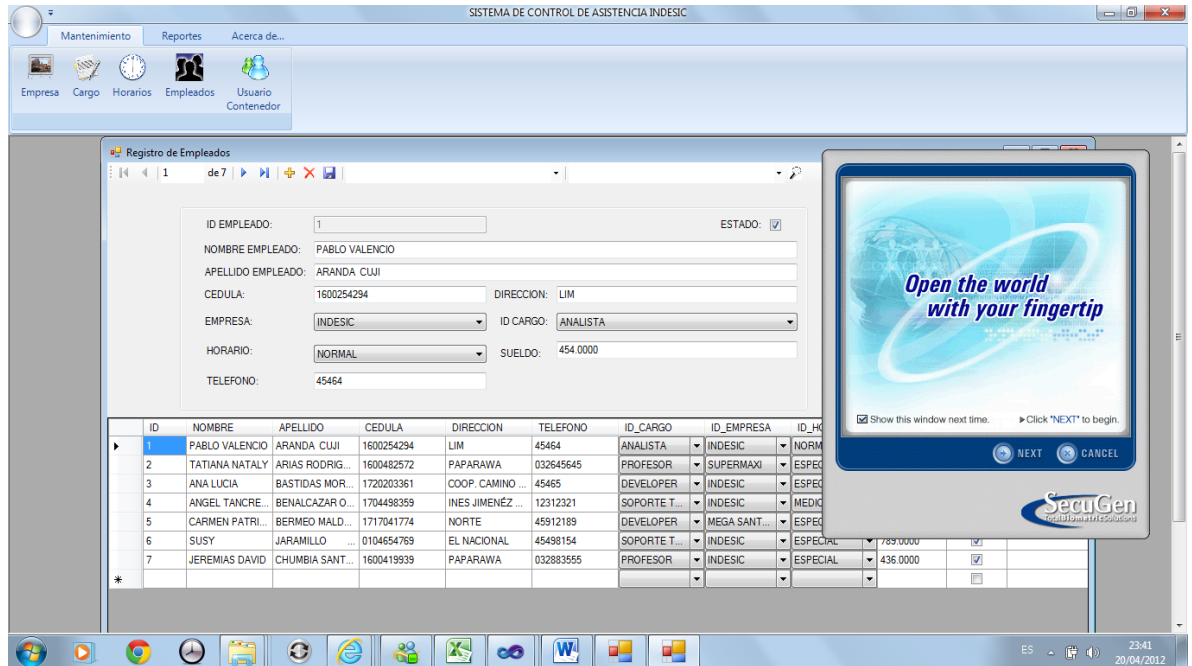
4. Una vez terminando el paso 4 estamos listos para que e pueda enrollar al empleado.



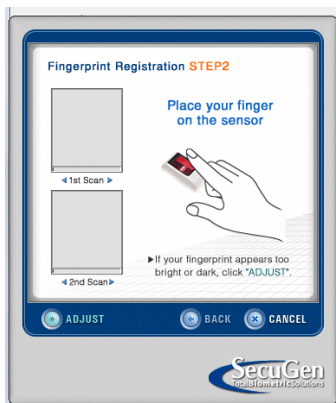
5. Damos clic en el botón Capturar Huellas y nos mostrará la siguiente ventana.



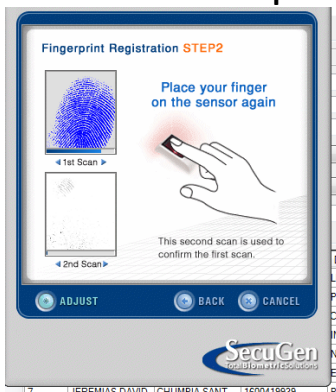
9. En esta ventana se debe seleccionar el dedo que va ser enrollado y presionamos en botón na Next.



10. En la siguiente ventana tenemos nos indica que debemos poner el dedo en el biométrico.

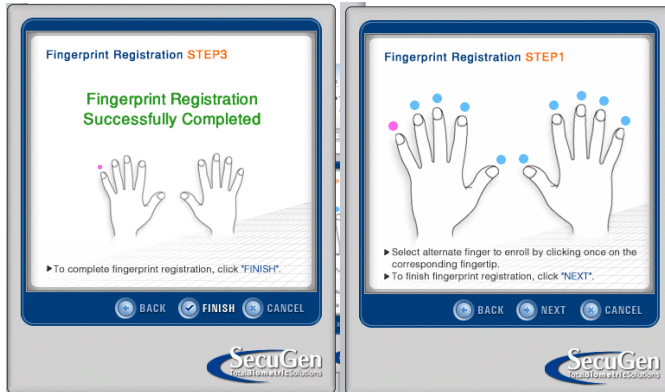


11. En esta ventana ponemos en dedo en el biométrico.

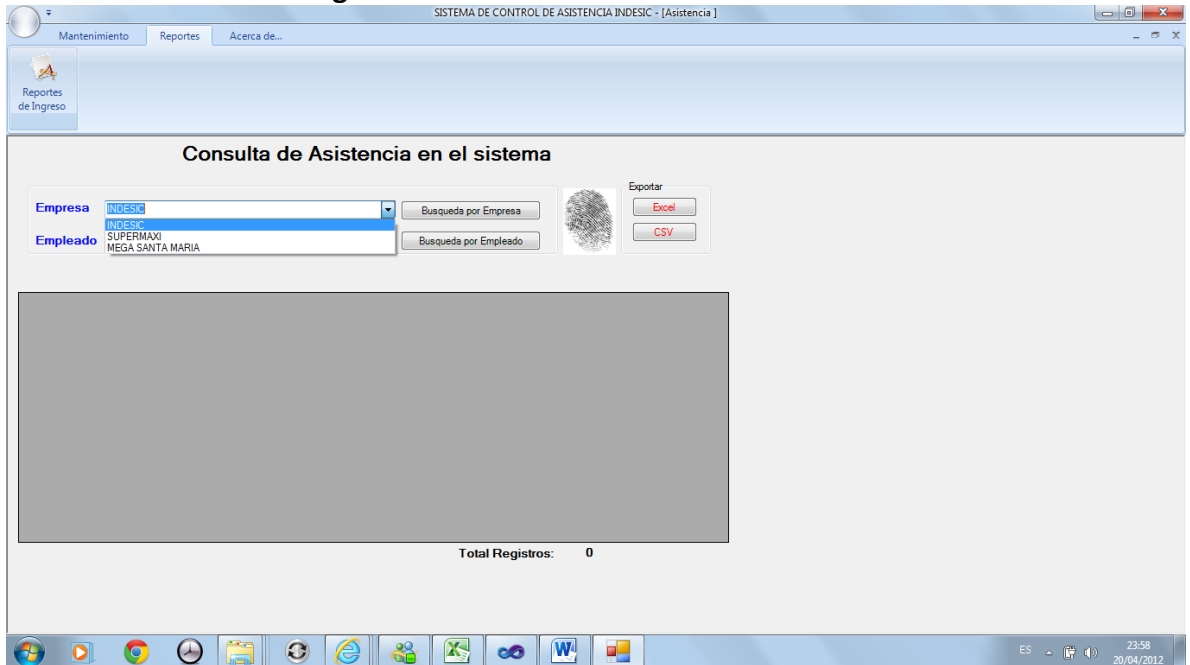




12. Ahora esperamos que el dispositivo genere un t mplate de la huella digital.



13. Ventana del fin de registro de la huella.



14. Para los reportes se debe seleccionar en el men  reportes. Seleccionamos el campo a consultar empresa y o empleado. Presionamos buscar y nos genera la siguiente pantalla.

**Consulta de Asistencia en el sistema**

Empresa: INDESIC      Busqueda por Empresa

Empleado: ANA LUCIA BASTIDAS MOROCHO      Busqueda por Empleado

RAZON_SOCIAL	CEDULA	NOMBRE_EMPLE	APELLIDO_EMPLE	DESCRIPCION_CA	NOMBRE_TIPO_R	FECHA_REGISTRO
INDESIC	1720203361	ANA LUCIA	BASTIDAS MOR...	DEVELOPER	ENTRADA	17/04/2012 18:08
INDESIC	1720203361	ANA LUCIA	BASTIDAS MOR...	DEVELOPER	SALIDA	17/04/2012 18:09
INDESIC	1720203361	ANA LUCIA	BASTIDAS MOR...	DEVELOPER	ENTRADA	20/04/2012 9:08
INDESIC	1720203361	ANA LUCIA	BASTIDAS MOR...	DEVELOPER	SALIDA	17/04/2012 18:09
INDESIC	1720203361	ANA LUCIA	BASTIDAS MOR...	DEVELOPER	ENTRADA	20/04/2012 9:00
INDESIC	1720203361	ANA LUCIA	BASTIDAS MOR...	DEVELOPER	SALIDA	19/04/2012 18:16
INDESIC	1720203361	ANA LUCIA	BASTIDAS MOR...	DEVELOPER	ENTRADA	17/04/2012 9:12
INDESIC	1720203361	ANA LUCIA	BASTIDAS MOR...	DEVELOPER	SALIDA	17/04/2012 18:06
INDESIC	1720203361	ANA LUCIA	BASTIDAS MOR...	DEVELOPER	ENTRADA	17/04/2012 9:13
INDESIC	1720203361	ANA LUCIA	BASTIDAS MOR...	DEVELOPER	SALIDA	17/04/2012 18:09
INDESIC	1720203361	ANA LUCIA	BASTIDAS MOR...	DEVELOPER	ENTRADA	17/04/2012 9:11
INDESIC	1720203361	ANA LUCIA	BASTIDAS MOR...	DEVELOPER	SALIDA	17/04/2012 18:10

Total Registros: 43

**15. Pantalla de visualización de los empleados.**

**Consulta de Asistencia en el sistema**

Empresa: INDESIC      Busqueda por Empresa

Empleado: ANA LUCIA BASTIDAS MOROCHO      Busqueda por Empleado

RAZON_SOCIAL	CEDULA	NOMBRE_EMPLE	APELLIDO_EMPLE	DESCRIPCION_CA	NOMBRE_TIPO_R	FECHA_REGISTRO
INDESIC	1720203361	ANA LUCIA	BASTIDAS MOR...	DEVELOPER	ENTRADA	17/04/2012 18:08
INDESIC	1720203361	ANA LUCIA	BASTIDAS MOR...	DEVELOPER	SALIDA	17/04/2012 18:09
INDESIC	1720203361	ANA LUCIA	BASTIDAS MOR...	DEVELOPER	ENTRADA	20/04/2012 9:08
INDESIC	1720203361	ANA LUCIA	BASTIDAS MOR...	DEVELOPER	SALIDA	17/04/2012 18:09
INDESIC	1720203361	ANA LUCIA	BASTIDAS MOR...	DEVELOPER	ENTRADA	20/04/2012 9:00
INDESIC	1720203361	ANA LUCIA	BASTIDAS MOR...	DEVELOPER	SALIDA	19/04/2012 18:16
INDESIC	1720203361	ANA LUCIA	BASTIDAS MOR...	DEVELOPER	ENTRADA	17/04/2012 9:12
INDESIC	1720203361	ANA LUCIA	BASTIDAS MOR...	DEVELOPER	SALIDA	17/04/2012 18:06
INDESIC	1720203361	ANA LUCIA	BASTIDAS MOR...	DEVELOPER	ENTRADA	17/04/2012 9:13
INDESIC	1720203361	ANA LUCIA	BASTIDAS MOR...	DEVELOPER	SALIDA	17/04/2012 18:09
INDESIC	1720203361	ANA LUCIA	BASTIDAS MOR...	DEVELOPER	ENTRADA	17/04/2012 9:11
INDESIC	1720203361	ANA LUCIA	BASTIDAS MOR...	DEVELOPER	SALIDA	17/04/2012 18:10

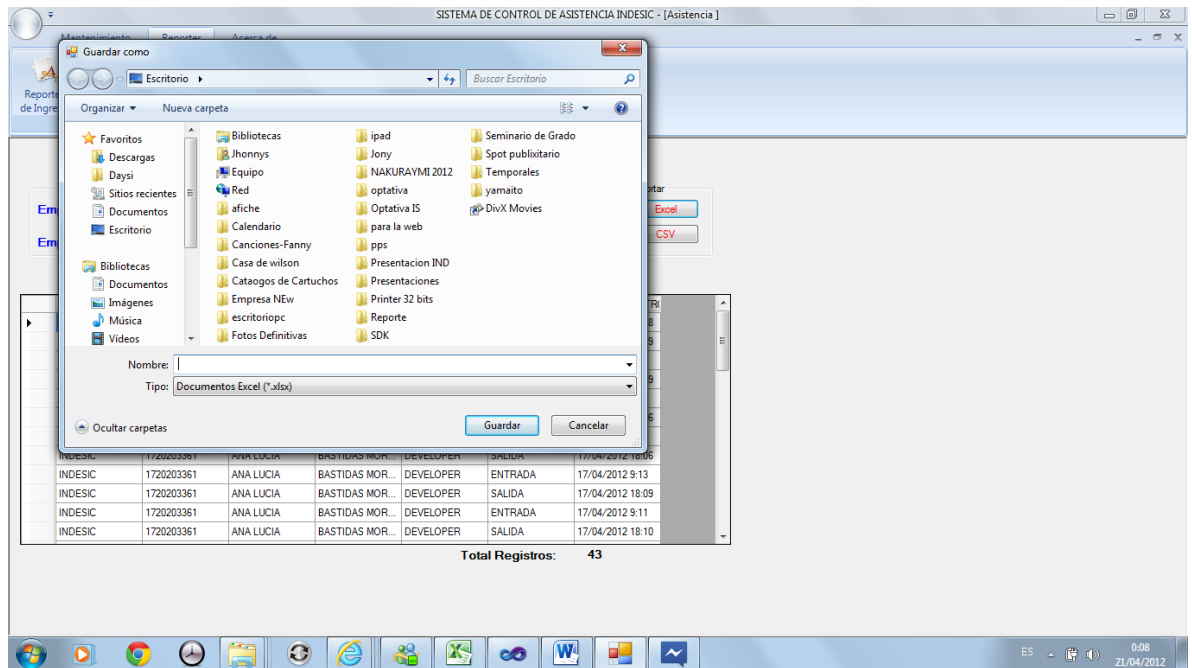
Total Registros: 43

Confirmación: Exportar a Excel (Si/No)

16. En esta pantalla procedemos a generar los documentos y archivos de exportación al sistema contable.

17. En el botón seleccionamos el tipo de archivo a generar.

18. Cuadro de diálogo aceptamos se visualizara la siguiente pantalla.



19. En esta pantalla procedemos a nombrar el archivo y el destino y listo.



## 7.2. Manual Técnico.

### //VALIDACIONES

```
8. using System;
9. using System.Collections.Generic;
10. using System.Linq;
11. using System.Text;
12. using System.Text.RegularExpressions;
13.
14. namespace Fingerprints.Entities
15. {
16.     public partial class CARGO
17.     {
18.
19.         partial void OnValidate(System.Data.Linq.ChangeAction action)
20.         {
21.             if (action != System.Data.Linq.ChangeAction.Delete)
22.             {
23.                 if (DESCRIPCION_CARGO == null || DESCRIPCION_CARGO.Trim() ==
String.Empty)
24.                 {
25.                     throw new Exception("\Nombre del cargo no puede estar
vacío\");
26.                 }
27.             }
28.
29.             if (Char.IsLower(DESCRIPCION_CARGO[0]))
30.             {
31.                 throw new Exception("\Nombre del Cargo debe iniciar con
Mayuscula\");
32.             }
33.         }
34.
35.     }
36.
37.     public partial class HORARIO
38.     {
39.         partial void OnValidate(System.Data.Linq.ChangeAction action)
40.         {
41.             if (action != System.Data.Linq.ChangeAction.Delete)
42.             {
43.                 if(DESCRIPCION_HORARIO==null ||
DESCRIPCION_HORARIO.Trim()==string.Empty)
44.                 {
45.                     throw new Exception("\La descripcion del horario no puede
estar vacío\");
46.                 }
47.                 if (Char.IsLower(DESCRIPCION_HORARIO[0]))
48.                 {
49.                     throw new Exception("\La descripcion del horario debe
iniciar con Mayuscula\");
50.                 }
51.             }
52.         }
53.     }
54.
55.     public partial class EMPLEADO
56.     {
```



```
57.
58.     partial void OnValidate(System.Data.Linq.ChangeAction action)
59.     {
60.         if (action != System.Data.Linq.ChangeAction.Delete)
61.         {
62.             if (NOMBRE_EMPLEADO == null || NOMBRE_EMPLEADO.Trim() ==
String.Empty)
63.             {
64.                 throw new Exception("\El nombre del empleado no puede
estar vacío\");
65.             }
66.             if (APELLIDO_EMPLEADO == null || APELLIDO_EMPLEADO.Trim() ==
String.Empty)
67.             {
68.                 throw new Exception("\El Apellido no puede estar vacío\");
69.             }
70.             if (CEDULA == null || CEDULA.Trim() == String.Empty)
71.             {
72.                 throw new Exception("\La cedula no puede estar vacío\");
73.             }
74.             if(!VerificaCedula(CEDULA))
75.             {
76.                 throw new Exception("\La cedula no es valida\");
77.             }
78.             if (DIRECCION == null || DIRECCION.Trim() == String.Empty)
79.             {
80.                 throw new Exception("\La direccion no puede estar
vacío\");
81.             }
82.             if (TELEFONO == null || TELEFONO.Trim() == String.Empty)
83.             {
84.                 throw new Exception("\El telefono no puede estar vacío\");
85.             }
86.             if (ID_CARGO == null )
87.             {
88.                 throw new Exception("\El empleado debe tener un cargo\");
89.             }
90.             if (ID_EMPRESA == null )
91.             {
92.                 throw new Exception("\El empleado debe pertenecer a una
empresa\");
93.             }
94.             if (SUELDO == null)
95.             {
96.                 throw new Exception("\El sueldo no puede estar
vacío\");
97.             }
98.         }
99.     }
100.
101.
102.
103.
104.
105.
106.
107.
108.
109.
110. }
```



```
111.
112.         NOMBRE_EMPLEADO = NOMBRE_EMPLEADO.ToUpper();
113.         //if (Char.IsLower(NOMBRE_EMPLEADO[0]))
114.         //{
115.         //    throw new Exception("\El Nombre del empleado debe iniciar
con Mayuscula\");
116.         //}
117.
118.         APELLIDO_EMPLEADO = APELLIDO_EMPLEADO.ToUpper();
119.         //if (Char.IsLower(APELLIDO_EMPLEADO[0]))
120.         //{
121.         //    throw new Exception("\El Apellido del empleado debe
iniciar con Mayuscula\");
122.         //}
123.         DIRECCION = DIRECCION.ToUpper();
124.         //if (Char.IsLower(DIRECCION[0]))
125.         //{
126.         //    throw new Exception("\La direccion iniciar con
Mayuscula\");
127.         //}
128.
129.     }
130.
131.
132.
133.     public bool TestEmailRegex(string emailAddress)
134.     {
135.
136.         const string patternStrict = @"^(([^<>()[\]\\\.,;:\s@\""]+)"
137.         +
138.         @"(\.([^<>()[\]\\\.,;:\s@\""]+)*|(\"\".+\"\"))@"
139.         + @"((\[[0-9]{1,3}\. [0-9]{1,3}\. [0-
9]{1,3}"
140.         + @"\. [0-9]{1,3}\)|((\[[a-zA-Z\-0-
9]+\.\.))+"
141.         + @"[a-zA-Z]{2,}))$";
142.
143.         var reStrict = new Regex(patternStrict);
144.         var isStrictMatch = reStrict.IsMatch(emailAddress);
145.         return isStrictMatch;
146.     }
147.
148.     public static bool VerificaCedula(string pCi)
149.     {
150.         var valor = false;
151.         if (pCi.Length >= 10)
152.         {
153.             var sum = 0;
154.             int dig;
155.             for (var i = 0; i < 9; i++)
156.             {
157.                 var aux = int.Parse(pCi.Substring(i, 1));
158.                 if ((i%2) == 0)
159.                     aux *= 2;
160.                 if (aux > 9)
161.                     aux -= 9;
162.                 sum += aux;
163.             }
164.             if ((sum%10) == 0)
165.                 dig = 0;
```



```
165.         else
166.             dig = 10 - (sum%10);
167.             if (dig == int.Parse(pCi.Substring(9, 1)))
168.                 valor = true;
169.         }
170.         return valor;
171.     }
172. }
173. }
174.
175. public partial class EMPRESA
176. {
177.     partial void OnValidate(System.Data.Linq.ChangeAction action)
178.     {
179.         if (action != System.Data.Linq.ChangeAction.Delete)
180.         {
181.             if (RAZON_SOCIAL == null || RAZON_SOCIAL.Trim() ==
182. String.Empty)
183.             {
184.                 throw new Exception("\"El nombre de la Razon social no
185. puede estar vacío\"");
186.             }
187.             if (RUC == null || RUC.Trim() == String.Empty)
188.             {
189.                 throw new Exception("\"El RUC no puede estar vacío\"");
190.             }
191.             if (TELEFONO == null || TELEFONO.Trim() == String.Empty)
192.             {
193.                 throw new Exception("\"El telefono no puede estar
194. vacío\"");
195.             }
196.             if (DIRECCION == null || DIRECCION.Trim() == String.Empty)
197.             {
198.                 throw new Exception("\"La direccion no puede estar
199. vacío\"");
200.             }
201.             if (EMAIL == null || EMAIL.Trim() == String.Empty)
202.             {
203.                 throw new Exception("\"La empresa debe tener un mail de
204. contacto, no puede estar vacío.\"");
205.             }
206.             if (!TestEmailRegex(EMAIL))
207.             {
208.                 throw new Exception("\"Ingrese correctamente la
209. direccion de mail.\"");
210.             }
211.             if (Char.IsLower(RAZON_SOCIAL[0]))
212.             {
213.                 throw new Exception("\"La Razon Social iniciar con
214. Mayuscula\"");
215.             }
216.             if (Char.IsLower(DIRECCION[0]))
217.             {
```



```
218.             throw new Exception("\"La direccion iniciar con
    Mayuscula.\"");
219.             }
220.         }
221.     }
222.
223.     public bool TestEmailRegex(string emailAddress)
224.     {
225.
226.         const string patternStrict = @"^(([^<>()[]\.\.,;\s@\""]+)"
227.         +
228.         @"(\.([^<>()[]\.\.,;\s@\""]+)*|(\"\".+\"\"))@"
229.         + @"((\[[0-9]{1,3}\. [0-9]{1,3}\. [0-
230.         9]{1,3}"
231.         + @"\. [0-9]{1,3}\])|((\[[a-zA-Z\-\0-
232.         9]+\.\.)*"
233.         + @"[a-zA-Z]{2,}))$";
234.
235.         var reStrict = new Regex(patternStrict);
236.         var isStrictMatch = reStrict.IsMatch(emailAddress);
237.         return isStrictMatch;
238.     }
239.
240.     public partial class USUARIO
241.     {
242.         partial void OnValidate(System.Data.Linq.ChangeAction action)
243.         {
244.             if (action != System.Data.Linq.ChangeAction.Delete)
245.             {
246.                 if(NOMBRE_USUARIO.Length<12)
247.                 {
248.                     throw new Exception("\"El tamaño minimo es de 12
    caracteres en el nombre de usuario\"");
249.                 }
250.                 if (PASSWORD.Length < 8)
251.                 {
252.                     throw new Exception("\"El tamaño minimo es de 8
    caracteres en la contraseña\"");
253.                 }
254.             }
255.         }
256.     }
257.
```

## // MANTENIMIENTO CARGO

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using Fingerprints.Entities;
using ProyectoFingerprints.Clases;
```



```
namespace ProyectoFingerprints.Mantenimiento
{
    public partial class Cargo : Form
    {
        private readonly FingerprintsDataContext ctx = new
FingerprintsDataContext();
        readonly Funciones _func = new Funciones();
        public Cargo()
        {
            InitializeComponent();
        }

        private void Cargo_Load(object sender, EventArgs e)
        {
            Refrescar();
        }

        private void Refrescar()
        {
            CARGOBindingSource.DataSource = ctx.CARGOs;

            var bloque = (from a in ctx.CARGOs
                        select a.DESCRIPCION_CARGO).Distinct();
            cmbBuscar.Items.Clear();
            foreach (var a in bloque)
            {
                cmbBuscar.Items.Add(a);
            }
            cmbBuscar.Text = String.Empty;
        }

        private void cARGODataGridView_UserDeletingRow(object sender,
DataGridViewRowCancelEventArgs e)
        {
            MessageBox.Show(@"Tecla Supr presionada, evite esta acción...", @"Mala
Práctica...", MessageBoxButtons.OK, MessageBoxIcon.Error);
            e.Cancel = true;
        }

        private void Cargo_FormClosing(object sender, FormClosingEventArgs e)
        {
            CARGOBindingNavigator.Refresh();
            var cs = ctx.GetChangeSet();
            var ins = cs.Inserts.Count();
            var upd = cs.Updates.Count();
            var del = cs.Deletes.Count();

            try
            {
                if (ins != 0 || upd != 0 || del != 0)
                {
                    var result = MessageBox.Show(string.Format("Han ocurrido los
siguientes cambios: \nNuevo(s): {0}\nModificado(s): {1}\nEliminado(s): {2}\nDesea
guardar los cambios?", ins, upd, del), "Guardar cambios",
MessageBoxButtons.YesNoCancel, MessageBoxIcon.Information);
                    switch (result)
                    {
                        case DialogResult.Yes:
                            ctx.SubmitChanges();
                    }
                }
            }
        }
    }
}
```



```
                break;
            case DialogResult.Cancel:
                e.Cancel = true;
                break;
        }
    }
}
catch (Exception ex)
{
    MessageBox.Show(string.Format("Error en actualizar los datos: {0}",
ex.Message));
    e.Cancel = true;
}
}

private void CARGOBindingNavigatorSaveItem_Click(object sender, EventArgs e)
{
    _func.GuardarDatos(ctx, CARGOBindingSource, this);
    Refrescar();
}

private void btnBuscar_Click(object sender, EventArgs e)
{
    var busqueda = from cargo in ctx.CARGOS
                    where
cargo.DESCRIPCION_CARGO.StartsWith(cmbBuscar.Text.Trim())
                    select cargo;

    CARGOBindingSource.DataSource = busqueda;
}

private void bindingNavigatorAddNewItem_Click(object sender, EventArgs e)
{
    DESCRIPCION_CARGOTextBox.Focus();
}
}
}
```

## //MANTENIMIENTO REGISTROS

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using Fingerprints.Entities;
using ProyectoFingerprints.Clases;

namespace ProyectoFingerprints.Mantenimiento
{
    public partial class ConsultaRegistros : Form
    {
        public ConsultaRegistros()
        {
```



```
        InitializeComponent();
    }

    private void ConsultaRegistros_Load(object sender, EventArgs e)
    {
        using (var ctx = new FingerprintsDataContext())
        {
            cmbEmpresa.DataSource = ctx.EMPRESAS;
            cmbEmpresa.DisplayMember = "RAZON_SOCIAL";
            cmbEmpresa.ValueMember = "ID_EMPRESA";
            cmbEmpresa_SelectedIndexChanged(null, null);
        }
    }

    private void cmbEmpresa_SelectedIndexChanged(object sender, EventArgs e)
    {
        if (string.IsNullOrEmpty(cmbEmpresa.SelectedValue.ToString())) return;
        short pEmpresa;
        if (!short.TryParse(cmbEmpresa.SelectedValue.ToString(), out pEmpresa))
return;
        using (var ctx = new FingerprintsDataContext())
        {
            cmbEmpleado.DataSource = ctx.usp_EmpleadosByEmpresa(pEmpresa);
            cmbEmpleado.DisplayMember = "NOMBRECOMPLETO";
            cmbEmpleado.ValueMember = "ID_EMPLEADO";
            cmbEmpleado.SelectedIndex = -1;
        }
    }

    private void btnEmpresa_Click(object sender, EventArgs e)
    {
        short pEmpresa;
        if (!short.TryParse(cmbEmpresa.SelectedValue.ToString(), out pEmpresa))
return;
        using (var ctx = new FingerprintsDataContext())
        {
            dgConsulta.DataSource = ctx.usp_AsistenciaByEmpresa(pEmpresa);
            lbltotalRegistros.Text = dgConsulta.Rows.Count.ToString();
        }
    }

    private void btnEmpleado_Click(object sender, EventArgs e)
    {
        short pEmpleado;
        if (!short.TryParse(cmbEmpleado.SelectedValue.ToString(), out
pEmpleado)) return;
        using (var ctx = new FingerprintsDataContext())
        {
            dgConsulta.DataSource = ctx.usp_AsistenciaByEmpleado(pEmpleado);
            lbltotalRegistros.Text = dgConsulta.Rows.Count.ToString();
        }
    }

    private void dgConsulta_CellDoubleClick(object sender,
DataGridViewCellEventArgs e)
    {
    }

    private void btnExportarCSV_Click(object sender, EventArgs e)
```



```
    {
        var fnc=new Funciones();
        sfdRegistros.InitialDirectory =
Environment.SpecialFolder.Desktop.ToString();
        sfdRegistros.DefaultExt = ".csv";
        sfdRegistros.Filter = @"Documentos CSV (*.csv)|*.csv ";
        if (MessageBox.Show(@"Exportar a CSV", @"Confirmación",
MessageBoxButtons.YesNo,
                        MessageBoxIcon.Question) !=
                DialogResult.Yes) return;
        sfdRegistros.FileName = string.Empty;
        if (sfdRegistros.ShowDialog() != DialogResult.OK) return;
        try
        {
            fnc.ExportarCSV(dgConsulta, sfdRegistros.FileName);
            System.Diagnostics.Process.Start(sfdRegistros.FileName);
        }
        catch (Exception er)
        {
            MessageBox.Show(er.Message);
        }
    }

private void btnExcel_Click(object sender, EventArgs e)
{
    sfdRegistros.InitialDirectory =
Environment.SpecialFolder.Desktop.ToString();
    sfdRegistros.DefaultExt = ".xlsx";
    sfdRegistros.Filter = @"Documentos Excel (*.xlsx)|*.xlsx ";
    if (MessageBox.Show(@"Exportar a Excel", @"Confirmación",
MessageBoxButtons.YesNo,
                        MessageBoxIcon.Question) !=
            DialogResult.Yes) return;
    sfdRegistros.FileName = string.Empty;

    if (sfdRegistros.ShowDialog() != DialogResult.OK) return;
    try
    {
        dgConsulta.ExportarExcel(sfdRegistros.FileName, "Consolidado");
        System.Diagnostics.Process.Start(sfdRegistros.FileName);
    }
    catch (Exception er)
    {
        MessageBox.Show(er.Message);
    }
}
}
```

## //MANTENIMIENTO EMPLEADOS

```
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Linq;
using System.Windows.Forms;
using Fingerprints.Entities;
```



```
using ProyectoFingerprints.Clases;
using SecuGen.SecuBSPPro.Windows;

namespace ProyectoFingerprints.Mantenimiento
{
    public partial class Empleados : Form
    {
        private SecuBSPMx m_SecuBSP;
        private bool m_DeviceOpened;

        public Empleados()
        {
            InitializeComponent();

            Refrescar();
            m_DeviceOpened = false;
            m_SecuBSP = new SecuBSPMx();
            GetDispositivos();
        }

        private readonly FingerprintsDataContext _ctx = new
        FingerprintsDataContext();
        private readonly Funciones _func = new Funciones();
        private void eMPLEADODataGridView_DataError(object sender,
        DataGridViewDataErrorEventArgs e)
        {
        }

        private void GetDispositivos()
        {
            DeviceIDCombo.Items.Clear();
            DeviceIDCombo.Items.Add("0x00ff (Auto Detect)");
            var err = m_SecuBSP.EnumerateDevice();
            if (err != BSPError.ERROR_NONE) return;
            for (var i = 0; i < m_SecuBSP.DeviceNum; i++)
            {
                var deviceId = m_SecuBSP.GetDeviceID(i);

                var deviceIdInfo = "0x" + MakeHexadecimal(deviceId, 4) + " ("
                    + m_SecuBSP.GetDeviceName(deviceId) + ","
                    + m_SecuBSP.GetDeviceInstanceNum(deviceId) + ")";

                DeviceIDCombo.Items.Add(deviceIdInfo);
            }
        }

        private void Empleados_FormClosing(object sender, FormClosingEventArgs e)
        {
            eMPLEADOBindingNavigator.Refresh();
            var cs = _ctx.GetChangeSet();

            var ins = cs.Inserts.Count();
            var upd = cs.Updates.Count();
            var del = cs.Deletes.Count();
        }
    }
}
```



```
try
{
    if (ins != 0 || upd != 0 || del != 0)
    {
        var result = MessageBox.Show(string.Format("Han ocurrido los
siguientes cambios: \nNuevo(s): {0}\nModificado(s): {1}\nEliminado(s): {2}\nDesea
guardar los cambios?", ins, upd, del), "Guardar cambios",
MessageBoxButtons.YesNoCancel, MessageBoxIcon.Information);
        switch (result)
        {
            case DialogResult.Yes:
                _ctx.SubmitChanges();
                break;
            case DialogResult.Cancel:
                e.Cancel = true;
                break;
        }
    }
}
catch (Exception ex)
{
    MessageBox.Show(string.Format("Error en actualizar los datos: {0}",
ex.Message));
    e.Cancel = true;
}

private void eMPLEADODataGridView_UserDeletingRow(object sender,
DataGridViewRowCancelEventArgs e)
{
    MessageBox.Show(@"Tecla Supr presionada, evite esta acción...", @"Mala
Práctica...", MessageBoxButtons.OK, MessageBoxIcon.Error);
    e.Cancel = true;
}

private void cEDULATextBox_KeyPress(object sender, KeyPressEventArgs e)
{
    if (e.KeyChar == (char)Keys.Back) return;
    int isNumber;
    e.Handled = !int.TryParse(e.KeyChar.ToString(), out isNumber);
}

private void bindingNavigatorAddNewItem_Click(object sender, EventArgs e)
{
    nOMBRE_EMPLEADOTextBox.Focus();
}

private void eMPLEADOBindingNavigatorSaveItem_Click(object sender, EventArgs
e)
{
    _func.GuardarDatos(_ctx, eMPLEADOBindingSource, this);
    Refrescar();
}

private void Refrescar()
{
    CARGOBindingSource.DataSource = _ctx.CARGOs.OrderBy(c =>
c.DESCRIPCION_CARGO);
}
```



```
        EMPRESABindingSource.DataSource = _ctx.EMPRESAs.OrderBy(ee =>
ee.RAZON_SOCIAL);
        HORARIOBindingSource.DataSource = _ctx.HORARIOS.OrderBy(hh =>
hh.DESCRIPCION_HORARIO);
        EMPLEADOBindingSource.DataSource = _ctx.EMPLEADOS.OrderBy(eml =>
eml.ID_EMPLEADO);

        var pNombre = (from a in _ctx.EMPLEADOS
                        select a.NOMBRE_EMPLEADO).Distinct();
        cmbNombre.Items.Clear();
        foreach (var a in pNombre)
        {
            cmbNombre.Items.Add(a);
        }
        cmbNombre.Text = String.Empty;

        var pApellido = (from a in _ctx.EMPLEADOS
                          select a.APELLIDO_EMPLEADO).Distinct();
        cmbApellido.Items.Clear();
        foreach (var a in pApellido)
        {
            cmbApellido.Items.Add(a);
        }
        cmbApellido.Text = String.Empty;
    }

    private void toolStripButton1_Click(object sender, EventArgs e)
    {
        EMPLEADOBindingSource.DataSource =
        _ctx.EMPLEADOS.Where(ee=>ee.NOMBRE_EMPLEADO.StartsWith(cmbNombre.Text.Trim()) &&
ee.APELLIDO_EMPLEADO.StartsWith(cmbApellido.Text.Trim())).OrderBy(eml =>
eml.ID_EMPLEADO);
    }

    private void sUELDOTextBox_KeyPress(object sender, KeyPressEventArgs e)
    {
        if (e.KeyChar == (char)Keys.Back) return;
        int isNumber;
        e.Handled = !int.TryParse(e.KeyChar.ToString(), out isNumber);
    }

    static string MakeHexadecimal(Int32 numbers, Int32 digit)
    {
        var destStr = "0000000000000000"; // digit can not exceed 16

        var str = Convert.ToString(numbers, 16);
        var len = str.Length;

        if (len > digit)
            destStr = "";
        else
            destStr = destStr.Substring(0, digit - len) + str;

        return destStr;
    }

    private void btnDispositivo_Click(object sender, EventArgs e)
    {
        if (m_DeviceOpened)
        {
```



```
        m_SecuBSP.CloseDevice();
        m_DeviceOpened = false;
    }

    // Get Selected device by User
    var selectedDevice = DeviceIDCombo.Text;
    selectedDevice = selectedDevice.Substring(0, 6);
    var deviceId = Convert.ToInt16(selectedDevice.Substring(0, 6), 16);

    m_SecuBSP.DeviceID = deviceId;

    BSPError err = m_SecuBSP.OpenDevice();
    //DisplaySecuBSPErrMsg("OpenDevice", err);
    if (err != BSPError.ERROR_NONE)
        return;

    m_DeviceOpened = true;

    if (err != BSPError.ERROR_NONE) return;
    lblMensajeDispositivo.Text = @"Dispositivo Abierto";
    lblMensajeDispositivo.ForeColor = Color.Red;

    //MessageBox.Show(@"Abierto");
}

private void btnDispositivos_Click(object sender, EventArgs e)
{
    m_DeviceOpened = false;
    m_SecuBSP = new SecuBSPMx();
    GetDispositivos();
}

private void Empleados_Load(object sender, EventArgs e)
{
}

private void btnCapturarHuellas_Click(object sender, EventArgs e)
{
    if (lblMensajeDispositivo.Text != @"Dispositivo Abierto")
    {
        MessageBox.Show(@"No este disponible el dispositivo",
@"Advertencia", MessageBoxButtons.OK,
                        MessageBoxIcon.Stop);
        return;
    }

    //Habilita el mensaje de bienvenida
    m_SecuBSP.EnrollWindowOption.WelcomePage = false;
    //Verificacion de existencia de huellas\
    short idUsuario;
    if(short.TryParse(id_EMPLEADOTextBox.Text,out idUsuario))
    {
        //Consulta si la huella para este usuario si existe????
        var uspConsulta =
        _ctx.usp_ConsultaRegistroHuella(idUsuario).SingleOrDefault();
    }
}
```



```
var figuraHuella = string.Empty;
if (uspConsulta != null)
    figuraHuella = uspConsulta.FIGURA_HUELLA;

const FIRFormat format = FIRFormat.STANDARD;
m_SecuBSP.SetFIRFormat(format);

BSPError err = figuraHuella != "" ?
m_SecuBSP.Enroll(id_EMPLEADOTextBox.Text, figuraHuella) :
m_SecuBSP.Enroll(id_EMPLEADOTextBox.Text);

if (err == BSPError.ERROR_NONE)
{
    _ctx.usp_RegistroHuella(short.Parse(id_EMPLEADOTextBox.Text),
m_SecuBSP.FIRTextData);
}
}

private void id_HORARIOComboBox_SelectedIndexChanged(object sender,
EventArgs e)
{
    if (id_HORARIOComboBox.SelectedValue == null) return;

    var pHorario=new List<HorarioEntradaSalida>();
    short pValor;
    if (!short.TryParse(id_HORARIOComboBox.SelectedValue.ToString(), out
pValor)) return;
    var pH = _ctx.HORARIOConsultas.Where(h => h.ID_HORARIO == pValor);
    if (pH.Count() <= 0) return;
    pHorario.AddRange(pH.Select(horario => horario.HORA_INGRESO_TRABAJO !=
null ? (horario.HORA_SALIDA_TRABAJO != null ? (horario.HORA_ENTRADA_ALMUERZO != null
? (horario.HORA_SALIDA_ALMUERZO != null ? new HorarioEntradaSalida { EntradaTrabajo
= horario.HORA_INGRESO_TRABAJO.Value.ToShortTimeString(), EntradaAlmuerzo =
horario.HORA_ENTRADA_ALMUERZO.Value.ToShortTimeString(), SalidaAlmuerzo =
horario.HORA_SALIDA_ALMUERZO.Value.ToShortTimeString(), SalidaTrabajo =
horario.HORA_SALIDA_TRABAJO.Value.ToShortTimeString() } : null) : null) : null) :
null));
    dgHorario.DataSource = pHorario;
    LblHorario.Text = @"HORARIO: " + id_HORARIOComboBox.Text;
}

private void cEDULATextBox_Validating(object sender,
System.ComponentModel.CancelEventArgs e)
{
    var fnc=new Funciones();

    if (!fnc.VerificaCedula(cEDULATextBox.Text))
    {
        // Cancel the event and select the text to be corrected by the user.
        e.Cancel = true;
        cEDULATextBox.Select(0, cEDULATextBox.Text.Length);

        // Set the ErrorProvider error with the text to display.
        errorProvider1.SetError(cEDULATextBox, "La cedula no es valida");
    }
    else
    {

```



```
        errorProvider1.SetError(cEDULATextBox, "");
    }
}
}
```

## //MANTENIMIENDO EMPRESA

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using Fingerprints.Entities;
using ProyectoFingerprints.Clases;
```

```
namespace ProyectoFingerprints.Mantenimiento
{
```

```
    public partial class Empresa : Form
    {
```

```
        private readonly FingerprintsDataContext _ctx = new
FingerprintsDataContext();
```

```
        readonly Funciones _func = new Funciones();
```

```
        public Empresa()
```

```
        {
            InitializeComponent();
        }
```

```
e) private void eMPRESABindingNavigatorSaveItem_Click(object sender, EventArgs
```

```
    {
        _func.GuardarDatos(_ctx, eMPRESABindingSource, this);
        Refrescar();
    }
```

```
private void Empresa_Load(object sender, EventArgs e)
{
    Refrescar();
}
```

```
private void Refrescar()
{
    eMPRESABindingSource.DataSource = _ctx.EMPRESAs;
```

```
    var bloque = (from a in _ctx.EMPRESAs
                  select a.RAZON_SOCIAL).Distinct();
```

```
    CmbBuscar.Items.Clear();
```

```
    foreach (var a in bloque)
```

```
    {
        CmbBuscar.Items.Add(a);
    }
```

```
    CmbBuscar.Text = String.Empty;
}
```

```
private void btnBuscar_Click(object sender, EventArgs e)
{
```

```
    var busqueda = from bloque in _ctx.EMPRESAs
```



```
        where
bloque.RAZON_SOCIAL.StartsWith(CmbBuscar.Text.Trim())
        select bloque;

        eMPRESABindingSource.DataSource = busqueda;
    }

    private void eMPRESADataGridView_UserDeletingRow(object sender,
DataGridViewRowCancelEventArgs e)
    {
        MessageBox.Show(@"Tecla Supr presionada, evite esta acción...", @"Mala
Práctica..", MessageBoxButtons.OK, MessageBoxIcon.Error);
        e.Cancel = true;
    }

    private void Empresa_FormClosing(object sender, FormClosingEventArgs e)
    {
        eMPRESABindingNavigator.Refresh();
        System.Data.Linq.ChangeSet cs = _ctx.GetChangeSet();

        int ins = cs.Inserts.Count();
        int upd = cs.Updates.Count();
        int del = cs.Deletes.Count();

        try
        {
            if (ins != 0 || upd != 0 || del != 0)
            {
                var result = MessageBox.Show(string.Format("Han ocurrido los
siguientes cambios: \nNuevo(s): {0}\nModificado(s): {1}\nEliminado(s): {2}\nDesea
guardar los cambios?", ins, upd, del), "Guardar cambios",
MessageBoxButtons.YesNoCancel, MessageBoxIcon.Information);
                switch (result)
                {
                    case DialogResult.Yes:
                        _ctx.SubmitChanges();
                        break;
                    case DialogResult.Cancel:
                        e.Cancel = true;
                        break;
                }
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(string.Format("Error en actualizar los datos: {0}",
ex.Message));
            e.Cancel = true;
        }
    }

    private void rUCTextBox_KeyPress(object sender, KeyPressEventArgs e)
    {
        if (e.KeyChar == (char)Keys.Back)return;
        int isNumber;
        e.Handled = !int.TryParse(e.KeyChar.ToString(), out isNumber);
    }

    private void TELEFONOTextBox_KeyPress(object sender, KeyPressEventArgs e)
    {
```



```
        if (e.KeyChar == (char)Keys.Back) return;
        int isNumber;
        e.Handled = !int.TryParse(e.KeyChar.ToString(), out isNumber);
    }

    private void bindingNavigatorAddNewItem_Click(object sender, EventArgs e)
    {
        rAZON_SOCIALTextBox.Focus();
    }
}
}
```

## //MANTENIMIENTO HORARIOS

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using Fingerprints.Entities;
using ProyectoFingerprints.Clases;

namespace ProyectoFingerprints.Mantenimiento
{
    public partial class Horarios : Form
    {
        public Horarios()
        {
            InitializeComponent();
        }

        private readonly FingerprintsDataContext ctx = new
FingerprintsDataContext();
        private readonly Funciones func = new Funciones();
        private void hORARIOBindingNavigatorSaveItem_Click(object sender, EventArgs
e)
        {
            func.GuardarDatos(ctx, hORARIOBindingSource, this);
            Refrescar();
        }

        private void Horarios_Load(object sender, EventArgs e)
        {
            Refrescar();
        }

        private void Refrescar()
        {
            hORARIOBindingSource.DataSource = ctx.HORARIOS;

            var horario = (from a in ctx.HORARIOS
                select a.DESCRIPCION_HORARIO).Distinct();
            cmbBuscar.Items.Clear();
        }
    }
}
```



```
        foreach (var a in horario)
        {
            cmbBuscar.Items.Add(a);
        }
        cmbBuscar.Text = String.Empty;
    }

    private void hORARIODataGridView_UserDeletingRow(object sender,
DataGridViewRowCancelEventArgs e)
    {
        MessageBox.Show(@"Tecla Supr presionada, evite esta acción...", @"Mala
Práctica..", MessageBoxButtons.OK, MessageBoxIcon.Error);
        e.Cancel = true;
    }

    private void btnBuscar_Click(object sender, EventArgs e)
    {

    }

    private void Horarios_FormClosing(object sender, FormClosingEventArgs e)
    {
        hORARIOBindingNavigator.Refresh();
        var cs = ctx.GetChangeSet();

        var ins = cs.Inserts.Count();
        var upd = cs.Updates.Count();
        var del = cs.Deletes.Count();

        try
        {
            if (ins != 0 || upd != 0 || del != 0)
            {
                var result = MessageBox.Show(string.Format("Han ocurrido los
siguientes cambios: \nNuevo(s): {0}\nModificado(s): {1}\nEliminado(s): {2}\nDesea
guardar los cambios?", ins, upd, del), "Guardar cambios",
MessageBoxButtons.YesNoCancel, MessageBoxIcon.Information);
                switch (result)
                {
                    case DialogResult.Yes:
                        ctx.SubmitChanges();
                        break;
                    case DialogResult.Cancel:
                        e.Cancel = true;
                        break;
                }
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(string.Format("Error en actualizar los datos: {0}",
ex.Message));
            e.Cancel = true;
        }
    }

    private void bindingNavigatorAddNewItem_Click(object sender, EventArgs e)
    {
        DESCRIPCION_HORARIOTextBox.Focus();
    }
}
```



```
        private void label1_Click(object sender, EventArgs e)
        {
            }
    }
}
```

## //MANTENIMIENTO HUELLA

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using Fingerprints.Entities;
using ProyectoFingerprints.Clases;

namespace ProyectoFingerprints.Mantenimiento
{
    public partial class Huella_Empleado : Form
    {
        public Huella_Empleado()
        {
            InitializeComponent();

            readonly Funciones func = new Funciones();
            private void Huella_Empleado_Load(object sender, EventArgs e)
            {
                //using (var ctx = new FingerprintsDataContext())
                //{
                //    cmbEmpresa.DataSource = ctx.EMPRESAs;
                //    cmbEmpresa.DisplayMember = "RAZON_SOCIAL";
                //    cmbEmpresa.ValueMember = "ID_EMPRESA";
                //    cmbEmpresa.SelectedIndex = -1;
                //}

                private void cmbEmpresa_SelectedIndexChanged(object sender, EventArgs e)
                {
                    //short pEmpresa;
                    //if (!short.TryParse(cmbEmpresa.SelectedValue.ToString(), out
                    pEmpresa)) return;
                    //using (var ctx = new FingerprintsDataContext())
                    //{
                    //    cmbEmpleado.DataSource = ctx.usp_EmpleadosByEmpresa(pEmpresa);
                    //    cmbEmpleado.DisplayMember = "NOMBRECOMPLETO";
                    //    cmbEmpleado.ValueMember = "ID_EMPLEADO";
                    //    cmbEmpleado.SelectedIndex = -1;
                    //}

                    private void Huella_Empleado_Fill_Panel_Paint(object sender, PaintEventArgs
                    e)
                    {

```



```
    }
  }
}

using System;
using System.Linq;
using System.Windows.Forms;
using Fingerprints.Entities;
using ProyectoFingerprints.Clases;

namespace ProyectoFingerprints.Mantenimiento
{
    public partial class UsuarioContenedor : Form
    {
        private FingerprintsDataContext _ctx = new FingerprintsDataContext();
        private Funciones _func = new Funciones();

        public UsuarioContenedor()
        {
            InitializeComponent();
        }

        private void uSUARIOBindingNavigatorSaveItem_Click(object sender, EventArgs e)
        {
            _func.GuardarDatos(_ctx, uSUARIOBindingSource, this);
        }

        private void UsuarioContenedor_FormClosing(object sender, FormClosingEventArgs e)
        {
            uSUARIOBindingNavigator.Refresh();
            var cs = _ctx.GetChangeSet();

            var ins = cs.Inserts.Count();
            var upd = cs.Updates.Count();
            var del = cs.Deletes.Count();

            try
            {
                if (ins != 0 || upd != 0 || del != 0)
                {
                    var result =
                        MessageBox.Show(
                            string.Format(
                                "Han ocurrido los siguientes cambios: \nNuevo(s):
                                {0}\nModificado(s): {1}\nEliminado(s): {2}\nDesea guardar los cambios?",
                                ins, upd, del), @"Guardar cambios",
                                MessageBoxButtons.YesNoCancel,
                                MessageBoxIcon.Information);
                    switch (result)
                    {
                        case DialogResult.Yes:
                            _ctx.SubmitChanges();
                            break;
                        case DialogResult.Cancel:
                            e.Cancel = true;
                            break;
                    }
                }
            }
        }
    }
}
```



```
    }
    }
    catch (Exception ex)
    {
        MessageBox.Show(string.Format("Error en actualizar los datos: {0}",
ex.Message));
        e.Cancel = true;
    }
}

private void UsuarioContenedor_Load(object sender, EventArgs e)
{
    EMPRESABindingSource.DataSource = _ctx.EMPRESAs.OrderBy(em =>
em.RAZON_SOCIAL);
    USUARIOBindingSource.DataSource = _ctx.USUARIOS.OrderBy(u =>
u.ID_USUARIO);
    //bindingNavigatorAddNewItem.Enabled = true;
}

private void bindingNavigatorAddNewItem_Click(object sender, EventArgs e)
{
    id_USUARIOTextBox.Focus();
}

}
}
```

```
using System;
using System.Collections.Generic;
using System.Data.Linq;
using System.IO;
using System.Linq;
using System.Runtime.InteropServices;
using System.Text;
using System.Windows.Forms;
using Microsoft.Office.Interop.Excel;

namespace ProyectoFingerprints.Clases
{
    public class Funciones
    {

        public bool VerificaCedula(string pCi)
        {
            var valor = false;
            if (pCi.Length >= 10)
            {
                var sum = 0;
                int dig;
                for (var i = 0; i < 9; i++)
                {
                    var aux = int.Parse(pCi.Substring(i, 1));
                    if ((i % 2) == 0)
                        aux *= 2;
                    if (aux > 9)

```



```
        aux -= 9;
        sum += aux;
    }
    if ((sum % 10) == 0)
        dig = 0;
    else
        dig = 10 - (sum % 10);
    if (dig == int.Parse(pCi.Substring(9, 1)))
        valor = true;
    }
    return valor;
}

public String[] LeerArchivoString(string nfile)
{
    string path = string.Format("{0}\\{1}", Environment.CurrentDirectory,
nfile);
    string[] linea;
    if (File.Exists(path))
    {
        linea = File.ReadAllLines(path);
    }
    else
    {
        linea = null;
        File.Create(path);
        MessageBox.Show(@"No existe el archivo de conexión: ", @"Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    return linea;
}

public void GuardarDatos(DataContext dc, BindingSource bs, Form fm)
{
    fm.Validate();
    bs.EndEdit();
    try
    {
        dc.SubmitChanges();

        MessageBox.Show(@"Datos Actualizados");
    }
    catch (DuplicateKeyException)
    {
        MessageBox.Show(@"Error en actualizar los datos: \"""Codigo Dia ya
existe\"""");
    }
    catch (Exception ex)
    {
        MessageBox.Show(string.Format(" Error en actualizar los datos: {0}",
ex.Message));
    }
}

public void Posicionar(BindingSource bs)
{
```



```
        if (bs.Position + 1 == bs.Count)
        {
            bs.Position -= 1;
            bs.Position += 1;
        }
        else if (bs.Position == 0)
        {
            bs.Position += 1;
            bs.Position -= 1;
        }
    }

    public bool IsNumeric(object valor)
    {
        double isItNumeric;
        var isNumber = Double.TryParse(Convert.ToString(valor),
System.Globalization.NumberStyles.Any,
System.Globalization.NumberFormatInfo.InvariantInfo, out isItNumeric);
        return isNumber;
    }

    public static string GetCodString(List<int> codigos)
    {
        var str = codigos.Aggregate(string.Empty, (current, i) => current +
string.Format("{0},", i.ToString()));
        if (str.Length > 0)
        {
            str = str.Remove(str.Length - 1, 1);
        }
        return str;
    }

    public void ExportarCSV(DataGridView dat, string medicion)
    {
        const string pCorte = " ";
        var strExport =
dat.Columns.Cast<DataGridViewColumn>().Aggregate(string.Empty, (current, dc) =>
current + (dc.Name + pCorte));

        strExport = strExport.Substring(0, strExport.Length - 3) +
Environment.NewLine;

        foreach (DataGridViewRow dr in dat.Rows)
        {
            strExport = dr.Cells.Cast<DataGridViewCell>().Where(dc => dc.Value
!= null).Aggregate(strExport, (current, dc) => current + (dc.Value + pCorte));
            strExport += Environment.NewLine.ToString();
        }
        strExport = strExport.Substring(0, strExport.Length - 3) +
Environment.NewLine;
        StreamWriter tw = new StreamWriter(medicion);
        tw.Write(strExport);
        tw.Close();
    }
}

using System;
using System.Collections.Generic;
```



```
using System.Drawing;
using System.IO;
using System.Linq;
using System.Runtime.InteropServices;
using System.Text;
using System.Windows.Forms;
using Microsoft.Office.Interop.Excel;
using Application = Microsoft.Office.Interop.Excel.Application;

namespace ProyectoFingerprints.Clases
{
    public static class FuncExportar
    {
        public static void ExportarExcel(this DataGridView dataGridView, string
fileName, string worksheetName)
        {
            // todos los archivos declarados son con la ext xlsx
            if (dataGridView == null)
            {
                throw new ArgumentNullException("dataGridView");
            }

            var excel = new Application();
            Workbook book = null;
            Worksheet sheet = null;
            try
            {
                book = File.Exists(fileName) ? excel.Workbooks.Open(fileName) :
excel.Workbooks.Add();

                sheet = book.Sheets.Add();
                sheet.Name = worksheetName;
                for (var i = 0; i < dataGridView.Columns.Count; i++)
                {
                    sheet.Cells[1, i + 1] = dataGridView.Columns[i].HeaderText;
                    sheet.Cells[1, i + 1].Font.Bold = true;
                    sheet.Cells[1, i + 1].Font.Color = Color.Red;
                }
                for (var row = 0; row < dataGridView.Rows.Count; row++)
                {
                    for (int column = 0; column < dataGridView.Columns.Count;
column++)
                    {
                        sheet.Cells[row + 2, column + 1] =
dataGridView.Rows[row].Cells[column].Value;
                    }
                }

                book.SaveAs(fileName);
                book.Close();
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
            finally
            {
                excel.Workbooks.Close();
                excel.Quit();
                if (sheet != null) Marshal.ReleaseComObject(sheet);
            }
        }
    }
}
```



```
        if (book != null) Marshal.ReleaseComObject(book);
        Marshal.ReleaseComObject(excel);
    }
}
}

namespace Fingerprints.Entities
{
    using System.Data.Linq;
    using System.Data.Linq.Mapping;
    using System.Data;
    using System.Collections.Generic;
    using System.Reflection;
    using System.Linq;
    using System.Linq.Expressions;
    using System.ComponentModel;
    using System;

    [global::System.Data.Linq.Mapping.DatabaseAttribute(Name="Fingerprints")]
    public partial class FingerprintsDataContext : System.Data.Linq.DataContext
    {
        private static System.Data.Linq.Mapping.MappingSource mappingSource =
        new AttributeMappingSource();

        #region Definiciones de métodos de extensibilidad
        partial void OnCreated();
        partial void InsertHORARIOConsulta(HORARIOConsulta instance);
        partial void UpdateHORARIOConsulta(HORARIOConsulta instance);
        partial void DeleteHORARIOConsulta(HORARIOConsulta instance);
        partial void InsertCARGO(CARGO instance);
        partial void UpdateCARGO(CARGO instance);
        partial void DeleteCARGO(CARGO instance);
        partial void InsertEMPLEADO(EMPLEADO instance);
        partial void UpdateEMPLEADO(EMPLEADO instance);
        partial void DeleteEMPLEADO(EMPLEADO instance);
        partial void InsertEMPRESA(EMPRESA instance);
        partial void UpdateEMPRESA(EMPRESA instance);
        partial void DeleteEMPRESA(EMPRESA instance);
        partial void InsertHORARIO(HORARIO instance);
        partial void UpdateHORARIO(HORARIO instance);
        partial void DeleteHORARIO(HORARIO instance);
        partial void InsertHUELLA_EMPLADO(HUELLA_EMPLADO instance);
        partial void UpdateHUELLA_EMPLADO(HUELLA_EMPLADO instance);
        partial void DeleteHUELLA_EMPLADO(HUELLA_EMPLADO instance);
        partial void InsertREGISTRO(REGISTRO instance);
        partial void UpdateREGISTRO(REGISTRO instance);
        partial void DeleteREGISTRO(REGISTRO instance);
        partial void InsertTIPO_REGISTRO(TIPO_REGISTRO instance);
        partial void UpdateTIPO_REGISTRO(TIPO_REGISTRO instance);
        partial void DeleteTIPO_REGISTRO(TIPO_REGISTRO instance);
        partial void InsertUSUARIO(USUARIO instance);
        partial void UpdateUSUARIO(USUARIO instance);
        partial void DeleteUSUARIO(USUARIO instance);
        #endregion

        public FingerprintsDataContext() :
```



```
        base(global::Fingerprints.Entities.Properties.Settings.Default.FingerprintsCo
nnectionString, mappingSource)
    {
        OnCreated();
    }

    public FingerprintsDataContext(string connection) :
        base(connection, mappingSource)
    {
        OnCreated();
    }

    public FingerprintsDataContext(System.Data.IDbConnection connection) :
        base(connection, mappingSource)
    {
        OnCreated();
    }

    public FingerprintsDataContext(string connection,
System.Data.Linq.Mapping.MappingSource mappingSource) :
        base(connection, mappingSource)
    {
        OnCreated();
    }

    public FingerprintsDataContext(System.Data.IDbConnection connection,
System.Data.Linq.Mapping.MappingSource mappingSource) :
        base(connection, mappingSource)
    {
        OnCreated();
    }

    public System.Data.Linq.Table<HORARIOConsulta> HORARIOConsultas
    {
        get
        {
            return this.GetTable<HORARIOConsulta>();
        }
    }

    public System.Data.Linq.Table<CARGO> CARGOs
    {
        get
        {
            return this.GetTable<CARGO>();
        }
    }

    public System.Data.Linq.Table<USUARIO> USUARIOS
    {
        get
        {
            return this.GetTable<USUARIO>();
        }
    }

    public System.Data.Linq.Table<EMPLEADO> EMPLEADOs
    {
        get
```



```
        {
            return this.GetTable<EMPLEADO>();
        }
    }

    public System.Data.Linq.Table<EMPRESA> EMPRESAS
    {
        get
        {
            return this.GetTable<EMPRESA>();
        }
    }

    public System.Data.Linq.Table<HORARIO> HORARIOS
    {
        get
        {
            return this.GetTable<HORARIO>();
        }
    }

    public System.Data.Linq.Table<HUELLA_EMPLEADO> HUELLA_EMPLEADOS
    {
        get
        {
            return this.GetTable<HUELLA_EMPLEADO>();
        }
    }

    public System.Data.Linq.Table<REGISTRO> REGISTROS
    {
        get
        {
            return this.GetTable<REGISTRO>();
        }
    }

    public System.Data.Linq.Table<TIPO_REGISTRO> TIPO_REGISTROS
    {
        get
        {
            return this.GetTable<TIPO_REGISTRO>();
        }
    }

    [global::System.Data.Linq.Mapping.FunctionAttribute(Name="dbo.usp_RegistroHue
    lla")]
    public int
    usp_RegistroHuella([global::System.Data.Linq.Mapping.ParameterAttribute(DbType="SmallInt")] System.Nullable<short> pIdEmpleado,
    [global::System.Data.Linq.Mapping.ParameterAttribute(DbType="VarChar(8000)")] string pHuella)
    {
        IExecuteResult result = this.ExecuteMethodCall(this,
        ((MethodInfo)(MethodInfo.GetCurrentMethod)), pIdEmpleado, pHuella);
        return ((int)(result.ReturnValue));
    }
}
```



```
[global::System.Data.Linq.Mapping.FunctionAttribute(Name="dbo.usp_AssistenciaByEmpresa")]
public ISingleResult<usp_AssistenciaByEmpresaResult>
usp_AssistenciaByEmpresa([global::System.Data.Linq.Mapping.ParameterAttribute(DbType="SmallInt")] System.Nullable<short> pEmpresa)
{
    IExecuteResult result = this.ExecuteMethodCall(this,
((MethodInfo)(MethodInfo.GetCurrentMethod)), pEmpresa);
    return
((ISingleResult<usp_AssistenciaByEmpresaResult>)(result.ReturnValue));
}

[global::System.Data.Linq.Mapping.FunctionAttribute(Name="dbo.usp_ConsultaRegistroHuella")]
public ISingleResult<usp_ConsultaRegistroHuellaResult>
usp_ConsultaRegistroHuella([global::System.Data.Linq.Mapping.ParameterAttribute(DbType="SmallInt")] System.Nullable<short> pIdEmpleado)
{
    IExecuteResult result = this.ExecuteMethodCall(this,
((MethodInfo)(MethodInfo.GetCurrentMethod)), pIdEmpleado);
    return
((ISingleResult<usp_ConsultaRegistroHuellaResult>)(result.ReturnValue));
}

[global::System.Data.Linq.Mapping.FunctionAttribute(Name="dbo.usp_EmpleadosByEmpresa")]
public ISingleResult<usp_EmpleadosByEmpresaResult>
usp_EmpleadosByEmpresa([global::System.Data.Linq.Mapping.ParameterAttribute(DbType="SmallInt")] System.Nullable<short> pEmpresa)
{
    IExecuteResult result = this.ExecuteMethodCall(this,
((MethodInfo)(MethodInfo.GetCurrentMethod)), pEmpresa);
    return
((ISingleResult<usp_EmpleadosByEmpresaResult>)(result.ReturnValue));
}

[global::System.Data.Linq.Mapping.FunctionAttribute(Name="dbo.usp_IdEmpleadoByHuella")]
public ISingleResult<usp_IdEmpleadoByHuellaResult>
usp_IdEmpleadoByHuella([global::System.Data.Linq.Mapping.ParameterAttribute(DbType="VarChar(500)")] string pHuella)
{
    IExecuteResult result = this.ExecuteMethodCall(this,
((MethodInfo)(MethodInfo.GetCurrentMethod)), pHuella);
    return
((ISingleResult<usp_IdEmpleadoByHuellaResult>)(result.ReturnValue));
}

[global::System.Data.Linq.Mapping.FunctionAttribute(Name="dbo.usp_ListadoHuellas")]
public ISingleResult<usp_ListadoHuellasResult> usp_ListadoHuellas()
{
    IExecuteResult result = this.ExecuteMethodCall(this,
((MethodInfo)(MethodInfo.GetCurrentMethod)));
}
```



```
        return
        ((ISingleResult<usp_ListadoHuellasResult>)(result.ReturnValue));
    }

    [global::System.Data.Linq.Mapping.FunctionAttribute(Name="dbo.usp_RegistroAsi
stencia")]
    public ISingleResult<usp_RegistroAsistenciaResult>
usp_RegistroAsistencia([global::System.Data.Linq.Mapping.ParameterAttribute(DbType="
SmallInt")] System.Nullable<short> pUsuario,
[global::System.Data.Linq.Mapping.ParameterAttribute(DbType="TinyInt")]
System.Nullable<byte> pTipoRegistro)
    {
        IExecuteResult result = this.ExecuteMethodCall(this,
((MethodInfo)(MethodInfo.GetCurrentMethod)), pUsuario, pTipoRegistro);
        return
        ((ISingleResult<usp_RegistroAsistenciaResult>)(result.ReturnValue));
    }

    [global::System.Data.Linq.Mapping.FunctionAttribute(Name="dbo.usp_AsistenciaB
yEmpleado")]
    public ISingleResult<usp_AsistenciaByEmpleadoResult>
usp_AsistenciaByEmpleado([global::System.Data.Linq.Mapping.ParameterAttribute(DbType
="SmallInt")] System.Nullable<short> pEmpleado)
    {
        IExecuteResult result = this.ExecuteMethodCall(this,
((MethodInfo)(MethodInfo.GetCurrentMethod)), pEmpleado);
        return
        ((ISingleResult<usp_AsistenciaByEmpleadoResult>)(result.ReturnValue));
    }
}

[global::System.Data.Linq.Mapping.TableAttribute(Name="dbo.HORARIO")]
public partial class HORARIOConsulta : INotifyPropertyChanging,
INotifyPropertyChanged
{
    private static PropertyChangingEventArgs emptyChangingEventArgs = new
PropertyChangingEventArgs(String.Empty);

    private short _ID_HORARIO;

    private string _DESCRIPCION_HORARIO;

    private System.Nullable<System.DateTime> _HORA_INGRESO_TRABAJO;

    private System.Nullable<System.DateTime> _HORA_ENTRADA_ALMUERZO;

    private System.Nullable<System.DateTime> _HORA_SALIDA_ALMUERZO;

    private System.Nullable<System.DateTime> _HORA_SALIDA_TRABAJO;

    private System.Nullable<byte> _TOTAL_HORAS_MES;

    private System.Nullable<bool> _ESTADO;

    private System.Nullable<System.DateTime> _FECHA_CREACION;
#region Definiciones de métodos de extensibilidad
partial void OnLoaded();
```



```
partial void OnValidate(System.Data.Linq.ChangeAction action);
partial void OnCreated();
partial void OnID_HORARIOChanging(short value);
partial void OnID_HORARIOChanged();
partial void OnDESCRIPCION_HORARIOChanging(string value);
partial void OnDESCRIPCION_HORARIOChanged();
partial void OnHORA_INGRESO_TRABAJOChanging(System.Nullable<System.DateTime>
value);
partial void OnHORA_INGRESO_TRABAJOChanged();
partial void OnHORA_ENTRADA_ALMUERZOChanging(System.Nullable<System.DateTime>
value);
partial void OnHORA_ENTRADA_ALMUERZOChanged();
partial void OnHORA_SALIDA_ALMUERZOChanging(System.Nullable<System.DateTime>
value);
partial void OnHORA_SALIDA_ALMUERZOChanged();
partial void OnHORA_SALIDA_TRABAJOChanging(System.Nullable<System.DateTime>
value);
partial void OnHORA_SALIDA_TRABAJOChanged();
partial void OnTOTAL_HORAS_MESChanging(System.Nullable<byte> value);
partial void OnTOTAL_HORAS_MESChanged();
partial void OnESTADOChanging(System.Nullable<bool> value);
partial void OnESTADOChanged();
partial void OnFECHA_CREACIONChanging(System.Nullable<System.DateTime> value);
partial void OnFECHA_CREACIONChanged();
#endregion

public HORARIOConsulta()
{
    OnCreated();
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_ID_HORARIO",
AutoSync=AutoSync.OnInsert, DbType="SmallInt NOT NULL IDENTITY", IsPrimaryKey=true,
IsDbGenerated=true)]
public short ID_HORARIO
{
    get
    {
        return this._ID_HORARIO;
    }
    set
    {
        if ((this._ID_HORARIO != value))
        {
            this.OnID_HORARIOChanging(value);
            this.SendPropertyChanging();
            this._ID_HORARIO = value;
            this.SendPropertyChanged("ID_HORARIO");
            this.OnID_HORARIOChanged();
        }
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_DESCRIPCION_HORAR
IO", DbType="VarChar(50)")]
public string DESCRIPCION_HORARIO
{
    get
```



```
        {
            return this._DESCRIPCION_HORARIO;
        }
        set
        {
            if ((this._DESCRIPCION_HORARIO != value))
            {
                this.OnDESCRIPCION_HORARIOChanging(value);
                this.SendPropertyChanging();
                this._DESCRIPCION_HORARIO = value;
                this.SendPropertyChanged("DESCRIPCION_HORARIO");
                this.OnDESCRIPCION_HORARIOChanged();
            }
        }
    }

    [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_HORA_INGRESO_TRABAJO", DbType="Time")]
    public System.Nullable<System.DateTime> HORA_INGRESO_TRABAJO
    {
        get
        {
            return this._HORA_INGRESO_TRABAJO;
        }
        set
        {
            if ((this._HORA_INGRESO_TRABAJO != value))
            {
                this.OnHORA_INGRESO_TRABAJOChanging(value);
                this.SendPropertyChanging();
                this._HORA_INGRESO_TRABAJO = value;
                this.SendPropertyChanged("HORA_INGRESO_TRABAJO");
                this.OnHORA_INGRESO_TRABAJOChanged();
            }
        }
    }

    [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_HORA_ENTRADA_ALMUERZO", DbType="Time")]
    public System.Nullable<System.DateTime> HORA_ENTRADA_ALMUERZO
    {
        get
        {
            return this._HORA_ENTRADA_ALMUERZO;
        }
        set
        {
            if ((this._HORA_ENTRADA_ALMUERZO != value))
            {
                this.OnHORA_ENTRADA_ALMUERZOChanging(value);
                this.SendPropertyChanging();
                this._HORA_ENTRADA_ALMUERZO = value;
                this.SendPropertyChanged("HORA_ENTRADA_ALMUERZO");
                this.OnHORA_ENTRADA_ALMUERZOChanged();
            }
        }
    }
}
```



```
[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_HORA_SALIDA_ALMUERZO", DbType="Time")]
public System.Nullable<System.DateTime> HORA_SALIDA_ALMUERZO
{
    get
    {
        return this._HORA_SALIDA_ALMUERZO;
    }
    set
    {
        if ((this._HORA_SALIDA_ALMUERZO != value))
        {
            this.OnHORA_SALIDA_ALMUERZOChanging(value);
            this.SendPropertyChanging();
            this._HORA_SALIDA_ALMUERZO = value;
            this.SendPropertyChanged("HORA_SALIDA_ALMUERZO");
            this.OnHORA_SALIDA_ALMUERZOChanged();
        }
    }
}
```

```
[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_HORA_SALIDA_TRABAJO", DbType="Time")]
public System.Nullable<System.DateTime> HORA_SALIDA_TRABAJO
{
    get
    {
        return this._HORA_SALIDA_TRABAJO;
    }
    set
    {
        if ((this._HORA_SALIDA_TRABAJO != value))
        {
            this.OnHORA_SALIDA_TRABAJOChanging(value);
            this.SendPropertyChanging();
            this._HORA_SALIDA_TRABAJO = value;
            this.SendPropertyChanged("HORA_SALIDA_TRABAJO");
            this.OnHORA_SALIDA_TRABAJOChanged();
        }
    }
}
```

```
[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_TOTAL_HORAS_MES", DbType="TinyInt")]
public System.Nullable<byte> TOTAL_HORAS_MES
{
    get
    {
        return this._TOTAL_HORAS_MES;
    }
    set
    {
        if ((this._TOTAL_HORAS_MES != value))
        {
            this.OnTOTAL_HORAS_MESChanging(value);
            this.SendPropertyChanging();
            this._TOTAL_HORAS_MES = value;
        }
    }
}
```



```
        this.SendPropertyChanged("TOTAL_HORAS_MES");
        this.OnTOTAL_HORAS_MESChanged();
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_ESTADO",
DbType="Bit")]
public System.Nullable<bool> ESTADO
{
    get
    {
        return this._ESTADO;
    }
    set
    {
        if ((this._ESTADO != value))
        {
            this.OnESTADOChanging(value);
            this.SendPropertyChanging();
            this._ESTADO = value;
            this.SendPropertyChanged("ESTADO");
            this.OnESTADOChanged();
        }
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_FECHA_CREACION",
DbType="DateTime")]
public System.Nullable<System.DateTime> FECHA_CREACION
{
    get
    {
        return this._FECHA_CREACION;
    }
    set
    {
        if ((this._FECHA_CREACION != value))
        {
            this.OnFECHA_CREACIONChanging(value);
            this.SendPropertyChanging();
            this._FECHA_CREACION = value;
            this.SendPropertyChanged("FECHA_CREACION");
            this.OnFECHA_CREACIONChanged();
        }
    }
}

public event PropertyChangingEventHandler PropertyChanging;
public event PropertyChangedEventHandler PropertyChanged;

protected virtual void SendPropertyChanging()
{
    if ((this.PropertyChanging != null))
    {
        this.PropertyChanging(this, emptyChangingEventArgs);
    }
}
```



```
protected virtual void SendPropertyChanged(String propertyName)
{
    if ((this.PropertyChanged != null))
    {
        this.PropertyChanged(this, new
PropertyChangeEventArgs(propertyName));
    }
}

[global::System.Data.Linq.Mapping.TableAttribute(Name="dbo.CARGO")]
public partial class CARGO : INotifyPropertyChanging, INotifyPropertyChanged
{
    private static PropertyChangingEventArgs emptyChangingEventArgs = new
PropertyChangingEventArgs(String.Empty);

    private short _ID_CARGO;

    private string _DESCRIPCION_CARGO;

    private System.Nullable<System.DateTime> _FECHA_CREACION;

    private EntitySet<EMPLEADO> _EMPLEADO;

    #region Definiciones de métodos de extensibilidad
    partial void OnLoaded();
    partial void OnValidate(System.Data.Linq.ChangeAction action);
    partial void OnCreated();
    partial void OnID_CARGOChanging(short value);
    partial void OnID_CARGOChanged();
    partial void OnDESCRIPCION_CARGOChanging(string value);
    partial void OnDESCRIPCION_CARGOChanged();
    partial void OnFECHA_CREACIONChanging(System.Nullable<System.DateTime> value);
    partial void OnFECHA_CREACIONChanged();
    #endregion

    public CARGO()
    {
        this._EMPLEADO = new EntitySet<EMPLEADO>(new
Action<EMPLEADO>(this.attach_EMPLEADO), new Action<EMPLEADO>(this.detach_EMPLEADO));
        OnCreated();
    }

    [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_ID_CARGO",
AutoSync=AutoSync.OnInsert, DbType="SmallInt NOT NULL IDENTITY", IsPrimaryKey=true,
IsDbGenerated=true)]
    public short ID_CARGO
    {
        get
        {
            return this._ID_CARGO;
        }
        set
        {
            if ((this._ID_CARGO != value))
            {
                this.OnID_CARGOChanging(value);
                this.SendPropertyChanging();
            }
        }
    }
}
```



```
        this._ID_CARGO = value;
        this.SendPropertyChanged("ID_CARGO");
        this.OnID_CARGOChanged();
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_DESCRIPCION_CARGO",
DbType="VarChar(50)")]
public string DESCRIPCION_CARGO
{
    get
    {
        return this._DESCRIPCION_CARGO;
    }
    set
    {
        if ((this._DESCRIPCION_CARGO != value))
        {
            this.OnDESCRIPCION_CARGOChanging(value);
            this.SendPropertyChanging();
            this._DESCRIPCION_CARGO = value;
            this.SendPropertyChanged("DESCRIPCION_CARGO");
            this.OnDESCRIPCION_CARGOChanged();
        }
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_FECHA_CREACION",
DbType="DateTime")]
public System.Nullable<System.DateTime> FECHA_CREACION
{
    get
    {
        return this._FECHA_CREACION;
    }
    set
    {
        if ((this._FECHA_CREACION != value))
        {
            this.OnFECHA_CREACIONChanging(value);
            this.SendPropertyChanging();
            this._FECHA_CREACION = value;
            this.SendPropertyChanged("FECHA_CREACION");
            this.OnFECHA_CREACIONChanged();
        }
    }
}

[global::System.Data.Linq.Mapping.AssociationAttribute(Name="CARGO_EMPLEADO",
Storage="_EMPLEADO", ThisKey="ID_CARGO", OtherKey="ID_CARGO")]
public EntitySet<EMPLEADO> EMPLADO
{
    get
    {
```



```
        return this._EMPLEADO;
    }
    set
    {
        this._EMPLEADO.Assign(value);
    }
}

public event PropertyChangingEventHandler PropertyChanging;

public event PropertyChangedEventHandler PropertyChanged;

protected virtual void SendPropertyChanging()
{
    if ((this.PropertyChanging != null))
    {
        this.PropertyChanging(this, emptyChangingEventArgs);
    }
}

protected virtual void SendPropertyChanged(String propertyName)
{
    if ((this.PropertyChanged != null))
    {
        this.PropertyChanged(this, new
PropertyChangedEventArgs(propertyName));
    }
}

private void attach_EMPLADO(EMPLADO entity)
{
    this.SendPropertyChanging();
    entity.CARGO = this;
}

private void detach_EMPLADO(EMPLADO entity)
{
    this.SendPropertyChanging();
    entity.CARGO = null;
}
}

[global::System.Data.Linq.Mapping.TableAttribute(Name="dbo.EMPLADO")]
public partial class EMPLADO : INotifyPropertyChanging,
INotifyPropertyChanged
{
    private static PropertyChangingEventArgs emptyChangingEventArgs = new
PropertyChangingEventArgs(String.Empty);

    private short _ID_EMPLADO;

    private string _NOMBRE_EMPLADO;

    private string _APELLIDO_EMPLADO;

    private string _CEDULA;

    private string _DIRECCION;
```



```
private string _TELEFONO;

private System.Nullable<short> _ID_CARGO;

private System.Nullable<short> _ID_EMPRESA;

private System.Nullable<short> _ID_HORARIO;

private System.Nullable<decimal> _SUELDO;

private System.Nullable<bool> _ESTADO;

private System.Nullable<System.DateTime> _FECHA_CREACION;

private EntitySet<HUELLA_EMPLEADO> _HUELLA_EMPLEADO;

private EntitySet<REGISTRO> _REGISTRO;

private EntityRef<CARGO> _CARGO;

private EntityRef<EMPRESA> _EMPRESA;

private EntityRef<HORARIO> _HORARIO;

#region Definiciones de métodos de extensibilidad
partial void OnLoaded();
partial void OnValidate(System.Data.Linq.ChangeAction action);
partial void OnCreated();
partial void OnID_EMPLEADOChanging(short value);
partial void OnID_EMPLEADOChanged();
partial void OnNOMBRE_EMPLEADOChanging(string value);
partial void OnNOMBRE_EMPLEADOChanged();
partial void OnAPELLIDO_EMPLEADOChanging(string value);
partial void OnAPELLIDO_EMPLEADOChanged();
partial void OnCEDULACHanging(string value);
partial void OnCEDULACHanged();
partial void OnDIRECCIONChanging(string value);
partial void OnDIRECCIONChanged();
partial void OnTELEFONOCChanging(string value);
partial void OnTELEFONOCChanged();
partial void OnID_CARGOChanging(System.Nullable<short> value);
partial void OnID_CARGOChanged();
partial void OnID_EMPRESAChanging(System.Nullable<short> value);
partial void OnID_EMPRESAChanged();
partial void OnID_HORARIOChanging(System.Nullable<short> value);
partial void OnID_HORARIOChanged();
partial void OnSUELDOChanging(System.Nullable<decimal> value);
partial void OnSUELDOChanged();
partial void OnESTADOChanging(System.Nullable<bool> value);
partial void OnESTADOChanged();
partial void OnFECHA_CREACIONChanging(System.Nullable<System.DateTime> value);
partial void OnFECHA_CREACIONChanged();
#endregion

public EMPLEADO()
{
    this._HUELLA_EMPLEADO = new EntitySet<HUELLA_EMPLEADO>(new
Action<HUELLA_EMPLEADO>(this.attach_HUELLA_EMPLEADO), new
Action<HUELLA_EMPLEADO>(this.detach_HUELLA_EMPLEADO));
```



```
        this._REGISTRO = new EntitySet<REGISTRO>(new
Action<REGISTRO>(this.attach_REGISTRO), new Action<REGISTRO>(this.detach_REGISTRO));
        this._CARGO = default(EntityRef<CARGO>);
        this._EMPRESA = default(EntityRef<EMPRESA>);
        this._HORARIO = default(EntityRef<HORARIO>);
        OnCreated();
    }

    [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_ID_EMPLEADO",
AutoSync=AutoSync.OnInsert, DbType="SmallInt NOT NULL IDENTITY", IsPrimaryKey=true,
IsDbGenerated=true)]
    public short ID_EMPLEADO
    {
        get
        {
            return this._ID_EMPLEADO;
        }
        set
        {
            if ((this._ID_EMPLEADO != value))
            {
                this.OnID_EMPLEADOChanging(value);
                this.SendPropertyChanging();
                this._ID_EMPLEADO = value;
                this.SendPropertyChanged("ID_EMPLEADO");
                this.OnID_EMPLEADOChanged();
            }
        }
    }

    [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_NOMBRE_EMPLEADO",
DbType="VarChar(50)")]
    public string NOMBRE_EMPLEADO
    {
        get
        {
            return this._NOMBRE_EMPLEADO;
        }
        set
        {
            if ((this._NOMBRE_EMPLEADO != value))
            {
                this.OnNOMBRE_EMPLEADOChanging(value);
                this.SendPropertyChanging();
                this._NOMBRE_EMPLEADO = value;
                this.SendPropertyChanged("NOMBRE_EMPLEADO");
                this.OnNOMBRE_EMPLEADOChanged();
            }
        }
    }

    [global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_APELLIDO_EMPLEADO",
DbType="NVarChar(50)")]
    public string APELLIDO_EMPLEADO
    {
        get
        {
```



```
        return this._APELLIDO_EMPLEADO;
    }
    set
    {
        if ((this._APELLIDO_EMPLEADO != value))
        {
            this.OnAPELLIDO_EMPLEADOChanging(value);
            this.SendPropertyChanging();
            this._APELLIDO_EMPLEADO = value;
            this.SendPropertyChanged("APELLIDO_EMPLEADO");
            this.OnAPELLIDO_EMPLEADOChanged();
        }
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_CEDULA",
DbType="VarChar(10)")]
public string CEDULA
{
    get
    {
        return this._CEDULA;
    }
    set
    {
        if ((this._CEDULA != value))
        {
            this.OnCEDULACHanging(value);
            this.SendPropertyChanging();
            this._CEDULA = value;
            this.SendPropertyChanged("CEDULA");
            this.OnCEDULACHanged();
        }
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_DIRECCION",
DbType="VarChar(50)")]
public string DIRECCION
{
    get
    {
        return this._DIRECCION;
    }
    set
    {
        if ((this._DIRECCION != value))
        {
            this.OnDIRECCIONChanging(value);
            this.SendPropertyChanging();
            this._DIRECCION = value;
            this.SendPropertyChanged("DIRECCION");
            this.OnDIRECCIONChanged();
        }
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_TELEFONO",
DbType="VarChar(50)")]
```



```
public string TELEFONO
{
    get
    {
        return this._TELEFONO;
    }
    set
    {
        if ((this._TELEFONO != value))
        {
            this.OnTELEFONOChanging(value);
            this.SendPropertyChanging();
            this._TELEFONO = value;
            this.SendPropertyChanged("TELEFONO");
            this.OnTELEFONOChanged();
        }
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_ID_CARGO",
DbType="SmallInt")]
public System.Nullable<short> ID_CARGO
{
    get
    {
        return this._ID_CARGO;
    }
    set
    {
        if ((this._ID_CARGO != value))
        {
            if (this._CARGO.HasLoadedOrAssignedValue)
            {
                throw new
System.Data.Linq.ForeignKeyReferenceAlreadyHasValueException();
            }
            this.OnID_CARGOChanging(value);
            this.SendPropertyChanging();
            this._ID_CARGO = value;
            this.SendPropertyChanged("ID_CARGO");
            this.OnID_CARGOChanged();
        }
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_ID_EMPRESA",
DbType="SmallInt")]
public System.Nullable<short> ID_EMPRESA
{
    get
    {
        return this._ID_EMPRESA;
    }
    set
    {
        if ((this._ID_EMPRESA != value))
        {
            if (this._EMPRESA.HasLoadedOrAssignedValue)
            {
```



```
                throw new
System.Data.Linq.ForeignKeyReferenceAlreadyHasValueException();
            }
            this.OnID_EMPRESAChanging(value);
            this.SendPropertyChanging();
            this._ID_EMPRESA = value;
            this.SendPropertyChanged("ID_EMPRESA");
            this.OnID_EMPRESAChanged();
        }
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_ID_HORARIO",
DbType="SmallInt")]
public System.Nullable<short> ID_HORARIO
{
    get
    {
        return this._ID_HORARIO;
    }
    set
    {
        if ((this._ID_HORARIO != value))
        {
            if (this._HORARIO.HasLoadedOrAssignedValue)
            {
                throw new
System.Data.Linq.ForeignKeyReferenceAlreadyHasValueException();
            }
            this.OnID_HORARIOChanging(value);
            this.SendPropertyChanging();
            this._ID_HORARIO = value;
            this.SendPropertyChanged("ID_HORARIO");
            this.OnID_HORARIOChanged();
        }
    }
}

[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_SUELDO",
DbType="Money")]
public System.Nullable<decimal> SUELDO
{
    get
    {
        return this._SUELDO;
    }
    set
    {
        if ((this._SUELDO != value))
        {
            this.OnSUELDOChanging(value);
            this.SendPropertyChanging();
            this._SUELDO = value;
            this.SendPropertyChanged("SUELDO");
            this.OnSUELDOChanged();
        }
    }
}
```



```
[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_ESTADO",
DbType="Bit")]
public System.Nullable<bool> ESTADO
{
    get
    {
        return this._ESTADO;
    }
    set
    {
        if ((this._ESTADO != value))
        {
            this.OnESTADOChanging(value);
            this.SendPropertyChanging();
            this._ESTADO = value;
            this.SendPropertyChanged("ESTADO");
            this.OnESTADOChanged();
        }
    }
}
```

```
[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_FECHA_CREACION",
DbType="DateTime")]
public System.Nullable<System.DateTime> FECHA_CREACION
{
    get
    {
        return this._FECHA_CREACION;
    }
    set
    {
        if ((this._FECHA_CREACION != value))
        {
            this.OnFECHA_CREACIONChanging(value);
            this.SendPropertyChanging();
            this._FECHA_CREACION = value;
            this.SendPropertyChanged("FECHA_CREACION");
            this.OnFECHA_CREACIONChanged();
        }
    }
}
```



### **7.3. Nombramiento Dirección Ejecutiva.**





**7.4. Cedula: Director.**





**7.5. Ruc: INDESIC**



## 7.6. Otros Anexos